

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Ю.С. Гришук

МІКРОПРОЦЕСОРНІ ПРИСТРОЇ

НАВЧАЛЬНИЙ ПОСІБНИК
по дисципліні
«Мікропроцесорні пристрої»
для студентів електромеханічних
спеціальностей

Затверджено
редакційно-видавничою
радою університету
протокол № 2 від 17.06.2005 р.

Харків НТУ «ХПІ» 2007

ББК 32.973.26-04

Г85

УДК 004.315(075)

Рецензенти: *А.Д. Черенков*, доктор техн.наук, професор ХНТУСГ;
О.Б. Богаєвський, канд. техн. наук, доцент ХНАДУ;
М.І. Шника, канд. техн. наук, зав. проектно-
конструкторським і науково-дослідним відділом
проектування електроприводів і систем управління
ДП заводу “Електротяжмаш”

Гриф надано Міністерством освіти і науки України,
лист № 1.4/18-Г-1305 від 25.07.2007 р.

Г85 **Грищук Ю.С.** Мікропроцесорні пристрої: Навчальний посібник. –
Харків: НТУ “ХПІ”, 2007.– 280с.

ISBN 978-966-593-696-1

Викладено навчально-теоретичний матеріал, що описує основні терміни і поняття, арифметичні, логічні, фізичні основи і архітектуру мікропроцесорів і однокристалевих мікроконтролерів, інтерфейси, запам'ятовуючі і спеціалізовані периферійні пристрої. Посібник містить лабораторний практикум, особливості і приклади практичного застосування мікроконтролерів, контрольні запитання і завдання.

Призначено для студентів спеціальностей 7092206 – “Електричні машини і апарати”, 7092205 – “Електропобутова техніка” і може бути корисною для студентів технічних вузів, аспірантів та інженерно-технічних працівників.

Іл. 72. Табл. 40. Бібліогр. 35 назв.

ББК 32.973.26-04

ISBN 978-966-593-696-1

© Ю.С. Грищук, 2007

ЗМІСТ

Передмова	5
Умовні скорочення і позначення	9
Вступ.....	16
Основні терміни і визначення.....	18
1. Арифметичні, логічні і фізичні основи	27
1.1. Системи числення	27
1.2. Переклад чисел з однієї системи числення в іншу	32
1.3. Арифметичні дії з двійковими числами.....	35
1.4. Одиниці інформації.....	39
1.5. Логічні основи і алгебра логіки	40
1.6. Способи представлення складних функцій	47
1.7. Мінімізація складних виразів.....	48
1.8. Фізичні основи реалізації логічних функцій.....	48
1.9. Логічні схеми з напівпровідниковими елементами.....	50
Контрольні питання і завдання.....	52
2. Архітектура мікропроцесорів.....	55
2.1. Типова структура мікропроцесора	55
2.2. Мікропроцесорний комплект серії KP580.....	70
Контрольні питання і завдання.....	78
3. Запам'ятовуючі пристрої мікропроцесорних систем.....	81
3.1. Класифікація запам'ятовуючих пристроїв	81
3.2. Оперативні запам'ятовуючі пристрої.....	84
3.3. Постійні запам'ятовуючі пристрої.....	87
3.4. Зовнішні запам'ятовуючі пристрої	91
Контрольні питання і завдання	95
4. Інтерфейс мікропроцесорних систем	97
4.1. Визначення і функції інтерфейсу	97
4.2. Пристрої вводу-виводу інформації	101

4.3. Інтерфейс каналу мікроЕОМ	107
4.4. Спеціалізовані периферійні пристрої	109
Контрольні питання і завдання.....	114
5. Однокристальні мікроконтролери	115
5.1. Мікроконтролери сімейства МК51	115
5.2. Однокристальний мікроконтролер КМ1816ВЕ51	119
5.3. Мікроконтролери фірми Atmel.....	160
5.4. Мікроконтролери фірми Microchip	167
5.5. Мікроконтролери фірми Motorola.....	184
Контрольні питання і завдання.....	200
6. Лабораторний практикум і стенд МК51	203
6.1. Лабораторна робота № 1. Дослідження лабораторного стенду мікроконтролера МК51	203
6.2. Лабораторна робота № 2. Програмування і ввід програм з клавіатури стенду МК51	212
6.3. Лабораторна робота № 3. Складання і відладка програм	218
6.4. Лабораторна робота № 4. Дослідження аналого-цифрового перетворювача	233
7. Застосування мікроконтролерів.....	240
7.1. Застосування мікроконтролерів в електромеханічних системах	240
7.2. Застосування мікроконтролерів в електроапаратобудуванні.....	242
7.3. Застосування МК51 при випробуваннях електричних апаратів захисту.....	245
7.4. Структурна схема АСУТПВ з паралельними АЦП.....	252
Контрольні питання і завдання.....	261
Список літератури	263
Додаток 1. Таблиця Д 1 – Система команд мікропроцесора КР580	266
Додаток 2. Приклади виконання завдань.....	271
Додаток 3. Система команд PIC16X17XX.....	272
Додаток 1. Таблиця Д 4 – Команди Асемблера МК51	274

ПЕРЕДМОВА

Найважливішою умовою подальшого соціально-економічного розвитку країни є стабільне зростання енергетики, що забезпечує роботу промисловості, транспорту і інших галузей народного господарства.

Конкретні завдання науково-технічного прогресу в паливно-енергетичному, електротехнічному, машинобудівному і транспортному комплексах висувають в даний час на перше місце проблему забезпечення необхідної надійності, живучості і безаварійності складних технічних об'єктів і їх систем управління. До таких об'єктів відносяться сучасні потужні електроенергетичні системи і їх елементи (наприклад, електричні станції, дальні лінії електропередачі, електричні розподільні системи крупних промислових підприємств, автономні електроенергетичні, перетворювальні, електротехнічні і електромеханічні установки і системи управління ними), що включають пристрої релейного захисту і протиаварійної автоматики, контактні розподільні пристрої. Комутація, захист, управління, регулювання і інші функції в цих об'єктах, як правило, виконуються різними електричними апаратами (ЕА).

Останніми роками у зв'язку з появою мікропроцесорів (МП) намітилася тенденція до широкого використання їх в електро побутовій техніці (ЕПТ) і електроапаратобудуванні, до безпосереднього впровадження в пристрої управління ЕА, техніку релейного захисту, системи протиаварійної автоматики, в автоматизовані системи управління технологічними процесами (АСУТП) виробництва, випробування і дослідження ЕПТ і ЕА. Це викликано появою

нових підвищених і різноманітних вимог до систем автоматизації управління об'єктами. У зв'язку з цим традиційні ЕА і системи управління і регулювання, виконані на аналогових елементах, опинилися не в змозі конкурувати з цифровими пристроями, в яких використовуються мікропроцесори.

Відмінними рисами пристроїв і систем, виконаних на базі МП, є: можливість значного розширення функцій шляхом додавання нових алгоритмів і програм в систему програмного забезпечення, високий рівень уніфікації елементів, можливість перепрограмування, для реалізації тих або інших функцій без зміни комплексу технічних і апаратних засобів і автоматизації процесів діагностики і настройки апаратури. Сучасні МП мають малі габаритні розміри, і можуть розміщатися поруч з керованими об'єктами, володіють високою надійністю і розвиненими логічними можливостями, характеризуються низькими енергоспоживанням і вартістю. У багатьох застосуваннях сучасні МП можуть складатися тільки одну ВІС з одним рівнем напруги +5 В. Крім того, їх застосування в системах управління забезпечує високу швидкодію, гнучкість і ефективність.

Застосування мікропроцесорів вимагає від розробників корінного перегляду традиційних методів при проектуванні систем управління ЕПТ, ЕА і електронними апаратами, заміни у багатьох випадках проектування схем і систем управління розробкою програм настройки мікропроцесорної апаратури на виконання певних функцій. Рішення цих задач вимагає підготовки кваліфікованих фахівців, здатних проектувати, розробляти, експлуатувати і обслуговувати таку складну МП техніку, як ЕА і ЕПТ з мікропроцесорним управлінням.

Для підготовки таких фахівців виникає необхідність у виданні додаткової навчальної літератури, що містить одночасно основи мікропроцесорної техніки і нову інформацію про розвиток мікропроцесорних засобів. Зокрема, потрібна інформація про однокристальні мікроконтролери (МК), які часто в публікаціях називають мікроЕОМ, і їх застосування в ЕПТ, електроапаратобудуванні і інших галузях. Доцільним і корисним, на думку автора, є включення інформації

про лабораторний практикум, що виконується на реальних лабораторних стендах, створених на базі МК, що дозволяє досліджувати їх будову, принцип дії і різні режими роботи найбільш широко вживаних МК.

Мета посібника – сприяти поглибленню знань студентів, що навчаються за спеціальностями 092206 “Електричні машини і апарати”, 092205 “Електропобутова техніка” і іншими технічними спеціальностями в області мікропроцесорних систем, і отриманню ними практичних навичок у роботі з реальними мікропроцесорними пристроями і однокристальними МК. Посібник складений за матеріалами лекцій, лабораторних і практичних занять, курсового проектування по дисциплінах, що пов'язані з мікропроцесорною технікою і її використанням в ЕПТ і ЕА, які включені в навчальний процес НТУ «ХП» і що проводяться автором з 1984 р. по теперішній час.

Зміст посібника визначений, перш за все, прагненням викласти ті аспекти МП пристроїв і їх застосування в ЕПТ, електроапаратобудуванні і в інших напрямках, які недостатньо відображені в літературі, що є на даний час. Зокрема, в посібнику описаний один з дев'яти лабораторних стендів, розроблених на основі однокристальних мікроконтролерів сімейства MCS51 (МК51) на кафедрі “Електричні апарати” НТУ «ХП» за ініціативою і при особистій участі автора посібника. Стенд дозволяє досліджувати 24 режими роботи МК. Тут же подано лабораторний практикум до вивчення МК, складений автором і апробований у навчальному процесі. У лабораторному практикумі передбачається дослідження лабораторного стенда і ознайомлення із структурою однокристальних МК та їх системою команд, дослідження режимів їх роботи, складання і виконання програм, програмування МК і вводу програм з клавіатури лабораторного стенда, складання і вводу програм на мові Асемблер з використанням ПЕОМ, відладка програм крос-засобами, дослідження роботи аналого-цифрових перетворювачів.

У посібнику викладені відомості про сучасні МК провідних зарубіжних виробників: сімейства MCS151 і MCS251 фірми Intel, серії AT89 і AT90 фірми Atmel, сімейств PIC16/17 і PIC16X7XX фірми Microchip, сімейств HC05, HC08,

HC11 фірми Motorola, сімейство цифрових сигнальних процесорів (ЦСП) TMS320Cxxx (платформи C2000, C5000 і C6000 продуктивністю від 20 MIPS до 8800 MIPS) фірми TEXAS INSTRUMENT. Розглянуті їх архітектура, структурні схеми, особливості різних модифікацій сімейств, вказані їх основні технічні характеристики, а також області їх практичного застосування. Приведені приклади застосування різних МК в системах управління транспорту, ЕА, комплектними розподільними пристроями (КРП), гнучкими системами релейного захисту (ГСПЗ), технологічними процесами випробувань і досліджень електричних апаратів, в управлінні електромеханічними системами і електроприводами і т.ін.

Автор вважає своїм приємним обов'язком подякувати ст.н.с. *В.М. Лещенко*, ас. *А.В. Дудник* кафедри «Автоматики і управління в технічних системах» за консультації і допомогу в розробці і налазці мікроконтролерних лабораторних стендів, студентам і співробітникам кафедри «Електричні апарати» за надану допомогу при створенні лабораторних стендів і лабораторії МПС, зав. лабораторією кафедри інформаційних систем і технологій в міському господарстві Харківської Національної академії міського господарства *С.Ю. Грицуку* за надану істотну допомогу в підготовці рукопису і оригінал-макету посібника.

Автор виражає глибоку подяку рецензентам:

Черенкову О.Д., доктору техн. наук, професору кафедри теоретичної електротехніки Харківського Національного технічного університету сільського господарства, ім. Петра Василенка;

Богаєвському О.Б., канд. техн. наук, доценту кафедри автомобільної електроніки Харківського Національного автодорожнього університету;

Шника М.І., канд. техн. наук, зав. проектно-конструкторським і науково-дослідним відділом проектування електроприводів і систем управління ГП заводу «Електротяжмаш».

Ряд цінних порад і зауважень, які були зроблені ними при рецензуванні рукопису, в суттєвій мірі сприяли поліпшенню даного посібника.

УМОВНІ СКОРОЧЕННЯ І ПОЗНАЧЕННЯ

1. Українська нотація (у дужках приведена англійська нотація)

А	– акумулятор (див. АСС)
АВ	– автоматичний вимикач
АЛП	– арифметично – логічний пристрій
АСУТПВ	– автоматизована система управління технологічним процесом випробувань (досліджень)
ВІС	– велика інтегральна схема
ШЗ	– швидкодіючий запобіжник
ВК	– вибір корпусу (див. СЕ)
ЗПД	– зовнішня пам'ять даних
ЗПП	– зовнішня пам'ять програм
ВХПР	– вхід приймача УАПП (см. RXD)
ВИХПЕР	– вихід передавача УАПП (див. TXD)
ЗП	– сигнал запису, що управляє (див. WR)
ІС	– інтегральна схема
ЗПР	– запит переривання(див. INT)
КОП	– код операції (поле в тілі команди)
КРП	– комплектно – розподільний пристрій
МК	– мікроконтролери однокристальні
МК51	– мікроконтролери серії 1816: КМ1816ВЕ51
МП	– мікропроцесор
МПРЗ	– мікропроцесорний релейний захист
МПС	– мікропроцесорна система
ОЗП	– оперативний запам'ятовуючий пристрій (див. RAM)
ВРПП	– сигнал відключення резидентної пам'яті програм, (див. EA)
ПП	– пам'ять програм
ПРОГ	– сигнал, що управляє програмуванням резидентної

	пам'яті програм (см.PROG)
РА	– реєстр адреси (див. RAR)
РВВ	– ВІС розширювача вводу/виводу
ДЗПП	– сигнал дозволу зовнішньої пам'яті програм (див. PSEN)
РК	– реєстр команд (див. IR)
РМП	– реєстр маски переривань (див. IE)
РЗП	– реєстр загального призначення
РП	– реєстр пріоритетів (див. IP)
РПД	– резидентна пам'ять даних
РПП	– резидентна пам'ять програм
РРТЛ	– реєстр режиму таймера/лічильника (див. TMOD)
РСФ	– реєстри спеціальних функцій (PSW, TMOD, TCON, SCON, PCON, IE, IP)
РУД	– реєстр покажчика даних (див. DPTR)
РУП	– реєстр управління потужністю (див. PCON)
РУПП	– реєстр управління послідовного порту (див. SCON)
РПС	– реєстр покажчика стека (див. SP)
РУСТ	– реєстр управління/стану таймера (див. TCON)
САЗП	– сигнал строба адреси зовнішньої пам'яті (див. ALE)
НВІС	– надвелика інтегральна схема
СКД	– керуючий сигнал скидання (див. RST)
ЛК	– лічильник команд (див. PC)
ССП	– слово стану програми (див. PSW)
Т/Л	– таймер/лічильник подій (див. TCNT)
УАПП	– універсальний асинхронний приймач-передавач (послідовний порт МК51)
ЧТ	– керуючий сигнал читання (див. RD)
ША, ШД	– шина адреси, шина даних
КРОК	– керуючий сигнал покрокового (покомандного) режиму, роботи (див. SS)

EA	– електричний апарат
ЕПТ	– електропобутова техніка

2. Англійська нотація

A	– реєстр-акумулятор
AC	– допоміжне перенесення (Auxiliary Carry flag in PSW)
ACC	– символічне ім'я реєстра A
ad (dir)	– пряма 8-бітова адреса байта РПД (0-127), порту або РСФ
ADC	– аналого-цифровий перетворювач
add	– пряма 8-бітова адреса призначення
ads	– пряма 8 бітова адреса джерела
ad11	– пряма 11-бітова адреса передачі управління
ad16	– пряма 16-бітова адреса передачі управління
ad16h	– старший байт прямої 16-бітової адреси
ad16l	– молодший байт прямої 16-бітової адреси
B	– реєстр-розширювач акумулятора
bit	– пряма 8-бітова адреса біта (МК51)
C	– прапор перенесення
CAN	– контролер локальної мережі (Controller Area Network)
CE (CS)	– Chip Enable (Chip Select) (див. ВК)
CLK	– синхросигнал (Clock)
CPU	– центральний процесорний пристрій (Central Processor Unit)
C/\bar{T}	– керуючий біт вибору режиму таймера/лічильника, (Timer or Counter selector in TMOD)
#	– 8-бітовий безпосередній операнд (константа)
#d16	– 16-бітовий безпосередній операнд (константа)
#d16h	– старший байт 16-бітового безпосереднього операнда
#d16l	– молодший байт 16-бітового безпосереднього операнда
DAC	– цифро-аналоговий перетворювач

DPH	– Data Pointer High (старший байт РУД)
DPL	– Data Pointer Low (молодший байт РУД)
EA	– керуючий біт зняття блокування всіх переривань (Enable All Control bit in IE)
ЕССР	– модернізований РСР
\overline{EA}/VPP	– External Address/Voltage Power Programming (див. ОРПП)
EPROM	– Erasable Programmable Read Only Memory (див. РПП)
ES	– керуючий біт дозволу переривання від УАПП (Enable Serial port control bit in IE)
ET	– керуючий біт дозволу переривання від таймера (Enable Timer port control bit in IE)
EX	– керуючий біт дозволу зовнішнього переривання (Enable External interrupt control bit in IE)
F0, F1	– прапори, які специфікуються користувачем
GATE	– біт управління блокуванням Т/С (Gating control bit in TMOD)
GFLOPS	– мільярд операцій з плаваючою крапкою в секунду
GFO,GF1	– прапори користувача (General Flags in PCON) в МК51
i	– біт в КОП, що визначає регістр непрямої адреси: i=0,1 (R0, R1)
I ² C	– інтерфейс інтегральних схем
IDL	– керуючий біт холостого ходу (Idle mode in PCON)
IE	– 1) Interrupt Enable register (див. РМП) 2) прапор зовнішнього переривання, встановлений по спаду з сигналу ЗІР (Interrupt Edge flag in TCON)
IP	– Interrupt Priority control register (див. РПП)
INT	– Interrupt (див. ЗІР)
IR	– регістр команд (Instruction Register) (див. РК)
IT	– біт вибору типу (рівень/спад) керуючого сигналу ЗІР (Interrupt Type control bit in TCON)
I/O	– ввід-вивід

M0, M1	– керуючі біти вибору режиму роботи Т/С (Operating Mode in TMOD)
MI ² C	– ведучий I ² C
MIPS	– мільйон операцій операцій в секунду
OV	– прапор переповнювання (Overflow flag in PSW)
P	– прапор паритету (Parity flag in PSW)
PC	– лічильник команд (Program Counter) (див. ЛК)
PCON	– регістр управління потужністю Power Control register (див. РУП)
PD	– біт управління потужністю (Power Done in PCON)
PROG	– Programming EPROM (див. ПРОГ)
PS	– керуючий біт пріоритету УАПП (Serial port Priority control bit in IP)
PSEN	– Program Store Enable (див. ДЗПП)
PSP	– паралельний ведений порт
PSW	– Program Status Word (див. ССП)
PT	– керуючий біт пріоритету таймера (Timer Priority control bit in IP)
PWM	– широтно-імпульсний модулятор
PX	– керуючий біт пріоритету зовнішнього переривання (External interrupt Priority control bit in IP)
PX.Y	– символічне ім'я біта Y порту X; Y=0 ÷ 7 для МК51: X=0,1,2,3
RAM	– Random Access Memory (див. ОЗП)
RAR	– RAM Address Register (див. РА, укр.)
RB8	– дев'ятий (bit 8) прийнятий біт (Receive Bit 8 in SCON)
RD	– Read (див. ЧТ)
rel	– 8-бітова відносна адреса передачі управління (-127÷+128) у МК51
REN	– керуючий біт дозволу прийому в УАПП (Receive Enable Control bit in SCON)
RI	– прапор переривання від приймача (Received Interrupt flag)

in SCON)

- Ri – узагальнене ім'я регістра непрямої адреси (R0 або R1)
- Rn – узагальнене ім'я робочого регістра ($n=0 \div 7$)
- rrr – 3-бітове поле в коді операції, що визначає регістр загального призначення (R0-R7)
- RS – керуючий біт вибору банку регістрів (Register bank Select in PSW)
- RST – Reset (див. СКД)
- RST/VPD – Reset/Voltage Power Done (див. СКД)
- RTI – система переривань реального часу (Real Time Interrupt)
- RXD – Receive Data pin (див. ВХІР)
- Sn – стан пристрою управління МК51 $n=1 \div 6$ (State)
- SCON – Serial Port Control/status register (див. РУПП)
- SCSI – високошвидкісний інтерфейс призначений для жорстких дисків, а також деяких зовнішніх пристроїв призначених для вводу даних в комп'ютер.
- SM0, SM1,
SM2 – керуючі біти режиму роботи УАПП (Serial port Mode control bits in SCON)
- SMOD – керуючий біт подвійної швидкості передачі (Double Baud rate in PCON)
- SP – Stack Pointer (див. РУС)
- SPI – синхронний послідовний інтерфейс
- SS – Single Step (див. КРОК)
- STB – Strobe (див. СТБ)
- SXPY – пояснення на тимчасових діаграмах і схемах, що прив'язують сигнали до станів S пристрою управління і фаз P синхросигналів ($X=1 \div 6; Y=1,2$) для МК51
- T0, T1 – тест-входи МК
- TB8 – дев'ятий біт (bit 8), що передається (Transmit Bit8 in SCON)

TCNT	– Timer/Counter events (див. T/C)
TCON	– Timer/Counter Control/status register (див. PУСТ)
TF	– прапор переповнювання таймера (Timer overflow Flag in TCON)
TH	– старший байт таймера (Timer High byte)
TI	– прапор переривання від передавача (Transmit Interrupt flag in SCON)
TL	– молодший байт таймера (Timer Low byte)
TMOD	– Timer/counter Mode register (див. PPTC, укр.)
TR	– керуючий біт пуску таймера (Timer Run control bit in TCON)
TXD	– Transmit Date (см. ВыхПЕР)
USART	– універсальний синхронно-асинхронний приємопередавач
USB	– універсальна послідовна шина
Vref	– джерело опорної напруги
WDT	– сторожовий таймер
WR	– Write (див. ЗП)

3. Спеціальні символи

←	– оператор привласнення (заміщення)
↔	– оператор взаємного обміну
∧, ∨, ∇	– оператори логічних операцій: I (кон'юнкція) АБО (диз'юнкція), виключаюче АБО
@	– префікс непрямої адресації
#	– префікс безпосереднього операнда: В, Н – суфікси відповідно двійкового (Binary) і шістнадцятирічного (Hex.) кодів
(Y)	– вміст регістра або елемента пам'яті з ім'ям Y
((Y))	– вміст елемента пам'яті, що адресується вмістом Y (непряма адресація)
⌘	– поточний вміст лічильника команд МК

ВСТУП

У розвитку електронних цифрових обчислювальних машин відповідно до використовуваних технологічних принципів розрізняють п'ять поколінь цифрових машин: 1) на електронних лампах; 2) на транзисторах; 3) на інтегральних схемах (ІС); 4) на інтегральних схемах з великим ступенем інтеграції (ВІС); 5) на інтегральних схемах з надвеликим ступенем інтеграції (НВІС).

Перший проект електронно-цифрової машини розробив в 1937–1939 рр. в місті Еймс (штат Айова, США) професор фізики і математики Джон В. Анатасов, болгарин за походженням [1]. За цим проектом в керованій частині машини містилося 300 електронних ламп і 1642 конденсатори [1]. У 1945 р. американський математик Джо фон Нейман істотно розвинув ідею програмного управління обчислювальним процесом і сформулював принципи організації пам'яті. Після винаходу транзистора в 1948 р. почався процес заміни електронних ламп дискретними напівпровідниковими пристроями, а потім інтегральними схемами (ІС) і великими інтегральними схемами (ВІС).

Створення мікропроцесора (програмно-керованого пристрою, що здійснює процес обробки інформації і управління, побудованого на одній або декількох ВІС) почалося порівняно недавно і явилось слідством розвитку і вдосконалення технології виробництва ІС і ВІС. Перше повідомлення про розробку мікропроцесора 1-4004 опублікувала фірма Intel в 1971 р.

Створення мікропроцесора як функціонально закінченої частини ЕОМ на одній інтегральній схемі є одним із найбільших досягнень минулого двадцятого сторіччя. Здатність програмувати послідовності виконуваних функцій, тобто здатність працювати за заданою програмою, є основною відмінністю МП від елементів «жорсткої» логіки (інтегральних схем малого і середнього ступеня інтеграції). Удосконалення технологій виробництва ІС і ВІС привело до того, що за порівняно невеликий час з'явилися чотири покоління МП, що відрізняються своїми технічними характеристиками [2]:

перше – повільно діючі (час виконання команди 10-20мкс), чотирьохрозрядні МП, що мають відносно обмежені набір команд, об'єм пам'яті і види адресації;

друге – чотирьох- і восьмирозрядні МП з часом виконання команди 2–5 мкс, розширеним набором команд і об'ємом пам'яті, і різними видами адресації;

третє – швидкодіючі (час виконання команди 100-300 нс), секціоновані МП, виконані з використанням біполярної технології і мікропрограмним принципом управління, а також 16-розрядні процесори і спецпроцесори;

четверте – однокристальні мікроЕОМ з вбудованими портами вводу-виводу і пристроями, що запам'ятовують, 32-розрядні МП.

Сучасні МП – 64-розрядні двохядерні МП, однокристальні мікроконтролери, що містять в одному корпусі НВІС наступне: МП, роздільні пам'ять даних і пам'ять програм, паралельні і послідовні порти вводу-виводу, таймери/лічильники, АЦП, ЦАП, сторожовий таймер, внутрішній температурний сенсор і інші елементи, необхідні для ефективного використання МП в системах управління промисловим і побутовим устаткуванням.

Основні причини широкого впровадження мікропроцесорної техніки:

використання в мікропроцесорних системах цифрового способу подання інформації, що дозволяє значно підвищити швидкість її передачі і перешкодостійкість; застосування програмного способу обробки інформації, компактність, висока надійність і низьке споживання електроенергії.

ОСНОВНІ ТЕРМІНИ І ВИЗНАЧЕННЯ

Мікропроцесорна техніка (МПТ) має свою специфічну термінологію, в якій постійно з'являються і входять в практику нові терміни, визначення і поняття. Нижче наводяться рекомендовані Міжнародним центром наукової і технічної інформації, Міжнародним науково-дослідним інститутом проблем управління [3] і загальноприйняті в сучасній літературі по МП [1,2,4,5] основні терміни і визначення, які використані в даному посібнику.

Знання основних термінів украй важливо, оскільки є істотним чинником для студентів на початковому етапі вивчення основ МП техніки і дозволяє надалі якісно сприймати матеріал цього курсу та самостійно вивчати сучасну літературу з МП.

Основні терміни, які використовуються в МПТ

ASCII – стандартизована система позначень різних символів (букв і цифр клавіатури комп'ютера) для пересилання інформації (American Standart Code for Information Interchange).

FLASH-пам'ять – зовнішня для мікропроцесора пам'ять, яка реалізована мікросхемно і виконує функції жорсткого диска. Від іншої відрізняється високою стійкістю від ударів і вібрації.

Адреса – вказівка місце розташування об'єкта в пам'яті ЕОМ.

Адресний простір – максимальне число секцій пам'яті (у кілобайтах, мегабайтах, гігабайтах), які можна пронумерувати (проадресувати) за допомогою шини адреси, реалізованої в мікропроцесорі. Якщо шина адреси має 16 дротів (16 розрядів), то адресний простір дорівнює 2^{16} байтам, тобто 64 Кбайта.

Алгоритм – набір розпоряджень, що однозначно визначають зміст і послідовність виконання операцій для систематичного вирішення певного завдання.

Аналого-цифровий перетворювач (АЦП) – пристрій, що перетворює безперервний (аналоговий) сигнал в дискретні цифрові величини.

Арифметико-логічний пристрій (АЛП) – функціональна частина процесора, що виконує арифметичні і логічні дії над даними.

Архітектура мікропроцесора – складові частини мікропроцесора, а також їх взаємне з'єднання і взаємодія між ними. Архітектура включає: 1) структурну схему самого МП; 2) програмну модель МП (опис функцій регістрів); 3) інформацію про організацію пам'яті (місткість пам'яті і способи її адресації); 4) опис організації процедур вводу-виводу і управління; 5) опис системи команд і ін.

Асемблер – системна обслуговуюча програма, що перетворює символічні інструкції в команди машинної мови і що дозволяє проводити діагностику, формування посилань для редактора зв'язків і т.ін.

Байт – оброблюваний як єдине ціле елемент даних, що складається з послідовності двійкових розрядів. У мікроЕОМ, як правило, використовують восьмибітовий байт (Кбайт – $2^{10} = 1024$ байт; Мбайт – мегабайт = 2^{20} байт; Гбайт – гігабайт = 2^{30} байт $\approx 1\,000\,000$ Кбайт).

Бейсік – популярна алгоритмічна мова програмування високого рівня, спочатку створена для цілей навчання програмуванню. Мова має відносно простий синтаксис, що полегшує його швидке освоєння.

Біт – один двійковий розряд машинного слова або одиниця інформації, що приймає значення 0 або 1.

Бод – одиниця швидкості передачі інформації послідовним двійковим кодом (біт в секунду).

Буфер – запам'ятовуючий пристрій для тимчасового зберігання даних з метою узгодження асинхронно працюючих пристроїв, або область ОЗП, що тимчасово резервується для виконання процедури вводу-виводу.

Вбудовувана мікроЕОМ – мікроЕОМ, конструктивно пристосована для роботи у складі приладів і устаткування.

Відеотермінал — пристрій, що забезпечує можливість обміну даними по каналу зв'язку з віддаленою ЕОМ. Включає клавіатуру для вводу і дисплей для виведення інформації.

Графічний пристрій – пристрій виводу, призначений для представлення даних у вигляді графічного зображення на папері.

Дані – інформація (числа), призначена для обробки в ЕОМ.

Діалоговий режим – режим взаємодії користувача з ЕОМ, при якому кожен запит користувача викликає негайну у відповідь дію ЕОМ.

Дисплей – пристрій, що забезпечує візуальне представлення цифрової, алфавітно-цифрової і (або) графічної інформації на екрані електронно-променевої трубки, в плазмових панелях, на рідких кристалах, світлодіодах і т.ін. у формі, зручній для оператора.

Довжина слова – кількість бітів в одному машинному слові.

Доступ (звернення) – процедура встановлення зв'язку з пристроєм (ЗП), що запам'ятовує, для вибірки / запису даних.

Емуляція – імітація функціонування однієї системи засобами іншої системи без втрати функціональних можливостей або спотворення одержуваних результатів.

Завантажувач – обслуговуюча програма для завантаження об'єктної програми в пристрій, що оперативно запам'ятовує (ОЗП).

Запам'ятовуючий пристрій (ЗП) – виріб, що реалізовує функціональну частину ЕОМ, яка призначена для запам'ятовування і (або) видачі інформації.

Інтерпретатор – обслуговуюча програма, що здійснює пооператорну трансляцію і виконання початкової програми.

Інтерфейс – сукупність уніфікованих технічних і програмних засобів, необхідних для підключення даних пристроїв до системи або однієї системи до іншої.

Канал передачі даних – сукупність технічних засобів і пристроїв, що забезпечують передачу інформації і перетворення сигналів.

Канал прямого доступу до пам'яті - сукупність технічних засобів і пристроїв, що забезпечують прямий доступ до пам'яті без використання ЦП.

Кеш-пам'ять – допоміжна оперативна пам'ять, недоступна для програміста. Вона розміщується функціонально між процесором і оперативним

пристроєм, що запам'ятовує, і служить для підвищення швидкодії мікропроцесорної системи. У ній зберігається і оновлюється вміст секцій пам'яті, що здубльований з пристроєм ОЗП з командами, які найбільш часто вживаються в програмі.

Команда – розпорядження, що визначає крок процесу виконання програми; містить вказівку операції, адреси операндів і інші службові ознаки.

Компілятор – обслуговуюча програма, що виконує трансляцію на машинну мову програми, записаної на початковій мові програмування.

Контролер – пристрій, що виконує функції управління, передачі даних і що звільняє від цих функцій процесор.

Контроль парності – метод контролю даних, при якому сума по модулю 2 двійкових одиниць в машинному слові, включаючи контрольний розряд, повинна мати певну парність, тобто бути завжди парною або непарною.

Користувач – особа, що використовує даний обчислювальний пристрій для виконання необхідних йому робіт.

Лічильник команд – регістр, на основі вмісту якого виробляється адреса наступної команди.

Магістраль – сукупність шин, що зв'язують між собою всі пристрої мікропроцесорної системи.

Маркер (курсор) – спеціальний знак на екрані дисплея для вказівки певних позицій або елементів.

Машинне слово – послідовність бітів або знаків, яка трактується в процесі обміну або обробки як єдиний елемент даних.

Машинний код – двійковий код, в якому за специфічними для даної ЕОМ правилами кодується її система команд.

Мережа мікроЕОМ – система сполучених між собою, мікроЕОМ, що обмінюються інформацією.

Мікро С, мікро Паскаль, мікро Бейсік – мови високого рівня, які призначені для програмування МК за допомогою ЕОМ.

Мікропроцесор (МП) – програмно-керований пристрій, що здійснює

процес обробки цифрової інформації і управління ним, побудований, як правило, на одній або декількох великих інтегральних схемах.

Мікропроцесорний комплект – сукупність мікропроцесорних і інших інтегральних мікросхем, сумісних по конструктивно-технологічному виконанню і призначених для сумісного застосування.

мікроЕОМ – ЕОМ, що складається з мікропроцесора, напівпровідникової пам'яті, засобів зв'язку з периферійними пристроями і при необхідності – пульта управління і джерела живлення, об'єднаних загальною конструкцією.

Місткість пам'яті – найбільший об'єм даних, виражений в одиницях інформації, який може одночасно зберігатися в ЗП.

Мова Асемблер – символічна мова програмування, структура операторів якого визначається форматами команд і даними машинної мови.

Мова високого рівня – мова програмування, засоби якого допускають опис проблеми в наочному, легко сприйманому вигляді.

Мова програмування низького рівня – машинний код, машинна мова, Асемблер.

Модем – модулятор і демодулятор, що об'єднаний в одному пристрої, який здійснює перетворення сигналів для передачі їх по лінії зв'язку.

Монітор – записана в ПЗП системна програма, що реалізовує операції обміну із зовнішніми пристроями і що допомагає здійснити відладку програм.

Накопичувач на гнучкому магнітному диску – зовнішнє ЗП, в якому носіями інформації є змінні, гнучкі магнітні диски.

Накопичувач на компакт-дисках (CD-R, CD-RW, і DVD) – зовнішній ЗП в якому, носієм інформації є рельєфна підкладка з полікарбонату, на яку нанесений тонкий шар металу (звичайно алюмінію CD-R, CD-RW), що відображає світло, і спеціального матеріалу (у дисках DVD), який під впливом лазерного променя змінює свій стан, переходячи з кристалічного в аморфний. Місткість DVD дисків значно перевищує місткість накопичувачів на магнітних дисках і може досягати 17 Гбайт.

Непряма адресація – система адресації, при якій адресна частина

інструкції містить адресу елемента пам'яті, що містить пряму адресу або іншу непрямую адресу.

Обмін – обмін інформацією, процедура пересилки інформації у формі паралельного або послідовного коду.

Однокристална ЕОМ – мікроЕОМ, побудована у вигляді однієї великої або надвеликої інтегральної схеми.

Операнд – елемент даних, над яким виконується операція.

Оперативний запам'ятовуючий пристрій (ОЗП) – ЗП з прямою адресацією, що відрізняється швидкістю доступу.

Оператор – допустима в мові програмування синтаксична конструкція, що відображає певну дію в програмі (привласнення значення, передачу управління і т. д.).

Операційна система – комплекс взаємозв'язаних керуючих і обслуговуючих програм, що забезпечують автоматичне управління обчислювальними процесами і ресурсами ЕОМ при рішенні задач.

Паралельний інтерфейс – мікросхема, що реалізовує пересилку інформації в паралельному двійковому коді по шині з 8, 12, 16 паралельних електричних дротів і що забезпечує ввід-вивід інформації, наприклад з комп'ютера на принтер.

Паралельний порт – порт вводу-виводу, через яке дані передаються і приймаються паралельно, тобто одночасно всі розряди, що відносяться до даного символу або блоку даних.

Перепрограмований постійний запам'ятовуючий пристрій (ППЗП) – ЗП, в який інформація, що підлягає зберіганню, заноситься багато разів, але при цьому час запису значно перевищує час вибірки.

Переривання – тимчасове припинення виконання поточної програми і перехід до виконання програми обслуговування пристрою, що викликав переривання.

Підпрограма – частина програми, що допускає багатократне звернення до неї з різних точок програми.

Показчик стека – реєстр, що визначає адресу верхнього осередку використовуваного стека.

Порт вводу-виводу – засіб для підключення периферійних пристроїв до ЕОМ.

Послідовний інтерфейс – мікросхема, яка реалізує пересилку інформації в послідовному кодi по двопровідній лінії. Імпульси напруги двійкових кодованих чисел (біти) передаються послідовно один за іншим (біт за бітом). Послідовний інтерфейс служить для пересилки інформації між окремими комп'ютерами (по телефонних лініях) або між керуючим комп'ютером, і окремим об'єктом управління (більш ніж на 5–10 м). Як правило, послідовний інтерфейс використовується в парі з модемом – модулятором-демодулятором, а також може використовуватися для зв'язку МК з комп'ютером.

Послідовний порт – порт вводу-виводу, через який дані передаються і приймаються послідовно розряд за розрядом по двопровідній лінії.

Постійний запам'ятовуючий пристрій (ПЗП) – ЗП з незмінним змістом пам'яті.

Пристрій вводу-виводу – пристрій, що забезпечує обмін даними між оперативною пам'яттю ЕОМ і периферійними пристроями.

Програма – послідовність інструкцій, що реалізують алгоритм. Програми звичайно можуть бути написані: а) у двійковому або шістнадцятирічному (машинному) кодi, який безпосередньо сприймається процесором; б) на мові типу Асемблер; в) на мові високого рівня.

Програма на початковій мові – програма, представлена в системі в початковому вигляді, тобто написана на одній з мов програмування. Вимагає для свого виконання попереднього перетворення, наприклад трансляції.

Програмактор – спеціальний пристрій для запису підготовлених користувачем програм в ПЗП або РПЗП.

Програмований постійний запам'ятовуючий пристрій (ППЗП) – ПЗП, в який інформація заноситься одноразово користувачем і після цього не міняється.

Програмна сумісність – можливість виконання одних і тих же програм на ЕОМ різних типів з отриманням ідентичних результатів.

Програмне забезпечення – сукупність програм, що забезпечують реалізацію функцій мікроЕОМ, мікропроцесорного пристрою або системи.

Пряма адресація – система адресації, при якій адресна частина інструкції містить адресу, що визначає безпосередньо елемент пам'яті або місце на носії, що містить необхідний операнд.

Прямий доступ в пам'ять – метод, що дозволяє з великою швидкістю здійснювати завантаження даних з периферійного пристрою прямо в оперативний ЗП.

Регістр – функціональний блок для зберігання машинного слова або його частини.

Редактор – обслуговуюча програма для редагування, набору даних з метою представлення їх у вигляді, що сприймається засобами обробки, або у відповідному форматі виводу.

Режим роботи в реальному масштабі часу – режим роботи системи, що забезпечує прийом до обробки даних у міру їх надходження без яких-небудь обмежень і видачу результатів в необхідні інтервали часу.

Розряд – позиція для запису цифр числа в якій-небудь системі числення. В МПТ і ЕОМ використовуються двійкова, вісімкова, десяткова, двійково-десяткова і шістнадцятирічна системи числення. Команди, що виконуються ЕОМ, зберігаються в пам'яті, пересилаються і виконуються тільки в двійковому коді. Кожному розряду двійкового числа відповідає провідник (лінія) шини, по якій передається це число в паралельному коді.

Розрядність мікропроцесора – одна з основних характеристик МП, яка визначається в загальному випадку розрядністю двійкового числа, що обробляється на основних операціях в АЛУ за один раз і визначає його швидкодію. Як правило, розрядність МП відповідає розрядності регістрів загального призначення і шини даних.

Секційний мікропроцесор – мікропроцесор, одержаний на основі з'єднання

однотипних 2-, 4-, 8- або 16 - розрядних мікропроцесорних інтегральних схем, кожна з яких має в своєму складі АЛП і декілька загальних регістрів. Паралельне з'єднання цих мікросхем дозволяє побудувати мікроЕОМ з будь-якою бажаною довжиною машинного слова.

Символ – окремий знак із заданого набору умовних позначень, використовуваних для представлення даних в ЕОМ.

Система команд – повний набір всіх інструкцій, допустимих в машинній мові даної ЕОМ.

Стек — пам'ять магазинного типу.

Таймер – пристрій (мікросхема), призначений для реалізації функцій, пов'язаних з відліком часу, наприклад для підрахунку кількості імпульсів напруги, їх затримки на заданий час, організації часових інтервалів, реалізації серії імпульсів заданої частоти і т.ін.

Тактова частота – основний параметр МП, який визначає швидкість обробки інформації процесором, а також рядом інших мікросхем і окремих плат, які входять до складу комп'ютера.

Технічне забезпечення – сукупність технічних компонентів мікроЕОМ, мікропроцесорного пристрою або системи.

Утиліта - допоміжна невелика програма, що входить в системну програму, яка забезпечує нормальну роботу комп'ютера, його обслуговування і настройку. До таких системних програм відноситься в першу чергу операційна система комп'ютера.

Файл – послідовність записів, що розміщується на зовнішніх ЗП і яка розглядається в процесі пересилки і обробки як єдине ціле.

Центральний процесор (ЦП) – процесорна ВІС, що безпосередньо здійснює процес обробки даних.

Цифроаналоговий перетворювач (ЦАП) – пристрій, що перетворює дискретний цифровий сигнал в безперервний аналоговий сигнал.

Шина – група ліній передачі інформації, об'єднаних загальною функціональною ознакою (наприклад, шина даних, адреси, управління).

1. АРИФМЕТИЧНІ, ЛОГІЧНІ І ФІЗИЧНІ ОСНОВИ

У світовій практиці відомі і використовуються безліч різних систем числення. Найширше при обчисленнях використовується десяткова система числення. Ця система, хоч і найпоширеніша в світі, але не єдина. Наприклад, в МП і в ЕОМ для представлення інформації використовується двійкова система числення, тобто всі арифметичні і логічні операції виконуються із застосуванням двійкових чисел.

1.1. Системи числення

Системою числення називається сукупність правил запису чисел обмеженою кількістю символів, званих цифрами. Розрізняють *позиційні* і *непозиційні* системи числення. *Розрядом цифри* називають місце цифри в позиційній системі. *Основою n системи числення* називається кількість різних цифр, які застосовують для написання чисел: $0, 1, 2, \dots, n-1$.

Числа в десятковій системі числення можуть бути представлені у формі суми добутоків, які утворюються множенням однієї з цифр від 0 до 9 на відповідний ступінь числа 10. Розглянемо число 2745. Його можна представити в наступній формі: $2 \cdot 1000 + 7 \cdot 100 + 4 \cdot 10 + 5 \cdot 1$, або $2 \cdot 10^3 + 7 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0$.

Аналогічно можна утворити будь-яку іншу систему числення. Для цього необхідно спочатку вибрати число для основи системи. Цифри, які ми помножимо на ступені основи, починаються з нуля і досягають числа, рівного основі системи мінус одиниця. Для прикладу виберемо як основу число 3. Одержуємо трійкову систему, а цифри перед ступенями числа 3 будуть 0, 1 і 2. Систему числення можна утворити від будь-якого числа, відмінного від 0 і 1.

У дискретній техніці використовуються наступні системи числення: *двійкова* (позначається індексом b), *вісімкова* (індексом q), *шістнадцятирична* (індексом h), *двійково-десяткова* (індексом bdc або ДДК).

Для представлення інформації в сучасних МП і ЕОМ використовується, проста зі всіх можливих систем – двійкова. На користь цієї системи є ряд міркувань, детально викладених в [1,6,7], які ми розглянемо в наступних підрозділах і які пов'язані з арифметичними, логічними і фізичними основами МП і ЕОМ. У двійковій системі числення основою є число 2, а для запису якого-небудь числа використовуються тільки цифри 0 і 1. Такі цифри називають бітами. Це добре узгоджується з технічними характеристиками цифрових систем, що мають, як правило, лише два стійкі стани – 0 і 1.

Фізично в цифрових електронних системах значення 0 відповідає напрузі низького рівня (L-рівня), а значення 1 – напрузі високого рівня (H- рівня).

Проведемо порівняння запису чисел в двох системах – десятковій і двійковій, почавши з найменших чисел, значення яких в цих системах числення приведені в табл. 1.1.

Таблиця 1.1 – Значення чисел в десятковій і двійковій системах

Десяткова система	Двійкова система	Десяткова система	Двійкова система
0	0	10	1010
1	1	11	1011
2	10	12	1100
3	11	13	1101
4	100	14	1110
5	101	15	1111
6	110	16	10000
7	111	17	10001
8	1000	18	10010
9	1001	19	10011

Для подання чисел в двійковій системі, наведемо значення числа 2, піднесеного до ступеня:

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}
1	2	4	8	16	32	64	128	256	512	1024

Представимо для прикладу число 344 як суму ступенів числа 2:

$$344 = 256 + 64 + 16 + 8 = 2^8 + 2^6 + 2^4 + 2^3$$

або точніше

$$344_{10} = 1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 101011000_2$$

(числа, написані справа вниз, указують основи систем числення).

При записі двійкового числа кожна позиція зайнята двійковою цифрою – *бітом*. Розглядаючи двійкові числа, часто користуються поняттями – найменший значущий біт (найменший двійковий розряд) і найбільший значущий біт (найбільший двійковий розряд). Аналогічно з найбільшою і найменшою значущими цифрами десяткового числа, найменший значущий біт двійкового числа має найменшу вагу, а найбільший значущий біт – найбільшу вагу. Зазвичай двійкове число записується так, що найбільший значущий біт є крайнім зліва. Як і в десятковій системі, цифри, якими написані числа в двійковій системі, називають розрядами або позиціями і відлічують від двійкової коми справа наліво; перший, другий, третій і т.д. – для цілої частини; зліва направо: перший, другий, третій і т.д. – для дробової частини. Як і в десятковій системі, в двійковій для відділення дробової частини від цілої використовується кома (крапка) і кожній позиції числа (розряду) привласнена певна вага. Якщо в десятковій системі вага дорівнює числу 10 в деякому ступені, то в двійковій системі замість числа 10 використовується число 2.

Розглянемо, як виглядають дробові числа в двійковій системі. Напишемо ступені числа 2 при негативних показниках ступеня:

$$2^{-1} = 1/2^1 = 1/2 = 0,5; \quad 2^{-2} = 1/2^2 = 1/4 = 0,25; \quad 2^{-3} = 1/2^3 = 1/8 = 0,125;$$
$$2^{-4} = 1/2^4 = 1/16 = 0,0625; \quad 2^{-5} = 1/2^5 = 1/32 = 0,03125 \text{ і т.д.}$$

За загальним правилом подання двійкових чисел як сум добутків ступенів числа 2 на числа 0 і 1 можна записати наступні дробові числа:

$$0,25_{10} = 0,01_2, \text{ оскільки } 0,01_2 = 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2};$$

$$0,625_{10} = 0,101_2, \text{ оскільки } 0,101_2 = 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}.$$

З порівняння чисел, записаних в двійковій і десятковій системах числення видно, що для запису чисел в двійковій системі необхідно в 3–4 рази більше

цифр, чим для запису того ж числа в десятковій системі. Крім цієї незручності виникають труднощі і при читанні числа.

Щоб уникнути цих незручностей, в обчислювальній техніці використовують вісімкову і шістнадцятирічну системи. Перевага цих систем полягає в тому, що їх основою служать числа 8 і 16, які є точним ступенем числа 2 ($8 = 2^3$, $16 = 2^4$). Як ми побачимо трохи пізніше, і переклад чисел з десяткової у вісімкову систему проводиться достатньо легко.

Нижче в табл. 1.2 наводяться числа від 0 до 20, записані в двійковій, вісімковій, шістнадцятирічній і десятковій системах числення. Ця таблиця дозволяє провести порівняння десяткових чисел з їх відповідними (2, 8, 16) еквівалентами. Останнє число в третьому стовпці 24_8 (табл. 1.2) може бути представлено у вигляді $24_8 = 2 \cdot 8^1 + 4 \cdot 8^0 = 20_{10}$.

З табл. 1.2 видно перевагу вісімкової системи. Вона полягає в тому, що будь-який розряд числа, записаного у вісімковій системі, може бути представлений трьома розрядами числа, написаного в двійковій системі.

Якщо кількість розрядів числа, записаного в двійковій системі, не кратна трьом, то попереду воно доповнюється необхідною кількістю нулів (для цілого числа).

Наприклад, останнє число в третьому стовпці табл. 1.2 можна подати так: 010|100. Тут перші три розряди, приведені зліва (010), дають число 2, а три інші розряди (100) – дають число 4, або разом 24_8 , що дорівнює числу 20_{10} . Останнє число в другому стовпці табл. 1.2 може бути представлено у вигляді $14_{16} = 1 \cdot 16^1 + 4 \cdot 16^0 = 20_{10}$.

Аналогічну перевагу має і шістнадцятирічна система. Будь-який розряд числа, записаного в шістнадцятирічній системі, може бути представлений чотирма розрядами числа, написаного в двійковій системі.

Якщо кількість розрядів числа, записаного в двійковій системі, не кратна чотирьом, то попереду воно доповнюється необхідною кількістю нулів (для цілого числа). Наприклад число 100110 буде мати вид 0010|0110.

Таблиця 1.2 – Співвідношення чисел в десятковій, шістнадцятирічній, вісімковій і двійковій системах

Десяткова система	Шістнадцятирічна система	Вісімкова система	Двійкова система
0	0	0	0
1	1	1	1
2	2	2	10
3	3	3	11
4	4	4	100
5	5	5	101
6	6	6	110
7	7	7	111
8	8	10	1000
9	9	11	1001
10	A	12	1010
11	B	13	1011
12	C	14	1100
13	D	15	1101
14	E	16	1110
15	F	17	1111
16	10	20	10000
17	11	21	10001
18	12	22	10010
19	13	23	10011
20	14	24	10100

Наприклад, останнє число в третьому стовпці табл. 1.2 можна представити так: 0001|0100. Перші чотири розряди зліва (0001) дають число 1, а другі чотири розряди (0100) дають число 4, або разом 14. Це дорівнює числу 20_{10} , написаному в шістнадцятирічній системі.

У двійково-десятковій системі числення кожна десяткова цифра від 0 до 9 представляється чотирьохрозрядним двійковим еквівалентом, наприклад, $942_{10} = 1001\ 0100\ 0010_{2-10}$. Це дозволяє скоротити програмні і апаратні витрати в МП при виведенні на екран десяткових чисел.

Вісімкова і шістнадцятирічна системи числення дозволяють здійснювати запис числа, представленого в двійковій системі, в більш стислому вигляді.

1.2. Переклад чисел з однієї системи числення в іншу

Переклад цілих десяткових чисел в двійкову систему або переклад чисел з двійкової системи у вісімкову і назад є окремими випадками перекладу чисел з однієї системи в іншу. Існують загальні правила перекладу чисел з однієї системи числення в іншу (наприклад “метод ділення”) [1,6,7]. Розглянемо найбільш часто використовувані в обчислювальній техніці системи числень: двійкову, вісімкову, десяткову, шістнадцятирічну і двійково-десяткову.

Розглянемо приклад перекладу числа 49 з десяткової системи в двійкову форму (приведений нижче).

Для цього представимо число 49 у вигляді суми добутків числа 2 в ступені, помноженого відповідно на 1 або 0 [1]. Віднявши з 49 число $2^5 \cdot 1$, рівне 32, одержимо в залишку 17 і запишемо 1 в п'ятому розряді. Після другого віднімання з числа в залишку 17 числа $2^4 \cdot 1 = 16$ одержимо одиницю і запишемо 1 в четвертому розряді. Оскільки в залишку 1 не міститься ні $8 = 2^3$, ні $4 = 2^2$, ні $2 = 2^1$, ці числа можемо представити як добуток кожного з них на 0, записавши нулі у відповідних розрядах. Наступна одиниця стоїть в наймолодшому розряді двійкового числа: $2^0 = 1$.

Отже, $49_{10} = 110001_2$.

$\begin{array}{r} 49 \\ - 32 \\ \hline 17 \\ - 16 \\ \hline 1 \\ - 1 \\ \hline 0 \end{array}$	<table style="border-collapse: collapse;"> <tr> <td></td> <td style="text-align: center;">2^5</td> <td style="text-align: center;">2^4</td> <td style="text-align: center;">2^3</td> <td style="text-align: center;">2^2</td> <td style="text-align: center;">2^1</td> <td style="text-align: center;">2^0</td> </tr> <tr> <td style="text-align: right;">2^5</td> <td style="text-align: center;">— 1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> </tr> <tr> <td style="text-align: right;">2^4</td> <td colspan="2" style="border-top: 1px solid black; border-left: 1px solid black;"></td> <td></td> <td></td> <td></td> <td style="border-left: 1px solid black;"></td> </tr> <tr> <td style="text-align: right;">2^0</td> <td colspan="5" style="border-top: 1px solid black; border-left: 1px solid black;"></td> <td style="border-left: 1px solid black;"></td> </tr> </table>		2^5	2^4	2^3	2^2	2^1	2^0	2^5	— 1	1	0	0	0	1	2^4							2^0						
	2^5	2^4	2^3	2^2	2^1	2^0																							
2^5	— 1	1	0	0	0	1																							
2^4																													
2^0																													

Існує ще один метод перекладу цілих чисел з однієї системи в іншу – метод ділення. Згідно йому, десяткове число ділять на основу нової системи.

Якщо при цьому одержане число більше основи, ділення продовжують до отримання частки, меншої за основу. Частка від останнього ділення є старший розряд числа в новій системі. За ним йдуть залишки від попередніх ділень, починаючи з останнього залишку.

У математиці переклад числа з однієї системи в іншу для більшої наочності записують так:

$$\begin{array}{r} \underline{183 \text{ } | \text{ } 8} \\ \underline{176} \quad \underline{22 \text{ } | \text{ } 8} \\ 7 \quad \underline{16} \quad 2 \\ \quad \quad \quad 6 \end{array}$$

$$183_{10} = 267_8$$

$$\begin{array}{r} 183 \text{ } | \text{ } 16 \\ \underline{176} \quad 11 \\ 7 \end{array}$$

$$183_{10} = B7_{16}$$

Переклад десяткового числа в двійкову систему можна виконати за допомогою приведеного правила послідовного ділення на 2. Наприклад, число 11 можна записати так:

$$\begin{array}{r} \underline{11 \text{ } | \text{ } 2} \\ \underline{10} \quad \underline{5 \text{ } | \text{ } 2} \\ 1 \quad \underline{4 \text{ } | \text{ } 2} \\ \quad \quad \underline{2 \text{ } | \text{ } 2} \\ \quad \quad \quad \underline{1 \text{ } | \text{ } 2} \\ \quad \quad \quad \quad \underline{0} \end{array}$$

Отже, $11_{10} = 1011_2$.

Коли необхідно перевести велике число, зручніше спочатку перевести його у вісімкову систему і після цього кожен розряд вісімкового числа представити трьома розрядами двійкової системи. Як ми знаємо, $183_{10} = 267_8$.

Тоді $267_8 = 010110111_2$.

Якщо число в десятковій системі менше основи системи, в яку ми хочемо перевести число, то в написанні немає різниці:

$$1_{10} = 1_8 = 1_2 ; 6_{10} = 6_8.$$

Для перекладу правильного десяткового дробу (що містить тільки дробову частину без цілої частини) в іншу систему, необхідно послідовно

помножити цей дріб на основу нової системи до тих пір, поки дробова частина не стане рівна нулю. При цьому помножують тільки дробову частину. Дріб в новій системі складається з цілих частин добутоків, починаючи з першого добутку [1,6,7].

Наприклад, переклад правильного десяткового дробу 0,3125 у вісімкову систему проводиться так :

$$\begin{array}{r} \times 0,3125 \\ \quad \quad \quad \underline{8} \\ \times 2,5000 \\ \quad \quad \quad \underline{8} \\ \times 4,0000 \end{array}$$

Отже, $0,3125_{10} = 0,24_8$.

Здійснимо переклад в двійкову систему:

$$\begin{array}{r} \times 0,3125 \\ \quad \quad \quad \underline{2} \\ \times 0,6250 \\ \quad \quad \quad \underline{2} \\ \times 1,2500 \\ \quad \quad \quad \underline{2} \\ \times 0,5000 \\ \quad \quad \quad \underline{2} \\ \times 1,0000 \end{array}$$

Отже, $0,3125_{10} = 0,0101_2$.

Цього результату можна досягти швидше, якщо в числі $0,24_8$ кожен розряд представити трьома двійковими розрядами:

$$0,24_8 = 0,010100_2.$$

	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	0	1	0	1
× 0,3125				
× 0,25				
× 0,0625				
× 0,0625				
× 0,0000				
0,0000				

Таким чином, ми одержали той самий результат.

Зрозуміло, зліва від найбільшого ступеня числа 2 (якщо він менше, ніж 0,5) ми напишемо необхідні нулі. Для цього необхідно мати в своєму розпорядженні таблицю ступенів числа 2 з негативними показниками ступеня.

Якщо дріб неправильний, тобто, якщо число містить як цілу, так і дробову частини, то ці частини переводять в нову систему окремо за описаними вище правилами.

1.3. Арифметичні дії з двійковими числами

Складання. Для виконання складання використовується наступна таблиця:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

Останній рядок таблиці показує, що при складанні двох одиниць в результаті виходить одиниця в більш старшому розряді.

Застосуємо цю таблицю для додавання декількох пар двійкових чисел:

$$\begin{array}{r} + 10 \\ 1 \\ \hline 11 \end{array}$$

$$\begin{array}{r} + 101 \\ 10 \\ \hline 110 \end{array}$$

$$\begin{array}{r} + 100 \\ 10 \\ \hline 110 \end{array}$$

$$\begin{array}{r} + 101 \\ 100 \\ \hline 1001 \end{array}$$

Таким же чином проводиться складання трьох і більше двійкових чисел:

$$\begin{array}{r} + 100 \\ 11 \\ 10 \\ \hline 1001 \end{array}$$

$$\begin{array}{r} + 1100 \\ 11 \\ 110 \\ \hline 10101 \end{array}$$

Поки не буде добре засвоєне складання трьох і більше чисел, рекомендується проводити перевірку шляхом складання тих же чисел в

десятковій системі, чи ж складати спочатку два числа, потім до одержаної суми додавати третє число, потім – четверте і т.д.

Так само виконується складання в МП і ЕОМ, де для цієї мети використовують спеціальні пристрої, названі двійковими суматорами.

Віднімання. При відніманні використовують наступну таблицю:

$$\begin{aligned} 1 - 1 &= 0 \\ 0 - 0 &= 0 \\ 1 - 0 &= 1 \\ 10 - 1 &= 1 \end{aligned}$$

(останній рядок таблиці показує, що для віднімання 1 з 0 необхідно зайняти 1 з більш старшого розряду).

Розглянемо декілька прикладів:

$$\begin{array}{r} \underline{110} \\ \underline{10} \\ 100 \end{array} \quad \begin{array}{r} \underline{111} \\ \underline{110} \\ 1 \end{array} \quad \begin{array}{r} \underline{100} \\ \underline{11} \\ 1 \end{array} \quad \begin{array}{r} \underline{101} \\ \underline{11} \\ 10 \end{array} \quad \begin{array}{r} \underline{1001} \\ \underline{11} \\ 110 \end{array}$$

І тут бажано виконувати перевірки в десятковій системі. Для полегшення віднімання вводиться поняття доповнення. Необхідність цього поняття розглянемо в десятковій системі. Можна помітити, що в десятковій системі дане число легко віднімається з числа, яке утворене одиницею в старшому розряді і нулями в решті розрядів ($1 \cdot 10^n$), а число вигляду $1 \cdot 10^n$ також легко віднімається з інших чисел: $100 - 76 = 24$; $2498 - 1000 = 1498$.

Доповненням даного цілого десяткового числа називають інше ціле десяткове число, яке, будучи складеним з першим, дає число, що складається з одиниці в більш старшому розряді (в порівнянні із старшим розрядом даного числа) і нулів в решті розрядів.

Позначивши доповнення буквою d , одержимо для довільного числа (наприклад, для числа 45):

$$45 + d = 100; \quad d = 100 - 45.$$

Якщо нам необхідно відняти 45 з 95, ми поступимо так:

$$95 + 100 - 45 - 100 = 95 + d - 100 = 95 + 55 - 100 = 150 - 100 = 50.$$

Збільшення і віднімання одного і того ж числа (у нашому випадку 100) не змінює результату, а лише полегшує виконання дії.

Приведений приклад дозволяє вивести наступне *правило знаходження різниці з використанням доповнення*.

1. Визначити доповнення від'ємника.
2. Скласти одержане доповнення із зменшуваним.
3. З одержаної суми відняти число, що складається з одиниці в старшому (по відношенню до від'ємника) розряді і нулів в решті розрядів.

Аналогічно це виконується і в двійковій системі. Доповнення даного двійкового числа знаходять на практиці так: замінюють всі одиниці нулями, всі нулі – одиницями і до одержаного результату додають одиницю в наймолодшому розряді. Розглянемо це на прикладі двох двійкових чисел 11001 і 111. Від'ємником є число з меншою кількістю одиниць в старших розрядах, тобто 111. Доповненням до 111 є 1. Після визначення доповнення проводимо його складання із зменшуваним 11001, після чого з одержаного результату 11010 віднімаємо число, що складається з нулів і одиниці в більш старшому розряді по відношенню до числа, що віднімається, 1000.

$$\begin{array}{r} + 11001 \\ \quad 1 \\ \hline 11010 \end{array} \qquad \begin{array}{r} - 11010 \\ \quad 1000 \\ \hline 10010 \end{array}$$

Перевірку здійснюємо в десятковій системі: $25 - 7 = 18$.

Множення. Таблиця множення двійкових чисел, яка представлена нижче, має наступний вигляд:

$$\begin{array}{l} 0 \times 0 = 0 \\ 1 \times 0 = 0 \\ 0 \times 1 = 0 \\ 1 \times 1 = 1 \end{array}$$

Розглянемо декілька прикладів множення двійкових чисел:

$$\begin{array}{r} \times 10 \\ \underline{1} \\ 10 \end{array} \quad \begin{array}{r} \times 101 \\ \underline{1} \\ 101 \end{array} \quad \begin{array}{r} \times 110 \\ \underline{11} \\ 110 \\ 110 \\ \underline{10010} \end{array} \quad \begin{array}{r} \times 111 \\ \underline{10} \\ 000 \\ 111 \\ \underline{1110} \end{array}$$

$$\begin{array}{r} \times 11101 \\ \underline{101} \\ 11101 \\ 00000 \\ \underline{11101} \\ 10010001 \end{array} \quad \begin{array}{r} \times 1001 \\ \underline{111} \\ 1001 \\ 1001 \\ \underline{1001} \\ 111111 \end{array}$$

Зрозуміло, що і при множенні двох двійкових чисел ми могли б не писати цілий рядок нулів (коли є нуль у відповідному розряді множника), а відразу писати наступний рядок, зміщений на один розряд вліво. Множення можна описати наступним алгоритмом.

Крок 1. Дослідити молодший розряд множника:

а) якщо він рівний 1, то додати множене до результату і перейти до кроку 2;

б) якщо він рівний 0, перейти до кроку 2.

Крок 2. Змістити множене на один розряд вліво.

Крок 3. Дослідити наступний за старшинством розряд множника і виконати крок 1а або 1б.

Множення закінчується після того, як цей алгоритм буде застосований до самого старшого розряду множника.

Ділення. Так само, як множення може бути замінено багатократним складанням і зміщенням вліво, ділення може бути замінено багатократним відніманням і зміщенням вправо.

Наприклад, для того, щоб визначити, чому рівне $18:6$, потрібно кілька разів відняти 6 з 18. Число віднімань буде результатом ділення.

Розглянемо декілька прикладів ділення двійкових чисел.

$$\begin{array}{r} _ 110 : 10 = 11 \\ \underline{10} \\ _ 10 \\ \underline{10} \\ 0 \end{array} \qquad \begin{array}{r} _ 1100 : 11 = 100 \\ \underline{11} \\ 0 \end{array}$$

$$\begin{array}{r} _ 11100 : 100 = 111 \\ \underline{100} \\ _ 110 \\ \underline{100} \\ _ 100 \\ \underline{100} \\ 0 \end{array} \qquad \begin{array}{r} _ 100011 : 101 = 111 \\ \underline{101} \\ _ 111 \\ \underline{101} \\ _ 101 \\ \underline{101} \\ 0 \end{array}$$

В цьому випадку також корисно виконати перевірку в десятковій системі.

Слід зазначити, що коли ділення замінюється відніманням і зміщенням вправо в МП і ЕОМ, то для полегшення знаходження різниці використовують доповнення.

1.4. Одиниці інформації

Найменша одиниця інформації – 1 біт. Ця інформація виходить при описі події, яка має два рівноімовірні результати, наприклад підкидання монети. Оскільки ця одиниця мала, то використовують інші одиниці інформації:

$$1 \text{ байт} = 8 \text{ біт}; \quad 1 \text{ кілобіт} = 1000 \text{ біт};$$

$$1 \text{ кілобайт} = 1000 \text{ байт} = 8000 \text{ біт};$$

$$1 \text{ мегабайт} = 10^6 \text{ байт} = 8 \cdot 10^6 \text{ біт};$$

$$1 \text{ гігабайт} = 10^9 \text{ байт} = 8 \cdot 10^9 \text{ біт}.$$

З появою напівпровідникових ЗП, в яких число комірок рівне деякому степеню числа 2, почалося використання нової кратної одиниці інформації, яка позначається буквою К:

$$1 \text{ Кбіт} = 1024 \text{ біт}$$

$$1 \text{ Кбайт} = 1024 \text{ байт}$$

(з одного боку, 1 Кбіт близько до 1 кілобіт, а з іншого боку $1 \text{ Кбіт} = 2^{10} \text{ біт}$).

1.5. Логічні основи і алгебра логіки

Логічні основи МП і мікроЕОМ (мікроконтролерів) будуються на принципах математичної логіки. *Математична логіка* – наука яка не змінює основного змісту логіки, як науки про форми і закони мислення, але дозволяє точніше і вірно вирішувати задачі. Розглянемо лише один з розділів математичної логіки – алгебру логіки, яка в основних своїх рисах була створена англійським математиком Джорджем Булем [1,7].

Алгебра логіки оперує з двійковими змінними, тобто з такими змінними, які можуть приймати тільки одне з двох можливих значень. Іншими словами, наші вирази, незалежно від їх змісту, розглядаються тільки з погляду істинності: вірно або невірно, істинно або помилково.

Заперечення. Заперечення виразу одержимо, якщо використовуємо частинку НІ або вираз "невірно, що". Для позначення заперечення над прописною буквою ставиться риска (в цьому випадку читаємо: А з рискою або не А). Дія логічного заперечення виражається наступною таблицею:

A	\bar{A}
1	0
0	1

Або словами: заперечуючи вірний вираз, ми говоримо не правду; заперечуючи невірний вираз, ми говоримо істину.

Написавши над А дві риски, ми одержимо подвійне заперечення $\bar{\bar{A}} = A$.

Це правило відоме в логіці як *закон подвійного заперечення*. Заперечення 1 рівне 0 і навпаки.

Множення (кон'юнкція). Множення двох виразів зображається за допомогою союзу І. Результат множення називають *логічним добутком*. Найважливішим є питання про істинність логічного добутку і про те, як воно залежить від істинності окремих виразів. Незалежно від змісту, логічний

добуток є істинним тільки в тому випадку, якщо істинні складові частини добутку (окремі вирази).

Таблицю, по якій визначають істинність логічного добутку залежно від істинності окремих виразів, називають *таблицею істинності*. Вона має наступний вигляд:

A	B	AB
1	1	1
1	0	0
0	1	0
0	0	0

Таку ж таблицю істинності ми одержали б, якби написали спочатку B , а потім A . Отже, $AB = BA$.

Крім цього, $A \cdot A = A$; $A \cdot 1 = A$; $A \cdot 0 = 0$.

У істинності цих трьох рівностей (точніше тотожностей) ми переконаємося, якщо задаватимемо для A значення 0 або 1. Наприклад, першу рівність можна виразити так: якщо логічно помножити деякий вираз сам на себе, то добуток матиме значення цього виразу. І, дійсно, з таблиці істинності логічного твору можна бачити, що $1 \cdot 1 = 1$; $0 \cdot 0 = 0$.

Подібним же чином, задаючи для A значення 0 або 1, перевіримо і другу рівність: $0 \cdot 1 = 0$; $1 \cdot 1 = 1$.

Для третьої рівності маємо $1 \cdot 0 = 0$; $0 \cdot 0 = 0$.

Складання (диз'юнкція). Складання двох виразів виражається за допомогою союзу АБО. Результат складання називають *логічною сумою*.

І в цьому випадку найважливішим є питання про істинність логічної суми і її залежності від істинності окремих виразів.

Незалежно від змісту виразів, для істинності логічної суми достатньо, щоб хоч би один з виразів, був істинний.

Логічна сума допускає одночасну істинність двох виразів, оскільки її істинність визначається наступною таблицею:

A	B	$A+B$
1	1	1
1	0	1
0	1	1
0	0	0

Таблиця істинності не змінилася б, якби ми поміняли місцями A і B , тобто $A + B = B + A$. Крім того, $A + A = A$; $A + 1 = 1$; $A + 0 = A$.

Три приведених вище рівності (тотожності) можна перевірити, задаючи для A значення 0 або 1. Для першого можемо написати $0 + 0 = 0$; $1 + 1 = 1$. Або словами: якщо підсумовувати логічний вираз сам з собою, сума має значення цього виразу.

Аналогічним способом перевіримо другу тотожність: $0 + 1 = 1$; $1 + 1 = 1$.

Для третьої тотожності: $0 + 0 = 0$; $1 + 0 = 1$.

Складні вирази можуть містити більш ніж два прості вирази і можуть бути результатом більш ніж однієї логічної дії:

$$x = AB + \bar{C};$$

$$x = \overline{(A+B)C}$$

Таблиці істинності складних виразів одержують шляхом послідовного виконання вказаних дій.

Наприклад, для складного виразу, який має вигляд

$$x = \overline{AB + C},$$

одержані результати приведемо в табл. 1.3.

Є складні вирази, чия істинність постійна і не залежить від істинності вхідних в них простих виразів. Такі вирази називають *тотожними*, наприклад вираз, значення якого завжди рівне 0 (ніколи не істинно). У логіці цей закон носить назву *закону суперечності*.

Таблиця 1.3 – Результати логічних операцій функції $x = \overline{AB + C}$

A	B	C	AB	\overline{C}	$AB + \overline{C}$	$\overline{AB + \overline{C}}$
1	1	1	1	0	1	0
1	1	0	1	1	1	0
1	0	1	0	0	0	1
1	0	0	0	1	1	0
0	1	1	0	0	0	1
0	1	0	0	1	1	0
0	0	1	0	0	0	1
0	0	0	0	1	1	0

Вираз $x = A + \overline{A}$ також тотожний і має значення, рівне 1 (завжди істинно). Це відомий в логіці закон виключеного третього.

Для логічної суми справедливо наступна рівність:

$$A + (B + C) = (A + B) + C.$$

Для логічного добутку справедлива наступна рівність:

$$A(BC) = (AB)C; \quad A \cdot (B+C) = AB+AC.$$

Особливої уваги заслуговують два наступні складні вирази:

$$A + AB = A(1 + B) = A \cdot 1 = A; \quad AB + A\overline{B} = A(B + \overline{B}) = A \cdot 1 = A$$

Ці вирази є запис закону подвійності, який свідчить: якщо у формулі деякого складного виразу замінити всі знаки множення знаками складання і навпаки, одержимо вираз, еквівалентний первинному. Наприклад, якщо у виразу $AB + A\overline{B} = A$ замінити в лівій частині знаки, то одержимо $(A+B) \cdot (A+\overline{B}) = A$.

Заперечення логічної суми і логічного множення підкоряються формулі де Моргана: $\overline{A+B} = \overline{A}\overline{B}$. Дві приведені вище формули можна узагальнити на випадок більше двох виразів: $\overline{A+B+C+\dots+N} = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \dots \cdot \overline{N}$;
 $\overline{A \cdot B \cdot C \cdot \dots \cdot N} = \overline{A} + \overline{B} + \overline{C} + \dots + \overline{N}$.

Використання властивості складання виразів дає можливість

перетворювати ці вирази в простіші.

Розглянемо вираз $x = A + AB$. Перетворимо його таким чином:

$$x = A + AB; \quad x = A (I + B).$$

Але $I + B = I$, і тоді $x = A I = A$. Отже, $A + AB = A$. Ця операція, пов'язана із зникненням простого виразу B , називається *поглинанням*.

Застосовуючи закон подвійності, можемо записати $A (A + B) = A$.

Інша операція, пов'язана із спрощенням логічних формул, називається *злиттям*: $AB + A\bar{B} = A(B + \bar{B}) = A$, де $B + \bar{B} = 1$

Це перетворення називають злиттям по B . І тут можна застосувати закон подвійності: $(A + B) \cdot (A + \bar{B}) = A$.

Нижче приводяться розглянуті нами залежності Булевої алгебри в тому вигляді, як вони використовуються в спеціальній літературі.

Постулати:

$$\begin{array}{lll} 0 \cdot 0 = 0 & 0 + 0 = 0 & 0 = \bar{1} \\ 0 \cdot 1 = 0 & 0 + 1 = 1 & 1 = \bar{0} \\ 1 \cdot 0 = 0 & 1 + 0 = 1 & \\ 1 \cdot 1 = 1 & 1 + 1 = 1 & \end{array}$$

Теорема:

$$\begin{array}{lll} A \cdot 0 = 0 & A \cdot \bar{A} = 0 & A + A = A \\ 0 \cdot A = 0 & A + 0 = A & A + \bar{A} = 1 \\ A \cdot 1 = A & 0 + A = A & \bar{\bar{A}} = A \\ 1 \cdot A = A & A + 1 = 1 & \\ A \cdot A = A & 1 + A = 1 & \end{array}$$

Закони:

$$\begin{array}{lll} \text{Закон ідентичності} & A = A & \bar{\bar{A}} = A \\ \text{Закон комутативності} & AB = BA & A+B=B+A. \end{array}$$

Закон асоціативності	$A(BC)=ABC$	$A+(B+C)=A+B+C.$
Закон ідемпотентності	$AA = A$	$A+A=A.$
Закон дистрибутивності	$A (B+C) =AB+AC$	$A+BC=(A+B)(A+C).$
Закон поглинання	$A + AB= A$	$A (A + B) = A.$
Закон злиття	$AB + \bar{A}B = A$	$(A + B) \cdot (A + \bar{B}) = A .$
Закони де Моргана	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}.$

Деякі тотожності:

$$A(\bar{A} + B) = AB$$

$$A + \bar{A}B = A + B$$

$$(\bar{A}B)(A + B) = \bar{A}B$$

$$\overline{\bar{A}B + \bar{A}B} = AB + \bar{A}B$$

$$(A + B)(B + C)(A + C) = AB + BC + AC$$

$$(A + B) \cdot (\bar{A} + C) = AC + \bar{A}B$$

$$AC + AB + \bar{B}C = AC + \bar{B}C$$

$$(A + B) \cdot (B + C) \cdot (\bar{A} + C) = (A + B) \cdot (\bar{A} + C)$$

Операції І і АБО по суті є двійковими функціями, в яких число змінних як, мінімум, рівно двом. Можна довести, що з n двійкових змінних можна скласти 2^{2^n} функцій. Якщо приймемо, що $n = 2$, то одержимо $2^{2^2} = 2^4 = 16$ функцій двох двійкових змінних. У табл.1.4 приведені значення істинності цих функцій, а під таблицею дано їх опис. У дужках приведені англійські назви найпоширеніших функцій.

Слід особливо відзначити, що всі двійкові функції можуть виражатися одна через іншу.

Якщо узяти, для прикладу, розглянуті функції НІ, І і АБО, то можна стверджувати, що всі функції таблиці 1.4 можуть бути виражені за допомогою двох з цих трьох функцій, якщо одна з них буде НІ. Але функції F_8 і F_{14} точно відповідають цій умові. Отже, за допомогою будь-якої з них можна виразити всі

інші двійкові функції.

Таблиця 1.4 – Значення 16-ти функцій двох двійкових змінних X і Y

X	Y	F ₀	F ₁	F ₂	F ₃	F ₄	F ₅	F ₆	F ₇	F ₈	F ₉	F ₁₀	F ₁₁	F ₁₂	F ₁₃	F ₁₄	F ₁₅
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Символ оператора				/		/		⊕	+	↓	⊙	-	⊂	-	⊃	↑	

$$F_0 = 0$$

Функція еквівалентна 0

$$F_1 = xy$$

I (AND)

x і y

$$F_2 = x\bar{y}$$

ЗАБОРОНА

x, але не y

$$F_3 = x$$

Функція еквівалентна x

$$F_4 = \bar{x}y$$

ЗАБОРОНА

y, але не x

$$F_5 = y$$

Функція еквівалентна y

$$F_6 = \bar{x}y + x\bar{y}; x \oplus y$$

ВИКЛЮЧАЮЧЕ АБО (XOR)

x або y, але не обидва

$$F_7 = x + y$$

АБО (OR)

x АБО y

$$F_8 = \overline{(x + y)} = x \downarrow y$$

АБО-НІ (NOR)

Інвертоване АБО

$$F_9 = \overline{xy} + xy = x \odot y$$

ЕКВІВАЛЕНТНІСТЬ

x еквівалентне y

$$F_{10} = \bar{y}$$

ІНВЕРСІЯ

НЕ y

$$F_{11} = x + \bar{y} = x \subset y$$

ІМПЛІКАЦІЯ

Якщо y, то x

$$F_{12} = \bar{x}$$

ІНВЕРСІЯ

НЕ x

$$F_{13} = \bar{x} + y = x \supset y$$

ІМПЛІКАЦІЯ

Якщо x, то y

$$F_{14} = \overline{(xy)} = x \uparrow y$$

I-НІ (NAND)

Інвертоване I

$$F_{15} = 1$$

Функція еквівалентна 1

1.6. Способи представлення складних функцій

Коли складний вираз складається з суми добутків простих виразів (або їх заперечень), ми говоримо, що складний вираз написаний в *диз'юнктивній нормальній формі* (ДНФ).

Наприклад:

$$x = AB + A\bar{B}; \quad x = ABC\bar{C} + A\bar{B}C + ABC; \quad x = ABC + \bar{A}\bar{B} + A\bar{B}\bar{C}.$$

Третій складний вираз відрізняється від першого і другого тим, що в його другому доданку відсутнє C (або \bar{C}).

Якщо в кожному доданку (доданок – це добуток простих виразів) беруть участь всі прості вирази або їх заперечення, ми говоримо, що складний вираз написаний в *досконалій диз'юнктивній нормальній формі* (ДДНФ).

Для математичної логіки найбільший інтерес представляють (головним чином з погляду мінімізації) складні вирази, які написані в ДДНФ. Тому ми розглянемо спосіб переходу від ДНФ до ДДНФ.

Наприклад, якщо дано вираз, то перехід до ДДНФ проводиться шляхом множення першого доданку на $C + \bar{C} = 1$

$$x = A\bar{B}(C + \bar{C}) + ABC\bar{C}; \quad x = A\bar{B}C + A\bar{B}\bar{C} + ABC\bar{C}.$$

Коли складний вираз складається з добутків сум, в які входять прості вирази (або їх заперечення), ми говоримо, що воно написане в *кон'юнктивній нормальній формі* (КНФ). Наприклад:

$$x = (A + B) \cdot (\bar{B} + C); \quad x = (A + \bar{B}) \cdot (\bar{A} + B).$$

І тут під *досконалою кон'юнктивною нормальною формою* (ДКНФ) розуміють складний вираз у формі добутку, складеного з сум, в кожну з яких входять всі прості вирази (або їх заперечення).

Наприклад:

$$x = (A + B + \bar{C}) \cdot (A + \bar{B} + C) \cdot (\bar{A} + \bar{B} + C).$$

Перехід від КНФ до ДКНФ проводиться таким чином. Нехай $x = (A + B)\bar{A}$.
Множимо на A , тоді

$$x = (A + B) \cdot \bar{A} \cdot (B + \bar{B}) = (A + B)(\bar{A}B + \bar{A}\bar{B}).$$

На підставі закону подвійності запишемо

$$\bar{A}B + \bar{A}\bar{B} = (\bar{A} + B) \cdot (\bar{A} + \bar{B}).$$

Внаслідок чого одержимо $(A + \bar{B})\bar{A} = (A + B) \cdot (\bar{A} + B) \cdot (\bar{A} + \bar{B})$.

1.7. Мінімізація складних виразів

Досконалі диз'юнктивна і кон'юнктивна форми використовуються для мінімізації складних логічних виразів. Мінімізація - це таке перетворення даної формули, при якому нова формула містить менше число букв.

Розглянемо приклад $x = AB\bar{C} + ABC$. У цьому складному виразі можна винести AB за дужки, тобто провести злиття по букві C :

$$x = AB \cdot (\bar{C} + C); \quad x = AB.$$

Для складного виразу $x = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC$ можна виконати злиття

по C (для першого і другого добутоків) і по A (для третього і четвертого добутоків). Тоді одержимо

$$x = AB(\bar{C} + C) + BC(\bar{A} + A); \quad x = \bar{A}\bar{B} + BC.$$

1.8. Фізичні основи реалізації логічних функцій

Фізична реалізація математичних і логічних операцій може бути здійснена за допомогою самих різних елементів: механічних, пневматичних, гідравлічних, електричних, електронних і ін. Розглянемо приклад реалізації логічних функцій на основі електричних елементів: батареї, ключів (вимикачів) і лампочок [1]. Свічення лампочки буде доказом правильності виконання даної логічної операції.

1. Операція логічного складання (АБО) – лампа горить, коли включений тільки один з двох ключів (A або B) (рис. 1.1).

2. Операція логічного множення (І) – лампа горить тільки тоді, коли

включені обидва ключі (A і B) (рис. 1.2).

3. Операція логічного заперечення (НЕ) – лампа горить, коли не включений ключ (рис. 1.3). При включенні ключа лампа гасне, оскільки клема батареї при цьому замикаються накоротко. При використанні цієї схеми для демонстрації логічного заперечення, ключ слід тримати включеним дуже короткий час, щоб не пошкодити батарею.

4. Операція заперечення логічного підсумовування (АБО-НЕ) – лампа горить, якщо НЕ включений один з ключів A АБО B (рис. 1.4).

5. Операція заперечення логічного множення (І-НЕ) – лампа горить, якщо НЕ включені одночасно обидва ключі A і B (рис. 1.5).

Механічні і електричні реалізації вимагають втручання людини. Кроком на шляху до автоматизації пристроїв, що виконують логічні функції, з'явилися електромагнітні реле, що мають контакти трьох видів: 1) нормально розімкнені, які замикаються при протіканні струму через обмотку;

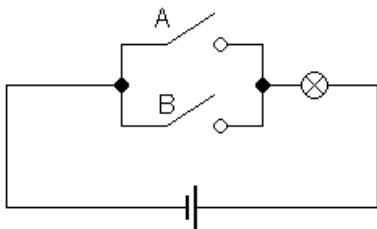


Рисунок 1.1 – Електрична реалізація логічної операції АБО

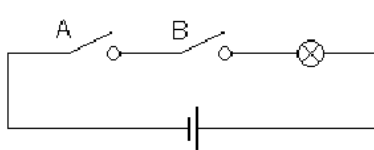


Рисунок 1.2 – Електрична реалізація логічної операції І

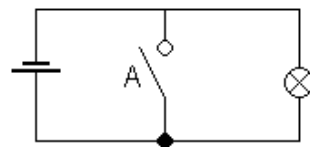


Рисунок 1.3 – Електрична реалізація логічної операції НЕ

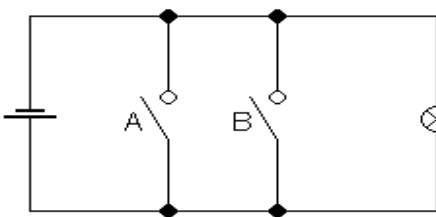


Рисунок 1.4 – Електрична реалізація логічної операції АБО-НЕ

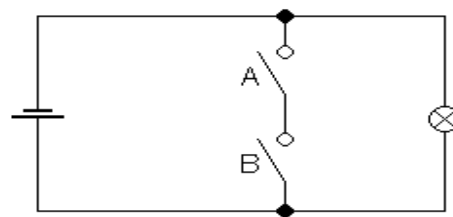


Рисунок 1.5 – Електрична реалізація логічної операції І-НЕ

2) нормально замкнуті, які розмикаються при протіканні струму через обмотку;
3) перемикальні, які складаються з пари нормально розімкнених і пари

нормально замкнених контактів.

У 1944 р. в Гарвардському університеті (США) вперше була створена релейна обчислювальна машина з роздільною пам'яттю даних і програм. Таке розділення пам'яті даних і програм одержало назву “Гарвардська архітектура”, яка в даний час широко використовується в сучасних МПС і МК, оскільки, дозволяє значно підвищити їх швидкодію.

Наступним кроком в автоматизації реалізації логічних функцій з'явилися електронні логічні елементи.

1.9. Логічні схеми з напівпровідниковими елементами

Перші електронні логічні елементи були побудовані на електронних лампах. Тут ми розглянемо тільки схеми з напівпровідниковими приладами, оскільки в сучасній обчислювальній техніці застосовуються виключно напівпровідникові логічні схеми.

Із-за стрибкоподібної зміни вхідних логічних сигналів напівпровідникові елементи працюють в режимі перемикачів, при якому опір електричного ланцюга змінюється від нуля до безкінечності. Строго кажучи, таких перемикальних елементів взагалі не існує, але реальні елементи більшою мірою наближаються до ідеальних. За допомогою діодів можна реалізувати логічні схеми І і АБО. Перевагою діодно-резисторних логічних схем є їх простота, а недоліком – ослаблення сигналу на виході, тобто те, що амплітуда вихідного сигналу стає менше амплітуди вхідного. Із-за ослаблення сигналу не прийнято сполучати послідовно більше двох діодно-резисторних логічних схем.

Транзистори є найважливішими елементами в схемах сучасних обчислювальних машин. У транзисторних логічних схемах вхідні сигнали також змінюються стрибкоподібно між двома рівнями, один з яких приймають за логічний 0, а інший – за логічну 1. Транзистори звичайно працюють в так званому ключовому режимі, який характеризується тим, що транзистор або повністю замкнений (режим відсічення), або повністю відкритий (режим

насичення). Опір ділянки “емітер – колектор” змінюється від доль ома (відкритий) до десятків і сотень кілоом (замкнутий). У схемах із загальним емітером, транзистор закривається шляхом подачі на його базу сигналів відповідної полярності (негативної для *n-p-n-транзисторів*).

Найбільше застосування в сучасній обчислювальній техніці транзистори знаходять не як окремі дискретні елементи, а як частини інтегральних схем.

Розробка багатоемітерного транзистора створила основу для побудови елементів транзисторно-транзисторної логіки (ТТЛ) і розвитку технології інтегральних схем останніми роками.

Оскільки електричні схеми не застосовуються для вирішення логічних завдань в МП, а застосовуються інтегральні мікросхеми, то для зображення логічних елементів зручно користуватися умовними позначеннями, приведеними на рис. 1.6.

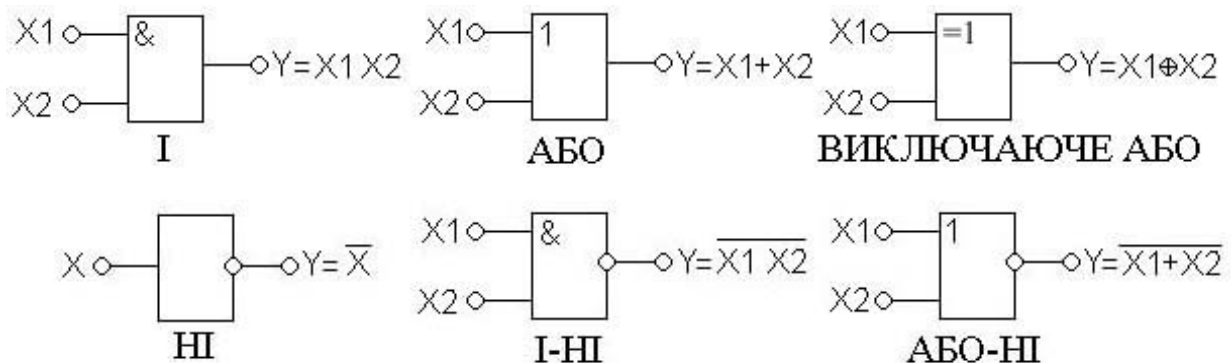


Рисунок 1.6 – Умовні позначення логічних схем

Використовуючи закони де Моргана, можна визначити еквівалентність деяких позначень вказаних на рис. 1.6. Якщо замкнути накоротко входи елементів I-НІ або АБО-НІ, то вони будуть виконувати функцію НІ. В [1] показано, що за допомогою елемента I-НІ можна реалізувати всі функції двох змінних. Логічні елементи входять до складу великих інтегральних схем і використовуються в арифметично-логічних пристроях, МП для виконання логічних операцій.

Контрольні питання і завдання

Варіанти завдань до даного розділу приведені в табл. 1.5.

Таблиця 1.5 – Варіанти даних для перекладу десяткових чисел в різні (2, 8, 16, 2–10) системи числення і виконання арифметичних і логічних операцій уручну і програмним способом

№ варіанту	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1-е число	171	172	173	174	175	180	181	182	183	184	185	190	191	192	193	194	195	196	197	198
2-е число	160	161	159	158	163	164	165	167	169	152	153	154	155	145	143	144	140	139	138	136

Арифметичні основи

1. Які системи числення ви знаєте і на чому вони засновані ?
2. Переведіть числа, приведені в табл. 1.5 (вибравши один з варіантів) в 2-, 8- і 16-річну системи числення.
3. Складіть і відніміть в двійковій системі два числа, вибрані з табл. 1.5 і перевірте результат в десятковій системі.
4. Які одиниці інформації використовуються в МП і з чим це зв'язано?
5. Проведіть віднімання двох чисел в додатковому коді і перевірте результат в десятковій системі.
6. Виконайте множення і ділення двох чисел (табл. 1.5 №4) в двійковій системі і перевірте результат в десятковій системі.
7. Викладіть правило знаходження різниці з використанням доповнення і приведіть приклади для двох чисел (табл. 1.5).
8. Представте двійкові числа в 8- і 16-річних системах і поясніть, для чого використовується таке представлення двійкових чисел.
9. Що означають біт і байт в двійковій системі числення?
10. Викладіть правило перетворення дробових чисел з десяткової системи в двійкову форму і приведіть приклади.
11. Переведіть число 1942_{10} в двійково-десяткову систему.

12. Переведіть число $1000\ 0011\ 0111\ 0101_{2-10}$ з двійково-десятькової в десяткову систему числення.

13. Назвіть перевагу і особливості двійково-десятькової системи.

14. Перетворіть в двійковий еквівалент числа: $1A_{16}$, $2B_{16}$, $5F_{16}$, FF_{16} , 29_8 , 123_8 , 1342_8 .

15. Переведіть число $10100011\ 01110101_2$ з двійкової в шістнадцятиричну систему числення.

16. Переведіть число $11111111\ 11111111_2$ з двійкової в шістнадцятиричну систему числення.

Логічні основи

1. На яких принципах будуються логічні основи МП?

2. Назвіть операції алгебри логіки.

3. Назвіть основні логічні функції.

4. Приведіть таблиці істинності основних логічних функцій.

5. Назвіть залежності Булевої алгебри: постулати, теореми, закони, тотожності.

6. Назвіть і визначте значення всіх 16-ти функцій двох двійкових змінних.

7. Викладіть суть мінімізації складних логічних виразів і приведіть приклади.

8. Проведіть логічні операції АБО, НЕ, І, ВИКЛЮЧАЮЧЕ АБО з двома числами, приведеними в табл. 1.5 для одного з варіантів.

9. Провести мінімізацію функції $x = ABC + AB\bar{C} + \bar{A}BC + \bar{A}\bar{B}\bar{C}$.

10. Побудуйте структурну схему з логічних елементів для виразу $x = (ABC + AB\bar{C} + \bar{A}BC + \bar{A}\bar{B}\bar{C}) \vee (A \wedge B)$.

Фізичні основи

1. Назвіть основні логічні елементи і вкажіть на особливості їх фізичної реалізації.

2. Приведіть приклади реалізації логічних функцій в електричних схемах.

3. Викладіть особливості реалізації логічних функцій за допомогою напівпровідникових елементів і інтегральних схем.

4. Назвіть існуючі технології виконання інтегральних схем і їх особливості.
5. Приведіть умовні позначення логічних схем І, АБО, НЕ, АБО-НЕ, І-НЕ і вкажіть, в яких пристроях МП вони використовуються.
6. Складіть структурну схему для логічного виразу і проведіть її мінімізацію: $x = (ABC + ABC\bar{C} + \bar{A}BC + \bar{A}\bar{B}\bar{C})VABC$.
7. Викладіть особливості великих і надвеликих інтегральних схем, і вкажіть їх перевагу перед інтегральними схемами.
8. Викладіть суть транзисторно-транзисторної логіки (ТТЛ), яка використовується в інтегральних мікросхемах.
9. Приведіть особливості логіки ТТЛШ.
10. Назвіть фізичні характеристики логічних елементів.
11. Поясніть суть перешкодостійкості, швидкодії і здатності навантаження, ВІС і НВІС, і вкажіть шляхи їх підвищення.
12. Вкажіть чинники, що впливають на перешкодостійкість ВІС і вкажіть способи захисту ВІС від перешкод.
13. Приведіть чинники, що впливають на здатність навантаження ВІС і приведіть приклади та способи її підвищення.
14. Назвіть основні характеристики, по яких здійснюється вибір ВІС і приведіть практичні приклади їх вибору для різних умов.
15. Вкажіть чинники, що впливають на швидкодію ВІС, і приведіть приклади для різних технологій їх виготовлення.
16. Викладіть особливості нанотехнологій і характеристики надвеликих інтегральних схем, одержаних на їх основі.
17. Які логічні елементи і для чого використовуються в арифметико-логічному пристрої МП?
18. В яких пристроях МП використовуються логічні елементи?
19. За допомогою якого логічного елемента можна реалізувати всі логічні функції двох змінних?

2. АРХІТЕКТУРА МІКРОПРОЦЕСОРІВ

Під *архітектурою мікропроцесора* мають на увазі складові частини мікропроцесора, а також їх взаємне з'єднання і взаємодію між ними. Архітектура включає: 1) структурну схему самого МП; 2) програмну модель МП (опис функцій регістрів); 3) інформацію про організацію пам'яті (місткість пам'яті і способи її адресації); 4) опис організації процедур вводу-виводу і управління; 5) опис системи команд і ін.

Існують два основних типи архітектури – *фоннейманівська* і *гарвардська*. *Фоннейманівську архітектуру* запропонував в 1945 р. американський математик Джо фон Нейман. Особливістю цієї архітектури є те, що програма і дані знаходяться в загальній пам'яті, доступ до яких здійснюється по одній шині даних і команд. Прикладом такої архітектури є класичний мікропроцесор КР580 (І 8080). *Гарвардська архітектура* вперше реалізована в 1944 р. в релейній обчислювальній машині Гарвардського університету (США). Особливістю цієї архітектури є те, що пам'ять даних і пам'ять програм розділені і мають окремі шини даних і шини команд, що дозволяє збільшити швидкість МП системи за рахунок можливості одночасного звернення по цих двох шинах до пам'яті програм і пам'яті даних. Прикладом такої архітектури є мікроконтролер К1816ВЕ51 і всі представники сімейства МК51 та інші сучасні МК.

2.1. Типова структура мікропроцесора

Розглянемо особливості організації процесу обробки інформації в цифрових пристроях (цифрових автоматах).

Завдання створення цифрового автомата, що виконує певні дії над двійковими сигналами, полягає у виборі елементів і способі їх з'єднання, що забезпечує задане функціональне перетворення. Ці завдання вирішують за допомогою математичної логіки або алгебри логіки.

Пристрої, що формують функції алгебри логіки, називають *логічними, або цифровими* і класифікують за різними відмітними ознаками.

По схемному рішенню і характеру зв'язку між вхідними і вихідними змінними з урахуванням їх зміни по тактах роботи, розрізняють два типи цифрових пристроїв – *комбінаційні і цифрові*.

У *комбінаційних цифрових пристроях* сукупність сигналів на виходах в кожен конкретний момент часу повністю визначається вхідними сигналами, що діють у цей момент на його входах. Алгоритм функціонування комбінаційних пристроїв може бути представлений у вигляді таблиці відповідності, що містить значення вихідних сигналів для всіх можливих комбінацій значень вхідних сигналів.

Цифрові пристрої послідовного типу істотно відрізняються від комбінаційних, перш за все, наявністю пам'яті. Їх вхідні сигнали є функцією не тільки вхідних сигналів, але і внутрішнього стану, в якому пристрій знаходився до надходження вхідних сигналів.

На основі цифрового пристрою послідовного типу може бути спроектовано пристрій, який залежно від послідовності вхідних сигналів виконуватиме один з багатьох алгоритмів. Ці вхідні сигнали можуть розміщуватися і послідовно витягуватися із зовнішнього блоку регістрів, названого керуючою пам'яттю. Деякі вихідні сигнали можуть використовуватися для синхронізації надходження вхідних сигналів з пам'яті, що управляє, і для їх адресації. Такий пристрій може бути названий пристроєм з програмованою логікою, або програмованим пристроєм. До таких пристроїв відноситься і мікропроцесор.

Архітектура мікропроцесора є логічна організація, що визначає можливості апаратної або програмної реалізації функцій, необхідних для побудови мікроЕОМ.

Мікропроцесори визначаються наступними характеристиками: розрядність адреси і даних, тип корпусу, кількість джерел живлення, потужність розсіяння, температурний діапазон, можливість розширення

розрядності, час циклу виконання команд (мікрокоманд), рівні сигналів, перешкодостійкість, здатність навантаження, об'єднання сигналів на виходах, надійність і т.ін.

По числу ВІС в комплекті (МПК) розрізняють однокристальні, багатокристальні, багатокристальні секційні мікропроцесори [8].

Однокристальні мікропроцесори утворюються при реалізації всіх апаратних засобів процесора в одній ВІС. У міру збільшення ступеня інтеграції елементів в кристалі і числа виведень корпусу, параметри однокристальних мікропроцесорів поліпшуються. Проте можливості однокристальних мікропроцесорів обмежені апаратними ресурсами кристала і корпусу. Тому поширеніші багатокристальні і багатокристальні секційні мікропроцесори.

Багатокристальні мікропроцесори виходять при розбитті його логічної структури на функціонально закінчені частини, які реалізують у вигляді ВІС. Функціональна закінченість ВІС багатокристального мікропроцесора означає, що його частини виконують наперед певні функції і можуть працювати автономно, а для побудови розвиненого процесора не вимагається організації великої кількості нових зв'язків і яких-небудь інших інтегральних схем (ІС).

Одним з можливих варіантів розбиття структури процесора є створення трьохкристального мікропроцесора, що містить ВІС операційного процесора, процесора, що управляє, і інтерфейсного процесора. Операційний процесор (ОП) служить для обробки даних, процесор, що управляє (УП), виконує функції вибірки, декодування і обчислення адрес операндів, а також генерує послідовності мікрокоманд. Автономність роботи і велика швидкодія ВІС дозволяють вибирати команди з пам'яті з більшою швидкістю, ніж ВІС ОП. При цьому в УП утворюється черга ще не виконаних команд, наперед готуються ті дані, які будуть потрібні ОП в наступних циклах роботи. Така випереджаюча вибірка команд економить час ОП на очікування операндів, необхідних для виконання команд програм. Інтерфейсний процесор (ІП) дозволяє підключити пам'ять і периферійні засоби до мікропроцесора. Велика інтегральна схема ІП виконує також функції каналу прямого доступу до пам'яті.

Вибрані з пам'яті команди розпізнаються і виконуються кожною частиною мікропроцесора автономно, і тому може бути забезпечений режим одночасної роботи всіх ВІС МП, тобто конвейерний *потоківий режим виконання послідовності команд програми* (виконання послідовності з невеликим зміщенням у часі). Такий режим роботи значно підвищує продуктивність мікропроцесора.

Багатокристалні секційні мікропроцесори виходять у тому випадку, коли у вигляді ВІС реалізуються частини (секції) логічної структури процесора. Мікропроцесорна секція – це ВІС, призначена для обробки декількох розрядів даних або виконання певних керуючих операцій. Секційність ВІС МП визначає можливість «нарощування» розрядності оброблюваних даних або ускладнення пристроїв управління мікропроцесором при «паралельному» включенні більшого числа ВІС. Багатокристалні секційні мікропроцесори мають розрядність від 2–4 до 8–16–32–64 біт і дозволяють створювати високопродуктивні процесори ЕОМ.

За призначенням розрізняють *універсальні і спеціалізовані* мікропроцесори. Універсальні мікропроцесори можна застосовувати для вирішення різноманітних завдань. Їх ефективна продуктивність мало залежить від проблемної специфіки вирішуваних задач. Спеціалізація МП, тобто його проблемна орієнтація на прискорене виконання певних функцій, дозволяє різко збільшити ефективну продуктивність при вирішенні тільки певних задач. Серед спеціалізованих мікропроцесорів можна виділити: мікроконтролери, орієнтовані на виконання складних послідовностей логічних операцій; математичні МП, призначені для підвищення продуктивності при виконанні арифметичних операцій за рахунок, наприклад, матричних методів їх виконання; МП для обробки даних в різних областях застосування і т.д. За допомогою спеціалізованих МП можна ефективно вирішувати складні задачі паралельної обробки даних.

По вигляду оброблюваних вхідних сигналів розрізняють *цифрові і аналогові* мікропроцесори. Самі мікропроцесори – це цифрові пристрої, проте

вони можуть мати вбудовані аналого-цифрові і цифро-аналогові перетворювачі. Вхідні аналогові сигнали передаються в МП після перетворення в цифрову форму, обробляються, і після зворотного перетворення в аналогову форму, поступають на вихід. З погляду архітектури такі мікропроцесори є аналогові функціональні перетворювачі сигналів і називаються *аналоговими мікропроцесорами*. Вони можуть виконувати функції будь-якої аналогової схеми. Застосування аналогового мікропроцесора значно підвищує точність обробки аналогових сигналів, а їх відтворення розширює функціональні можливості за рахунок програмного налагодження цифрової частини мікропроцесора на різні алгоритми обробки сигналів.

Звичайно до складу однокристальних аналогових МП входять декілька каналів аналого-цифрового і цифро-аналогового перетворювачів. У аналоговому мікропроцесорі розрядність оброблюваних даних досягає 24 біт і більше, велике значення приділяється збільшенню швидкості виконання арифметичних операцій.

По характеру тимчасової організації роботи розрізняють *синхронні і асинхронні* мікропроцесори. *Синхронні мікропроцесори* – це мікропроцесори, в яких початок і кінець виконання операцій задаються пристроєм управління (час виконання операцій в цьому випадку не залежить від виду виконуваних команд і величин операндів). *Асинхронні мікропроцесори* дозволяють початок кожної наступної операції визначити по сигналу фактичного закінчення виконання попередньої операції. Для ефективнішого використання кожного пристрою мікропроцесорної системи, до складу асинхронно працюючих пристроїв, вводять електронні ланцюги, що забезпечують автономне функціонування пристроїв. Закінчивши роботу над якою-небудь операцією, пристрій виробляє сигнал запиту, що означає його готовність до виконання наступної операції. При цьому функції природного розподільника робіт приймає на себе пам'ять, яка, відповідно до наперед встановленого пріоритету, виконує запити решти пристроїв по забезпеченню їх командною інформацією і даними.

По кількості виконуваних програм розрізняють *одно- і багатопрограмні* мікропроцесори.

У *однопрограмих мікропроцесорах* виконується тільки одна програма. Перехід до виконання іншої програми відбувається після завершення поточної програми.

У *багато- або мультипрограмих мікропроцесорах* одночасно виконуються декілька (звичайно декілька десятків) програм. Організація мультипрограмної роботи мікропроцесорних керуючих систем дозволяє здійснювати контроль за станом і управляти великим числом джерел або приймачів інформації.

Типова структура мікропроцесора приведена на рис. 2.1. [5]. Мікропроцесор складається з трьох основних блоків: арифметико-логічного пристрою (АЛП), блоку внутрішніх регістрів і пристрою управління. Для передачі даних між цими блоками використовується внутрішня шина даних.

Арифметико-логічний пристрій виконує одну з головних функцій мікропроцесора – обробку даних. Перелік функцій АЛП залежить від типу мікропроцесора. Деякі АЛП здатні виконувати безліч різних операцій, у інших набір операцій обмежений. Функції АЛП визначають архітектуру мікропроцесора в цілому. У більшості мікропроцесорів АЛП виконує наступні операції: складання, віднімання, І, АБО, виключаюче АБО, інверсію, зсув вправо, зсув вліво, прирости позитивні і негативні.

Важлива складова частина мікропроцесора – *регістр*. Кожен регістр мікропроцесора можна використовувати для тимчасового зберігання одного слова даних.

Деякі регістри мають спеціальне призначення, інші – багатозначне. Останні називаються *регістрами загального призначення (РЗН)* і можуть використовуватися програмістом на його розсуд.

Кількість і призначення регістрів в мікропроцесорі залежать від його архітектури.

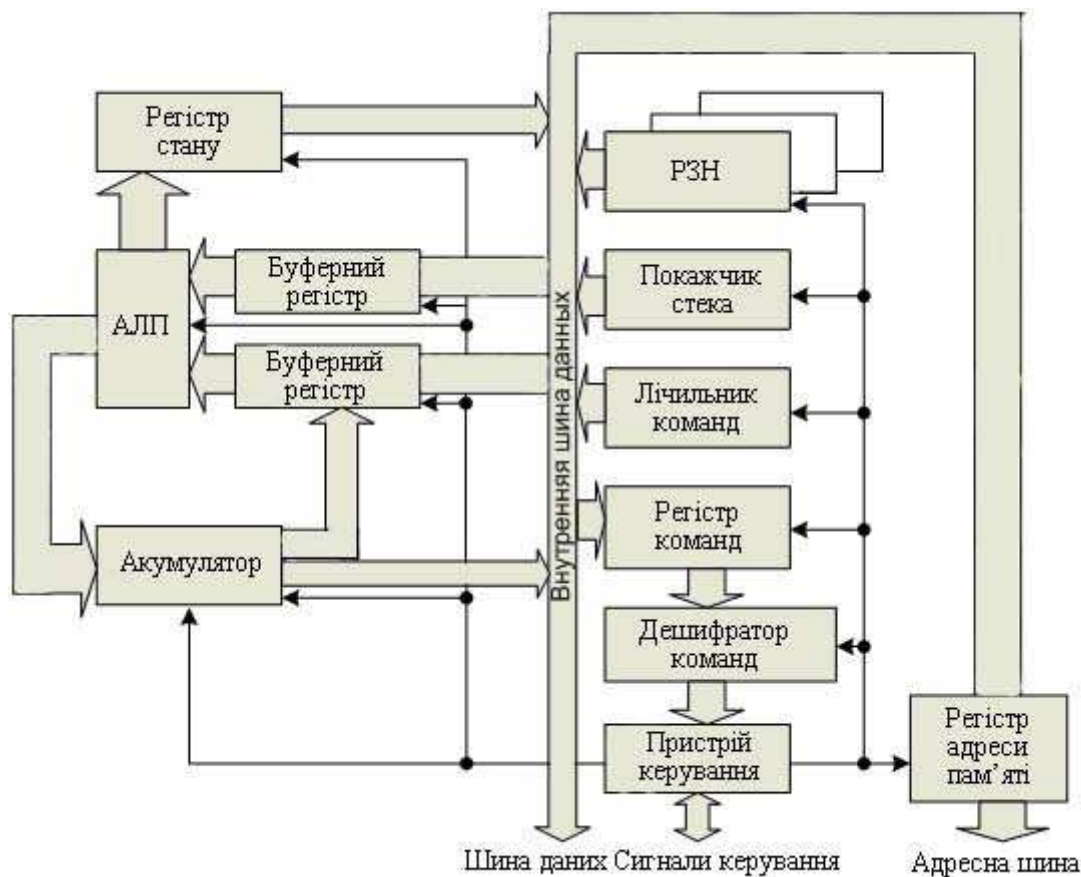


Рисунок 2.1 – Типова структурна схема мікропроцесора

Розглянемо призначення основних реєстрів, наявних майже у всіх мікропроцесорах.

Акумулятор – це головний реєстр мікропроцесора. Він використовується при різних маніпуляціях з даними. Більшість арифметичних і логічних операцій здійснюються шляхом використання АЛУ і акумулятора. Будь-яка з таких операцій над двома словами даних (операндами) припускає розміщення одного з них в акумуляторі, а іншого – в пам'яті або якому-небудь реєстрі. Так, при складанні двох слів, названих умовно А і В, і розташованих в акумуляторі та пам'яті відповідно, результуюча сума С завантажується в акумулятор, заміщаючи слово А. Результат виконання операції АЛУ теж звичайно розміщується в акумуляторі, вміст якого при цьому втрачається.

Операцією іншого типу, що використовує акумулятор, є програмована передача даних з однієї частини мікропроцесора в іншу. Наприклад, пересилка даних між портом вводу – виводу і пам'яттю, між двома областями пам'яті і т.ін. Виконання операції «програмована передача даних» здійснюється в два етапи: спочатку виконується пересилка даних з джерела в акумулятор, потім – з акумулятора в пункт призначення.

Мікропроцесор може виконувати деякі дії над даними безпосередньо в акумуляторі. Наприклад, акумулятор можна очистити шляхом запису двійкових нулів у всі його розряди, встановити в одиничний стан шляхом запису у всі його розряди двійкових одиниць. Вміст акумулятора можна зрушувати вліво або управо, набувати його інвертованого значення, а також виконувати інші операції.

Акумулятор є найбільш універсальним регістром мікропроцесора. Для виконання будь-якої операції з даними, перш за все, необхідно помістити їх в акумулятор. Дані поступають в нього з внутрішньої шини даних мікропроцесора. У свою чергу, акумулятор може посилати дані на цю шину.

Кількість розрядів акумулятора відповідає довжині слова мікропроцесора, проте деякі мікропроцесори мають акумулятори подвійної довжини. У додаткові розряди акумулятора записуються при цьому біти, що з'являються при виконанні деяких арифметичних операцій. Наприклад, при множенні двох 8-бітових слів результат (16-бітове число) розміщується в акумуляторі подвійної довжини.

Лічильник команд – це один з найбільш важливих регістрів мікропроцесора. Як відомо, програма – це послідовність команд (інструкцій), що зберігаються в пам'яті мікроЕОМ, і призначених для того, щоб інструктувати машину, як вирішувати поставлену задачу. Для коректного її виконання команди повинні поступати в строго певному порядку. Лічильник команд забезпечує формування адреси чергової команди, записаної в пам'яті.

Коли мікропроцесор починає працювати, то по команді початкової установки в лічильник команд завантажуються дані з області пам'яті, заданої

проектувальником мікропроцесора. Коли програма починає виконуватися, першим значенням вмісту лічильника команд є ця, наперед визначена, адреса.

На відміну від акумулятора, лічильник команд не може виконувати операції різного типу. Набір команд, що його використовують, украй обмежений в порівнянні з подібним набором для акумулятора.

Перед виконанням програми лічильник команд необхідно завантажити адресою, вказуючою на першу команду програми. Адреса першої команди програми посилається по адресній шині до схем управління пам'яттю, внаслідок чого прочитується її вміст за вказаною адресою. Далі ця команда передається в спеціальний реєстр мікропроцесора, званий *реєстром команд*.

Після витягання команди з пам'яті, мікропроцесор автоматично дає приріст вмісту лічильника команд. Цей приріст лічильник команд набуває в той момент, коли мікропроцесор починає виконувати команду, тільки що завантажену з пам'яті. Отже, з цієї миті лічильник команд містить адресу наступної команди.

Лічильник команд можна завантажити іншим вмістом при виконанні особливої групи команд. Може виникнути необхідність виконати частину програми, яка «випадає» з послідовності команд основної (головної) програми. Наприклад, таку частину програми, яка повторюється в процесі виконання всієї програми. Замість того щоб писати цю частину програми кожного разу, коли в ній виникає необхідність, програму записують один раз і повертаються до її повторного виконання, відступаючи від вказаної послідовності. Частина програми, що виконується шляхом відступу від послідовності команд головної програми, називається *підпрограмою*. В даному випадку в лічильник команд безпосередньо записується необхідна адреса. Часто лічильник команд має набагато більше розрядів, ніж довжина слова даних мікропроцесора. Так, у більшості 8-розрядних мікропроцесорів, число розрядів лічильника команд дорівнює 16.

Реєстр команд містить команду в процесі її дешифровки і виконання. Вхідні дані поступають в реєстр з пам'яті у міру послідовної вибірки команд.

Звичайно існує можливість запису даних в реєстр команд за допомогою набору перемикачів і кнопок на пульті управління ЕОМ. Як правило, цією можливістю користуються для передачі управління в початок програми.

Реєстр адреси пам'яті при кожному зверненні до пам'яті мікроЕОМ указує адресу області пам'яті, що підлягає використанню мікропроцесором. Реєстр адреси пам'яті містить двійкове число-адресу області пам'яті. Вихід цього реєстра називається *адресною шиною* і використовується для вибору області пам'яті або порту вводу-виводу.

Протягом вибірки команди з пам'яті, реєстри адреси пам'яті і лічильника команд мають однаковий зміст, тобто реєстр адреси пам'яті указує місцеположення команди, витягнутої з пам'яті.

Після декодування команди, лічильник команд одержує приріст на відміну від реєстра адреси пам'яті.

В процесі виконання команди зміст реєстра адреси пам'яті залежить від виконуваної команди. Якщо, відповідно до команди, мікропроцесор повинен провести ще одне звернення до пам'яті, то реєстр адреси пам'яті підлягає вторинному використанню в процесі обробки цієї команди. Для деяких команд, наприклад команди очищення акумулятора, адресація до пам'яті не потрібна. При обробці таких команд реєстр адреси пам'яті використовується лише один раз – протягом вибірки команди з пам'яті.

У більшості мікропроцесорів реєстри адреси пам'яті і лічильника команд мають однакову кількість розрядів. Як і лічильник команд, реєстр адреси пам'яті повинен мати в своєму розпорядженні кількість розрядів достатню для адресації будь-якої області пам'яті мікроЕОМ. У більшості 8-ми розрядних мікропроцесорів кількість розрядів реєстра адреси пам'яті дорівнює 16.

Оскільки реєстр адреси пам'яті підключений до внутрішньої шини даних мікропроцесора, він може завантажуватися від різних джерел. Більшість мікропроцесорів мають в своєму розпорядженні команди, що дозволяють завантажувати цей реєстр вмістом лічильника команд, реєстра загального призначення або якої-небудь області пам'яті. Деякі команди надають

можливість змінювати вміст регістра адреси пам'яті шляхом виконання обчислень: нове значення вмісту цього регістра виходить шляхом складання або віднімання вмісту лічильника команд з числом, вказаним в самій команді. Адресація такого типу називається *адресацією з використанням зсуву*.

Буферний регістр – призначений для тимчасового зберігання (буферування) даних.

Регістр стану – призначений для зберігання результатів деяких перевірок, здійснюваних в процесі виконання програми. Розряди регістра стану приймають те або інше значення при виконанні операцій, що використовують АЛП і деякі регістри. Запам'ятовування результатів згаданих перевірок дозволяє використовувати програми, що містять переходи (порушення природної послідовності виконання команд).

За наявності в програмі переходу за заданою ознакою, виконання команд починається з деякої нової області пам'яті, тобто лічильник команд завантажується новим числом. У разі умовного переходу така дія має місце, якщо результати певних перевірок співпадають з очікуваними значеннями. Вказані результати знаходяться в регістрі стану.

Регістр стану надає програмісту можливість організувати роботу мікропроцесора так, щоб за певних умов мінявся порядок виконання команд.

Розглянемо деякі найбільш часто використовувані розряди регістра стану.

1. *Перенесення/позика*. Даний розряд указує, що остання виконана операція супроводжувалася перенесенням або позикою (негативним перенесенням). Значення розряду перенесення встановлюється рівним 1, якщо в результаті складання двох чисел має місце перенесення із старшого розряду АЛП. Негативне перенесення (позика) фіксується в регістрі стану при відніманні більшого числа з меншого.

2. *Нульовий результат*. Приймає одиничне значення, якщо після закінчення операції у всіх розрядах регістра результату виявлені двійкові нулі. Установка цього розряду в 1 відбувається не тільки при негативному прирості

вмісту регістра, але і при будь-якій іншій операції, результат якої – число з двійкових нулів.

3. *Знаковий*. Приймає одиничне значення, коли старший значущий біт вмісту регістра, призначеного для запису результату операції, стає рівним 1. При виконанні арифметичних операцій з числами в додатковому коді одиничне значення старшого значущого біта показує, що в регістрі знаходиться негативне число.

Багато мікропроцесорів мають в своєму розпорядженні додаткові розряди станів. У деяких передбачені спеціальні команди для скидання або очищення всіх розрядів стану.

Регістри загального призначення (РЗП). Більшість МП мають в своєму складі набір регістрів, використовуваних як пристрої, що запам'ятовують. Оскільки АЛП може здійснювати операції з вмістом РЗП без виходу на зовнішню магістраль адрес і даних, то вони відбуваються набагато швидше, ніж операції із зовнішньою пам'яттю. Тому іноді РЗП *називають надоперативною пам'яттю*. Кількість РЗП і можливості програмного доступу до них у різних мікропроцесорів різні.

Показчик стека. *Стек* – це набір регістрів мікропроцесора або елементів оперативної пам'яті, звідки дані або адреси вибираються «зверху» за принципом: перший – що поступив останнім.

При записі в стек чергового слова всі раніше записані слова зміщуються на один регістр вниз. При вибірці слова із стека, слова, що залишилися, переміщуються вгору на один регістр. Вказані процедури ілюструє рис. 2.2 Тут стек складається з семи регістрів. Якщо в стек завантажується яке-небудь слово, наприклад А5, то воно записується у верхньому регістрі, а кожне із слів А1...А4 переміщується в сусідні нижні регістри. Якщо ж А5 витягується із стека, то кожне із слів А1..А4 переміщується в сусідні верхні регістри. Не можна витягнути А4 раніше А5, тобто автоматично реалізується відмічений вище принцип. Стек звичайно використовується в мікропроцесорах для

зберігання адрес повернення при зверненні до підпрограм, а також для запам'ятовування стану внутрішніх реєстрів при обробці переривань.

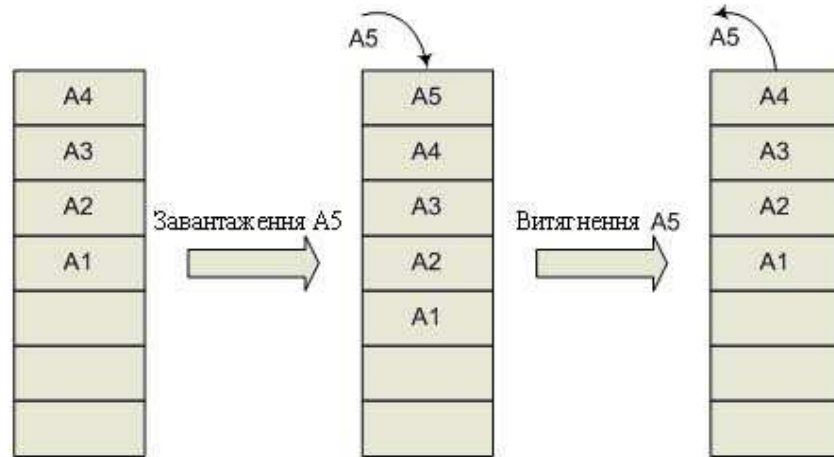


Рисунок 2.2 – Процедури роботи стека

При організації стека в пам'яті, час на звернення до нього буде дорівнювати циклу звернення до пам'яті. Ця операція виконується значно швидше, якщо стек у вигляді набору реєстрів входить до складу мікропроцесора. Важливим параметром у такому разі є число реєстрів стека. При спробі записати в стек більшої кількості слів, чим число його реєстрів, перше слово буде втрачено. У деяких мікропроцесорах при переповненні реєстрів стека відповідні слова записуються в стек пам'яті.

Часто стек реалізується таким чином, що процес його функціонування нагадує роботу з пачкою документів, коли кожен новий документ кладеться зверху пачки. При такій організації стека необхідний спеціальний реєстр – показчик стека (РПС) для зберігання адреси останнього, за часом надходження, елемента стека. Приведений на рис. 2.3. показчик стека є трьохрозрядний реєстр, з двійковим представленням інформації. Спочатку показчик стека містить число 011_2 . Це означає, що останній елемент – «верхівка стека» знаходиться в реєстрі з адресою 011_2 (або 3_{10}). При операції завантаження в реєстр 4_{10} записується число A5, а вміст показчика стека змінюється так, що

він указує на регістр 4₁₀. При операції витягання із стека проводяться зворотні дії.

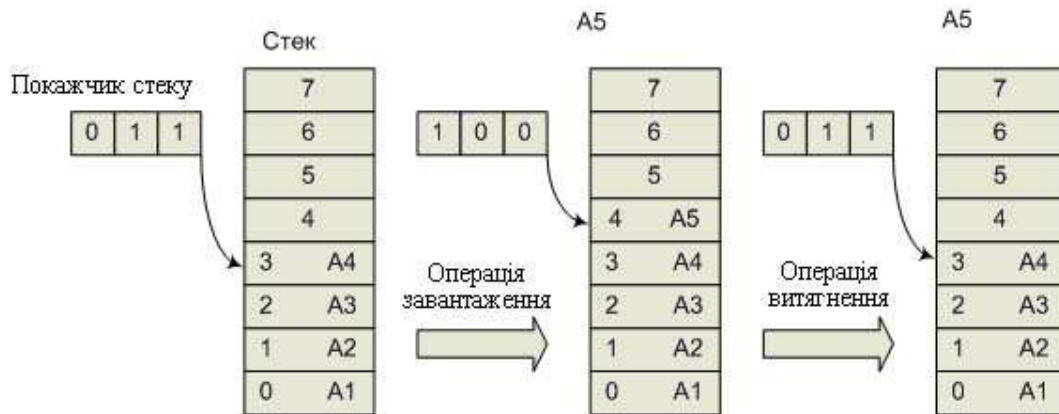


Рисунок 2.3 – Адресація елементу стека з використанням покажчика стека

Схеми управління. Роль схем управління в мікропроцесорі полягає в підтримці необхідної послідовності функціонування всієї решти його ланок.

По сигналах схем управління чергова команда витягується з регістра команд. При цьому визначається, що необхідно робити з даними, а потім забезпечується послідовність дій для виконання поставленого завдання.

Одна з головних функцій схем управління – декодування команди, що знаходиться в регістрі команд, за допомогою дешифратора команд, який в результаті видає сигнали, необхідні для її виконання.

Крім вказаних вище дій, схеми управління виконують деякі спеціальні функції керування послідовністю включення живлення і процесами переривань. Переривання – це свого роду запит, що поступає на схеми управління з інших пристроїв (пам'яті, вводу-виводу). Переривання пов'язане з використанням внутрішньої шини даних мікропроцесора. Схеми управління приймають рішення, коли і в якій послідовності інші пристрої можуть користуватися внутрішньою шиною даних.

Система шин. На характеристики мікропроцесора впливає спосіб організації його зв'язку із зовнішнім середовищем – пристроями вводу-виводу (ПВВ) і пристроями, що запам'ятовують. За способом організації зв'язків із зовнішнім середовищем розрізняють мікропроцесори з мультиплексованою шиною адреси і даних (рис. 2.4, а) і з роздільними шинами адреси і даних (рис. 2.4,б) [12]. Мікропроцесор з роздільними шинами адрес і даних зображений на рис. 2.1.

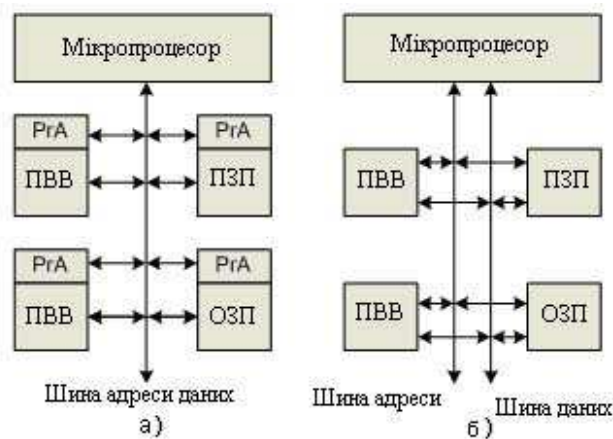


Рисунок 2.4 – Система шин мікроЕОМ

У мікропроцесорах з мультиплексованою шиною адреса зберігається на шині тільки короткий проміжок часу, тому пристроям, підключеним до шини, потрібні регістри адреси (РГА). Для організації обміну інформацією в таких мікропроцесорах необхідно використовувати керуючий сигнал «дані-адреси». При роздільних шинах адреси і даних сигнал, що управляє, не потрібен. Крім того, у пристроїв, підключених до шин, відпадає необхідність в регістрі адреси, оскільки він може бути розміщений безпосередньо на кристалі мікропроцесора. Розрядність адресної шини в таких мікропроцесорах не пов'язана з розрядністю шини даних. Характерним прикладом МП з роздільними шинами адрес і даних є мікропроцесорний комплект КР580, а з мультиплексованою шиною адреси і даних – мікропроцесорний комплект К588.

2.2. Мікропроцесорний комплект серії КР580

У мікропроцесорному комплекті КР580 центральний процесор, разом з пристроєм управління, реалізований у вигляді окремої ВІС і має фіксовану розрядність і вбудовану систему команд [4,10,14].

Мікропроцесорний комплект серії КР580 включає: мікропроцесорну ВІС КР580ВМ80; програмований пристрій вводу-виводу паралельної інформації різного формату КР580ВВ55; програмований блок пріоритетного переривання КР580ВН59; програмований пристрій прямого доступу до пам'яті КР580ВТ57; інтервальний таймер КР580ВІ53; універсальний синхронно-асинхронний програмований приймач-передавач КР580ВВ51; програмований контролер електронопроменевої трубки КР580ВГ75; програмований контролер клавіатури КР580ВВ79; системний контролер КР580ВК28; тактовий генератор КР580ГФ24; шинні формувачі КР580ВА86, 87; буферні регістри КР580ІР82, 83 [4,10,14].

Схеми виконані за n-МОП технологією, вхідні і вихідні сигнали відповідають рівням роботи ТТЛ схем. МікроЕОМ, побудована на базі комплекту, працює з тактовою частотою до 2 МГц. Схеми програмуються за допомогою фіксованого набору команд мікропроцесорних ВІС КР580ВМ80.

Мікропроцесорна ВІС КР580ВМ80 являє собою однокристальний 8-розрядний мікропроцесор з двома магістралями: однонаправленою 16-розрядною адресною магістраллю (МА), двонаправленою 8-розрядною магістраллю даних (МД) і 12 сигналами управління (шість вхідних і шість вихідних). Структурна схема мікропроцесора показана на рис. 2.5. Мікропроцесорна ВІС розрахована на виконання логічних і арифметичних операцій з 8-розрядними числами в двійковій і десятковій системах числення, а також операцій з подвійною розрядністю (з 16-розрядними числами).

Функціональне призначення зовнішніх виводів ВІС КР580ВМ80 наступне:

A0...A15 – адресна магістраль (виходи з трьома станами), що забезпечує адресацію до будь-якого з 2^{16} 8-розрядного елементу пам'яті або зовнішнього пристрою (ВУ);

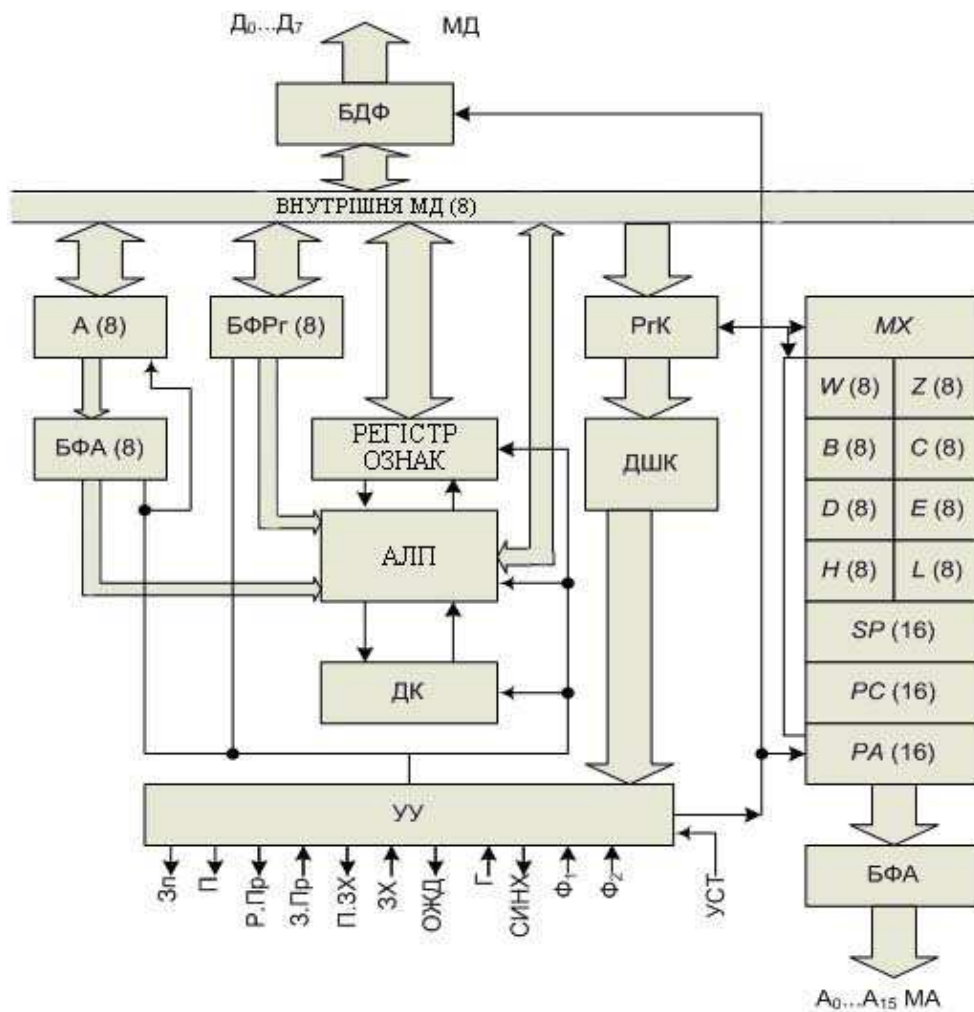


Рисунок 2.5 – Структурна схема мікропроцесора КР580ВМ80.

Д0...Д7 – двонаправлена магістраль даних, використовувана для обміну інформацією з пам'яттю або ВУ;

СИНХ – на цьому виході ВІС формується сигнал СИНХРОНІЗАЦІЯ на початку кожного машинного циклу;

П – сигнал ПРИЙОМ на цьому виході указує на готовність ВІС до прийому даних;

ОЖД – сигнал ОЧІКУВАННЯ на цьому виході указує, що мікропроцесор знаходиться в процесі очікування;

Зп – на цьому виході МП ВІС сигнал ЗАПИС указує, що дані видані мікропроцесорною ВІС і встановлені на МД, тобто можуть бути записані у ВУ;

ЗХ – вхід, використовуваний для подачі сигналу ЗАПИТ ЗАХОПЛЕННЯ на МП ВІС. Забезпечує перехід мікропроцесора в стан ЗАХОПЛЕННЯ, при якому МА і МД переходять в третій стан (високий опір). Звичайно цей стан використовується для організації обміну інформацією по каналу прямого доступу до пам'яті, а також в багатомікропроцесорних системах;

Р.Пр – на цьому виході сигнал ДОЗВІЛ ПЕРЕРИВАННЯ указує на стан внутрішнього тригера дозволу переривання МП ВІС. Стан тригера може бути встановлений програмно. При рівні «0» на виході *Р.Пр* прийом запитів переривання МП ВІС неможливий;

Г – сигнал ГОТОВИЙ на цьому вході інформує про готовність ВУ до обміну інформацією з МП ВІС. При рівні «0» МП ВІС знаходиться в стані ОЧІКУВАННЯ;

П.ЗХ – на цьому виході МП ВІС сигнал ПІДТВЕРДЖЕННЯ ЗАХОПЛЕННЯ з'являється у відповідь на сигнал *ЗХ* і указує, що МД і МА знаходяться в стані високого опору. Поки діє сигнал *ЗХ*, процесор припиняє виконання своїх операцій після закінчення поточного машинного циклу;

З.Пр – вхід, використовуваний для подачі сигналу ЗАПИТ ПЕРЕРИВАННЯ. Сигнал поступає від зовнішнього пристрою і вимагає переривання виконання основної програми, переходу на виконання підпрограм обслуговування переривання. Сигнал запиту переривання не сприймається МП ВІС при роботі його в режимах ЗАХОПЛЕННЯ, ОЧІКУВАННЯ або нульовому стані внутрішнього тригера дозволу переривання;

УСТ – вхід, по якому поступає сигнал на початкову установку МП, при цьому обнуляються його програмний лічильник, внутрішні тригери, що

формують сигнали P , Pr і $П$. $ЗХ$.

$\Phi 1$ і $\Phi 2$ – входи для подачі тактових сигналів $\Phi 1$ і $\Phi 2$, які є непересічними в часі сигналами з амплітудою 12В і визначають тактову частоту роботи МП. Для формування цих сигналів використовується ВІС КР580ГФ24.

Входи джерел живлення МП ВІС – нульовий потенціал (земля), +5В, –5В, +12В.

Мікропроцесорна ВІС, яка представлена на рис. 2.5, складається з наступних функціональних блоків: блоку реєстрів загального призначення; арифметико-логічного блоку; реєстра команд (PrK) і пристрою управління (ПУ); буферів МД (БФД) і МА (БФА).

Блок реєстрів загального призначення (РЗН) включає шість програмно доступних реєстрів, які складені в пари: B , C ; D , E ; H , L . Їх можна використовувати в програмах як окремі 8-розрядні реєстри, а також як три 16-розрядні реєстри (B ; D ; H). Реєстри W і Z не є програмно доступними і використовуються для виконання команд усередині МП.

Програмний лічильник PC містить поточну адресу пам'яті, до якої звертається програма. Зміст PC автоматично змінюється протягом кожного циклу команди. Показчик стека SP містить адресу пам'яті, починаючи з якої її можна застосовувати для зберігання і відновлення вмісту програмно доступних реєстрів мікропроцесора. Двонаправлений шинний мультиплексор MX призначений для зв'язку РОНів з шиною даних МП; PA – це реєстр, призначений для зберігання адреси пам'яті.

Арифметико-логічний блок (АЛБ) виконує арифметичні і логічні операції під впливом пристрою управління МП. Він включає 8-розрядне АЛП, схему десяткової корекції ДК, побудовану на базі ПЗП, 5-розрядний реєстр ознак, акумулятор A , буфер акумулятора БФА і буферний реєстр БФРг. Арифметико-логічний блок дозволяє здійснювати арифметичні операції складання, віднімання, а також основні логічні операції (І, АБО, ВИКЛЮЧАЮЧЕ АБО) і зсув. При проведенні операцій одне число завжди береться з буфера

акумулятора, а інше – з буферного регістра. По результату виконання арифметико-логічних операцій, АЛБ встановлює в регістрі ознак п'ять розрядів.

1. *Біт перенесення* встановлюється (рівний «1») і скидається (рівний «0») при виконанні команд складання, віднімання, зсуву і логічних команд, і може бути програмно проаналізований. Якщо в результаті виконання операції з'являється одиниця перенесення із старшого розряду, то біт перенесення встановлюється, якщо ж перенесення не виникає, то біт перенесення скидається.

2. *Додаткова ознака перенесення* встановлюється в одиницю, якщо при виконанні команд виникає одиниця перенесення з третього розряду числа. Стан розряду може бути проаналізований лише командою десяткової корекції числа.

3. *Біт знаку* забезпечує представлення в додатковому двійковому коді числа від -128_{10} до $+127_{10}$. При цьому сьомий біт представляє знак числа. Якщо він містить 1, байтом можна представити числа -128_{10} до -1_{10} , якщо 0, то – від 0 до $+127_{10}$. В кінці виконання деяких операцій біт знаку встановлюється по сьомому біту результату.

4. *Біт нульової ознаки* встановлюється, якщо результат певних операцій рівний 0. Біт нульової ознаки скидається, якщо результат ненульовий. Якщо операція дає нульовий байт результату і одиницю перенесення, то біт нуля також встановлюється.

5. *Розряд ознаки парності*. У нього записується одиниця, якщо при виконанні команди кількість одиниць в розрядах результату буде парною.

Регістр команд PгК і дешифратор команд ДШК використовуються в мікропроцесорі для отримання і дешифрації коду команди. При витяганні команди, перший байт коду, що містить її, поміщається в регістр команди і поступає на дешифратор команд.

Пристрій управління спільно з дешифратором формує сигнали, що управляють, для всіх внутрішніх блоків мікропроцесора, а також його вихідні сигнали управління і стану.

Буфери МД і МА забезпечують зберігання відповідно даних і адрес на якийсь час, необхідний для їх обробки.

Виконання кожної команди проводиться мікропроцесором в строгій послідовності, визначеній кодом команди, і синхронізується в часі сигналами $\Phi 1$ і $\Phi 2$ тактового генератора.

Період синхросигналів $\Phi 1$ і $\Phi 2$ називається *машинним тактом*. Тривалість машинного такту T можна встановити довільно в діапазоні 0,4...2мкс. *Машинний цикл* – час, потрібний для витягання одного байта інформації з пам'яті або виконання команди, визначуваної одним машинним словом. Машинний цикл може складатися з 3...5 машинних тактів. *Час виконання команди* – час отримання, декодування і виконання команди. Залежно від виду команди цей час може складатися з 1...5 машинних циклів. Для МП КР580ВМ80 існує *десять різних типів машинних циклів*: 1) витягання коду команди; 2) читання даних з пам'яті; 3) запис даних в пам'ять; 4) витягання даних із стека; 5) запис даних в стек; 6) вводу даних із зовнішнього пристрою; 7) запис даних в зовнішній пристрій; 8) цикл обслуговування переривання; 9) останов; 10) обслуговування переривання при роботі МП в режимі останову.

Першим машинним циклом при виконанні будь-якої команди завжди є цикл витягання коду команди. У першому такті кожного машинного циклу, МП указує тип виконуваного циклу за допомогою 8-розрядного слова стану процесора, що видається на МД.

Окремі розряди слова стану задіюються в мікроЕОМ для формування сигналів в лініях магістралі управління. Слово стану видається на МД лише на короткий проміжок часу, а використовується впродовж всього машинного циклу, тому його записують в спеціальний регістр слова стану (PrCC).

Зовнішні пристрої видають сигнали запиту на переривання асинхронно по відношенню до роботи мікропроцесора шляхом подачі рівня «1» на вхід *3.Пр*. Запит може бути задоволений тільки в тому випадку, якщо мікропроцесор знаходиться в стані дозволу сприйняття запитів переривання. Мікропроцесор автоматично встановлюється в стан заборони сприйняття запитів переривання

після виконання початкової установки, а також після початку обслуговування запиту переривання. Управління станом сприйняття запитів переривання може здійснюватися програмно за допомогою команд ДОЗВІЛ ПЕРЕРИВАНЬ і ЗАБОРОНА ПЕРЕРИВАНЬ.

Мікропроцесор указує на можливість сприйняття переривань сигналами на виході *P.Пр*. Якщо на цьому виході присутній рівень «1», то запити переривання можуть бути сприйняті, інакше (рівень «0» на виході *P. Пр*) запити сприйматися не будуть. Для забезпечення режиму переривань використовується програмований блок пріоритетного переривання – КР580ВН59, що забезпечує вісім рівнів запитів переривання.

Обмін інформацією між мікропроцесором і зовнішніми пристроями організовується за допомогою трьох шин: *адрес, даних і управління*.

Шина адреси є однонаправленою, видача адрес на неї здійснюється лише МП ВІС або схемою прямого доступу до пам'яті при роботі мікроЕОМ з каналом прямого доступу до пам'яті. Мала здатність навантаження шини адреси вимагає застосування буфера для тимчасового зберігання даних з метою узгодження роботи пристроїв.

Шина даних організує двонаправлений обмін даними між МП ВІС і пам'яттю або зовнішніми пристроями. Максимальна здатність навантаження її невелика, що не дозволяє безпосередньо підключати до неї системи з великою місткістю пам'яті і широким набором периферійних пристроїв. Для збільшення здатності навантаження шини даних, застосовують контроллер КР580ВК28.

Шина управління складається з п'яти ліній, сигнали на яких формуються із записаного слова стану МП ВІС і його вихідних сигналів ПРИЙОМ і ЗАПИС. Для формування сигналів шини управління також застосовується контроллер КР580ВК28.

Восьмирозрядне машинне слово мікропроцесора КР580ВМ80 [4,10,14] використовується для повідомлення МП ВІС як коду команд, так і даних. Набір команд мікропроцесора зафіксований і складається з 246 різних команд.

Команди МП КР580 можна розділити на п'ять груп:

1. Команди пересилки даних, що здійснюють пересилку даних між регістрами або регістрами і пам'яттю.

2. Арифметичні команди, що включають команди складання, віднімання, збільшення або зменшення на одиницю даних.

3. Логічні команди, що дозволяють здійснити логічні операції І, АБО, ВИКЛЮЧАЮЧЕ АБО, порівняння, зсув або узяття доповнення числа.

4. Команди передачі управління, що забезпечують безумовну передачу або передачу управління по умові, а також виклик або повернення з підпрограм.

5. Команди управління і роботи із стеком, організуючі ввід-вивід даних з мікроЕОМ, доступ до стека і внутрішнього регістра ознак МП ВІС, а також її управління.

Система команд Асемблера КР580 приведена в додатку 1.

Пам'ять мікроЕОМ на базі МП КР580ВМ80 організована у вигляді 8-розрядних комірок, кожна з яких має свою 16-розрядну адресу. Таким чином, МП ВІС може працювати з пам'яттю місткістю до 65536 байт. Багатобайтові команди записуються в послідовні елементи пам'яті. *Перший байт команди завжди містить код команди.* При трьохбайтових командах в другому і третьому байтах містяться дані (адреса), причому в другому байті записується молодший байт даних, а в третьому – старший байт даних (адреси).

Мікропроцесорна ВІС має чотири типи адресації до пам'яті або регістрів:

1. *Пряма адресація* – другий і третій байти команди містять вказівку адреси пам'яті: молодший байт адреси в другому байті команди, а старший в третьому.

2. *Адресація до регістра* – команда указує на регістр або пару регістрів, в яких записані дані.

3. *Непряма адресація за допомогою регістрів* – команда указує на пару регістрів, в яких записана адреса пам'яті, що містить дані.

4. *Безпосередня адресація* – команда сама містить дані. При двобайтових даних в другому байті команди записується молодший байт, а в третьому –

старший. За винятком режимів обслуговування переривань і виконання команд передачі управління МП ВІС виконує програми, прочитуючи команди послідовно з елементів пам'яті. У командах передачі управління адреса, до якої звертатиметься МП ВІС, може указуватися за допомогою прямої або непрямой адресації. При виконанні команд передачі управління по умові МП ВІС перевіряє наявність цієї умови і у разі її підтвердження здійснює передачу управління. При невиконанні умови програма продовжуватиметься з команди, записаної в елементі пам'яті, наступному після команди передачі управління по умові.

Контрольні питання і завдання

1. Назвіть три основні функціональні вузли структурної схеми МП.
2. Викладіть призначення і принцип дії акумулятора.
3. Назвіть операції, що виконуються АЛП.
4. Викладіть призначення і функціонування програмного лічильника.
5. Що міститься в реєстрі адреси пам'яті?
6. Назвіть ознаки реєстра стану і їх призначення.
7. Яка команда міститься в реєстрі команд в процесі виконання поточної команди і коли змінюється адреса в покажчику стека?
8. Приведіть принцип дії стека і призначення покажчика стека.
9. Для чого призначені буферні реєстри і реєстри загального призначення?
10. Викладіть призначення схем управління і синхронізації МП.
11. На протязі якого часу МП витягує команду з пам'яті і виконує її?
12. Що забезпечує таймер в схемах управління МП?
13. Куди поміщається результат виконання операції АЛП?
14. Назвіть шість основних реєстрів МП і дайте їм коротку характеристику.
15. Дайте визначення архітектури МП.
16. Приведіть відомі архітектури МП і вкажіть їх відмінності.
17. Приведіть відмінність акумулятора від інших реєстрів.

18. Назвіть функції, що виконуються пристроєм управління.
19. Що визначає вміст лічильника команд і як він змінюється?
20. Приведіть визначення стекової пам'яті і вкажіть призначення покажчика стека.
21. Що таке набір команд МП?
22. Якою може бути довжина команди 8-розрядного МП?
23. З яких двох частин складається команда і яке їх призначення?
24. Дайте характеристики безпосередньої, прямої, непрямої і реєстрової адресації.
25. При використанні, якого способу адресації для виконання команди потрібна найбільша кількість мікроциклів?
26. Перерахуйте способи адресації, і вкажіть, який з вказаних способів адресації дозволяє проводити звернення до будь-якої області пам'яті за допомогою команди завдовжки в одне слово.
27. Перерахуйте арифметичні і логічні команди і дайте їм характеристику.
28. Назвіть команди пересилки даних і дайте характеристику команд завантаження, пересилки і запису в пам'ять.
29. Назвіть команди переходу і виклику підпрограм і поясніть їх роботу.
30. Назвіть команди вводу-виводу.
31. Назвіть команди роботи із стеком.
32. Назвіть команди умовних переходів.
33. Назвіть команди пересилки даних з акумулятора в ОЗП і навпаки.
34. Складіть програму складання двох чисел і запису результату в комірку ОЗП.
35. Складіть програму віднімання двох чисел і запису результату в реєстр D і в елемент пам'яті з адресою 8020.
36. Складіть програму виконання операцій АБО, І, ВИКЛЮЧАЮЧЕ АБО над двома числами і занесення результатів в пам'ять.
37. Назвіть команди безпосереднього завантаження реєстрових пар і покажчика стека двобайтовими словами.
38. Назвіть команди обміну двобайтовими словами між реєстрами *HL* і *DE*, *HL* і

вершиною стека.

39. Поясніть роботу команд безумовного і умовного повернень з підпрограм.

40. Команди переходу до підпрограм обслуговування переривань.

41. Поясніть роботу команд дозволу і заборони переривань.

42. Призначення команд *HLT*, *NOP*, *CMC*, *CMA*, *CPI*.

43. Назвіть функції, що виконуються командами *OUT* і *IN*.

44. Назвіть команди пересилки даних з пам'яті в акумулятор.

45. Поясніть роботу команди повернення з підпрограм *RET*.

46. Викладіть роботу стека при виконанні команди виклику підпрограм *CALL* і вкажіть формат цієї команди.

47. Що таке розрядність мікропроцесора і структурою яких його пристроїв вона визначається?

48. Що знаходиться в першому байті трьохбайтної команди?

49. Скільки розрядів має шина даних КР580 і для чого вона призначена?

50. Із скількох ліній складається шина керування і які сигнали і чим на ній формуються?

51. Дайте характеристику шини адреси і вкажіть її призначення.

52. В яких випадках і де встановлюється та скидається біт переносу?

53. В яких випадках і де встановлюється та скидається біт додаткового переносу?

54. В яких випадках і де встановлюється та скидається біт знаку?

55. Яке призначення біта ознаки нуля? Коли і де він встановлюється і скидається?

56. Що характеризує ознаку парності? Коли і де він встановлюється і скидається?

57. Які функції виконує регістр команд (РГК) і дешифратор команд (ДШК)?

58. Назвіть десять різних типів машинних циклів в МП КР580.

59. Для чого призначений пристрій керування?

60. Приведіть визначення періоду синхросигналів, машинного такту, машинного циклу.

3. ЗАПАМ'ЯТОВУЮЧІ ПРИСТРОЇ МІКРОПРОЦЕСОРНИХ СИСТЕМ

Система пам'яті є функціональною частиною мікропроцесорної системи, призначеної для запису, зберігання і видачі інформації. Технічні засоби, що реалізують функції пам'яті називаються *пристроями, що запам'ятовують або запам'ятовуючі пристрої*.

3.1. Класифікація запам'ятовуючих пристроїв

Пристрої, що запам'ятовують (ЗП), є найважливішою складовою частиною будь-якого обчислювального пристрою, у тому числі і побудованого на мікропроцесорних ВІС.

По функціональному призначенню ЗП, використовувані в мікропроцесорних системах, можна розділити на наступні групи:

надоперативні ЗП, що є набір регістрів, вміст яких безпосередньо використовується при обробці інформації в мікропроцесорі;

кеш-пам'ять, допоміжна оперативна пам'ять, недоступна для програміста, служить для підвищення швидкодії МП;

оперативні ЗП, що зберігають оперативну інформацію (операнди, частини програми), потрібну в процесі роботи;

постійні ЗП, призначені для тривалого зберігання незмінної в процесі роботи мікроЕОМ інформації (програм, мікропрограм, констант);

напівпостійні ЗП, по виконуваних функціях повністю відповідають постійним ЗП, але відрізняються від останніх можливістю щодо швидкої зміни інформації, що зберігається в них, при необхідності зміни програм або констант, за обмежений час;

зовнішні ЗП, призначені для зберігання великих об'ємів інформації, з невеликою питомою вартістю біта інформації, що зберігається;

буферні ЗП, призначені для узгодження різних рівнів системи пам'яті мікроЕОМ між собою і зовнішніх пристроїв з системою пам'яті.

Запам'ятовуючі пристрої мікроЕОМ, можуть бути реалізовані на основі тільки ВІС ОЗП. Проте втрата інформації при відключенні живлення, вища питома вартість біта інформації, що зберігається, і ряд інших причин привели до широкого використання в мікроЕОМ постійних і напівпостійних ЗП.

За способом організації обміну інформацією між окремими ЗП і мікропроцесорами розрізняють ЗП: *з довільною вибіркою (ЗПДВ)*; *з послідовним зверненням (ЗППЗ)*; *з послідовно-паралельною організацією обміну*.

Переважаюча частина ЗП пристроїв мікропроцесорних запам'ятовуючих систем, – це ЗП з довільною вибіркою (або з довільним доступом). Якщо пам'ять не забезпечує довільного доступу до будь-якої її частини, то це пам'ять з послідовним доступом; вона рідко використовується як основна пам'ять мікропроцесорної системи. При роботі з такою пам'яттю для звернення до потрібної області необхідно заздалегідь звернутися до всіх областей, що знаходяться між тією, до якої відбулося звернення в даний момент, і необхідною областю. У послідовній формі зберігається інформація на магнітній стрічці. Послідовний доступ застосовують для запам'ятовування значних об'ємів даних, час звернення до яких не є критичним.

Пристрої, що запам'ятовують, характеризуються рядом якісних показників.

1. *Ємність ЗП* визначається максимально можливою кількістю бітів інформації, що зберігається.

2. *Ширина вибірки* визначається кількістю інформації, записуваної ЗП або витягуваної з нього за одне звернення. Ця характеристика достатньо важлива, оскільки при однаковому часі звернення, ЗП з більшою шириною вибірки має більшу інформаційну ефективність.

3. *Час звернення* визначається з моменту подачі в пристрій сигналу запису або читання до того моменту, коли закінчаться всі дії, пов'язані з виконуваною операцією, і пристрій буде готовий прийняти і реалізувати наступну операцію звернення до накопичувача інформації. Цей час називається також тривалістю циклу звернення до ЗП, протягом якого можна вибрати інформацію (читання), ввести інформацію (запис), відновити або модифікувати стан деякого елемента ЗП.

4. *Швидкість обміну інформацією між ЗП і іншими пристроями* є важливим параметром пристроїв, що запам'ятовують. Вона визначається числом бітів (байтів), передаваних в одиницю часу.

5. *Показник питомої вартості* застосовується для оцінки економічних характеристик ЗП. Він визначається відношенням його вартості до інформаційної місткості, тобто вартості біта інформації, що зберігається.

6. *Надійність*, а для систем спеціального призначення – *масо-габаритні показники і споживана потужність*, є для ЗП істотними характеристиками.

7. Одна з характеристик ЗП – здатність зберігати інформацію при відключенні джерел живлення. В цьому випадку розрізняють *енергозалежну або незалежну пам'ять*. У незалежній пам'яті при порушеннях в роботі системи живлення дані не руйнуються, а в енергозалежній – руйнуються.

Для мікропроцесорних систем найбільш характерно використання оперативних пристроїв, що запам'ятовують, на МОП- транзисторах. Існує *два способи побудови інтегральних схем пам'яті по МОП- технології*, залежно від яких пам'ять на МОП- структурах може бути *статичною* або *динамічною*.

Статична пам'ять простіше з погляду організації, що особливо явно на пристроях, що запам'ятовують, невеликого об'єму. Управляти статичною пам'яттю легше.

Інтегральні схеми, вживані для побудови динамічної пам'яті, відносно дешеві, але для організації її роботи необхідні додаткові мікросхеми. Крім того, *вміст мікросхем динамічної пам'яті необхідно періодично регенерувати*. На динамічному принципі будуються пристрої, що запам'ятовують, великого об'єму.

У деяких мікропроцесорних системах знаходять застосування ще два види напівпровідникової пам'яті: на приладах із зарядовим зв'язком (ППЗЗ) і на циліндрових магнітних доменах (ПЦМД). Обидва типи пам'яті орієнтовані на послідовний доступ. Як і інші пристрої послідовного доступу, вони мають невисоку швидкодію, проте дозволяють реалізувати запам'ятовуючі пристрої значно більшого об'єму, ніж пристрої пам'яті на біполярних і МОП-транзисторах.

Достоїнства ППЗЗ – надзвичайно мале споживання потужності, простота в застосуванні. Пам'ять на ППЗЗ енергозалежна, але мала споживана потужність робить можливим збереження в ній інформації за допомогою резервних акумуляторів при відмові на тривалий час основного джерела живлення.

Для ПЦМД потрібно набагато більше число допоміжних схем, чим для пам'яті на приладах із зарядовим зв'язком, але вона є енергонезалежною.

Таким чином, цей тип пам'яті дозволяє реалізувати ті енергозалежні пристрої, що запам'ятовують, великого об'єму, для яких не потрібні складні механічні стрічкопротяжні механізми і дисководи.

У подальших підрозділах розглядаються основні типи пристроїв, що запам'ятовують, мікропроцесорних систем і їх характеристики.

3.2. Оперативні запам'ятовуючі пристрої

За принципом зберігання інформації напівпровідникові оперативні ЗП діляться на динамічні і статичні.

Динамічні пристрої, що запам'ятовують, будуються на основі елемента, що запам'ятовує, зберігають свій стан тільки певний проміжок часу і тому вимагають періодичного відновлення. Елементом динамічних напівпровідникових ЗП служить конденсатор, в якому інформація зберігається у формі наявності або відсутності заряду. Із-за витоків поступово зменшується заряд на конденсаторі, що запам'ятовує; для відновлення заряду, конденсатор, що запам'ятовує, періодично підключають до джерела живлення.

Регенерація (відновлення даних) елемента динамічної пам'яті проводиться при кожному зверненні до неї. Проте при звичайній інтенсивності роботи пам'яті мікропроцесорної системи така регенерація не гарантує збереження всіх бітів інформації, що зберігалися в ній. Мікропроцесор може, наприклад, витратити час, що перевищує декілька мілісекунд, на виконання простого циклу, призначеного для вироблення часової затримки, і використовувати при цьому лише декілька елементів пам'яті. Протягом всього цього тимчасового проміжку всі інші слова пам'яті не піддаватимуться регенерації. Тому пристрої пам'яті, виконані на динамічних ЗП, потребують логічної схеми регенерації, яка автоматично звертається до кожного стовпця пам'яті з інтервалами в декілька десятих долей мілісекунди. Динамічний пристрій, що запам'ятовує, побудований так, що саме звернення до стовпця забезпечує регенерацію інформації у всіх його комірках. Робота логічної схеми регенерації повинна координуватися з іншими діями мікропроцесора. Якщо, наприклад, мікропроцесор намагається звернутися до пам'яті в мить, коли в ній здійснюється регенерація, то схема регенерації повинна віддати пріоритет саме мікропроцесору.

Динамічний принцип зберігання інформації забезпечує ряд переваг динамічних ЗП. Динамічні напівпровідникові ЗП з довільною вибіркою мають високий рівень інтеграції і швидкодії, низьку вартість і тому знаходять широке застосування в мікропроцесорних обчислювальних системах. На динамічних ВІС ЗП, що є функціонально закінченими пристроями, можна порівняно просто будувати ОЗП різної інформаційної місткості.

Схеми управління ЗП (мультиплексор, управління регенерацією) можуть бути виконані на стандартних або спеціалізованих ІС.

Недолік динамічних ЗП – необхідність регенерації компенсується більшою, ніж в статичних ЗП, інформаційною місткістю.

Статичні пристрої, що запам'ятовують, є найбільш поширеним видом пам'яті мікропроцесорних систем. Більшість статичних пристроїв, що запам'ятовують, реалізуються на основі МОП-технології, але існують і статичні ЗП на біполярних схемах.

Елемент пам'яті статичного ЗП є звичайний тригер. Він може бути встановлений або в стан «1», або в стан «0». Якщо тригер встановлений в «1», то цей стан зберігається до тих пір, поки не буде проведене скидання тригера або не буде вимкнене живлення.

Подібні елементи пам'яті об'єднуються в матричну структуру, тобто розміщуються по рядках і стовпцях. При побудові статичних ЗП найбільшого поширення набули ВІС ЗП з конфігурацією $(n \times 1)$ бітів, де n - кількість комірок, $n = 256, 512, 1024, 2048, \dots, 2K$. Випускаються ВІС ЗП і з іншою конфігурацією. Пам'ять мікроЕОМ звичайно має розрядність, рівну або кратну розрядності мікропроцесора.

Необхідна довжина слова пам'яті досягається паралельним включенням m ВІС пам'яті, де m – довжина слова в бітах. У 8-розрядному мікропроцесорі для побудови пам'яті з організацією $n \times 8$ необхідно включити паралельно вісім ВІС ОЗП з організацією $n \times 1$. Сукупність восьми однорозрядних ліній даних всіх восьми ВІС утворює 8-розрядну шину даних пристрою, що запам'ятовує. Лінії адреси і читання/запису всіх ВІС включаються паралельно для одночасного звернення до всіх ВІС ЗП.

Часто розрядність шини адреси мікроЕОМ більше розрядності її шини даних. Так, в 8-розрядній мікроЕОМ застосування 16-розрядної адреси забезпечує адресацію 65 536 байт пам'яті. Два байти називають молодшим і старшим байтами адреси. Молодші розряди адреси $A_0 \dots A_7$ називають *адресою*

слова (байта), а старші розряди A8...A15 називають *адресою сторінки*. Таким чином, можна адресувати 256 сторінок по 256 слів (байт) в кожній.

Типові ВІС динамічного ОЗП: КР565РУ6 місткістю 16384x1 біт; К565РУ5 місткістю 65536x1 біт.

Типові ВІС статичного ОЗП: КР537РУ14 – місткість 4096x1 біт; КР132РУ6А – місткість 16384x1 біт; К537РУ9 – місткість 2048x8 біт;

3.3. Постійні запам'ятовуючі пристрої

Постійні запам'ятовуючі пристрої (ПЗП) в мікропроцесорних обчислювальних системах служать для зберігання програм і іншої незмінної інформації. Важлива перевага ПЗП в порівнянні з ОЗП – збереження інформації при виключенні живлення. Вартість біта інформації, що зберігається в ПЗП, може бути майже на порядок нижче, ніж в ОЗП. Постійні ЗП можуть бути реалізовані на основі різних фізичних принципів і елементів і відрізняються способом занесення інформації, кратністю занесення, способом стирання.

В даний час застосовуються наступні види ПЗП: програмовані на заводі-виготовнику або масочні ПЗП (МПЗП); програмовані користувачем [15,16]; перепрограмовані ПЗП [13]. Перші два види ПЗП допускають тільки одноразове програмування, третій вид ПЗП дозволяє змінювати інформацію, що зберігається в ньому, багато разів.

Розглянемо докладніше кожного з типів ПЗП.

Програмовані масочні ПЗП програмуються їх виготовником, який за підготовленою користувачем інформацією робить фотошаблони, за допомогою яких заносить цю інформацію в процесі виробництва на кристал ПЗП. Цей спосіб найдешевший і призначений для великосерійного виробництва ПЗП.

Масочні ПЗП будуються на основі діодів, біполярних і МДП-транзисторів. У діодних ПЗП діоди включені в тих перетинах матриці, які відповідають запису «1», і відсутні в місцях, де повинен бути записаний «0». Зовнішні ланцюги управління діодних ПЗП дуже прості. Оскільки діодна

матриця є елемент з гальванічними зв'язками, то вихідні сигнали мають ту ж форму, що і вхідні. Таким чином, якщо на входи подаються напруги постійних рівнів, то і на виходах рівні будуть також постійними, тому відпадає необхідність у вихідному регістрі для зберігання інформації.

Масочні ПЗП на біполярних і МДП-транзисторах також будуються у вигляді матриць. Постійні ЗП на МДП-транзисторах дещо простіше у виготовленні, чим біполярні.

Масочні ПЗП характеризуються великою надійністю, але при їх виготовленні виникає ряд незручностей для замовника і для виготовника. Велика номенклатура ПЗП і мала їх тиражність, тому від виготовника потрібні підвищені витрати на фотошаблони, що збільшує вартість ПЗП. Відсутня можливість оперативно змінювати інформацію в ПЗП без виготовлення нової ІС, що особливо незручно на етапі розробки програм системи.

Програмовані користувачем ПЗП є більш універсальними і, отже, дорожчими приладами. Вони являються матрицями біполярних приладів, зв'язки яких з адресними і розрядними шинами руйнуються при занесенні, на спеціальних програмуючих пристроях, відповідних кодових комбінацій. Ці пристрої виробляють напруги, необхідні і достатні для перепалювання плавких перемичок у вибраних елементах ПЗП. Можливість програмування користувачем зробила ПЗП цього типу надзвичайно зручними при розробці мікроЕОМ [13,16].

Найбільшого поширення набули ПЗП з *ультрафіолетовим стиранням* серії K573, з плавкими перемичками серії K556 і K541, з електричним стиранням і записом інформації серій K558, K1601, K1609.

У всіх перерахованих типах пристроїв, що запам'ятовують, елементи, що зберігають інформацію, також розташовуються у вигляді комірок двовимірної матриці.

Кожна комірка може зберігати один біт інформації, тобто в змозі набувати значення логічного «0» або «1».

Фізично на кристалі мікросхеми ПЗП осередки розташовуються на

перетині «словарних ліній», що йдуть від дешифратора, і розрядних ліній, перпендикулярних словарним, які під'єднуються до входів мультиплектора. На дешифратор і мультиплексор подаються розряди адреси. При подачі адреси на дешифратор збуджується одна із словарних ліній і всі елементи, що запам'ятовують, розташовані на ній паралельно видають інформацію, що зберігається в них, на всі розрядні лінії. Вибірка потрібного числа бітів для подачі на вихід мікросхеми ЗП здійснюється мультиплексором. Залежно від організації мікросхеми, мультиплексор і дешифратор можуть мати різну розрядність. Наприклад, мікросхема місткістю (2x8) кбіт може бути організована як матриця розміром 128x128, що означає використання усередині мікросхеми дешифратора «1-з-128» для збудження словарних ліній і восьми мультиплексорів «16-в-1» для прочитування розрядних ліній.

З урахуванням топологічних і технологічних особливостей кожного типу мікросхем, можна провести ділення матриці осередків, що запам'ятовують, на блоки інших розмірів. Подібна побудова пристроїв, що запам'ятовують, є загальною для всіх типів. Відмінності між ними – в організації осередків, що запам'ятовують, розташованих на перетині «словарної» і «розрядних» ліній.

Мікросхеми з плавкими перемичками, виконані по ТТЛ- або ТТЛШ-технологіях, застосовуються там, де необхідна висока швидкодія. На їх основі створюється пам'ять мікропрограм для мікропроцесорних пристроїв з розрядно-модульною архітектурою (серія К589 і ін.), пристрої перемножування і функціонального перетворення сигналів. Елементом, що запам'ятовує, в мікросхемах даного типу є – *n-p-n-транзистор*, приєднаний базою до «словарної лінії», колектором до $+U_{п}$, а емітером, через плавку перемичку, – до «розрядної» лінії. Як плавка перемичка використовується полікристалічний кремній або ніхром, намілені при виготовленні мікросхеми.

Протікання струму програмування через ніхромову перемичку викликає часткове випаровування і окислення ніхрому, що приводить до розриву перемички. Проте, після закінчення деякого часу, така перемичка може відновитися, тому для підвищення надійності програмування проводять

електротермотренування мікросхем. Подібного недоліку позбавлені мікросхеми з перемичками із полікристалічного кремнію, в яких процес необоротного переходу полікремнія з провідного стану в непровідний відбувається під дією нагріву, викликаного протіканням струму.

При збудженні «словарної лінії» активізуватимуться (переходитимуть в стан «1») лише ті «розрядні» лінії, до яких приєднані транзистори з невиплавленими перемичками. Таким чином, процес програмування для мікросхем даного типу зводиться до видалення плавких перемичок в необхідних місцях.

Схеми підтримки режиму програмування звичайно розташовуються на самому кристалі мікросхеми, і процес програмування протікає таким чином. На адресні входи подається адреса вибраного осередку. Напруга живлення мікросхеми підвищується до напруги програмування, необхідної для створення струму, достатнього для виплавлення перемички. Далі на виходах мікросхеми, шляхом задання струму, указуються ті розряди слова, в яких виплавлятимуться перемички. В процесі занесення інформації в мікросхему, необхідна послідовність подачі імпульсів напруги на певні виводи забезпечується програмуючим пристроєм, який паралельно контролює правильність програмування, прочитуючи інформацію з ПЗП. Постійні ЗП даного типу допускають тільки одноразовий запис інформації в комірку.

Мікросхеми, в яких інформація стирається за допомогою ультрафіолетового випромінювання (УФППЗП), мають: можливість багатократного програмування, достатньо малі час вибірки і енергоспоживання, велику місткість. Це робить їх переважнішими для застосування як пам'ять мікропроцесорних систем із збереженням інформації після відключення живлення. Мікросхеми даного типу використовують в блоках ПЗП більшості мікроЕОМ.

Елементом, що запам'ятовує, в ПЗП з УФ-стиранням є МОП-транзистор, розташований на перетині відповідних «словарної» і «розрядної» ліній. Інформація про вміст даної комірки зберігається у вигляді заряду на другому

(плаваючому) затворі МОП-транзистора. Затвор називається *плаваючим*, якщо він розміщений між затвором даного транзистора, що управляє, і його каналом та оточений високоомним діелектриком.

Перепрограмуючі ПЗП – це ПЗП із змінним вмістом. На затворах матриці МОП-транзисторів тривалий час можуть зберігатися заряди, які створюють заданий код. Всі перепрограмуючі ПЗП являються МОП-приладами [15,16].

При необхідності в перепрограмуванні мікросхеми заздалегідь записану інформацію стирають ультрафіолетовим променем через прозоре кварцеве віконце на поверхні корпусу мікросхеми. Потрапляючи на плаваючий затвор і вибиваючи з нього фотоелектрони, УФ випромінювання розряджає плаваючий затвор МОП-транзистора. Час збереження інформації в мікросхемах ПЗП даного типу визначається якістю призатворного діелектрика і для сучасних мікросхем складає десять років і більше.

Мікросхеми ПЗП з електричним стиранням інформації популярні у розробників мікропроцесорної техніки завдяки можливості швидкого стирання і запису, великим допустимим числом циклів перезапису інформації (10000 разів і більше). Проте вони достатньо дорогі і складні в порівнянні з мікросхемами ПЗП з УФ стиранням і тому поступаються останнім по ступеню використання в мікропроцесорній апаратурі.

Основу комірки, що запам'ятовує, в ПЗП з електричним стиранням складає МОП-транзистор з плаваючим затвором, такий же, як і в ПЗП з УФ стиранням. Але в мікросхемах даного типу технологічними методами забезпечена можливість зворотнього тунелювання, тобто відбору електронів з плаваючого затвора, що дозволяє вибірково стерати занесену інформацію.

3.4. Зовнішні запам'ятовуючі пристрої

Зовнішні запам'ятовуючі пристрої (ЗЗП) – пристрої великого об'єму (сотні тисяч байтів і більше), призначені для тривалого зберігання даних, створення архіву даних (у тому числі і на знімних носіях) і обміну цими даними

з оперативними пристроями мікропроцесорних систем, що запам'ятовують [8]. У загальному випадку ЗЗП складається з накопичувача і контролера.

Накопичувачем ЗЗП називається частина зовнішнього пристрою, що запам'ятовує, що безпосередньо приймає інформацію, та перетворює її (при необхідності) у форму, зручну для зберігання (наприклад, в магнітне поле), що здійснює зберігання цієї інформації і перетворення в електричні сигнали при прочитуванні.

Накопичувач ЗЗП складається з пристрою перетворення інформації, механізму (при необхідності) і носія інформації. Пристрій перетворення інформації включає підсилювачі запису і прочитування, магнітні головки (для магнітних ЗЗП) і інші засоби, необхідні для прямого і зворотнього перетворення цифрових кодів у форму, зручну для зберігання. Механізм, іноді званий приводом, забезпечує переміщення носія інформації щодо органів запису і прочитування. Носієм інформації є виріб, що здійснює безпосереднє зберігання інформації на основі певних фізичних явищ. Носій може бути магнітним (диск, стрічка, барабан), оптичним (фотоплівка, голограма) і т.п. Носій може бути постійним, тобто невіддільним від накопичувача, або з'ємним, що забезпечує можливість створення архіву даних довільного об'єму.

Контролером ЗЗП називається пристрій, що забезпечує управління роботою накопичувача і сполучення з інтерфейсом мікропроцесорної системи. Специфіка ЗЗП поміщена в його накопичувачі. Розглянемо типові накопичувачі ЗЗП.

Накопичувач на магнітній стрічці (НМС) – накопичувач ЗЗП, носієм інформації якого є магнітна стрічка. Такі накопичувачі котушкові і касетні. У мікропроцесорних системах застосовують переважно касетні накопичувачі на магнітній стрічці (КНМС), тобто накопичувачі, магнітна стрічка яких поміщена в спеціальну з'ємну касету. Існує два види таких касет: компакт-диск-касета, ідентична компакт-диск-касеті побутових магнітофонів, і спеціальна цифрова касета типу Картрідж.

У касеті компакт-диска приймальна і подаюча котушки магнітної стрічки розміщені в одній площині, привід здійснюється провідним валиком за магнітну стрічку з фрикційною підмоткою обох котушок.

У касетах типу Картридж приймальна і подаюча котушки магнітної стрічки розміщені в паралельних площинах одна під іншою. Привід здійснюється провідним валиком за допомогою нескінченної стрічки, що огинає приймальну і таку, що подає, котушки. Такий привід значно спрощує механізм (виключається фрикційний привід котушок) і підвищує стабільність швидкості обертання стрічки, але ускладнює касету.

Накопичувач на магнітних дисках (НМД) – накопичувач ЗЗП, носієм інформації якого є диск, що обертається, покритий тонкою плівкою магнітного матеріалу. Такі накопичувачі бувають з жорсткими або гнучкими дисками. У НМД з жорсткими дисками носієм інформації є металевий диск (звичайно з міцного алюмінієвого сплаву) або пакет дисків. Ці накопичувачі відрізняються найвищою швидкістю обміну інформації зі всіх видів магнітних ЗЗП з рухомим носієм. У мікропроцесорних системах широке розповсюдження одержали НМД типу Вінчестер, які відрізняються тим, що жорсткий диск з полегшеними плаваючими головками виконаний в єдиному (іноді знімному) герметизованому кожусі, що значно спрощує його експлуатацію і забезпечує можливість отримання максимально можливої щільності запису інформації.

Накопичувач на гнучких магнітних дисках (НГМД) – НМД, змінним носієм інформації якого є магнітний диск, виконаний на гнучкій полімерній основі.

Накопичувач на магнітних картах – накопичувач ЗЗП, носієм інформації якого є змінна магнітна карта, виконана на гнучкій полімерній основі.

Накопичувачі на циліндрових магнітних доменах (ЦМД) – накопичувач ЗЗП, носієм інформації якого є циліндрові магнітні утворення (доменні структури), сформовані в монолітному кристалі. Носії інформації на ЦМД виконуються у вигляді інтегральних мікросхем, що містять 256К, 1М, 4М біт

інформації і більш.

Накопичувач на компакт-дисках (CD - дисках) – накопичувач ЗЗП, носієм інформації якого є рельєфна підкладка з полікарбонату, на яку нанесений тонкий шар металу (зазвичай – алюмінію), що відбиває світло. При записі матриці компакт-диска лазерний промінь пропалює в ній крихітні ямки. Після цього з матриці штамнуються в цеху безліч полікарбонатних копій. Потім рельєфна основа металізується, додається ще один, тонший шар лаку, що захищає тонку металеву поверхню.

При читанні диска інший промінь, що «читає», відображається від ямок і «чистих» ділянок по-різному. Ямка дає сигнал «нуль», оскільки ямки розсіюють промінь і не дають йому повернутися, а ділянка, що відображає світло, – «одиницю». Читання даних, які записані на компакт-диску, проводиться за допомогою пристрою (дискководу), що задовільняє виробленому єдиному стандарту CD-ROM. «Класичним» вважається компакт-диск діаметром 12 см, який може вміщувати до 650 Мб даних або 74 хвилини аудіо інформації. На ринку є компакт-диски із збільшеною місткістю, вміщаючих 700 Мб даних або 80 хвилин і 800 Мб даних або 90 хвилин, і диски-малятка з діаметром 8 см і місткістю 340 Мб даних.

Більш ємкими в порівнянні з CD-ROM і перспективними є *нові носії інформації на DVD-дисках*. Накопичувачі на DVD-дисках навіть в найпростішому варіанті – у вигляді одностороннього одношарового диска мають місткість від 3,2 Гб до 4,7 Гб, що у вісім разів перевищує місткість сьогоденних дисків. Місткість диска DVD, багат шарового і двостороннього, може досягати 17 Гб.

Два нині використовуваних стандарти оптичних носіїв CD і DVD базуються на червоних лазерах з довжиною хвилі 789 нм і 650 нм відповідно. Для того щоб збільшити ємність диску і зберігти його невеликий розмір, застосовують короткохвильові лазери. Нові стандарти – Blu-ray і HD DVD – передбачають на випромінювання з довжиною хвилі 405 нм, яка знаходиться у блакитному секторі спектру. Ця специфікація HD DVD-ROM у версії 1.0

вийшла в світ в червні 2004 року. Диски HD DVD-ROM мають ємність 15 Гб на шар. Таким чином, односторонній двохшаровий носій здатний вмістити 30 Гб. Такі ж показники мають заготовки HD DVD-R і HD DVD-R DL. Перезаписувані HD DVD-RW дозволяє зберігати до 20 Гб на одній стороні. Перевагою дисків є те, що в них використовуються захисний шар товщиною 0,6 мм, такий же як і в звичайних DVD. Тому оптична система головки HD DVD майже не відрізняється від DVD.

Тоді ж, у вересні 2004 року, появився формат Blu-ray Disc (BD). BD-ROM версії 1.0 має ємність одного шару 25 Гб, а двох – 50 Гб, що значно більше ніж у носіїв HD DVD. Другою важливою відмінністю BD від попередників є те що захисний шар дисків з великою ємністю, під яким зберігається реєстраційна плівка, значно тонший – всього 0,1 мм. Тому оптична головка приводів Blu-ray має більш складний устрій, здатний випромінювати лазерний промінь на трьох довжинах хвиль (не одночасно). В перспективі, до 2007 року, передбачається розробити чотирьох шарові диски Blu-ray, які після ще одного подвоєння числа шарів здатні вмістити 200 Гб.

Flash пам'ять – зовнішня для МП пам'ять, що реалізована мікросхемно і виконує функції твердого жорсткого диску та має високу вібро і ударостійкість.

Контрольні питання і завдання

1. Викладіть принцип дії і структуру ОЗП.
2. Назвіть особливості енергозалежної і незалежної динамічної і статичної пам'яті.
3. Перерахуйте типи ПЗП і дайте їм характеристику.
4. Для чого використовується прямий доступ до пам'яті?
5. Приведіть класифікацію систем пам'яті і порівняйте їх швидкодію.
6. На яких елементах побудовані статичні і динамічні ОЗП?
7. Назвіть основні характеристики пристроїв пам'яті та приведіть приклади сучасних пристроїв пам'яті і їх питомі показники.

8. Визначте інформаційну місткість в бітах і кількість ліній шини даних для пристроїв пам'яті, позначених: 1) $1024 \cdot 8$; 2) $4048 \cdot 16$.

9. Визначте питому вартість модуля динамічної пам'яті, якщо її місткість складає 128 Мб, а вартість рівна 166 грн.

10. Назвіть особливості CD-R, CD-RW і DVD компакт- дисків.

11. Викладіть особливості накопичувачів на магнітних картах.

12. Приведіть пристрій, принцип дії і особливості накопичувачів на магнітних дисках типу Вінчестер.

13. Назвіть відмінні особливості накопичувачів на гнучких дисках.

14. Назвіть особливості Flash-пам'яті.

15. Який об'єм пам'яті вміщається на класичному компакт-диску CD-ROM діаметром 12 см?

16. Назвіть місткість одношарових і багатшарових дисків DVD.

17. Викладіть технологію запису інформації на компакт-диск CD-R, CD-RW, і DVD.

18. Викладіть призначення і особливості кеш-пам'яті.

19. Приведіть типи сучасних вінчестерів і їх характеристики.

20. Які пристрої використовуються для запису інформації на дискети і компакт-диски?

21. Назвіть пристрої, використовувані для прочитування інформації з дискет, компакт-дисків і порівняйте їх технічні характеристики.

22. Охарактеризуйте пристрої, використовувані для читання і запису інформації на DVD диски, і приведіть їх технічні параметри.

23. Викладіть призначення, побудову і особливості надоперативної пам'яті.

24. Порівняйте місткості компакт-дисків CD-R і DVD.4.

25. Які фізичні процеси проходять у Flash-пам'яті при занесенні даних?

26. Як розміщується інформація в постійних запам'ятовуючих пристроях?

27. Що може зберігати кожна комірка пам'яті?

4 ІНТЕРФЕЙС МІКРОПРОЦЕСОРНИХ СИСТЕМ

Основною функцією інтерфейсу є організація взаємодії із *зовнішніми пристроями* (джерелами і приймачами даних). МП вступає у взаємодію із зовнішнім середовищем за допомогою периферійних пристроїв (пристроїв вводу-виводу). До найбільш часто використовуваних пристроїв *вводу* відносяться: клавіатура, перемикачі, двійкові датчики, аналого-цифрові перетворювачі, а до пристроїв *виводу* – дисплеї, індикатори, світлодіоди, цифроаналогові перетворювачі, друкарські пристрої, реле. Прикладом пристроїв вводу-виводу є накопичувачі на магнітній стрічці, а також гнучких і твердих магнітних дисках і ін. Пристрої вводу-виводу відрізняються швидкодією, розрядністю, протоколами обміну, сигналами, що управляють, особливостями програмування. Звичайно інтерфейс складається з одного або декількох портів вводу-виводу і їх схем управління.

4.1. Визначення і функції інтерфейсу

Для включення мікропроцесора в будь-яку мікропроцесорну систему необхідно встановити єдині принципи і засоби його сполучення з рештою пристроїв системи. Для цих цілей служить *уніфікований інтерфейс*, що є сукупність правил, які встановлюють єдині принципи взаємодії пристроїв мікропроцесорної системи. До складу інтерфейсу входять: апаратурні засоби з'єднання (роз'єм і зв'язки), номенклатура і характер зв'язків, програмні засоби, що описують характер сигналів інтерфейсу і їх тимчасову діаграму, а також опис електрофізичних параметрів сигналів.

Для інтерфейсу вводу-виводу характерні чотири функції [4]: буферування інформації; дешифрування адреси або вибір пристрою; дешифрування команди; синхронізація і управління. Буферування необхідне для синхронізації обмінів даними між процесором і периферійними пристроями. Дешифрування адреси потрібне для вибору конкретного пристрою вводу-виводу в системах, що

використовують декілька периферійних пристроїв. Дешифрування команди проводиться для пристроїв, які окрім передачі даних виконують і інші дії, наприклад зворотнє перемотування стрічки. При реалізації будь-якої з перерахованих вище функцій, проводиться синхронізація і управління процесами.

Складність виконання розгалужених зв'язків між різними вузлами при проектуванні ВІС і пристроїв на їх основі привела до того, що практично реалізовані і набули широкого поширення магістральні структури зв'язків, до яких підключені входи і виходи електронних вузлів (блоків).

Єдина інформаційна магістраль мікропроцесорної системи зв'язує між собою всі пристрої і функціонально складається з інформаційних шин адрес, даних і сигналів управління. Часто обмін адресами і даними здійснюється по одній групі шин в режимі розділення часу.

Шина адрес. У простій мікропроцесорній системі тільки мікропроцесор може виробляти адреси передаваної в системі інформації. Тому шина адрес однонаправлена. Мікропроцесор генерує сигнали коду адреси, а решта пристроїв, підключених до шини адрес, може тільки сприймати їх, виконуючи безперервно мікрооперацію пізнання коду адреси. Кількість ліній шини адрес співпадає з розрядністю передаваного коду адреси. Якщо використовується 16-розрядний код, то в системі дозволяється вироблення $2^{16} = 65536$ адрес. Вони можуть всі відноситися до адрес елементів пам'яті або до адрес елементів пам'яті і адрес регістрів даних пристроїв вводу-виводу.

Шина даних. Мікропроцесор, а також ОЗП, ЗЗП, дисплеї можуть сприймати або передавати дані. Інші пристрої можуть або тільки приймати дані (наприклад, пристрій друку), або тільки видавати їх (наприклад, ПЗП).

Шина даних є двонаправленою, оскільки необхідно забезпечити всі можливі зв'язки системи. Її розрядність визначається розрядністю мікропроцесора і дорівнює 2, 4, 8, 16, 32 біт. Якщо в мікропроцесорі обробляються дані за програмами подвійної розрядності, то подвійне слово пересилається за два цикли, тобто в цьому випадку має місце тимчасове

мультиплексування. Воно застосовується також, коли використовується загальна (мультиплексована) шина адрес і даних.

Шина сигналів управління. Мікропроцесор, а також деякі з пристроїв вводу-виводу, генерують сигнали, що управляють, призначені для синхронізації і визначення типу операцій, що виконуються пристроями. Ці сигнали передаються по сукупності ліній, в цілому створюючих шину сигналів управління. Всі сигнали управління в електронній системі узгоджені з системними сигналами синхронізації. Вони задають початок і послідовність спрацьовування різних пристроїв системи, блоків і вузлів усередині всіх кристалів ВІС. Для задання головної послідовності синхронізуючих імпульсів звичайно використовують зовнішній кварц або генератор на його основі. Видавані мікропроцесором сигнали синхронізації бувають однофазними, рідше двофазними.

Кожен мікропроцесор має свою унікальну систему сигналів управління. Але практично всі мікропроцесори мають загальні сигнали, серед них, наприклад, вхідний сигнал СКИДАННЯ, який виробляється на пульті управління системи. Він приводить до скидання всіх внутрішніх регістрів мікропроцесора і завантаження лічильника команд – вузла, що визначає послідовність виконання команд програми, – початковим значенням адреси, де записана перша команда програми.

Найважливіша функція мікропроцесора, що управляє, – визначення напряму потоків даних в системі. Мікропроцесор викликає слова команд з пам'яті в процесі їх читання, звертається в пам'ять за операндами або до зовнішніх пристроїв за новими даними, може записати результат операції в пам'ять або, сформувавши масив даних, визначити необхідність їх виводу на зовнішні пристрої. Коли мікропроцесор посилає дані якомусь пристрою, відбувається операція запису даних, а коли одержує дані від якогось пристрою, то прочитає дані з його інформаційного регістра і виконує операцію їх читання. Щоб задати напрям передачі по шині даних, мікропроцесор генерує сигнали ЧИТАННЯ/ЗАПИС, передавані по одній з ліній шини управління.

Специфіка деяких пристроїв вводу-виводу даних така, що інформація може бути втрачена, якщо МП своєчасно не здійснить зв'язок з пристроєм. Тому ці пристрої генерують сигнали ЗАПИТ ПЕРЕРИВАННЯ ПРОЦЕСОРА, що звертають увагу мікропроцесора на стан своєї готовності (або несправності). Мікропроцесор має вхід для прийому, принаймні, одного сигналу ЗАПИТ ПЕРЕРИВАННЯ ПРОЦЕСОРА. Якщо ж запит приймається, то МП інформує систему, виробляючи у відповідь сигнал, ЗАПИТ ПЕРЕРИВАННЯ ЗАДОВОЛЕНИЙ.

Різна швидкість роботи пристроїв вводу-виводу і мікропроцесора породжує необхідність припинення роботи процесора на час підготовки даних в зовнішньому пристрої. Для цього застосовується режим роботи «Очікування» мікропроцесора, який визначається зовнішнім сигналом, ДАНІ ПІДГОТОВЛЕНІ (ДАНІ НЕ ПІДГОТОВЛЕНІ). Всього по шині управління передається до десяти різноманітних сигналів управління і більше.

Якщо необхідно перейти від одного виду інтерфейсу до іншого, то застосовують спеціальні апаратні засоби у вигляді перетворювача інтерфейсів або інтерфейсного контролера.

При побудові мікропроцесорних систем найчастіше здійснюються перетворення, пов'язані з різними форматами електричних сигналів.

Всі МП обробляють цифрові дані, представлені в паралельній формі. В цьому випадку розряди слів даних передаються одночасно по інформаційній шині і обробляються паралельно у всіх розрядах арифметико-логічного блоку мікропроцесора, тому усередині електронної системи, всі передачі даних також проводяться в паралельному форматі. Але в периферійній частині електронних систем можуть бути різноманітні формати інформаційних сигналів, серед яких найбільш важливими є аналогові і цифрові послідовні.

Аналого-цифрові і цифро-аналогові перетворювачі у вигляді ВІС вирішують задачі перетворення аналогових сигналів в паралельні коди і навпаки. Розвиток засобів цих перетворювачів, що управляють, дозволяє не тільки спростити їх інтерфейс з МП, але практично забезпечити пряме з'єднання

без додаткових апаратурних засобів. Дані в цифровому послідовному форматі передаються по одній інформаційній лінії. Це знижує кількість зв'язків в периферійній частині систем у випадках, коли не потрібне сполучення з швидкодіючими периферійними пристроями. Ці дані можуть прямо вводитися (або виводитися) до МП, для чого необхідно розробити програмні модулі прийому і перетворення форматів даних з відповідною синхронізацією МП і зовнішніх пристроїв. Тимчасові витрати і низьку ефективність такого рішення можна подолати переходом до апаратурної реалізації системи вводу-виводу даних на основі використання спеціальних ВІС контролерів – перетворювачів форматів даних, які одержали назву універсальних асинхронних приймачів-передавачів.

Асинхронна передача даних означає, що приймач (наприклад, МП) і передавач (наприклад, телетайп) здійснюють зв'язок в умовах, коли кожен має свою власну систему синхронізації, тому передавач посилає свої дані у будь-який момент незалежно від тимчасового стану приймача.

У приймачі повинні бути передбачені засоби аналізу і входження в «тимчасове зачеплення», тобто засоби синхронізації своєї роботи з роботою передавача.

4.2. Пристрої вводу-виводу інформації

При проектуванні мікропроцесорної системи дуже важливий правильний вибір методів управління вводом-виводом даних. Мікропроцесор має широкі можливості по організації вводу-виводу. Проте на ефективність вводу-виводу істотно впливають особливості технічних засобів вводу (виводу) даних, що є джерелами (приймачами). Мікропроцесор може взаємодіяти з різними типами джерел (приймачів) даних. У деяких зміна даних відбувається у будь-який момент часу і займає достатньо великий інтервал (в порівнянні з швидкодією мікропроцесора). Встановлені дані прочитують (записують) через достатньо широкий інтервал часу після установки. До подібних засобів вводу-виводу

даних відносяться засоби, призначені для взаємодії оператора і системи (наприклад, дисплей, клавіатура).

Є пристрої, що працюють значно повільніше, ніж мікропроцесор, але в періодичному режимі (наприклад, накопичувачі даних на магнітних дисках). У них дані можуть бути записані (прочитані) тільки в певні проміжки часу. Якщо саме в цей проміжок часу мікропроцесор не здійснить сеанс взаємодії, то дані будуть втрачені. В той же час, якщо мікропроцесор тільки стежитиме за станом таких пристроїв, то він не зможе виконувати інші функції в системі. Тому в системі повинно бути вирішено завдання розділення пасивних і активних інтервалів взаємодії. Наприклад, в зовнішні пристрої можна ввести додаткові технічні засоби індикації активного стану, а в МП – засоби відповідної реакції на сигнал виникнення активного стану в системі.

У системі можуть бути дані, що знімаються з датчиків стану керованих об'єктів (процесів), швидкість зміни і граничні значення яких повинні бути предметом особливого контролю з боку мікропроцесора.

У реальній системі можуть бути об'єднані численні сигнали різноманітних даних, що вводяться і виводяться. Один з раціональних методів здійснення вводу-виводу в такій системі полягає в організації асинхронного режиму роботи мікропроцесора і джерел (приймачів) інформації. У цьому випадку до обміну даними завжди відбувається обмін асинхронними керуючими сигналами. Обмін по перериваннях роботи мікропроцесора може початися тільки тоді, коли зовнішній пристрій виробить сигнал готовності (ЗАПИТ НА ПЕРЕРИВАННЯ), а МП, завершивши обов'язкові дії над виконуваною командою програми, виробить у відповідь сигнал ЗАПИТ ЗАДОВОЛЕНИЙ. Потім мікропроцесор переходить до підпрограми обслуговування запиту переривання, яка містить послідовність команд обміну даними. Після закінчення підпрограми переривання мікропроцесор повертається до роботи за раніше перерваною програмою.

У всіх мікроЕОМ застосовується програмно-керована передача даних. При даному методі передачі, спеціальна програма управляє обміном даними.

Відомі три типи програмно-керованої передачі даних: синхронна, асинхронна і з перериванням програми.

Синхронна передача даних характерна для периферійних пристроїв, для яких відомі часові співвідношення. При цьому типі передачі пристрій вводу-виводу повинен бути готовий до прийому або передачі даних за час, рівний часу виконання певної команди процесора. Синхронна передача реалізується при мінімальних витратах технічних і програмних засобів.

Асинхронна передача даних, іноді звана обміном за допомогою «рукостискання», широко використовується в мікроЕОМ. При такій передачі даних ЕОМ перед виконанням операції вводу-виводу перевіряє стан периферійного пристрою. Звичайно при обміні необхідно: 1) перевірити стан пристрою; 2) активізувати пристрій, якщо останній готовий до обміну; 3) передати дані (ввести або вивести); 4) дезактивувати пристрій. На першому кроці виконується команда вводу інформації про стан вибраного пристрою. Далі виконується умовний перехід по вмісту розряду стану пристрою. Якщо пристрій зайнятий, в програмі реалізується перехід для повторної перевірки його стану. У разі готовності пристрою видається команда на ввід або вивід даних. При завершенні передачі пристрій дезактивується. Асинхронна передача є ідеальною в сенсі узгодження тимчасових відмінностей між периферійними пристроями і процесором. Недолік її в тому, що процесор вимушений чекати, поки периферійний пристрій не буде готовий до обміну. Це приводить не тільки до непродуктивних витрат часу мікроЕОМ (за наявності тривалих затримок), але у багатьох випадках є просто неприпустимим. Наприклад, у процесах управління, в цьому випадку, виникає необхідність збереження рівня сигналів управління на час очікування передачі. Методом, що дозволяє усунути подібні труднощі, є передача даних з перериванням програми.

Передача даних з перериванням програми – це такий тип обміну даними, при якому для виконання операції вводу-виводу проводиться переривання програми мікроЕОМ. Такий тип обміну особливо зручний при роботі з периферійними пристроями з низькою швидкодією, а також у випадках, коли

момент передачі даних у мікроЕОМ непередбачуваний, наприклад, при роботі з каналами зв'язку. Основна характерна риса даної передачі така – обмін даними між мікроЕОМ і периферійними пристроями ініціюється самими зовнішніми пристроями. Для реалізації даного типу обміну необхідно цикл очікування при асинхронній передачі даних замінити еквівалентним циклом перевірки наявності зовнішнього переривання, що виконується за допомогою апаратних засобів. За час виконання кожної машинної операції мікропроцесор автоматично перевіряє наявність сигналу переривання.

При передачі даних з перериванням програми реалізується така послідовність дій:

1. Периферійний пристрій запрошує переривання програми, що виконується процесором.
2. Після виконання поточної команди процесором, останній видає сигнал підтвердження готовності до переривання.
3. Процесор запам'ятовує вміст лічильника команд і здійснює перехід за адресою підпрограми обробки переривання.
4. Запам'ятовується вміст внутрішніх реєстрів (робочих реєстрів і реєстра стану) і виконується передача даних під управлінням спеціальної програми.
5. Здійснюється повернення до продовження виконання основної програми.

У мікропроцесорах звичайно, використовується два способи реалізації описаної послідовності дій: переривання з опитуванням і переривання по вектору. У разі переривання з опитуванням за допомогою технічних або програмних засобів опитуються периферійні пристрої до тих пір, поки не виявиться той, що запрошує переривання. Далі МП переходить на відповідну підпрограму обслуговування переривання, яка і виконує обмін даними. Пріоритет пристрою визначається його місцем у послідовності опиту. У разі переривання по вектору управління по запиту пристрою безпосередньо передається на відповідну підпрограму обслуговування. У системах з

перериванням по вектору всі пристрої володіють однаковим пріоритетом. У даному випадку опитування не вимагається, тому час реалізації переривання менший, ніж при виконанні переривання з опитом.

Багаторівневі, або каскадовані, переривання можуть застосовуватися в багатьох типах мікропроцесорів при використанні прапорців дозволу для маскуванню і демаскуванню окремих рівнів. Така можливість особливо корисна в системах, де використовуються периферійні пристрої, як з низькою, так і з високою швидкістю. Розглянуті методи обміну даними призначені для передачі інформації між зовнішніми пристроями і мікропроцесором. Для обміну даними між зовнішнім пристроєм і пам'яттю немає необхідності пересилати дані через мікропроцесор. Звичайно такий обмін полягає в пересилці масивів інформації. Якщо проводити читання даних в пам'яті (зовнішніх пристроях), заносити дані в акумулятор мікропроцесора, а потім записувати в регістри зовнішніх пристроїв (або пам'ять), то витрати часу мікропроцесора будуть дуже великими. Можна ввести в систему контролер прямого доступу в пам'ять, який бере на себе управління передачею, звільнивши від цих функцій системний мікропроцесор. Прямий доступ в пам'ять при виконанні операцій вводу-виводу значно збільшує швидкість передачі даних і підвищує ефективність використання засобів мікропроцесора. Побудова каналу прямого доступу в пам'ять є альтернативою програмному обміну, тому справедливі загальні закономірності балансування програмно-апаратних засобів. Якщо певні функції виконуються апаратно, то це спрощує програмне забезпечення, значно скорочує витрати часу на виконання функцій в порівнянні з їх програмною реалізацією, але вимагає використання додаткових апаратних витрат. Останнє збільшує вартість, габаритні розміри і потужність споживання системи, знижує її надійність.

У мікропроцесорних системах завдання розділення єдиного інформаційного каналу між мікропроцесором і каналом прямого доступу в пам'ять розв'язується шляхом використання властивостей трирівневого стану інформаційних магістралей.

Мікропроцесор під час передачі інформації по каналу прямого доступу переводить вихідні схеми управління магістралями даних, адрес і керуючих сигналів, у *високоімпедансний стан*, тим самим ізолюється від решти частини системи, що аналогічно обриву його інформаційного каналу.

Ступінь впливу операцій вводу-виводу в каналі прямого доступу на основний обчислювальний процес в системі визначається складністю апаратних засобів контролера цього каналу. У простому випадку контролер бере на себе управління передачею даних по інформаційній магістралі в умовах, коли мікропроцесор одночасно з видачею сигналу задоволення запиту каналу, переводиться в режим очікування завершення операцій в каналі. За рахунок перекладу вихідних каскадів управління шинами інформаційної магістралі у *високоімпедансний стан* мікропроцесор ефективно електрично і інформаційно відключається від системи. Стан внутрішніх регістрів «заморожується», тобто мікропроцесор в режимі очікування зберігає той інформаційний стан, який виник в ньому до моменту задоволення запиту каналу прямого доступу.

Побудова контролером каналу прямого доступу, що функціонує паралельно з мікропроцесором, дозволяє не відключати останній на час операцій вводу-виводу. Проте, операції вводу-виводу, з використанням передачі по інформаційній магістралі, треба синхронізувати з іншими операціями в системі і виконувати їх при вільній магістралі.

Мікропроцесор після звернення до пам'яті за черговою командою розшифровує її і аналізує, в цей проміжок часу адреси звернення до пам'яті ще не відомі. Тому канал прямого доступу може «зайняти» цикл звернення до пам'яті і здійснити операцію вводу або виводу слова даних. Цей метод заняття циклу звернення до пам'яті, контролер прямого доступу здійснює так, що мікропроцесор навіть «не знає», що одночасно з його функціонуванням відбуваються інші операції в системі, оскільки вони ніякого гальмування роботи мікропроцесора не викликають.

До істотних недоліків каналу прямого доступу із заняттям циклу відноситься обмеження його продуктивності швидкодією мікропроцесора. Це

обумовлено тим, що обмін даними між пам'яттю і зовнішніми пристроями здійснюється з швидкістю, визначуваною частотою вибірки команд.

4.3. Інтерфейс каналу мікроЕОМ

Як приклад розглянемо інтерфейс серійних мікроЕОМ [9].

Інтерфейс такого типу застосовується у ряді вітчизняних і зарубіжних мікроЕОМ і мікропроцесорних комплектів, наприклад, в МПК серії K588, опис якого приведений в [17].

Канал обміну інформацією мікроЕОМ є простою, швидкодіючою системою зв'язку, що сполучає центральний процесор, пам'ять і всі зовнішні пристрої. Всі модулі, підключені до каналу мікроЕОМ, використовують одні і ті ж канали зв'язку. Канал містить 38 ліній зв'язку, з яких 31 лінія є двонаправленою.

Зв'язок між двома пристроями, підключеними до каналу, здійснюється за принципом «керівник – керований» (активний – пасивний). У будь-який момент часу тільки один пристрій є активним.

Активний пристрій управляє циклами звернення до каналу, задовільняє при необхідності вимоги переривання від зовнішніх пристроїв, контролює надання прямого доступу до пам'яті. Пасивний пристрій (керований) є тільки виконавчим пристроєм і може приймати або передавати інформацію тільки під управлінням активного пристрою. Типовий приклад таких відносин – центральний процесор (активний пристрій), що вибирає команду з пам'яті (пасивний пристрій). Іншим прикладом може служити пристрій, що працює в режимі прямого доступу до пам'яті (пристрій ПДП) (активний пристрій) і пам'ять (пасивний пристрій).

Зв'язок через канал замкнутий, тобто на керуючий сигнал, передаваний активним пристроєм, повинен поступити у відповідь сигнал від пасивного пристрою. Тому процес обміну між пристроями залежить від довжини каналу і часу відгуку пасивного пристрою (в межах 10 мкс).

Асинхронне виконання операції передачі даних усуває необхідність в тактових імпульсах, внаслідок чого обмін з кожним пристроєм може відбуватися з максимально можливою швидкістю. Обмін між двома пристроями може виконуватися як 16-розрядними словами, так і байтами (вісім розрядів).

Канал мікроЕОМ забезпечує програмний обмін, включаючи режим переривання програми і обмін в режимі прямого доступу до пам'яті.

Як адреса, так і дані (слова або байти) передаються по одних і тих же 16 лініях адреси/даних.

Будь-який цикл звернення до каналу починається з адресації пасивного пристрою. Після завершення адресної частини циклу активний пристрій асинхронно приймає або передає дані і вимагає відповіді від адресованого пристрою.

Для виконання будь-якої команди процесору необхідно звернутися хоч би один раз до каналу. Для деяких команд потрібно виконати декілька операцій. Першою такою операцією для всіх команд є ввід даних з елемента пам'яті, адреса якої визначається лічильником команд (ЛК). Всі операції звернення до каналу для вводу і виводу даних називаються *циклами звернення до каналу*. Обмін інформацією в каналі здійснюється за допомогою трьох циклів: ЧИТАННЯ, ЗАПИС, ЧИТАННЯ–ПАУЗА–ЗАПИС.

Цикл ЧИТАННЯ. Напрямо передачі при виконанні операцій обміну даними визначається по відношенню до активного пристрою. При виконанні циклу ЧИТАННЯ дані передаються від пасивного пристрою до активного.

Цикл ЗАПИС. При виконанні циклу ЗАПИС дані передаються від активного пристрою до пасивного, наприклад, при записі даних в пам'ять.

Цикл ЧИТАННЯ–ПАУЗА–ЗАПИС. При виконанні циклу ЧИТАННЯ–ПАУЗА–ЗАПИС активний пристрій прочитує, а потім передає модифіковані дані за тією ж адресою.

У режимі прямого доступу до пам'яті обмін даними відбувається між периферійним пристроєм і пам'яттю без втручання процесора. Пристрій,

здатний працювати в режимі ПДП, повинен виконувати всі функції активного пристрою: адресацію, синхронізацію, вироблення керуючих сигналів, і при необхідності — управління регенерацією ОЗП.

Обмін даними в режимі ПДП проводиться стандартними циклами звернення до каналу, розглянутими раніше.

З докладним описом ліній і сигналів каналу мікроЕОМ МПК серії K588 можна ознайомитися в [9, 17].

4.4. Спеціалізовані периферійні пристрої

Спеціалізовані пристрої призначені для перетворення, первинної обробки і реєстрації інформації. Такі пристрої або обробляють проміжну інформацію, або є безпосередніми джерелами різного роду інформації, що обробляється в мікропроцесорних системах. До спеціалізованих пристроїв відносяться: таймери, лічильники подій, датчики, аналого-цифрові і цифроаналогові перетворювачі, модеми, калькулятори.

Таймери і лічильники подій використовують в мікропроцесорних системах для підрахунку різних подій або для відліку часу. При виконанні певних умов вони переривають роботу мікропроцесора і ініціюють певні дії системи. У мікропроцесорних керуючих і контролюючих системах широко використовують датчики параметрів, що безперервно змінюються, та дискретних параметрів, аналого-цифрові і цифро-аналогові перетворювачі, реєструючі пристрої.

Велике значення мають пристрої зв'язку окремих підсистем в розподілених обчислювальних керуючих системах.

Для підключення окремих мікроЕОМ і зовнішніх пристроїв до ліній зв'язку використовують модеми (модулятори-демодулятори). Вони перетворюють двійкову інформацію у форму, зручну для передачі по дальніх лініях зв'язку, забезпечують необхідні характеристики передаваних сигналів і виконують зворотнє перетворення інформації, що приймається. Їх роль росте із

збільшенням ступеня децентралізації процесу обробки інформації.

Для забезпечення зв'язку мікропроцесора з різними зовнішніми пристроями використовуються два види інтерфейсу: паралельний і послідовний. При використанні паралельного інтерфейсу, 8-розрядний мікропроцесор за кожну операцію обміну забезпечує передачу 8 біт інформації, 16-розрядний – 16 біт і т.д.

Багато пристроїв пов'язані з мікропроцесором за допомогою ліній послідовної передачі даних. Широке застосування способу послідовної передачі даних пояснюється обмеженнями, властивими способу паралельної передачі даних. Робоча відстань для лінії паралельного вводу-виводу обмежується 1–2 м. При збільшенні довжини кабелю зростає його місткість, тому передача даних на високій швидкості стає неможливою. Довжину ліній паралельної передачі даних можна збільшити до 10...20 м шляхом використання спеціальних формувачів і зменшення швидкості передачі. Проте подальше збільшення довжини ліній при паралельній передачі даних практично неможливе.

При послідовній передачі даних жорстких обмежень на довжину лінії не накладається. Але перш, ніж почати послідовну передачу даних, необхідно перетворити дані з паралельної форми в послідовну. Простий спосіб такого перетворення – перетворення за допомогою сдвигового регістра. Дані завантажуються в сдвиговий регістр, потім його вміст зсувається на один розряд під час вступу кожного тактового імпульсу. Дані на виході сдвигового регістра матимуть послідовну форму.

Щоб прийняти дані в послідовній формі і перетворити їх в паралельну форму, необхідно виконати дії, зворотні по відношенню до описаних вище. Дані, що поступають в послідовній формі, вводяться біт за бітом в сдвиговий регістр. Після заповнення сдвигового регістра, дані з нього в паралельній формі передаються в мікропроцесорну систему.

Пристрій, що забезпечує перетворення даних з паралельної форми в послідовну і зворотнє перетворення, називають *універсальним синхронно-асинхронним програмованим приймачем-передавачем* (УСАПП). Такий

пристрій часто реалізується у вигляді ВІС (наприклад, КР580ВВ51). Окрім перетворення форми представлення даних, приймач-передавач виконує функції контролю і управління. Для виявлення помилок при передачі даних УСАПП може використовувати контроль на парність або непарність. Після прийому даних приймач-передавач перевіряє в ньому контрольний біт. Якщо виявляється порушення парності (непарності), то приймач-передавач записує в свій регістр стану ознаку помилки. Потім УСАПП може видати запит на повторну передачу даних.

Деякі УСАПП виробляють сигнали, використовувані для встановлення зв'язку між двома модемами. Модем (модулятор –демодулятор) перетворить логічний сигнал певного рівня в сигнал, що модулюється зрушенням частоти. Модем може, наприклад, перетворювати двійковий передаваний сигнал в тоновий сигнал частотою 1270 Гц для логічної 1 і в тоновий сигнал частотою 1070 Гц для логічного 0. Сигнали таких частот можуть передаватися по телефонних лініях і по деяких довгих лініях послідовної передачі даних. Модульований сигнал можна пропускати через пристрої зв'язку змінного струму, наприклад трансформатори, які часто застосовують в довгих телефонних ланцюгах. Модеми використовують для модуляції і демодуляції сигналів, призначених для передачі по радіоканалу. Передача даних по радіоканалу за допомогою модемів часто застосовується в системах збору даних, для управління якими служить мікроЕОМ.

У мікропроцесорних системах управління широко застосовують *програмовані таймери* – пристрої, що формують задані, тимчасові інтервали. З їх допомогою здійснюється, наприклад, така операція, як реалізація обчисленого кута управління тиристорами при управлінні різним технологічним устаткуванням. За допомогою програмованого таймера виділяють часовий інтервал відповідно до заданого значення, яке у вигляді коду записується в пам'ять таймера.

Таймер можна реалізувати за допомогою центрального процесора системи (програмно), а також додатковими технічними засобами (апаратно).

Другий спосіб застосовується частіше, оскільки дозволяє значно зменшити непродуктивні витрати машинного часу.

Розрізняють *два типи таймерів: з попереднім і без попереднього накопичення сигналу.*

У таймері з *попереднім накопиченням сигналу* інформація, відповідна тривалості відмірюваного часового інтервалу, записується тільки на початку відліку цього інтервалу; по кожному рахунковому імпульсу вміст таймера зменшується на одиницю. Коли вміст таймера стає рівним нулю, на його виході з'являється сигнал, використовуваний для управління периферійними пристроями або для переривання програми мікроЕОМ.

У таймері *без попереднього накопичення сигналу* інформація може бути записана в довільний момент часу усередині часового інтервалу, при цьому можлива сигналізація про те, що інтервал більший, менший або рівний заданому.

Промисловість випускає *однокристальні ВІС таймерів*, наприклад К588ВІ1 [17]. Ця ВІС має два незалежні канали, організовані на базі двох 16-розрядних лічильників, що підсумовують, з послідовним перенесенням. До складу таймера входить незалежний 7-розрядний дільник частоти, виконаний на основі 7-розрядного двійкового лічильника, що підсумовує, з послідовним перенесенням і що має фіксовані коефіцієнти ділення 2, 4, 8, 16, 32, 64, 128.

В більшості випадків в мікропроцесорних системах управління сигнали зворотного зв'язку про регульовані параметри і задаючі дії представлені в аналоговому вигляді. Для вводу в мікроЕОМ і подальшої обробки таких сигналів їх представляють у вигляді цифрового коду. Для перетворення аналогових сигналів в цифровий еквівалент-код застосовуються *аналого-цифрові перетворювачі (АЦП).*

Існує декілька методів аналого-цифрового перетворення, що визначають в основному технічні характеристики виконаних на їх основі АЦП [18]. Технічні характеристики АЦП можна розбити на ряд груп: точності; часові; надійності; узагальнені вартісні.

У якості точностних характеристик використовують: кількість достовірних двійкових розрядів на виході АЦП, відносну погрішність і ін.

Часові характеристики або параметри в тій або іншій формі визначають швидкодію АЦП. Розрізняють три часові характеристики АЦП:

- *період квантування* – це інтервал між двома послідовними перетвореннями (величину, зворотню періоду квантування, називають *частотою квантування*);
- *тривалість циклу перетворення* – це затримка між моментом подачі вхідної величини на АЦП і моментом видачі коду;
- *час перетворення* – це часовий інтервал, протягом якого вхідний сигнал безпосередньо взаємодіє з АЦП.

Надійності характеристики і параметри оцінюють здібність АЦП до роботи під впливом різних дестабілізуючих чинників (часу, температури, вологості і ін.). Як параметри надійності можна брати самі різні величини (інтенсивність відмов, час напрацювання на відмову і ін.).

Узагальнені вартісні характеристики включають всю решту вимог.

У мікропроцесорних системах управління технологічним устаткуванням часто застосовують багатоканальні АЦП (АЦП з мультиплексованим входом). При цьому один АЦП послідовно перетворює в код ряд аналогових сигналів.

Для зв'язку АЦП з мікропроцесором часто застосовують спеціальні ВІС – контролери АЦП. У їх склад, як правило, входять: каналний інтерфейс для стиковки з системною магістраллю мікроЕОМ; мультиплексор для підключення до АЦП різних вхідних аналогових сигналів; буферний запам'ятовуючий пристрій для зберігання результатів перетворень (це дозволяє процесору прочитувати результат перетворення будь-якого сигналу в довільний момент часу); пристрій управління АЦП і інші сервісні пристрої.

Прикладом контролера АЦП, призначеного для підключення АЦП до системної магістралі мікроЕОМ, є ВІС К588ВГ4.

Для видачі інформації від мікроЕОМ в аналогові пристрої, зокрема на осцилографи, використовуються цифроаналогові перетворювачі (ЦАП). Для

підключення ЦАП до системної магістралі мікроЕОМ і виконання сервісних функцій, аналогічних виконуваним контроллером АЦП, існують ВІС контроллерів ЦАП.

Контрольні питання і завдання

1. Викладіть структуру паралельного інтерфейсу КР580ВВ55 і особливості його роботи в різних режимах.
2. Розкажіть про структуру послідовного інтерфейсу і особливості його застосування.
3. Назвіть інтерфейсні компоненти сучасних МП комплектів.
4. Розкажіть про обмін інформацією в режимах “Переривання” і “Прямий доступ в пам'ять”.
5. Назвіть склад і призначення периферійного устаткування мікропроцесорних систем.
6. Викладіть призначення регістра керуючого слова паралельного інтерфейсу МП КР580ВВ55 і приведіть його формат.
7. Складіть 16-річний код керуючого слова, і запрограмуйте паралельний інтерфейс МП КР580ВВ55 в режимі 0, канал А – ввід, канал В – вивід, молодші розряди каналу С – вивід, старші розряди каналу С – ввід.
8. Дайте характеристику режиму 0 паралельного інтерфейсу КР580ВВ55 і наведіть приклади його використання.
9. Приведіть приклади використання режиму 1 паралельного інтерфейса КР580 і вкажіть на особливості його роботи.
10. Викладіть особливості режиму 2 паралельного інтерфейса КР580ВВ55 і приведіть приклад його використання.
11. Назовіть основні технічні характеристики АЦП і ЦАП та приведіть приклади їх використання.
12. Викладіть особливості використання каналу С в режимах 0, 1, 2 паралельного інтерфейса КР580ВВ55 і наведіть приклади.

5. ОДНОКРИСТАЛЬНІ МІКРОКОНТРОЛЕРИ

Однокристалльні мікроконтролери (ОМК) – мікропроцесорні пристрої, призначені для регулювання і управління електротехнічними, електроенергетичними і іншими пристроями, а також технологічними процесами виробництва, випробувань і досліджень. Вони є пристроями, конструктивно виконаними в одному корпусі ВІС, яка містить всі складові мікропроцесорної системи: процесор, резидентну пам'ять даних і програм, програмовані інтерфейси. Високоінтегровані ОМК можуть мати АЦП, ЦАП, внутрішні температурні сенсори, послідовні порти UART, сторожові таймери і інші пристрої.

5.1. Мікроконтролери сімейства MCS51

Восьмирозрядні високопродуктивні однокристалльні мікроконтролери (ОМЕОМ) сімейства MCS51 виконані за високоякісною n-МОП технологією (серія 1816) і КМОП технологією (серія 1830). Сімейство MCS51 забезпечує: збільшення об'єму пам'яті команд і пам'яті даних, нові можливості вводу-виводу і периферійних пристроїв, розширює діапазон застосування і знижує загальні витрати системи. ОМЕОМ КР1816ВЕ51 і КР1830ВЕ51 містять програмовані в процесі виготовлення кристала ПЗП пам'яті програм, місткістю 4096 байт і розраховані на застосування в масовій продукції. За рахунок використання зовнішніх мікросхем пам'яті загальний об'єм пам'яті програм може бути розширений до 64 кбайт. ОМЕОМ КР1816ВЕ31 і КР1830ВЕ31 не містять вбудованої пам'яті програм, проте можуть використовувати до 64 кбайт зовнішньої постійної або перепрограмованої пам'яті програм і ефективно застосовуватися в схемах, що вимагають істотно більшого за об'ємом (чим 4 кбайт на кристалі) ПЗП пам'яті програм. Кожна з перерахованих вище мікросхем є відповідно аналогом ВІС 8051, 80С51, 8751, 8031, 80С31 сімейства MCS-51 фірми Intel (США). Кожна ОМЕОМ даного сімейства містить

вбудоване ОЗП пам'яті даних місткістю 128 байт з можливістю розширення загального об'єму оперативної пам'яті даних до 64 кбайт за рахунок використання зовнішніх мікросхем ЗПДВ. Порівняльні дані мікросхем приведені в табл.5.1.

Таблиця 5.1 – Порівняльні дані мікросхем

Мікросхема	Аналог	Об'єм внутрішньої пам'яті програм Кбайт	Тип пам'яті програм	Об'єм внутрішньої пам'яті даних байт	Максимальна частота проходження тактових сигналів МГц	Струм споживання мА
KP1816BE31	8031 AH	-	зовнішн.	128	12,0	150,0
KP1816BE51	8051 AH	4	ПЗП	128	12,0	150,0
KM1816BE751	8751H	4	ППЗУ	128	12,0	220,0
KP1830BE31	80C31BH	-	зовнішн.	128	12,0	18,0
KP1830BE51	80C51BH	4	ПЗП	128	12,0	18,0

Однокристалний 8-розрядний мікроконтролер МК1816BE51 (МК51) виконаний на основі Гарвардської архітектури і високорівневої n-МОП технології, що дозволяє одержати високий ступінь інтеграції, порівняно з біполярними структурами, і забезпечує підвищену швидкодію. Докладний опис архітектури МК51 приведений в п. 5.2. Основні характеристики і відмінні особливості мікроконтролерів сімейства MCS51 (MCS-51, MCS-151 і MCS-251) приведені табл. 5.2. Мікроконтролери сімейства 8XC151SX (MCS-151) за системою команд, набору програмно доступних ресурсів, системою переривань, набору блоків вводу-виводу і функціям виведень корпусу сумісні з мікроконтролерами 8XC51FX (MCS-51). Удосконалення торкнулися, в основному, операційного ядра. Введені конвеєр команд, режим сторінкової адресації пам'яті і ін. В результаті при конвеєрній вибірці в межах однієї сторінки час виконання команди складає два періоди частоти задаючого кварцевого генератора (замість 12 періодів у попереднього сімейства MCS-51).

Таблиця 5.2 – Характеристика мікроконтролерів сімейства MCS-51

Контролер	ROM/EPROM кбайт	RAM, байт	T/C	Макс.Fosc, МГц	Особливості групи
8031AH		128	2	12	HMOS технологія, базова конфігурація, 4 порти
8051AH	4K ROM	128	2	12	
8751H	4K EPROM	128	2	12	
80C31BH	-	128	2	12,16	CMOS технологія, режим зниженого енергоспоживання, 3 біти захисту
80C51BH	4K ROM	128	2	12,16	
87C51BH	4K EPROM	128	2	12,16,20	
8032AH	-	256	3	12	HMOS технологія, 3 таймери/лічильники, 4 порти, 3 біта захисту
8052AH	8K ROM	256	3	12	
8752BH	8K EPROM	256	3	12	
80C32	-	256	3	16,20,24	CMOS технологія, таймер/лічильник з прямим і зворотним рахунком, 3 біта захисту
80C52	8K ROM	256	3	16,20,24	
87C52	8K EPROM	256	3	16,20,24	
80C54	8K ROM	256	3	16,20,24	
87C54	16K EPROM	256	3	16,20,24	
80C58	32K ROM	256	3	16,20,24	
87C58	32K EPROM	256	3	16,20,24	
80L52	8K ROM	256	3	16,20	Версія із зниженою напругою живлення 2,7-3,6 В
87L52	8K EPROM	256	3	16,20	
80L54	8K ROM	256	3	16,20	
87L54	16K EPROM	256	3	16,20	
80L58	32K ROM	256	3	16,20	
87L58	32K EPROM	256	3	16,20	
80C51FA	-	256	3	16,20,24	CMOS технологія, модуль PCA, таймер/лічильник з прямим і зворотним рахунком, режими зниженого енергоспоживання, сторожовий таймер, 3 біти захисту
83C51FA	8K ROM	256	3	16,20,24	
87C51FA	8K EPROM	256	3	16,20,24	
83C51FB	16K ROM	256	3	16,20,24	
87C51FB	16K EPROM	256	3	16,20,24	
83C51FC	32K ROM	256	3	16,20,24	
87C51FC	32K EPROM	256	3	16,20,24	

Продовження табл. 5.2

80L51FA	-	256	3	16,20	CMOS технологія, версія із зниженою напругою живлення 2,7-3,6 В
83L51FA	8K ROM	256	3	16,20	
87L51FA	8K EPROM	256	3	12,16	
83L51FB	16K ROM	256	3	16,20	
87L51FB	16K EPROM	256	3	16,20	
83L51FC	32K ROM	256	3	16,20	
87L51FC	32K EPROM	256	3	16,20	
80C51GB	-	256	3	16,20,24	CMOS технологія, АЦП (8 кан/8 розряд), 2 РСА, 6 портів I/O, сторожовий таймер
83C51GB	8K ROM	256	3	16,20,24	
87C51GB	8K EPROM	256	3	16,20,24	

Мікроконтролери сімейства MCS-251 є розвитком архітектури сімейств MCS-51 і MCS-151. У основу покладена "стара" система команд і сталий набір блоків вводу-виводу: три таймери-лічильника, блок РСА, послідовний порт і сторожовий таймер. Центральний процесор мікроконтролерів MCS-251 побудований з використанням конвейера команд (час виконання команд – 2 періоди частоти кварцевого генератора) і регістрового файлу. Система команд доповнена інструкціями, оперуючими 16 і 32-розрядними операндами. Нове сімейство 8-бітових мікроконтролерів підвищує функціональність і продуктивність широко поширених мікроконтролерів MSC-51, при збереженні сумісності на рівні двійкових кодів. Завдяки сумісності по контактах з 8XC51FX, МК 8XC251SB може служити засобом підвищення продуктивності існуючих апаратно-програмних систем. До типових областей застосування 8XC251SB можна віднести системи управління. Всім МК сімейства MSC-251 властиві такі загальні особливості:

24-бітова лінійна адресація до 16 Мбайт пам'яті;

ЦПУ регістрової архітектури з регістрами, що адресуються як байти, слова і подвійні слова;

сторінковий режим, прискорюючий вибірку команд із зовнішньої пам'яті;

конвертер команд;

розширена система команд, що включає 16-бітові арифметичні і логічні команди;

64 - кбайтовий зовнішній стек;

мінімальний час виконання команд (два такти по порівнянню з 12 тактами у МК MCS-51);

двійкова сумісність з МК MCS-51.

Першим МК в сімействі MCS-251, розробленим компанією Intel, є мікроконтролер 8XC251SB. Опис його архітектури детальніше приведений в п. 7.3.

5.2. Однокристалний мікроконтролер КМ1816ВЕ51

Однокристалний мікроконтролер КМ1816ВЕ51 (МК51) виконаний в корпусі ВІС на основі високорівневої n-МОП технології, що дозволяє одержати вищий ступінь інтеграції в порівнянні з біполярними структурами. Корпус ВІС має 40 зовнішніх виводів. Цоколювання корпусу і назва виводів приведені на рис. 5.1. Для роботи потрібне одне джерело живлення +5В. Через чотири програмовані порти вводу-виводу МК51 взаємодіє з середовищем в стандарті TTL-схем з трьома станами виходу.

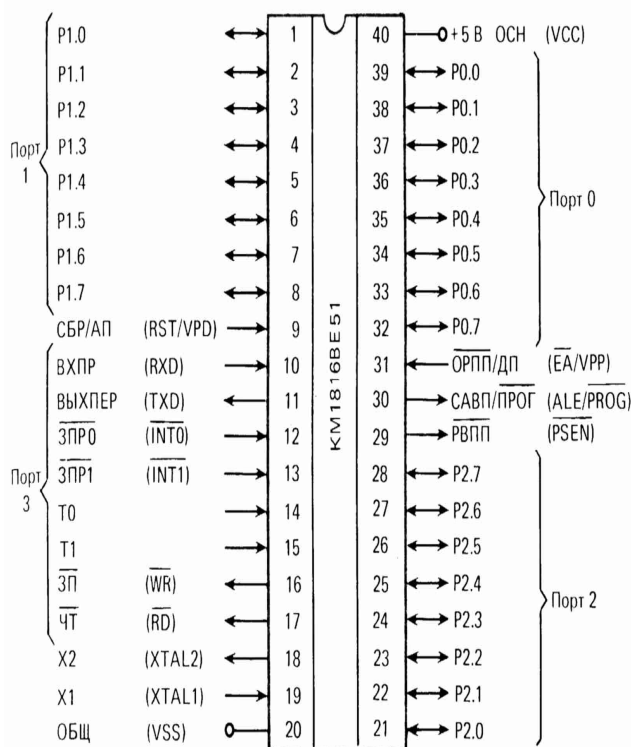


Рис.5.1– Корпус мікросхеми МК51 і найменування виводів

Корпус МК51 має два виводи для підключення кварцевого резонатора, чотири виводи для сигналів, що управляють режимом роботи МК, і 8 ліній порту 3, які можуть бути запрограмовані користувачем на виконання спеціалізованих (альтернативних) функцій обміну інформацією з середовищем. Назва виводів і значення вхідних і вихідних рівнів приводяться нижче:

ТТЛ: $U_{in} \geq 2,0V$ $U_{il} \leq 0,8 V$ – вхідні рівні;

$U_{oh} \geq 2,4V$ $U_{ol} \leq 0,4V$ – вихідні рівні;

СБР (RST) – керуючий сигнал скидання;

ВхПр(RxD) вхід приймача універсального приймача-передавача (УАПП);

ВихПер(TxD) – вихід передавача УАПП;

ЗПр(INT) – запит переривання;

T0, T1 – таймер/лічильник подій;

X1, X2 – входи кварцевого резонатора;

ЗП(WR) – керуючий сигнал запису;

ЧТ(RD) – керуючий сигнал читання;

ОРПП(EA) – керуючий сигнал відключення резидентної пам'яті програм;

САВП(ALE) – керуючий сигнал скидання адреси зовнішньої пам'яті;

ПРОГ(PROG) – керуючий сигнал програмування РПП;

ДЗПП(PSEN) – дозволи зовнішньої пам'яті програм;

ОБЩ(VSS) – потенціал землі;

+5V(VCC) – напруга живлення +5В;

X1(XTAL1) – вхід для підключення виведення кварцевого резонатора або вхід для сигналу від зовнішньої початкової сигналізації;

X2(XAL2) вхід для підключення другого виводу резонатора;

Структурна схема МК51

Узагальнена структурна схема представлена на рис. 5.2, а детальніша – на рис. 6.8. Її основу складає внутрішня двонаправлена 8-бітова шина, яка зв'язує між собою основні вузли і пристрої. До них відносяться: 8-розрядний центральний процесор (ЦП) (включає АЛУ, акумулятор з розширювачем, регістр слова стану, блок регістрів спеціальних функцій, пристрій управління і

синхронізації), розміщені на кристалі (всередині); пам'ять програм – ПЗП місткістю 4 кбайта і пам'ять даних – ОЗП місткістю 128 байт, які складають резидентну пам'ять; 32 лінії чотирьох паралельних портів (P0, P1, P2, P3) вводу-виводу (ВВ); канал послідовного порту ВВ; два 16-розрядні таймери/лічильники і логіку дворівневої системи переривань з п'ятьма або шістьма джерелами запитів.

Всі ці засоби утворюють резидентну частину МК, розміщену безпосередньо на кристалі. Окрім цього, є можливість реалізувати поза кристалом пам'ять програм і пам'ять даних до 64 кбайт кожна, шляхом підключення зовнішніх ВІС. Для скорочення ширини фізичного інтерфейсу більшість логічних ліній поєднуються. Так, при зверненнях до зовнішньої пам'яті порт P0 виконує роль суміщеної шини адреси даних, а P2–шини старшої частини адреси. Всі лінії порту P3 можуть виконувати альтернативні функції ліній управління.

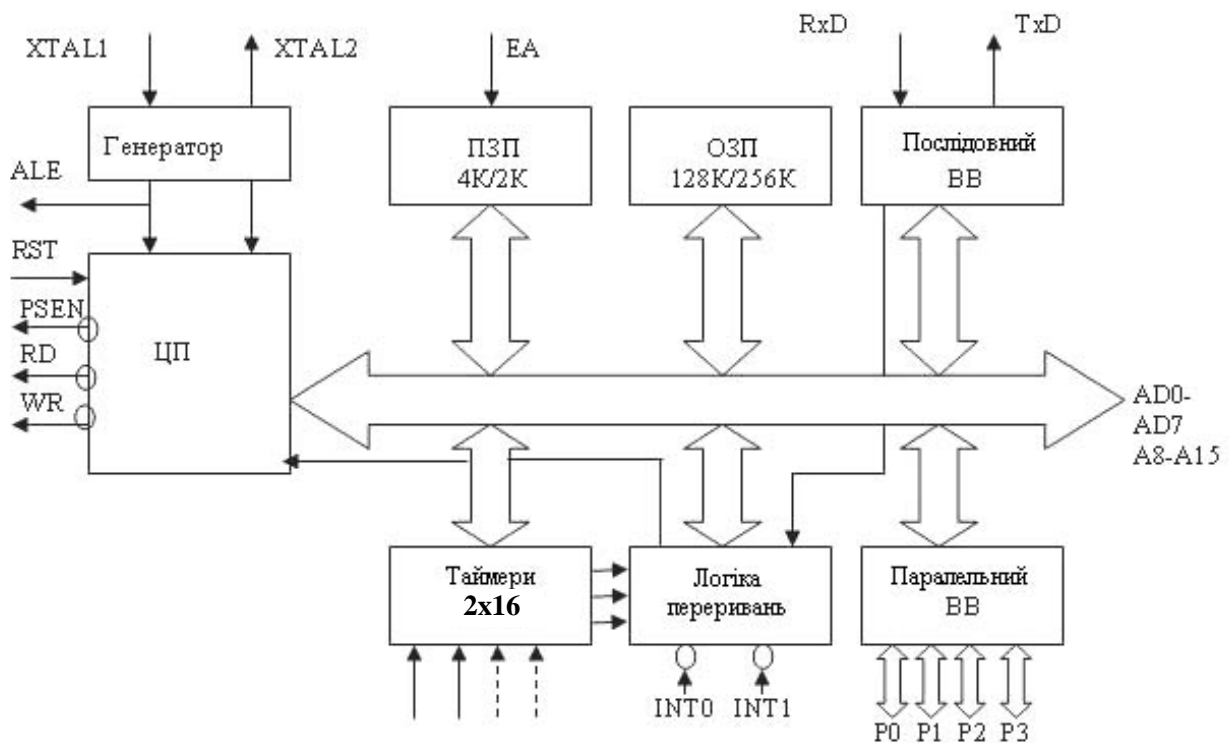


Рисунок 5.2 – Узагальнена структурна схема МК51

У архітектурі МК51 використаний принцип незалежності середовищ для зберігання програм і даних, тобто принцип Гарвардської архітектури.

Резидентна пам'ять. Пам'ять програм і пам'ять даних, розміщені на кристалі ВЕ51, фізично і логічно розділені, мають різні механізми адресації, працюють під управлінням різних сигналів і виконують різні функції.

Пам'ять програм (ПЗП або СПЗУ) має місткість 4Кбайта і призначена для зберігання команд, констант, слів ініціалізації, таблиць перекодування вхідних змінних і т.п., керована РПП має 16-бітову шину адреси, через яку забезпечується доступ з лічильника команд або з регістра покажчика даних (РПД).

Пам'ять даних (ОЗП) призначена для зберігання змінних в процесі виконання прикладної програми, адресується одним байтом і має місткість 128 байт. До адресного простору РПД примикають адреси регістрів спеціальних функцій (РСФ), РПП, РПД. Організація доступу до них представлена на рис. 5.3. Набір програмно доступних регістрів приведений на рис. 5.4. Оскільки структура відноситься до класу акумуляторних, з банками перемикачів робочих регістрів, то центральним регістром набору вважається акумулятор.

Акумулятор А – 8-розрядний регістр, виконує звичайні функції основного арифметичного регістра, є джерелом операнда і приймачем (місцем фіксації) результату арифметичних і логічних операцій, а також ряду операцій передачі даних. Тільки з використанням акумулятора можуть бути виконані операції зсуву, перевірки на нуль, з прапором паритету і ін.

Регістр В – 8-розрядний, служить розширенням акумулятора А, необхідний для здійснення операцій множення і ділення, де виступає як джерело і приймач операндів. У всіх інших операціях регістр В виконує функції, які визначені користувачем. При скиданні А і В встановлюють в нуль.

Контролер МК51 може виконувати безліч команд без участі акумулятора. Дані можуть бути передані з будь-якої комірки РПД в будь-який регістр, останній може бути завантажений безпосередньо операндом і т.д.

Ряд логічних операцій може бути виконаний без участі акумулятора. Змінні можуть бути інтерпретовані і перевірені (ТЕСТ) без участі акумулятора.

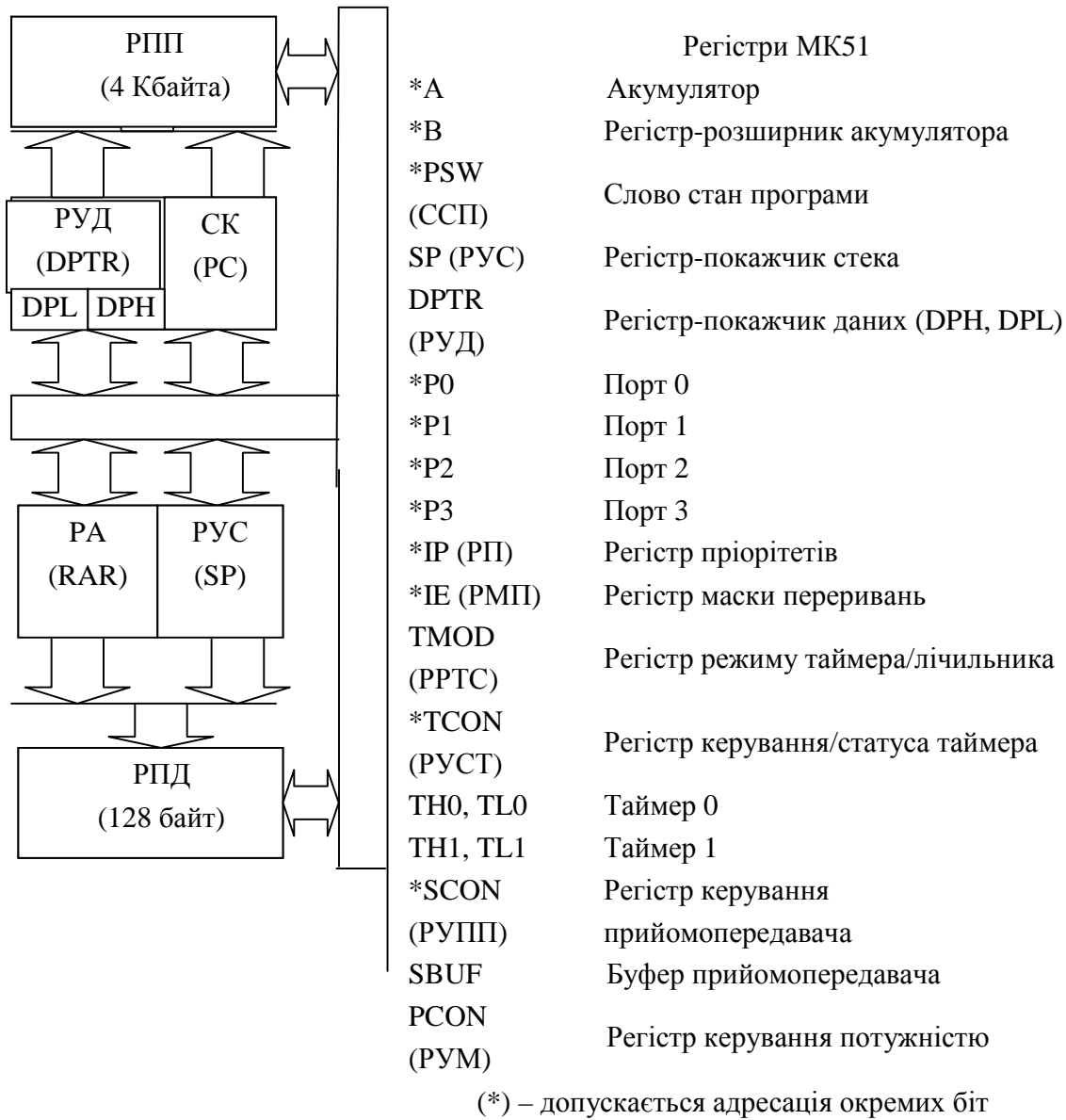


Рисунок 5.3 – Організація доступу до регістрів

Аналогічно можуть бути перевірені й зміннені прапори й керуючі біти.

Регістр слова стану програми (ССП) фіксує ряд ознак операцій (прапорів), які формуються при виконанні ряду операцій АЛУ (див. рис. 5.5). При скиданні в ССП установлюється 00Н.

1. *Прапор переносу С*. Управляється програмно й апаратно при виконанні арифметичних і логічних операцій, виконує функції "булевого акумулятора" у командах роботи з бітами.

2. *Прапор допоміжного переносу АС*. Установлюється апаратно при виконанні команд додавання й віднімання (перенос або позика в біті 3).

3. *Прапор, обумовлений користувачем F0*. Програмно керований.

4. *Прапори вибору банку регістрів RS0-RS1*, установлюються й скидаються програмно для вибору одного із чотирьох робочих банків регістрів.

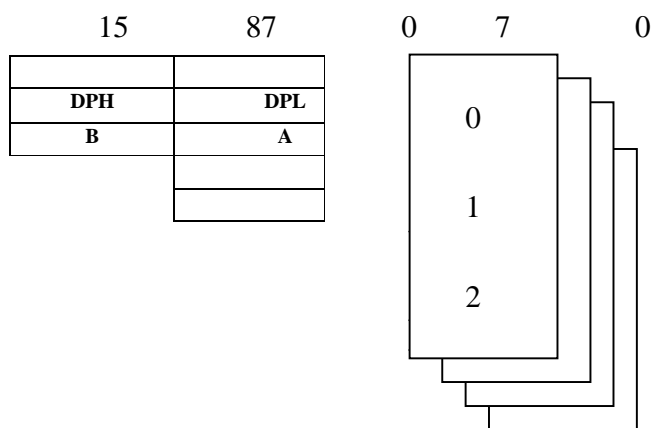


Рис. 5.4 – Набір програмно доступних регістрів

5. *Прапор переповнювання OV*. Встановлюється і скидається апаратно при виконанні арифметичних операцій над цілими числами із знаком.

6. *Прапор паритету P*. Виконується контроль по парності вмісту акумулятора.

Регістр покажчик стека (PUC) може адресувати будь-яку область РПД. Його вміст інкрементується перш, ніж дані будуть записані в стек командою PUSH і CALL. Вміст PUC декрементується після виконання команд POP і RET. PUC є 8-бітовим, що дозволяє утворити системний стек 256 байт. Активною

завжди є адреса останнього занесеного в стек байта. Стек росте у бік збільшення вмісту РУС. Після сигналу СБР, в РУС, автоматично завантажується код 07Н. Це значить, що якщо прикладна програма не перевизначає стек, то перший елемент даних в стеку розташовуватиметься в комірці РПД з адресою 08Н.



Рисунок 5.5 – Призначення бітів регістру слово стану програми (ССП)

Регістр – показчик даних (РПД) – 16 - розрядний, кожна його 8-бітова половина (DPL і DPH) може використовуватися (бути адресована) при необхідності незалежно одна від одної.

РУД використовується як адреса при пересилці констант з пам'яті програм і доступі до змінних із зовнішньої пам'яті даних, а також для організації передачі управління. При скиданні DPTR встановлюється 000H.

У МК51 передбачено 4 банки по 8 робочих регістрів R0-R7 у кожному, перераховуваних полем RsiCCP. Регістри виконують спільноцільові функції зберігання даних.

Регістри R0 і R1 кожного банку реалізують також функції 8-розрядних покажчиків даних.

Використання наборів робочих регістрів дозволяє істотно зменшити тривалість перемикання контекстів ЦП, що дуже важливо для мікросистем реального часу.

Таймер/лічильник. У складі засобів МК51 є регістрові пари TH0, TL0 і TH1, TL1, на основі яких функціонують 2 незалежних програмно керованих 16-бітових таймера/лічильника подій.

Буфер послідовного порту SBUF. Регістр SBUF має 2 незалежних регістра – буфер приймача і буфер передавача. Завантаження байта в SBUF негайно викликає початок процесу передачі через послідовний порт (ПП). Прочитування вмісту SBUF позначає прийом даних, джерелом яких є приймач ПП.

Регістри спеціальних функцій. Регістри з іменами IP, IE, TMOD, TCON, SCON, PCON використовують для фіксації і програмної зміни керуючих біт, і біт стану схеми переривання, таймера/лічильника, приймача-передавача послідовного порту і для управління споживаною потужністю МК-51.

Арифметико-логічний пристрій (АЛП) – 8-розрядний, може виконувати:

- арифметичні операції складання, віднімання, множення і ділення;
- логічні операції І, АБО, виключаюче АБО;
- операції циклічного зрушення, скидання, інвертування і т.п.

У АЛП є програмно недоступні регістри T1 і T2, призначені для часового зберігання операндів, схема десяткової корекції і схема формування ознак.

Прості операції декремента і інкремента утворюють тандеми, необхідні для виконання таких операцій, як інкрементування 16-бітових регістрових пар.

АЛП дозволяє оперувати не тільки з байтами, але і бітами. Програмно доступні біти можуть бути встановлені, скинуті, інвертовані, передані, перевірені, використані в логічних операціях.

АЛП може оперувати з чотирма типами інформаційних об'єктів:

булевими (1 біт);

цифровими (4 біт);

байтними (8 біт);

адресними (16 біт).

У багатьох мікросистемах передбачені спеціальні засоби для роботи з десятковими числами. Обробка числових даних безпосередньо в числовій формі часто застосовується в системах з десятковим ВВ (вводом-виводом), оскільки не вимагає їх проміжного перетворення в двійковий формат і назад. Для зберігання десяткових чисел в пам'яті мікросистеми використовують двійково-десятковий код. Розрізняють два формати: упакований і розпакований.

У 2-10-кодi упакованого формату кожна десяткова форма від 0 до 9 представляється 4-розрядним двійковим еквівалентом від 0000 В до 1001 В коди 1010 В–1111 В. У одному байті вдається розмістити тільки дві десяткові цифри і представити ціле число з діапазону 0–99. Для представлення багаторозрядного десяткового числа відводиться більше число байтів (по байту на кожні дві цифри):

$$9272 = 1001\ 0010\ 0111\ 0010В$$

$$380 = 0011\ 1000\ 0000В$$

Перехід від однієї системи до іншої виконується записаною командою десяткової цифри її двійковим еквівалентом і навпаки.

У розпакованому форматі для представлення кожної десяткової цифри використовується один байт, молодша тетрада містить код цифри (двійковий), а старша – нулі. Наприклад:

$$9272 = 00001001\ 000000010\ 00000111\ 00000010В$$

$$380 = 00000011\ 00001000\ 00000000B$$

Розпакований формат, порівняно з упакованим, вимагає в два рази більше пам'яті. Проте він дуже добре узгоджується з тестовим представленням цифр в коді КОИ-7. Для перекладу розпакованого 2–10 коду в тестовий формат достатньо в старшу тетраду кожної цифри записати 0011В. При необхідності знак десяткового числа кодується окремою тетрадою або байтом. У обох випадках комбінація 0000В відповідає "+", а 1001В - " - ".

Наприклад, для упакованого формату:

$$+921 = 0000\ 1001\ 0010\ 0001B;$$

$$-350 = 1001\ 0011\ 0101\ 0000B.$$

У розпакованому форматі:

$$-350 = 00001001\ 00000011\ 00001101\ 00000000B.$$

Арифметична операція над десятковими числами виконується в два етапи: спочатку проводиться звичайна двійкова операція над десятковими числами, потім десяткова корекція одержаного результату. В цьому випадку можна скористатися схемою десяткової корекції АЛП, двійкової операції, що наводить результат, до десяткового вигляду.

Корекція виконується, якщо в результаті операції вміст перевершує 9. Для розпакованого формату корекція відбувається шляхом збільшення числа 0F6H, завдяки чому результат стає вірним, десяткове перенесення шифрується і може бути враховано при складанні більш старших десяткових цифр.

Пристрій управління і синхронізації

Кварцевий резонатор, що підключається до зовнішніх виведень X1 і X2 корпусу МК51, управляє роботою внутрішнього генератора, який в свою чергу формує сигнали синхронізації.

Пристрій управління МК51 на основі сигналів синхронізації формує машинний сигнал фіксованої тривалості, рівний 12 періодам резонатора або шести станам первинного керуючого автомата. Кожен стан первинного керуючого автомата містить дві фази сигналів резонатора.

У першій фазі виконується операція в АЛП, а в другій здійснюється мікрореєстрова передача.

Зовнішніми спостережуваними сигналами є тільки сигнали резонатора і САВП. САВП формується двічі за машинний цикл (S1P2-S1P1 і S4P2-S5P1).

Більшість команд виконуються за один машинний цикл. Деякі команди, що оперують з 2-байтними словами або пов'язані із зверненням до зовнішньої пам'яті, виконуються за два машинні цикли. Тільки команди ділення і множення вимагають чотирьох машинних циклів. Розрахунок часу виконання прикладних програм проводиться з урахуванням цих особливостей.

Порти вводу-виводу (ВВ)

Підсистема ВВ МК51 розміщується безпосередньо на кристалі. Для ВВ даних і управління процесом їх передачі до складу МК51 введений ряд портів даних і реєстрів керування/стану. Фізична система ВВ мікроконтролера складається з чотирьох двонаправлених 8-розрядних портів P0-P3, які призначені для забезпечення обміну інформацією мікроконтролера з зовнішніми пристроями, утворюючи 32 лінії ВВ і можуть виконувати ряд додаткових функцій. Всі порти відображені в просторі внутрішньої пам'яті і нічим не відрізняються від звичайних елементів пам'яті. Це дозволяє застосувати до вмісту портів команди пам'яті з операндом з внутрішньої пам'яті. Кожний з портів містить фіксатор-защипку, яка являє собою 8-розрядний реєстр, що має байтову і бітову адресацію, для скидання установа програмами засобами. Це забезпечує можливість доступу до окремих розрядів портів.

При зверненні до зовнішньої пам'яті програм або даних порти P0, P2 виконують функції AD і AB відповідно. Молодший байт адреси і дані передаються через P0 в мультиплексному режимі: спочатку вводиться адреса, потім порт використовується для передачі даних. Коли розрядність адреси рівна 16 біт, старший байт адреси формується через P2 .

Всі виведення порту P3 можуть бути використані для реалізації альтернативних функцій, приведених в табл. 5.3.

Альтернативні функції можуть бути задіяні шляхом запису 1 у відповідні біти регістра-защипки (P3.0-P3.7 порту 3).

Таблиця 5.3 – Альтернативні функції порту 3

Позиція	Символ	Ім'я і призначення
P3.0	RxD	Вхід приймача послідовного каналу порту. Ввод-вивід даних у режимі зсуваючого регістра
P3.1	TxD	Вихід передавача послідовного каналу порту. Вихід синхронізації у режимі зсуваючого регістра
P3.2	INT0	Вхід запиту на переривання 0. Сприймається сигнал низького рівня або зріз
P3.3	INT1	Вхід запиту на переривання 1. Сприймається сигнал низького рівня або зріз
P3.4	T0	Зовнішній вхід таймера/лічильника 0 або тест вхід
P3.5	T1	Зовнішній вхід таймера/лічильника 1 або тест вхід
P3.6	WR	Запис. Активний сигнал низького рівня формується при зверненні до ЗПД
P3.7	RD	Читання. Активний сигнал низького рівня формується при зверненні до ЗПД

Порт P0 є двонаправленим, а P1-P3 квазідвонаправленими.

Кожна лінія портів може бути використана незалежно для вводу або виводу інформації. Для режиму вводу в регістр-защипку записується 1.

Кожен порт містить керований регістр-защипку, вхідний буфер і вихідний драйвер.

У портах P0 і P2 у вихідному буфері передбачені ключі, сполучаючі їх виводи з внутрішніми шинами AD і A відповідно.

По сигналу СБР (RST) в регістри-защипки всіх портів автоматично записуються одиниці, настраюючи їх цим на режим вводу.

При роботі порту, коли він одночасно є операндом і місцем призначення результату, пристрій управління автоматично реалізує спеціальний режим "читання-модифікація-запис". При цьому ввод сигналів відбувається не із зовнішніх виводів порту, а з його регістра-защипки, що дозволяє виключити неправильне прочитування раніше виведеної інформації.

Цей механізм звернення до портів реалізується наступними командами, наприклад:

ANL P2, A; ORL P3, A; XRL P1, A; JBC P1. 1, LABEL; CPL P1.2; INC P1;

DEC P3; DJNZ P2, LABEL; MOV PX.Y,C; SET PX.Y; CLR PX.Y.

По цих командах спочатку прочитується байт з порту, а потім записується новий байт в регістр-защіпку.

Послідовний інтерфейс (УАПП)

Асинхронний приймач-передавач (УАПП) здійснює прийом і передачу інформації, представленої послідовним кодом в повному дуплексному режимі обміну.

УАПП (послідовний порт) включає:

- приймаючий здвиговий регістр;
- передавальний здвиговий регістр;
- спеціальний буферний регістр (SBUF) приймача-передавача.

Наявність буферного регістра приймача дозволяє суміщати операцію читання раніше прийнятого байта з прийомом чергового байта. Якщо до моменту закінчення прийому байта попередній байт не був зчитаний з SBUF, то він буде втрачений. Послідовний інтерфейс може працювати в чотирьох різних режимах.

Режим 0 – інформація передається і приймається через зовнішній вивід входу приймача (RxD). Приймаються або передаються 8-біт даних. Через зовнішній вивід виходу передавача (TxD) видаються імпульси зсуву, які супроводжують кожен байт. Частота передачі біта інформації рівна 1/12 частоти резонатора.

Режим 1 – передаються через TxD і приймається з RxD 10 біт інформації: старт-біт (0) , 8 біт даних і стоп-біт (1). Швидкість прийому/передачі – величина змінна і задається таймером.

Режим 2 – через вивід TxD передаються, а через RxD приймаються 11 біт інформації: старт-біт, 8 біт даних, програмований дев'ятий біт і стоп-біт. При передачі дев'ятий біт може приймати значення 0 або 1, або, наприклад, для підвищення достовірності передачі шляхом контролю по парності в нього може бути поміщено значення ознаки паритету з ССП. Частота прийому передачі вибирається програмою і може бути рівна 1/32 або 1/64 частоти резонатора

залежно від біта SMOD, що управляє.

Режим 3 – співпадає з режимом 2 в усіх відношеннях за винятком частоти прийому передачі, яка є величиною змінною і задається таймером.

Управління режимами роботи УАПП здійснюється через спеціальний регістр з символічним ім'ям SCON, приведений у табл. 5.4.

Прикладна програма шляхом завантаження в старші біти спеціального регістра SCON 2-бітового коду визначає режим роботи УАПП. У всіх чотирьох режимах роботи передача з УАПП ініціюється будь-якою командою, в якій буферний регістр SBUF вказаний як одержувач байта. Прийом з УАПП в режимі 0 здійснюється за умови, що RI=0 і REN=1. У режимах 1, 2, 3 прийом починається з приходом старт-біта, якщо REN=1.

Таблиця 5.4 – Призначення бітів регістра управління – статусу УАПП

Позиція	Символ	Ім'я і позиція
SCON.0	RI	Прапор переривання приймача. Встановлюється апаратно при прийомі байта. Скидається програмно після обслуговування переривання
SCON.1	TI	Прапор переривання передавача. Встановлюється апаратно при закінченні передачі байта. Скидається програмно після обслуговування переривання
SCON.2	RB8	Прийом біта 8. Встановлюється / скидається апаратно для фіксації дев'ятого біта, що приймається, в режимі УАПП – 9 біт
SCON.3	TB8	Передача біта 8. Встановлюється / скидається програмно для задання дев'ятого передаваного біта в режимі УАПП-9 біт
SCON.4	REN	Біт дозволу прийому. Встановлюється / скидається програмно для дозволу / заборони прийому послідовних даних
SCON.5	SM2	Біт управління режимом УАПП. Встановлюється програмно для заборони прийому повідомлення, в якому 9-й біт має значення 0
SCON.6	SM1	Біти управління режимом роботи УАПП. Встановлюється / скидається програмно
SCON.7	SM0	

Примітка. Вибір режимів роботи УАПП:

SM0	SM1	Режим роботи УАПП
0	0	Зрушуючий регістр розширення вводу/виводу.
0	1	УАПП-8 біт. Змінна швидкість передачі.
1	0	УАПП-9 біт. Фіксована швидкість передачі.
1	1	УАПП-9 біт. Змінна швидкість передачі.

У біті TB8 програмно встановлюється значення дев'ятого біта даних, який буде переданий в режимі 2 або 3. У біті RB8 фіксується в режимах 2 і 3 дев'ятий біт даних, що приймається. У режимі 1, якщо SM2=0, у біт RB8 заноситься стоп-біт. У режимі 0 біт RB8 не використовується.

Прапор переривання передавача TI встановлюється апаратно в кінці періоду передачі восьмого біта даних в режимі 0 і на початку періоду передачі стоп-біта в режимах 1, 2 і 3. Відповідна підпрограма обслуговування переривання повинна скидати біт TI. Прапор переривання приймача RI встановлюється апаратно в кінці періоду прийому восьмого біта даних в режимі 0 і у середині періоду прийому стоп-біта в режимах 1, 2 і 3. Підпрограма обслуговування переривання повинна скидати біт RI. Швидкість прийому-передачі, тобто частота роботи УАПІ в різних режимах, визначається різними способами.

У режимі 0 частота передачі залежить тільки від резонансної частоти кварцевого резонатора $f_0 = f_{рез} / 12$. За один машинний цикл послідовний порт передає один біт інформації.

У режимі 1, 2 і 3 швидкість прийому-передачі залежить від значення керуючого біта SMOD в регістрі спеціальних функцій РУМ (управління потужністю), приведену табл. 5.5.

Таблиця 5.5 – Призначення бітів регістра управління потужністю РУМ (PCON)

PCON.0	IDL	Біт холостого ходу. Якщо біт встановлений в 1, то МК переходить в режим холостого ходу
PCON.1	PD	Біт зниженої потужності. При установці біта в 1 МК переходить в режим зниженої споживаної потужності (його пріоритет, вище ніж у IDL)
PCON.2	GF0	Прапори, визначувані користувачем (прапори загального призначення)
PCON.3	GF1	
PCON.4	-	Не використовуються
PCON.5	-	
PCON.6	-	
PCON.7	SMOD	Подвоєна швидкість передачі. Якщо біт встановлений в 1, то швидкість передачі в двоє більше, ніж при SMOD=0

У режимі 2 частота передачі визначається виразом

$$f_2 = \left(2^{\text{SMOD}} / 64 \right) f_{\text{рез}}$$

При SMOD, рівному нулю, частота передачі рівна $(1/64)f_{\text{рез}}$, а при SMOD, рівному 1, частота передачі рівна $(1/32)f_{\text{рез}}$.

У режимах 1 і 3 у формуванні частоти передачі, окрім біта SMOD, що управляє, бере участь таймер 1. При цьому частота передачі залежить від частоти переповнювання OVT1 і визначається:

$$f_{1,3} = \left(2^{\text{SMOD}} / 32 \right) \cdot f_{\text{OVT1}}$$

Переривання від таймера 1 в цьому випадку повинне бути заблоковане. Сам таймер-лічильник 1 може працювати і як таймер, і як лічильник в одному з трьох режимів. Нижче в таблиці 5.6 приводиться опис способів настройки Т/С 1 для отримання типових частот передачі даних через УАПП.

Доступ до зовнішньої пам'яті. У мікроконтролерних системах, побудованих на основі МК51, можливо використання двох типів зовнішньої пам'яті: постійної пам'яті програм (ЗПП) і оперативної пам'яті даних (ЗПД).

Доступ до ЗПП здійснюється за допомогою керуючого сигналу, який виконує функцію строб-сигналу читання. При цьому $\overline{\text{ORPP}} (\overline{\text{EA}})$ активний.

Таблиця 5.6 – Типові частоти прийому – передачі даних через УАПП

Частоти прийому/передачі (BAUD RATE)	Частота резонатора, Мгц	SMOD	Таймер/лічильник 1		
			С/Т	Режим (MODE)	Перезавантажене число
Режим 0, max: 1 мГц	12	X	X	X	X
Режим 2, max: 375 кГц	12	1	X	X	X
Режим 1,3: 62,5 кГц	12	1	0	2	0FFH
19,2 кГц	11,059	1	0	2	0FDH
9,6 кГц	11,059	0	0	2	0FDH
4,8 кГц	11,059	0	0	2	0FAH
2,4 кГц	11,059	0	0	2	0F4H

1,2 кГц	11,059	0	0	2	0E8H
---------	--------	---	---	---	------

Продовження табл. 5.6

137,5 кГц	11,059	0	0	2	1DH
110 кГц	6	0	0	2	72H
110 кГц	12	0	0	1	0FEЕBH

Доступ до ЗПД забезпечується керуючими сигналами $\overline{CT}(\overline{RD})$, які формуються в лініях P3.7 і P3.6 при виконанні портом 3 альтернативних функцій.

При зверненні до ЗПП завжди використовується 16-бітова адреса (через DPTR).

При зверненні до ЗПД можливо використання 16-бітової або 8-бітової адреси (через DPTR або R0-R1).

У будь-яких випадках використання 16-бітової адреси, старший байт адреси фіксується (і зберігається незмінно протягом першого циклу читання запису) в регістрі-защипці порту 2.

Якщо черговий цикл зовнішньої пам'яті (MOVX A @DPTR) слідує не відразу ж за попереднім циклом зовнішньої пам'яті, то незмінний вміст регістра-защипки порту 2 відновлюється в наступному циклі.

Якщо використовується 8-бітова адреса (MOVX A @Ri), то вміст регістра-защипки порту 2 залишається незмінним на його зовнішніх виводах протягом всього циклу зовнішньої пам'яті. Через порт 0 в режимі тимчасового мультиплексування здійснюється видача молодшого байта адреси і передача байта даних. Сигнал САВП повинен бути використаний для запису байта адреси в зовнішній регістр. Потім в циклі запису байт даних, що виводиться, з'являється на зовнішніх виводах порту 0 тільки перед появою сигналу $\overline{3P}(\overline{WR})$. У циклі читання байт даних, що вводиться, приймається в порт 0 по фронту стробуючого сигналу $\overline{CT}(\overline{RD})$.

Сигнал САВП, що забезпечує тимчасове узгодження передачі з порту 0 на зовнішній регістр молодшого байта адреси, використовується тільки для читання з ЗПП.

Доступ до ЗПД можливий тільки в тому випадку, якщо сигнал САВП відсутній, тому при виконанні команди MOVX (звернення до ЗПД) він блокується.

У системах без ЗПД сигнал САВП можна використовувати для синхронізації зовнішніх пристроїв з частотою 1/16.

Особливий режим роботи МК51

На етапі відладки прикладних програм, а також при необхідності оперативного перевантаження і модифікації прикладних програм в реальних системах, зовнішня пам'ять мікроконтролера може бути модифікованою для поєднання адресного простору ЗПП і ЗПД. Це здійснюється шляхом підключення зовнішньої логіки, представлені нижче на рис. 5.6.

Слід пам'ятати, що способи адресації РПП, РПД, ЗПП, ЗПД і блоку регістрів спеціальних функцій різні, тому переміщувана версія прикладної програми, яка відладжується в середовищі зовнішньої пам'яті програм даних, відрізнятиметься від завантаженої в РП версії програми.

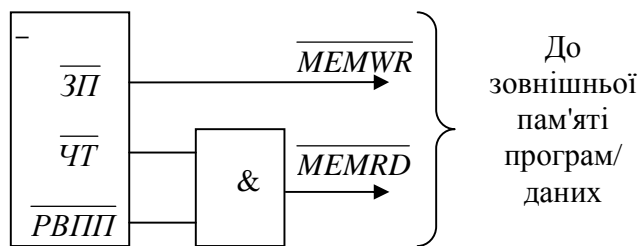


Рисунок 5.6 – Схема підключення зовнішньої пам'яті

Таймер / лічильник

Таймер / лічильник відноситься до стандартних засобів підтримки режиму реального часу. Він використовується для організації системних міток реального часу, формування тимчасових затримок і підрахунку зовнішніх подій.

До складу МК51 входять два 16-розрядні таймери/лічильника T0 і T1, стан яких відображається програмно доступними регістровими парами (TL0, TH0) і (TL1, TH1). T1 і T0 можуть бути запрограмовані для роботи або як таймер, або як лічильник. Функція таймера полягає в рахунку числа машинних циклів, слідуючих з частотою $f_p/12$. Функція лічильника полягає у відстежуванні переходів з 1 в 0 на відповідних входах T0 і T1. Опит значення зовнішнього вхідного сигналу виконується в кожному машинному циклі (S5S2). На розпізнавання переходу потрібно два машинні цикли.

Для управління режимами роботи Т/Л і для організації взаємодії таймера з системою переривань використовується два регістри: регістр режиму таймера/лічильника (PPTS) – рис. 5.7 і регістр управління/статусу таймера (PUS) – рис. 5.8. PPTS (TMOD) розбитий на два 4-розрядні підрегістри T0MOD і T1MOD, які відповідальні за управління T0 і T1, відповідно.

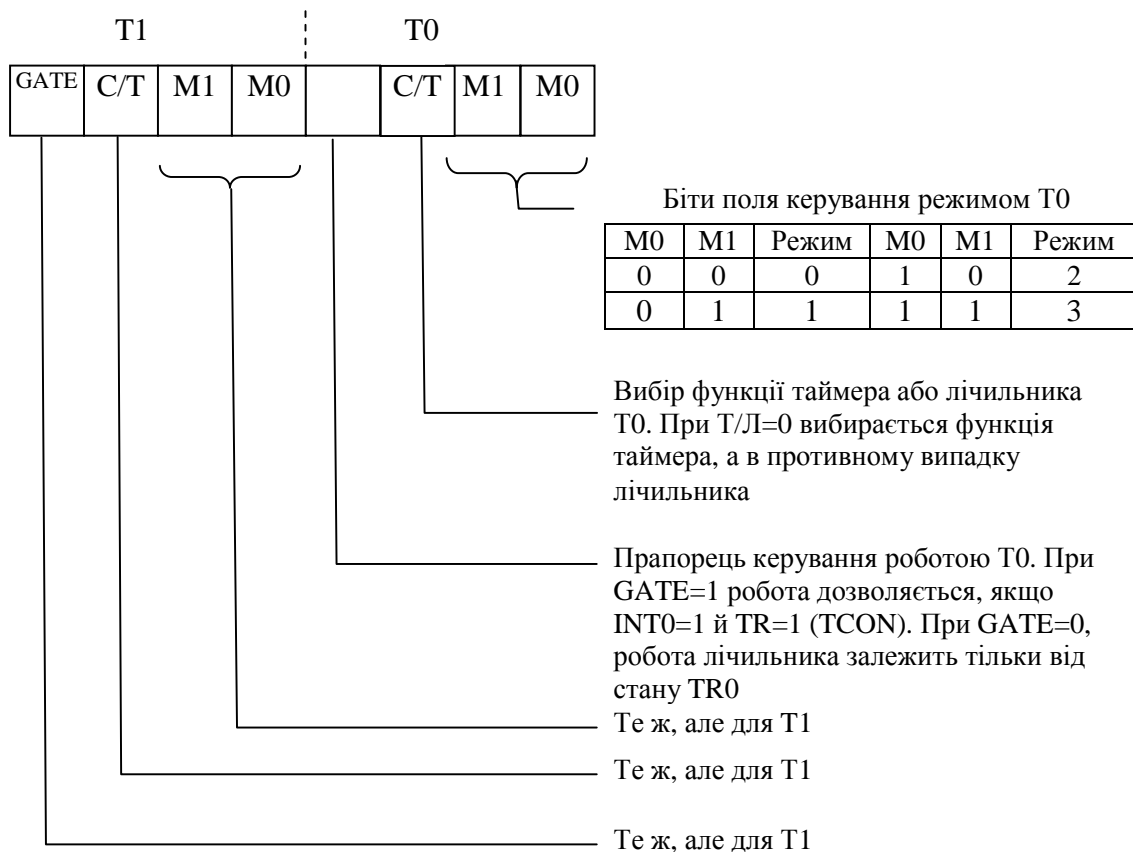


Рисунок 5.7 – Призначення бітів регістра режиму роботи таймера / лічильника (PPTC)

Як видно з опису керуючих бітів PPTC, приведеного на рис. 5.7, для обох таймерів/лічильників (Т/Л) режими роботи 0, 1 і 2 однакові. Режими 3 для Т/Л0 і Т/Л1 різні. Нижче приведено опис роботи Т/Л, у всіх чотирьох режимах 0, 1, 2, і 3.

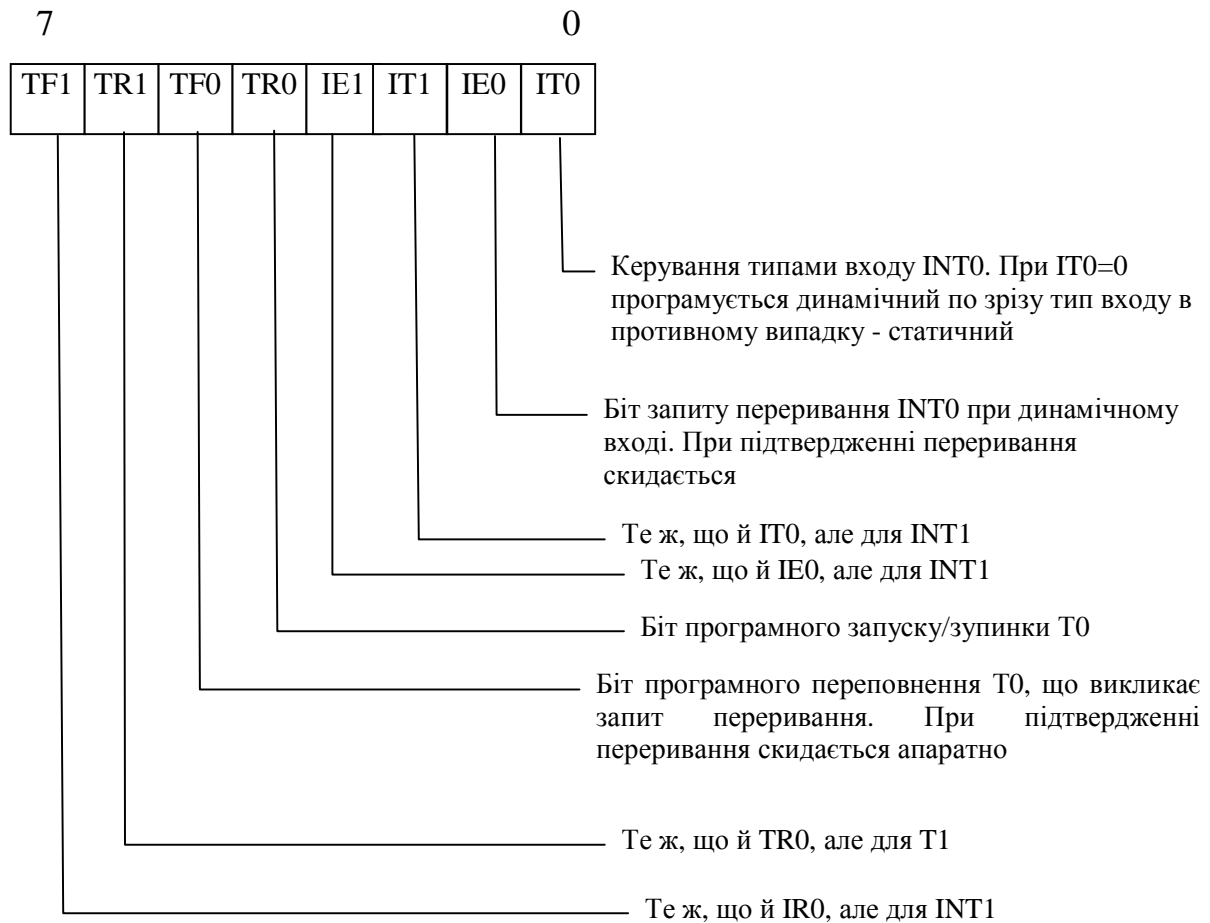


Рисунок 5.8 – Призначення бітів регістра управління/статусу таймера (PUCT)

Режим 0. У цьому режимі регістр таймера має розрядність 13 біт. При переході із стану "всі одиниці" в стан "всі нулі" встановлюється прапор переривання від таймера TF1. Установка GATE в одиницю дозволяє

використовувати таймер для вимірювання тривалості імпульсного сигналу, що подається на вхід запиту переривання.

Режим 1. Робота будь-якого Т/Л така ж як і в режимі 0, за винятком того, що регістр таймера має розрядність 16 біт.

Режим 2. Робота організована таким чином, що переповнювання (перехід із стану "всі одиниці" в стан "всі нулі") 8-бітового лічильника TL1 приводить не тільки до установки прапора TF1, але і автоматично перезавантажує в TL1 вміст старшого байта (ТН1) регістра таймера, який заздалегідь був заданий програмним шляхом. Перезавантаження залишає вміст ТН1 незмінним.

У цьому режимі обидва таймери/лічильники працюють однаково.

Режим 3. Робота Т/Л0 і Т/Л1 відбувається по-різному. Т/Л1 зберігає незмінним свій поточний стан, тобто так само, як при скиданні біта TR1, що управляє, в нуль. TL0 і ТН0 функціонують, як два незалежні 8-бітові лічильники. Роботу TL0 визначають біти GATE0 і TR0, вхідний сигнал $\overline{INT0}$ і прапор переповнювання TF0. Роботу ТН0, який може виконувати тільки функції таймера, визначає біт, що управляє TR1. При цьому ТН0 використовує прапор переповнювання TF1. Цей режим використовується, коли є необхідність наявності додаткового 8-бітового таймера або лічильника.

Система переривань

Система переривань призначена для реагування на зовнішні і внутрішні події шляхом встановлення відповідних прапорів IE0, IE1, TF0, TF1, RI, TI. Всі перераховані прапори можуть бути програмно встановлені або скинуті, що відповідно викликає або усуває переривання. Спрощена схема переривань приведена на рис. 5.9. Управління системою переривань здійснюється шляхом запису слів, що управляють, в регістри: IE – регістр дозволу переривань (рис. 5.10) і IP – регістр пріоритетів переривань (рис. 5.11). Регістр дозволу переривань IE призначений для дозволу або заборони переривань від відповідних джерел. Регістр IP призначений для встановлення рівня пріоритету переривання для кожного з п'яти джерел переривань. Зовнішні переривання INT0 і INT1 можуть бути викликані або рівнем, або переходом сигналу з 1 в 0

(регістр TCON). Прапори IE0 і IE1 (регістр TCON) встановлюються залежно від зовнішніх переривань і ініціюють відповідні програми обслуговування переривання. Скидання прапорів відбувається апаратно, тільки якщо переривання було викликано по зрізу (переходу) сигналу. Якщо виклик відбувається по рівню переривання, то скидання прапора здійснюється програмно. Прапори запитів переривання від таймера TF0 і TF1 скидаються автоматично під час переходу управління до підпрограми обслуговування. Прапори запитів переривання TI, RI від УАПП встановлюються блоком управління УАПП апаратно, але скидатися повинні програмно. Переривання можуть бути викликані або відмічені програмою, оскільки всі перераховані

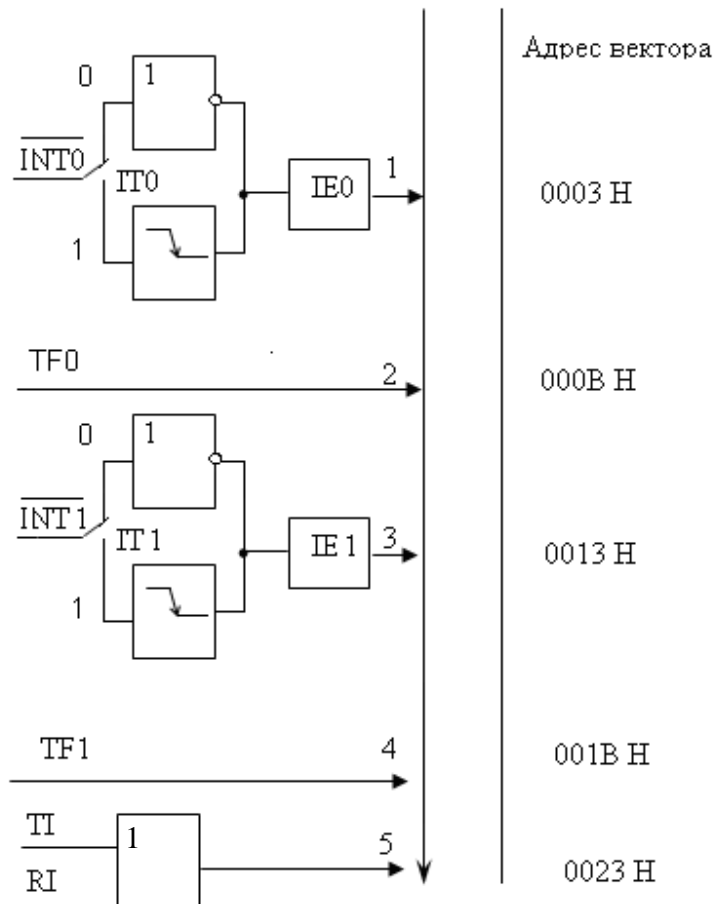


Рисунок 5.9 – Схема переривань МК51

прапори переривань опитуються на кожному машинному циклі (момент S5P2). Ранжирування переривань по рівню пріоритету виконується протягом наступного машинного циклу. Система переривань викликає (апаратно формує виклик LCALL) програму обслуговування переривань, якщо вона не заблокована однією з наступних умов:

- в даний момент обслуговується запит переривання рівного або вищого рівня пріоритету;
- поточний машинний цикл не останній в циклі виконуваної команди;
- виконується команда, пов'язана із зверненням до регістрів ІЕ і ІР.



Рисунок 5.10 – Призначення бітів регістра дозволу переривань (ІЕ)



Рисунок 5.11 – Призначення бітів регістра пріоритетів переривань (IP)

За адресою вектора переривання повинна бути розташована команда безумовної передачі управління (JMP) до початкової адреси підпрограми обслуговування переривання. Підпрограми обслуговування переривання обов'язково повинні закінчуватися командою RETI. По ній знімається блокування переривання і в лічильник команд із стека перезавантажується збережена адреса повернення в основну програму.

Скидання, режими холостого ходу і зниженого електроживлення

Скидання здійснюється шляхом подачі на вхід СБР (RST) сигналу 1. Для упевненого скидання сигнал 1 повинен утримуватися протягом двох машинних циклів (24 періоди резонатора). Квазідвонаправлені буферні схеми зовнішніх виведень ALE і \overline{PSEN} знаходяться при цьому в режимі вводу. Скидається вміст таких регістрів: PC, A, B, PSW, DPTR, TMOD, TCON T/C0, T/C1, IE, IP, SCON.

У режимі PCON скидається тільки старший біт. У регістр показчик стека SP завантажується 07H. У порти P0 і P3 коди 0FFH. Стан SBUF невизначений і

на вміст осередків РПД сигнал не впливає.

Режим холостого ходу. Будь-яка команда, по якій встановлюється біт IDL регістра PCON.0, переводить МК51 в режим холостого ходу. При цьому продовжує працювати внутрішній генератор синхросигналів, і всі регістри зберігають свої значення. На виходах всіх портів утримується той логічний стан, який на них був у момент переходу в режим холостого ходу. Вийти з режиму холостого ходу можна по сигналу СБР або по перериванню.

Режим зниженого енергоспоживання. Перехід в режим здійснюється по команді установки біта PD (PCON.1) в 1. При цьому відбувається зупинка генератора синхросигналів, зберігається вміст РПД, РСФ, утримується напруга на виходах портів. Напруга енергоживлення VCC +5В може бути відключена. Живлення здійснюється через виведення RST (СБР). Вихід з режиму зниженого енергоспоживання можливий тільки по сигналу СБР. При цьому вміст регістрів спеціальних функцій перевизначається, а вміст РПД не змінюється.

Примітка. Склад регістра PCON залежить від технології виготовлення контролера. При *n*-МОП технології PCON містить всього лише один прапорець SMOD. Повний склад прапорців можна зустріти тільки в КМОП варіантах.

Система команд МК51. Загальні відомості

Система команд, приведена у табл. 5.7, налічує 111 команд, з них: 49 – однобайтних; 45 – двобайтних; 17 – трьохбайтних.

Поява трьохбайтних команд пов'язана з введенням прямої адресації. Всі команди виконуються за один або два машинні цикли. Виняток становлять команди MUL і DIV, які вимагають чотири цикли.

При тактовій частоті 12 МГц тривалість машинного циклу дорівнює 1 мкс.

Більшість двобайтних команд – одноциклові, а всі трьохбайтні команди – двохциклові. Це пояснюється тим, що за один машинний цикл може вводиться до двох байтів програмного коду.

Всю безліч команд МК51 можна розділити на *n'ять груп*: пересилки (28); логічних операцій (25); арифметичних операцій (24); передачі управління (17); операцій з бітами (булевого процесора) (17).

Таблиця 5.7 – Команди Асемблера МК51

Команди пересилки, обміну і завантаження			
Мнемокод	КОП	Т Б Ц	Коментар
MOV A, Rn	11101rrr	1 1 1	A <= Rn;
MOV A, @Ri	1110011i	1 1 1	A <= (Ri);
MOV A, dir	11100101	3 2 1	A <= (dir);
MOV A, #DAT	01110100	2 2 1	A <=#DAT
MOV Rn, A	11111rrr	1 1 1	Rn <= A;
MOV @Ri, A	1111011i	1 1 1	(Ri) <= A;
MOV dir, A	11110101	3 2 1	(dir) <= A;
MOV DIR, #DAT	1111011i	7 3 2	(dir) <= #DAT;
MOV Rn, #DAT	01111rrr	2 2 1	Rn <= #DAT;
MOV @Ri, #DAT	0111011i	2 2 1	(Ri) <= #DAT;
MOV dir, dir	10000101	9 3 2	(dir) <= (dir);
MOV Rn, dir	10101rrr	3 2 2	Rn <= (dir);
MOV @Ri, dir	1010011i	3 2 2	(Ri) <= (dir);
MOV dir, Rn	10001rrr	3 2 2	(dir) <= Rn;
MOV dir, @Ri	1000011i	3 2 2	(dir) <= (dir);
MOV DPTR, # D16	10010000	3 2 2	DPTR <= #D16;
XCH A, Rn	11001rrr	1 1 1	A <=> Rn;
XCH A, @Rn	1100011i	1 1 1	A <=> (Rn);
XCH A, dir	11000101	3 2 1	A <=> (dir);
XCHD A, @Ri	1101011i	1 1 1	A <=> (Ri);
SWAP A	11000100	1 1 1	Обмін тетрад акумулятора
PUSH dir	11000000	3 2 2	SP <= SP+1; SP <= (dir);
POP dir	11010000	3 2 2	(dir) <= (SP); SP <= SP-1;
MOVX A, @DPTR	11100000	1 1 2	A <= (DPTR);
MOVX @DPTR, A	11110000	1 1 2	(DPTR) <= A;
MOVX A, @Ri	1110001i	1 1 2	A <= (Ri);
MOVX @Ri, A	1111001i	1 1 2	(Ri) <= A;
MOVC A, @A+DPTR	10010011	1 1 2	A <= ((A+DPTR));
MOVC A, @A+PC	10000011	1 1 2	A <= ((A+PC));
ADD A, Rn	00101rrr	1 1 1	A <= A+Rn;

Продовження табл. 5.7

Мнемокод	КОП	Т Б Ц	КОМЕНТАР
ADDC A, Rn	00111rrr	1 1 1	$A \leftarrow A + Rn + C;$
ADDC A, @RI	0011011i	1 1 1	$A \leftarrow A + (Ri) + C;$
ADDC A, dir	00110101	3 2 1	$A \leftarrow A + (dir) + C;$
ADDC A, #DAT	00110100	2 2 1	$A \leftarrow A + \#DAT + C;$
DA A	11010100	1 1 1	Десяткова корр. аккумуля.
Арифметичні і логічні команди			
SUBB A, Rn	10011rrr	1 1 1	$A \leftarrow A - Rn - C;$
SUBB A, @Ri	1001011i	1 1 1	$A \leftarrow A - (Ri) - C;$
SUBB A, dir	10010101	3 2 1	$A \leftarrow A - (dir) - C;$
SUBB A, #DAT	10010100	2 2 1	$A \leftarrow A - \#DAT - C;$
INC A	00000100	1 1 1	$A \leftarrow A + 1;$
INC Rn	00001rrr	1 1 1	$Rn \leftarrow Rn + 1;$
INC @Ri	0000011i	1 1 1	$(Ri) \leftarrow (Ri) + 1;$
INC dir	00000101	3 2 1	$(dir) \leftarrow (dir) + 1;$
INC DPTR	10100011	1 1 2	$DPTR \leftarrow DPTR + 1;$
DEC A	00010100	1 1 1	$A \leftarrow A - 1;$
DEC Rn	00011rrr	1 1 1	$Rn \leftarrow Rn - 1;$
DEC @Ri	0001011i	1 1 1	$(Ri) \leftarrow (Ri) - 1;$
DEC dir	00010101	3 2 1	$(dir) \leftarrow (dir) - 1;$
MUL AB	10100100	1 1 4	$BA \leftarrow A * B;$ Множення A на B $B \leftarrow$ старший байт, $A \leftarrow$ мл. байт
DIV AB	10000100	1 1 4	$A \bullet B \leftarrow A / B;$ Ділення A на B, $A \leftarrow$ байт частки $B \leftarrow$ остаток
ANL A, Rn	01011rrr	1 1 1	$A \leftarrow A \& Rn;$
ANL A, @Ri	0101011i	1 1 1	$A \leftarrow A \& (Ri);$
ANL A, dir	01010101	3 2 1	$A \leftarrow A \& (dir);$
ANL A, #DAT	01010100	2 2 1	$A \leftarrow A \& \#DAT;$
ANL dir, A	01010010	3 2 1	$(dir) \leftarrow (dir) \& A;$
ANL dir, #DAT	01010011	7 3 2	$(dir) \leftarrow (dir) \& \#DAT;$
ORL A, Rn	01001rrr	1 1 1	$A \leftarrow A \vee Rn;$
ORL A, @Ri	0100011i	1 1 1	$A \leftarrow A \vee (Ri);$
ORL A, dir	01000101	3 2 1	$A \leftarrow A \vee (dir);$
ORL A, #DAT	01000100	2 2 1	$A \leftarrow A \vee \#DAT$

Продовження табл. 5.7

ORL dir, A	01000010	3 2 1	(dir)<=(dir)/A;
ORL dir, #DAT	01000011	7 3 2	(dir)<=(dir)/ # DAT;
Логічні команди			
Мнемокод	КОП	Т Б Ц	Коментар
XRL A, Rn	01101rrr	1 1 1	A<=A (+) Rn
XRL A, @Ri	0110011i	1 1 1	A<=A (+) (Ri)
XRL A, dir	01100101	3 2 1	A<=A (+) (dir)
XRL A, #DAT	01100100	2 2 1	A<=A (+) # DAT
XRL dir, A	01100010	3 2 1	(dir)<=(dir) (+) A
XRL dir, #DAT	01100011	7 3 2	(dir)<=(dir) (+) # DAT
CLR A	11100100	1 1 1	A<=0; Скидання аккум.
CPL A	11110100	1 1 1	A<= \bar{A} ; Інверсія аккум.
RL A	00100011	1 1 1	Циклічне зрушення вліво
RLC A	00110011	1 1 1	Циклічне зрушення вліво через перенесення
RR A	00000011	1 1 1	Циклічне зрушення вправо
RRC A	00010011	1 1 1	Циклічне зрушення вправо через перенесення
Команди безумовних переходів і виклику підпрограми			
Мнемокод	КОП	Т Б Ц	КОМЕНТАР
LJMP 16ADR	00000010	12 3 2	PC[0 15]<=16ADR; довг. перехід
AJMP 11ADR	sss00001	6 2 2	PC[0 11]<=11ADR; абсолютний перехід всередині стор. в 2 кбайта
SJMP rel	10000000	5 2 2	PC<=PC+2+rel; відносний перехід
JMP @A+DPTR	01110011	1 1 2	PC<=A+DPTR; непрямий відносний перехід
LCALL 16ADR	00010010	12 3 2	(SP)<=PC+3; PC[0 15]<=16ADR; довгий виклик підпрограми
ACALL 11ADR	sss10001	6 2 2	(SP)<=PC+2; PC[0 10]<=11ADR; абс. виклик підпрограми в сторінці в 2 Кб

Продовження табл. 5.7

RET	00100010	1 1 2	PC<=(SP); повернення з підпрог.
RETI	00110010	1 1 2	PC<=(SP); повернення з підпрограми обробки переривань
NOP	00000000	1 1 1	PC<=PC+1; холоста команда
Команди умовних переходів			
Мнемокод	КОП	Т Б Ц	Коментар
JZ rel	01100000	5 2 2	PC<=PC+2+rel, якщо A=0
JNZ rel	01110000	5 2 2	PC<=PC+2+rel, якщо A<>0
JC rel	01000000	5 2 2	PC<=PC+2+rel, якщо прапор C=1
JNC rel	01010000	5 2 2	PC<=PC+2+rel, якщо прапор C=0
JB bit, rel	00100000	11 3 2	PC<=PC+3+rel, якщо (bit)=1
JNB bit, rel	00110000	11 3 2	PC<=PC+3+rel, якщо (bit)=0
JBCbit, rel	00010000	11 3 2	PC<=PC+3+rel, якщо (bit)=1
CJNE Rn,#DAT,rel	10111rrr	10 3 2	PC<=PC+3+rel, якщо Rn<>#DAT; C<=1, якщо Rn<#DAT; C<=0, якщо Rn>#DAT
CJNE @Ri, #DAT, rel	1011011i	10 3 2	PC<=PC+3+rel, якщо (@Ri)<>#DAT; C<=1, якщо (@Ri)<#DAT; C<=0, якщо (@Ri)>#DAT
CJNE A, #DAT,rel	10110100	10 3 2	PC<=PC+3+rel, якщо A<>#DAT; C<=1, якщо A<#DAT; C<=0, якщо A>#DAT
CJNE A, dir, rel	10110101	8 3 2	PC<=PC+3+rel, якщо A<>(dir); C<=1, якщо A<(dir); C<=0, якщо A>(dir)
DJNZ Rn, rel	11011rrr	5 2 2	PC<=PC+2+rel, якщо Rn-1<>0
DJNZ dir, rel	11010101	8 3 2	PC<=PC+3+rel, якщо (dir)-1<>0

Операції з бітами

Мнемокод	КОП	Т Б Ц	Коментар
CLR C	11000011	1 1 1	C<=0; скидання прапора C
SETB C	11010011	1 1 1	C<=1; встановлення прапора C
CPL C	10110011	1 1 1	C<= \bar{C} ; інверсія прапора C
CLR bit	11000010	4 2 1	(bit)<=0; скидання прямоадрес. біта

Продовження табл. 5.7.

SETB bit	11010010	4 2 1	(bit)<=1; установка “біта”
CPL bit	10110010	4 2 1	(bit)<= $\overline{\text{bit}}$; інверсія “біта”
ANL C, bit	10000010	4 2 2	$C \leq C \& (\text{bit})$
ANL C, /bit	10110000	4 2 2	$C \leq C \& (\overline{\text{bit}})$
ORL C, bit	01110010	4 2 2	$C \leq C \vee (\text{bit})$
ORL C, /bit	10100000	4 2 2	$C \leq C \vee (\overline{\text{bit}})$; лог. “АБО” C і інверсії прямоадресованного біта в прапор C
MOV C, bit	10100010	4 2 1	$C \leq (\text{bit})$ пересилка біта в прапор C
MOV bit, C	10010010	4 2 2	(bit)<=C; пересилка із C в “біт”

В процесі виконання команди впливають на ряд прапорців-ознак результату, що входять до складу ССП (PSW). Ознака P встановлюється всякий раз, коли приймачем результату служить акумулятор A, включаючи операції пересилки.

Склад операндів МК51 включає: *біти*, 4-бітові цифри, байти, 16-бітові слова.

МК51 має 128 програмно керованих прапорів користувача. Є також можливість адресації окремих бітів блоку регістрів спеціальних функцій і портів. Для адресації бітів використовується пряма 8-бітова адреса. Непряма адресація бітів неможлива.

Порти і РСФ адресуються тільки прямим способом. Байти пам'яті також можуть адресуватися непрямим чином через адресні регістри (R0, R1, DPTR і PC).

Способи адресації даних:

- *пряма* – в команді міститься адреса місця розташування даних;
- *безпосередня* – в команді міститься сам операнд, який слідує за кодом операції;
- *непряма* – в команді приведена адреса, по якій знаходиться операнд;

- *регістрова непряма* – в команді вказані регістри, в яких зберігається адреса, по якій знаходиться операнд; у МК51 використовуються всі 8 біт адресних регістрів R0 і R1;

- *неявна* – в команді адресне поле відсутнє, адресна інформація міститься в коді операції, тобто сама операція несе дані про місце знаходження одного або всіх операндів, використаних в ній.

Регістр “слово стану програми” (ССП) включає чотири прапори:

С–перенесення, АС– допоміжне перенесення ОV–переповнення, Р–паритет.

С – встановлюється, якщо в старшому біті результату виникає перенесення/ позика; при виконанні операцій множення і ділення *С* скидається.

АС – встановлюється, якщо при виконанні операції складання/віднімання між тетрадами байта виникло перенесення/позика.

OV – встановлюється, якщо результат складання/віднімання не укладається в семи бітах і старший (восьмий) біт не може інтерпретуватися як знаковий. При виконанні операцій ділення *OV* скидається, і встановлюється тільки при діленні на нуль. При множенні *OV* встановлюється, якщо результат більше 255.

Р – встановлюється і скидається апаратно в кожному циклі команди, фіксує парне/непарне число одиничних бітів в акумуляторі, тобто виконує контроль по парності.

Символічна адресація. При використанні асемблера МК51 (ASM51) для отримання об'єктних кодів програм допускається застосування в програмах символічних імен регістрів спеціальних функцій (РСФ), портів і їх окремих бітів.

Для окремих бітів РСФ і портів (така можливість не у всіх РСФ) можна використовувати символічне ім'я біта наступної структури:

<ім'я РСФ або порта>.<номер біта>.

Символічні імена РСФ і портів є зарезервованими словами і їх не треба перевизначати директивами асемблера.

Біти користувача (0-128), що адресуються, в бітах 32–47 внутрішнього ОЗУ.

Група команд передачі даних. Велику частину команд цієї групи, приведеної у табл. 5.7, складають команди передачі і обміну байтів. Всі команди даної групи не модифікують прапори результату, за винятком команд завантаження PSW і акумулятора (прапор паритету).

Команди логічних операцій виконують операції логічного І, логічного АБО, ВИКЛЮЧАЮЧОГО АБО, очищення, інверсії, зсуву.

Команди арифметичних операцій виконують операції складання, десяткової корекції, віднімання, множення, ділення, інкремента і декремента даних. У операціях цілочисельного множення і ділення без знаку бере участь акумулятор і регістр В. При множенні 8-розрядне значення А помножується на 8-розрядне значення В, а 16-розрядний результат записується в пару ВА. При діленні 8-розрядного значення А на 8-розрядне значення В частка записується в А, а залишок у В.

Група команд операцій з бітами. У командах цієї групи як операнди виступають окремі біти портів, деяких РСФ і 128 програмних прапорів користувача (байт 32–47). Для адресації бітів використовуються пряма 8-розрядна адреса. Непряма адресація бітів неможлива.

Група команд передачі управління включає команди, що забезпечують умовне і безумовне управління, виклик підпрограм і повернення з них, а також порожні операції NOP.

Довгий перехід. Перехід по всьому адресному простору ПП. В команді міститься повна 16-бітова адреса переходу. На практиці рідко виникає необхідність такого переходу.

Абсолютний перехід. Перехід в межах сторінки пам'яті програми розміром 2048 байт. Такі команди містять тільки 11 молодших бітів адреси переходу.

Відносний перехід. Короткий відносний перехід дозволяє передати управління в межах -128 – +127 байт щодо адреси команди, наступної за командою відносного переходу.

Непрямої перехід. Команда JMP @A+DPTR дозволяє передавати управління за прямою адресою, що обчислюється самою програмою і невідома при виконанні початкового тексту програми.

Умовні переходи. Галуження здійснюється по умові рівності або нерівності нулю вмісту акумулятора.

Команда DJNZ зручна при організації циклів. Лічильник циклів організовується в регістрі або прямо в байті, що адресується. При виконанні команди DJNZ проводиться декремент лічильника, перевірка на нуль і, якщо вміст не рівний нулю, то відбувається перехід до початку циклу, а при рівному 0 відбувається вихід з циклу.

Мітки для цієї команди повинні знаходитися в межах однієї сторінки пам'яті. Аналізовані ознаки не фіксуються в спеціальних тригерах (окрім прапорів C і F0), а представляються миттєвими значеннями сигналів в АЛП або на відповідних входах МК51.

Підпрограми. Для виклику їх використовують команду LCALL або ACALL. Ці команди зберігають в стеку адреси повернення в основну програму. Повернення з підпрограми відбувається при виконанні команди RET. Для повернення з підпрограм обслуговування переривання використовують команду RETI, яка відрізняється від команди RET тим, що дозволяє переривання обслуженого рівня.

Асемблер допускає використання узагальненого виду команд JMP і CALL, які в процесі трансляції замінюють оптимальними по формату командами переходу (AJMP, SJMP, LJMP) або виклику (ACALL і LCALL).

Засоби розробки і відладки програмного забезпечення

До цих засобів відносяться програми асемблювання, лінковки і компоновки, програми відладчики – емулятори.

Асемблювання

Асемблер – програма, створена для спрощення завдання написання програм на обчислювальних машинах. Він здійснює задання трансляції символічної програми у виконуваний об'єктний код.

Програма на асемблері складається з рядків трьох типів:

- команди процесора;
- директиви асемблера;
- параметри, що управляють.

Команди процесора – мнемонічне кодування машинних команд, які повинен виконати процесор.

Директиви асемблера – псевдокоманди, які при трансляції не породжують кодів машинних команд і використовуються для визначення змінних, структури програми, завдання констант і т.ін.

Команди, що управляють, – вказівки транслятору на виконання певних дій, наприклад, створення лістингу.

Програма асемблювання ASM51 дозволяє використовувати модульний принцип (розбиття складної програми на окремі функціональні частини, тобто модулі).

Після відладки всіх модулів окремо, вони можуть бути зібрані воєдино спеціальною програмою лінковки RL51.

Початкова програма створюється за допомогою редактора текстових файлів і містить коментарі, параметри асемблера і директиви, а також команди, що управляють, на мові асемблера.

Далі викликається програма асемблювання ASM51.EXE з вказівкою імені трансльованої програми, наприклад: `asm51 myprog.a51`

Результатом роботи асемблера буде:

- об'єктний файл (виконувана форма вихідних даних асемблера, що є абсолютний формат 16-річного коду), який може бути запрограмований на МК51 і включає необхідну для збірки і настройки інформацію;
- лістинговий файл (запис початкової програми об'єктного коду), в який асемблер вносить діагностичні повідомлення про синтаксичні та інші помилки кодування.

Директиви асемблера

Асемблер має декілька директив, які дозволяють користувачу: встановити значення символічних імен; зарезервувати і ініціалізувати місце пам'яті; управляти розміщенням програм.

Директиви визначення символічних імен дозволяють створювати символічні імена, які можуть застосовуватися для поліпшення читабельності програми. За допомогою цих директив можна визначати символічні імена для посилань на сегменти, адреси даних, адреси бітів, адреси зовнішніх даних:

```
SEGMENT  DATA  BIT
EQU      IDATA  CODE
SET      XDATA
```

Директиви вибору сегментів призначають подальший код або дані у вибраному сегменті до тих пір, поки інший сегмент не буде вибраний:

```
RSEG  CSEG  USING
BSEG  DSEG
ISEG  XSEG
```

Директиви управління станом асемблера:

- управління лічильником асемблера – ORG;
- міжмодульний зв'язок: PUBLIC, EXTRN, NAME;
- ініціалізація пам'яті:
DBIT DB
DW DS
- кінець програми (обов'язкова директива) – END.

Символічні імена можуть містити всі 28 латинських літер, цифри 0-9, знаки (? , –) . Символічні імена не можуть починатися з цифри. У імені можна використовувати до 255 символів, але значущими є лише перший 31 символ.

Мітка – символічне ім'я. Щоб виділити мітку, після неї потрібно ставити двокрапку (:).

Директиви, які не можуть мати мітку:

```
BIT; EQU; SET; CODE; IDATA; XDATA; DATA; ORG; END; SEGMENT.
```

Відладчик асемблерних програм дозволяє:

- завантажити для відладки HEX-файли, що виробляються трансляторами з мови асемблера (крос-засоби), а також файли чистого двійкового коду, що прочитуються з ПЗП;
- переглянути на екрані дизасембльований текст завантаженої програми, включаючи адреси і коди команд, область імітованого ОЗУ даних, зовнішньої пам'яті, пам'яті програм, вміст всіх регістрів;
- виконати завантажenu програму по кроках з прогляданням результатів після кожного кроку і в безперервному режимі з остановкою по крапках;
- внести зміни в завантажenu програму в мнемонічних позначеннях мови асемблера, а також в машинних кодах;
- внести зміни у вміст регістрів, прапорів і пам'яті.

Об'єктний файл перед відладкою на емуляторі слід перетворити в 16-річний. Для цього може бути використана крос-програма OH.EXE. Формат: oh турrog.obj. В цьому випадку одержимо результат турrog.hex.

Вид вікна відладчика-емюлятора FD51 приводиться нижче на рис. 5.12.

Для швидкого перегортання можна користуватися наступними клавішами:

- [Home] – перегортає вікно пам'яті даних вгору на одну сторінку;
- [End] – перегортає вікно пам'яті даних вниз на одну сторінку;
- [PgUp] – перегортає вікно пам'яті програм вгору на одну сторінку;
- [PgDn] – перегортає вікно пам'яті програм вниз на одну сторінку.

Призначення функціональних клавіш, за допомогою яких виконують команди, що найбільш часто зустрічаються, приведено нижче в табл. 5.8.

Команди відладчика

При описі команд відладчика використовують наступні позначення:

- параметри поміщені в кутові дужки, наприклад <адреса>;
- необов'язкові параметри поміщені в квадратні дужки, наприклад [<адреса>];
- всі числові значення повинні мати шістнадцятирічний формат, при

цьому не потрібно указувати букву "h".

H або Ctrl-H - швидке отримання довідки.

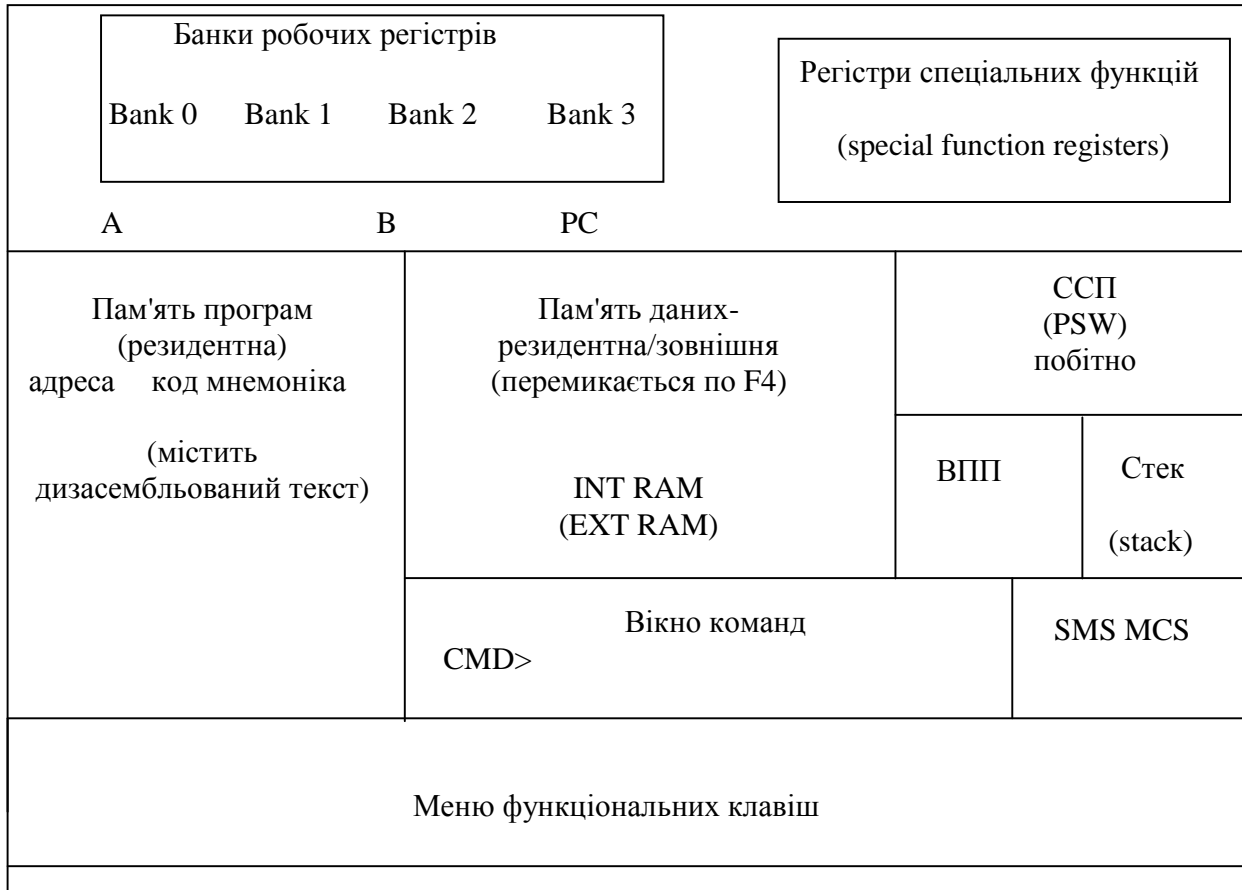


Рисунок 5.12 – Вікно програми відладчика – емулятора FD51.EXE

Таблиця 5.8 – Призначення функціональних клавіш

F1	Виконує поточну інструкцію завантаженої програми (поточною є інструкція, що підсвічується, у вікні дизасембльованого тексту)
F2	Виконує програму до наступної за адресою за поточною інструкцією
F3	Дозволяє представляти числову інформацію в десятковій, двійковій і шістнадцятиричній формах (для регістрів)
F4	Перемикає вікно пам'яті даних з внутрішньою (INT RAM) на зовнішню (EXT RAM) і назад

Продовження табл. 5.8

F5	Установка крапок переривання
F6	Перемикає форму представлення пам'яті у вікні в двійкову і шістнадцятирічну
F7	Перегортає вікно пам'яті даних вгору на один рядок
F8	Перегортає вікно пам'яті даних вниз на один рядок
F9	Перегортає вікно пам'яті програм вгору на один рядок
F10	Перегортає вікно пам'яті програм вниз на один рядок

L [<тип пам'яті><поч. адреса>] <файл. спец.>[/A] – *завантажити файл в пам'ять*. <Тип пам'яті> може бути I, E або P. Відповідно до цього параметра файл завантажується у внутрішню (Int), зовнішню (Ext) або програмну (Pgm) пам'ять. <Поч. адреса> і <тип пам'яті> указується тільки при завантаженні чистого двійкового коду. При завантаженні файлу, виробленого ISIS-II MACRO-ASSEMBLER'ом, потрібно вказати тільки специфікацію файлу і ключ /A. Приклад: L I 01F, A:\PGM\T1 – завантажити двійковий файл у внутрішню пам'ять з адреси 01F.

S <тип пам'яті><поч. адреса>-<кін. адреса>,<файл. спец.> – *зберегти область пам'яті в дисковому файлі* (взагалі кажучи, у всіх командах в якості <файл. спец.> допускається будь-яка коректна в DOS специфікація файлу, наприклад COM1), <поч.адреса> і <кін.адреса> вказують відповідно початок і кінець області, що зберігається. Збережений командою S файл можна потім знову завантажити командою L. Приклад: S P 20-642, C:\PGMLIB\MYFILE.

PRT <тип пам'яті><поч. адреса>-<кін. адреса>,[<файл. спец.>] – *роздрукувати дамп області пам'яті в шістнадцятирічному форматі*. Якщо не вказано <файл. спец.>, то дамп виводиться на принтер.

PRTD <поч. адрес>,<кількість команд>,[<файл. спец.>] – *роздрукувати дизасембльований текст, починаючи з <поч. адреси>*. Вивід за умовчанням на принтер.

R <номер регістру>=<число> – *занести число в регістр поточного банку*. Число повинне бути байтом. Приклад: R4=FF.

<Ім'я регістра>=<число> – занести число в регістр спеціального призначення. Можна використовувати наступні імена: А, В ТН0, ТН1, ТL0, ТL1, DPH, DPL, DPTR, SP, IP, ІЕ, ТMOD, ТCON, SCON, SBUF, РС. Число для РС і DPTR може бути і двобайтовою величиною. Приклад: SP=20 DPTR=FF00.

<Ім'я флага>=<число> – встановити або скинути прапор в PSW. Імена прапорів: С, АС, F0, S1, S0, OV, Р. Якщо число=0, то прапор скидається, інакше – встановлюється. Приклад: S1=0.

P0 <номер порта>=<число> – занести число в порт. Номер порту може бути 0-3. Приклад: P02=12.

D <адрес> – встановити адресу дизасембльованого тексту у вікні. Приклад: D 0240.

<Тип пам'яті><адреса>[-<кін. адреса>]=<число> – занести число в пам'ять. Якщо вказана <кін. адреса>, то цим числом заповнюється область пам'яті. Приклад: I 22=55 P 0-40=FF. Можливо виникнення неоднозначності при заповненні деяких елементів пам'яті. Наприклад, команда "PC=23" буде сприйнята не як занесення числа 23 в пам'ять програм за адресою 0С, а як команда установки лічильника команд (PC) в значення 23. В цьому випадку потрібно явно вказати, що це адреса P0C=23.

<Тип пам'яті><адреса>.<номер біта>=<число> – встановити або скинути біт в пам'яті. <Номер біта> може бути 7-0 (старший біт – 7). Приклад: I 20.6=1.

<Ім'я регістра>.<номер біта>=<число> – встановити або скинути біт в регістрі спеціального призначення (А,В,Р00-Р03,IP,ІЕ,ТMOD,ТCON,SCON).
Приклад: ТMOD.3=0.

M <тип пам'яті><поч. адрес> – встановити початкову адресу пам'яті у вікні. Приклад: M I 20 M E 0FF M P 0.

G [<поч. адреса>,<кін. адреса>] – виконати програму <поч. адреса> до <кін. адреса>. Якщо <поч. адреса> не вказаний, виконання починається з поточної команди (поточна команда виділена білим прямокутником). <Кін. адреса> можна не указувати, якщо використовуються точки переривання. Програму, що виконується, можна зупинити натисненням будь-якої клавіші. G

без параметрів можна ввести натисненням <Alt-F10>. Можна вказати тільки кінцеву адресу, але кома повинна бути присутньою. Приклад: G 100-FF0 G,2200.

T ON [,<файл. спец.>] – *включити трасування програми*. За умовчанням записи трасувань виводяться на принтер.

T OFF – *вимкнути трасування*.

INT <0/1>=<число> – *імітувати високий або низький рівень на входах INT0 або INT1*. Приклад: INT1=0.

BA=<адреса> – *встановити нову "точку відліку" для дизасемблювання*. Ця команда корисна при прогляданні таблиць, зашитих в пам'яті програм, коли при дизасемблюванні "назад" невідомо, звідки вести дизасемблювання.

RSTC – *скинути лічильник часу виконання програми*.

QUIT – *вихід в DOS*.

RST – *імітується скидання процесора*.

N – *очищення вмісту відладчика FD51*.

Відразу ж після запуску відладчик готовий до прийому команд користувача – курсор знаходиться в командному рядку.

Повноекранне редагування

Перехід в режим повноекранного редагування здійснюється натисненням клавіші [Enter] без вводу команди. Тепер можна переміщати курсор по екрану за допомогою клавіш управління курсором і змінювати вміст регістрів, пам'яті і прапори набором чисел на клавіатурі. Можна змінити також початкову адресу дизасембльованого тексту (поточної інструкції) і початкові адреси вікон пам'яті (у перших рядках вікон). Повноекранне редагування можна проводити і при десятковому, і при двійковому представленні інформації на екрані. Під час редагування залишаються доступними всі команди, що вводяться функціональними клавішами. Щоб повернутися в командний рядок, натисніть [Enter] знову. Для швидкого переміщення курсора по екрану можна користуватися клавішами [Tab] і [Shift]-[Tab].

Режим асемблера

Для переходу в режим асемблера (вводу команд відладжуваної програми в мнемонічних позначеннях) потрібно в режимі повноекранного редагування помістити курсор в полі поточної інструкції завантаженої програми. Тепер наберіть мнемоніку (наприклад, "MOV A,#45") і натисніть [Enter]. Якщо мнемоніка вірна, то відповідні їй коди заносяться в пам'ять програм, а вікно встановлюється на наступну адресу.

При асемблюванні підтримуються імена регістрів спеціального призначення. При виникненні неоднозначності слід числовим значенням передувати нулем.

Для виходу з режиму асемблера натисніть клавішу "Quit" або відведіть курсор з поля поточної інструкції.

Робота з крапками переривання

Меню крапок переривання викликається клавішею F5. Можна встановити одночасно 8 крапок переривання.

Переривання (остановка) програми, що виконується, відбувається при досягненні вказаної в колонці "PC" адреси і виконанні умови "Counter" = "Occur". "Counter" – це лічильник, значення якого визначає, скільки разів програма повинна пройти через вказану адресу, щоб відбулась остановка. "Occur" показує, скільки разів програма проходила через вказану адресу.

Закінчивши редагування, поточні значення крапок переривання можна зберегти на диску (клавіша F2). У користувача запрошується номер набору крапок переривання (0-9). Інформація записується у файл з ім'ям FD51.BRK. Відновити картинку можна клавішею F1, також вказавши її номер.

Для повернення в основне меню слід натиснути клавішу F5.

Визначивши крапки переривання, можна запустити програму командою G без параметрів.

При остановці програми по перериванню видається повідомлення з вказівкою номера крапки переривання.

Детальніше робота відладчика розглядається нижче в п. 6.3 (лабораторна робота № 3. Складання і відладка програм на мові Асемблер).

5.3. Мікроконтролери фірми Atmel

Фірма Atmel виробляє такі серії мікроконтролерів: серію AT89, сумісну з фактичним промисловим стандартом MCS-51™, і серію RISC-мікроконтролерів AT90 власної архітектури. Характерною відмінною особливістю мікроконтролерів фірми Atmel є пам'ять програм, що базується на FLASH-пам'яті. Це забезпечує швидкість програмування і зменшення циклу розробки. Додатково передбачений спеціальний, послідовний інтерфейс, що дозволяє програмувати і відладжувати мікроконтролери безпосередньо на монтажній платі.

Мікроконтролери серії AT89, сумісні з MCS-51™

Мікроконтролери з системою команд і архітектурою MCS-51™ представляти немає необхідності. Вони широко поширені, безліч фірм випускає їх модифікації і програмне забезпечення для них. Випускає такі мікроконтролери і фірма Atmel. Відмінною особливістю цих мікроконтролерів є застосування FLASH-пам'яті програм. Ця особливість дозволяє практично миттєво змінювати програмний код мікроконтролера, що істотно скорочує цикл розробки.

Мікроконтролери в корпусі з 40/44 виводами повністю сумісні по виводах з контролерами 80C51 і забезпечують можливість використання напрацьованих програм і прямої заміни. FLASH-пам'ять програм робить також можливою дистанційну зміну програмного коду вбудованих мікроконтролерів безпосередньо у замовника.

Мікроконтролери серії AT89 мають наступні основні особливості:

- 8-розрядний процесор, оптимізований для управління;
- обширні можливості побітової обробки;
- вбудовану FLASH -пам'ять програм;

- вбудовану оперативну пам'ять;
- двонаправлені лінії вводу-виводу, що індивідуально адресуються;
- один або декілька 16-розрядних таймерів/лічильників;
- повнодуплексний UART;
- розгалужену структуру переривань;
- вбудований тактовий генератор;
- економічні режими: IDLE і POWER DOWN ;
- вбудовану пам'ять EEPROM (AT89S);
- послідовний інтерфейс SPI (AT89S);
- сторожовий таймер (AT89S).

Склад сімейства мікроконтролерів представлений в таблиці 5.9.

Розглянемо детальніше особливості архітектури мікроконтролерів.

Для забезпечення економії споживання енергії мікроконтролери мають два програмно керовані режими роботи із зниженою потужністю.

У режимі IDLE процесор вимкнений, тоді як оперативна пам'ять і вбудовані периферійні пристрої продовжують функціонувати. У цьому режимі споживання струму зменшується приблизно на 15 % від споживання повністю активного пристрою.

У режимі POWER DOWN всі пристрої мікроконтролера вимкнені, проте дані в оперативній пам'яті продовжують зберігатися. У цьому режимі споживання мікроконтролера складає менше 15 мкА і у будь-якому випадку не більше 0,6 мкА. Крім того, мікроконтролери розроблені із застосуванням статичної логіки, яка не вимагає безперервної синхронізації. Тому частота тактового генератора може бути зменшена чи ж він може бути зупинений в очікуванні події, що вимагає обробки. Це також сприяє пониженому електроживленню.

Мікроконтролери мають окремі адресні простори для пам'яті програми і даних. Це дозволяє звертатися до пам'яті даних 8-бітовими адресами, чим забезпечується швидкість операцій, що виконуються з пам'яттю 8-розрядним процесором.

Таблиця 5.9 – Основні характеристики МК фірми Atmel

Тип мікроконтролера	AT89 C51	AT89 LV5	AT89 C52	AT89L V52	AT89 C2051	AT89 C151	AT89 S852
Пам'ять програм	4	4	8	8	2	1	8
Пам'ять даних	128	128	256	256	128	64	256
Пам'ять EEPROM	–	–	–	–	–	–	2
Виведення вводу/ виводу	32	32	32	32	15	15	32
16-бітові таймери / лічильники	2	2	3	3	2	1	3
UART	+	+	+	+	+	--	+
Джерела переривань	6	6	8			3	9
Біти захисту	3	3	3	3	8	6	3

Разом з тим, за допомогою регістра, ОПТК може згенерувати 16-бітову адресу даних. Таким чином, може бути адресовано до 64 кбайт зовнішньої пам'яті, для якої контролер генерує сигнали читання і запису RD і WR. Для пам'яті програми забезпечується тільки читання. Безпосередньо адресується до 64 кбайт пам'яті програм. Для читання зовнішньої пам'яті програм контролер генерує сигнал PSEN. Зовнішні пам'ять програми і пам'ять даних можуть бути об'єднані по логічному I для сигналів контролера RD і PSEN.

Після виконання процедури скидання, виконання програми починається з адреси 0000H. З адреси 0003H розташовуються блоки обробки переривань, що займають по 8 байтів. Якщо процедура обробки переривання займає не більше 8 байтів, вона може розташовуватися в цьому блоці. Процедури обробки переривань більшого розміру розміщуються в інших областях пам'яті програми, а управління передається їм з блоків обробки переривань командами безумовного переходу. Якщо переривання в програмі не використовуються, адреси, зарезервовані під блоки обробки переривань, можуть бути зайняті

кодом програми.

Нижні адреси пам'яті програми можуть адресувати як вбудовану FLASH-пам'ять програми, так і зовнішню пам'ять залежно від того, чи сполучене виведення управління зовнішнім доступом EA з ланцюгом живлення або із загальним дротом відповідно. Зовнішня пам'ять програми може бути адресована 8-розрядною адресою з використанням порту вводу-виводу P0 для організації мультиплексованої шини адреси даних або 16-розрядною адресою з використанням портів вводу-виводу P0 і P2, причому останній передає старший байт адреси.

Зовнішня пам'ять даних може мати об'єм 64 кбайт. Адресується вона також як і пам'ять програми 8 або 16-розрядною адресою. Простір пам'яті даних розділений на 3 блоки: нижні – 128 байтів, верхні – 128 байтів і область регістрів спеціальних функцій (Special Function Registers – SFR). Внутрішня пам'ять даних завжди адресується одним байтом, що відповідає максимальному об'єму пам'яті 256 байтів. Проте застосування різних способів адресації дозволяє використовувати до 384 байтів внутрішньої пам'яті. Це пов'язано з тим, що при прямій і непрямій адресації простору зверху FFH адресуються різні області пам'яті (верхні 128 байтів і область SFR).

У нижніх 128 байтах внутрішньої пам'яті перші 32 байти зайняті чотирма банками по 8 регістрів, що адресуються командами програми як R0...R7. Вибір банку, в якому адресуються регістри, забезпечується відповідною установкою двох бітів RS0 і RS1 у регістрі слова стану програми (PSW). Така архітектура дозволяє ефективніше використовувати кодовий простір, оскільки команди звернення до регістрів коротші за команди прямої адресації пам'яті. Наступні, після банків регістрів, 16 байтів внутрішньої пам'яті утворюють блок бітової адресації.

Система команд мікроконтролера включає широкий вибір команд бітової адресації, які можуть безпосередньо адресувати 128 біт в цій області. Біти, що адресуються, мають адреси 00H...7FH.

Всі байти в нижніх 128 байтах можуть адресуватися прямим і непрямим методом адресації. До верхніх 128 байтів, доступних тільки в контролерах з об'ємом пам'яті 256 байтів, можна звертатися тільки із застосуванням непрямой адресації. Простір *SFR* (регістрів спеціальних функцій) включає порти вводу-виводу, реєстри таймерів, реєстри управління периферійними пристроями і т.ін.. Ця область може адресуватися тільки прямою адресацією. Структура простору ідентична структурі аналогічного простору контролерів сімейства MCS-51™, проте є додаткові реєстри. 16 адрес в просторі 8РК можуть адресуватися і побайтно, і порозрядно. Адреси реєстрів, що розрядно адресуються, закінчуються трьома нулями. Адреси бітів цих реєстрів мають значення 80H...FFH.

Система команд мікроконтролерів оптимізована для 8-розрядних додатків управління, забезпечує ряд швидких способів адресації для доступу до внутрішньої оперативної пам'яті. Система команд забезпечує обширну підтримку для однобітових змінних як окремого типу даних, дозволяючи виконувати пряме розрядне маніпулювання в управлінні і логічних системах, які вимагають булевої обробки.

Слово стану програми (*PSW*) містить біти стану, які відображають поточний стан процесора, і розміщується в просторі *SFR*. Слово стану програми містить біт перенесення *CY*, біт додаткового перенесення (для операцій зі BCD – двійково-десятковим кодуванням) *AC*, біти вибору банку *RS0* і *RS1*, прапор переповнювання *OV*, біт контролю по парності *P* і два прапори стану, які визначаються користувачем.

Мікроконтролери мають команди прямої і непрямой адресації, реєстрові команди і спеціальні команди для деяких реєстрів. У останньому випадку код команди безпосередньо вказує на реєстр, з яким проводитиметься операція. При прямій адресації операнд визначений 8-розрядним полем адреси в команді. Цим методом може бути адресована тільки внутрішня оперативна пам'ять і простір *SFR*. При непрямій адресації в команді вказаний реєстр, який містить адресу операнда. Таким методом може адресуватися як внутрішня, так

і зовнішня оперативна пам'ять. Як регістр адреси для 8-розрядних адрес може бути або покажчик вершини стека, або регістри R0, або R1 вибраного банку. Регістром адреси для 16-розрядних адрес може бути тільки 16-розрядний регістр покажчика даних DPTR. До банків регістрів, які містять регістри R0...R7, можна звертатися командами, чий код операції включають 3-розрядну специфікацію регістра. Команди, які звертаються до регістрів цим способом, забезпечують ефективне використання коду програми, оскільки при цьому в команді відсутній байт адреси. Банк, в якому поточною командою адресується регістр, вибирається відповідною установкою двох бітів в слові стану програми PSW.

Значення константи може слідувати за кодом операції в пам'яті програми. До пам'яті програми можна звертатися тільки через індексну адресацію. 16-розрядний базовий регістр (або DPTR, або лічильник команд PC) указує на початок поточної команди.

Машинний цикл мікроконтролерів складається з 6 станів – S1...S6, кожне з яких, займає два такти тактового генератора. Таким чином, тривалість машинного циклу складає 12 тактів тактового генератора, або при тактовій частоті 12 МГц – 1 мкс. Команда програми може бути виконана протягом одного або декількох машинних циклів, наприклад команда MOV X займає два машинні цикли.

Стандартно мікроконтролери мають 5 джерел переривань: 2 зовнішніх переривання, 2 переривання по таймеру і переривання від послідовного порту. У деяких контроллерах є додаткові джерела переривання відповідно до особливостей їх архітектури. Переривання по кожному з джерел може бути індивідуально дозволено або заборонено шляхом установки або скидання відповідних бітів в регістрі дозволу переривань IE, розташованому в просторі SFR.

В процесі виконання програми, стани прапорів переривань прочитуються в п'ятому стані машинного циклу і опитуються в наступному циклі. Для

кожного з джерел переривань може бути запрограмований один з двох рівнів пріоритету шляхом установки або скидання відповідного біта в регістрі пріоритетів переривань *IP*, розташованому в просторі *SFR*. Низькопріоритетне переривання може бути перервано високопріоритетним перериванням, але не іншим низькопріоритетним перериванням. Виконання процедури високопріоритетного переривання не може бути перервано ніяким перериванням. Якщо одночасно надійшли два запити на переривання з різними рівнями пріоритету, то спочатку виконується процедура високопріоритетного переривання. Під час вступу запитів на переривання з однаковим рівнем пріоритету порядок виконання процедур обробки переривання визначається внутрішньою послідовністю опитування.

В процесі обробки переривання апаратна процедура *LCALL* поміщає вміст лічильника команд *PC* в стек і завантажує початковою адресою відповідного блоку обробки переривання. Окрім лічильника команд, автоматично в стеку не зберігаються ніякі інші регістри. За збереження інших необхідних регістрів відповідає програміст. У ряді випадків це дозволяє скоротити час обробки переривання. В результаті багато функцій обробки переривань, які є типовими в додатках управління (перемикання виведення порту, перезавантаження таймера або читання буфера послідовного порту), можуть бути завершені швидше, ніж це було б можливо при іншій архітектурі. Багато додатків вимагають більше двох рівнів пріоритетності переривань, які забезпечуються апаратними засобами мікроконтролерів. У такому разі можливо застосування простого програмного коду, за допомогою якого емулюється третій рівень пріоритетності переривання.

Послідовний порт мікроконтролерів – повнодуплексний, з буфером приймача. Доступ до регістрів прийому і передачі здійснюється через регістр *SBUF* в просторі *SFR*. При виконанні запису в цей регістр завантажується регістр передачі, читання забезпечує доступ до регістру прийому. При тактовій частоті мікроконтролера 12 МГц залежно від встановленого режиму послідовний порт забезпечує швидкість обміну до 1 Мбод. Проте для

забезпечення стандартних швидкостей обміну 1200. 19200 бод необхідно тактувати мікроконтролер з частотою 11,059 МГц, при цьому для отримання необхідної швидкості обміну використовується один з таймерів.

Мікроконтролери мають два режими роботи із зниженим споживанням по живленню – *idle* і *Power Down*, перехід в які забезпечується установкою відповідних бітів в регістрі *PCON* простору *SFR*.

У режимі *idle* ($IDL=1$) тактовий генератор продовжує працювати, і забезпечується робота периферійних пристроїв – таймерів, блоку обробки переривань і послідовного порту, при цьому процесор мікроконтролера зупиняється в очікуванні надходження переривання.

У режимі *Power Down* ($PD=1$) зупиняється тактовий генератор мікроконтролера, проте вміст вбудованої пам'яті і регістрів простору *SFR* зберігається.

Вихід із стану *Power Down* можливий тільки при виконанні апаратного скидання. При скиданні переініціалізуються всі регістри простору *SFR*, проте, вміст внутрішньої пам'яті даних не змінюється. Апаратне скидання запускає також тактовий генератор мікроконтролера. На час перебування МК в режимі *Power Down* напруга живлення може бути понижена 2В. Проте, напруга живлення не повинна бути знижена до того, як МК перейшов в режим *Power Down*, і повинно бути відновлено перед виконанням скидання. Сигнал скидання повинен бути достатньо тривалим (10 мс) для стабілізації роботи тактового генератора.

5.4. Мікроконтролери фірми Microchip

Мікроконтролери сімейств PIC об'єднують всі передові технології мікроконтролерів: електрично програмовані користувачем ППЗУ, мінімальне енергоспоживання, високу продуктивність, добре розвинену RISC- архітектуру, функціональну закінченість і мінімальні розміри. Широка номенклатура виробів забезпечує використання мікроконтролерів у виробках, призначених для

різноманітних сфер застосування. Ось короткий перелік виробів, побудованих на базі мікроконтролерів: мікро - АТС, автовідповідачі, АВНи, мобільні телефони, зарядні пристрої, системи сигналізації, вимірювальні прилади, лічильники води, газу і електроенергії, прилади сигналізації, регулятори температури, вологості, тиску і т.ін.

Такий широкий спектр їх застосування став можливий завдяки тому, що мікроконтролери сімейств PIC16/17 мають оптимальні, для побудови даних систем, архітектуру і параметри.

Нижче, в табл.5.10 приводиться короткий перелік мікроконтролерів даного сімейства і їх характерні архітектурні особливості.

У табл. 5.10 використовуються наступні умовні позначення: OTP – одноразово програмований кристал; ROM – пам'ять програм з масочним ПЗП; П – електрично програмована пам'ять; WDT – сторожовий таймер; I²C – послідовний інтерфейс; SPI – синхронний послідовний інтерфейс; USART – асинхронний послідовний інтерфейс; BROD – режим перезапуску при зменшенні напруги живлення; Комп – аналоговий компаратор; Uref – програмоване джерело опорної напруги; АЦП, ЦАП – аналого-цифровий і цифро-аналоговий перетворювачі; Slave port – режим обміну з контроллером по паралельному каналу (DATA0.. 7, WR, RD, CS); PWM – широтно-імпульсний модулятор; PCP – робота в режимі компаратора/ захвата/шим-модулятора; CAP – режим таймера; ICSP – програмування на платі; FLASH-пам'ять програм;

PIC16 – це сімейство мікроконтролерів, розроблене спеціально для застосування у вбудованих системах управління і контролю, де необхідні висока надійність і низька вартість. Для ілюстрації всього вищесказаного розглянемо молодшу групу даного сімейства – МК типу PIC16C5XX, представлену в табл. 5.11.

Архітектура всіх мікроконтролерів сімейства PIC16/17 заснована на концепції роздільних шин областей пам'яті для даних і для команд (Гарвардська архітектура). Шина даних і пам'ять даних (ОЗП) – мають ширину 8 біт, а програмна шина і програмна пам'ять (ПЗП) – 12 біт (PIC16C5XX).

Таблиця 5.10 – Сімейство мікроконтролерів PIC16/17

Назва	Пам'ять програм	OTP/ROM	RAM	МГц	I/O	Таймер	CCP/PWM	Послідовний інтерфейс	Особливості	Корпус
PIC16C177	12288x16	OTP	1024	40	33	3+WDT	2PWM	USART I ² C/SPI	Slave port 8р., АЦП 8р. BX..BROD, ICSP,(поліп- шений такто- вий генератор)	40P, 40JW. 4Lь, 44PT
PIC16C178	16384x16	OTP	1536	40	22	3+WDT	2PWM	USART I ² C/SPI	Slave Port, 8р. АЦП 5BX. BROD, ICSP, (поліпшений. тактовий генератор)	28P, 28S0, 28W
PIC16C179	16384x16	OTP	1536	40	33	3+WDT	2PWK	USART I ² C/ SPI	Slave Port, 8р. АЦП 8EX. .BROD, ICSP (поліпшений тактовий генератор)	40P,40JW. 44L,44PT
PIC16C185	8192x16	OTP	1024	40	50	4+WDT	3PWIK	USART I ² C/SPI	Slave Port, 10р. АЦП 5BX.. BROD, ICSP, (поліпшений тактовий генератор)	64SP.68CL, 68L,в4PT

Така концепція забезпечує просту, але потужну систему команд, розроблену так, що бітові, байтові і регістрові операції працюють з високою швидкістю і з перекриттям за часом вибірок команд і циклів виконання. 12-бітова ширина програмної пам'яті забезпечує вибірку 12-бітової команди в один цикл. Двоступінчатий конвеєр забезпечує одночасну вибірку і виконання команди. І дозволяє виконувати кожну команду за один цикл. Якщо команда змінює програмний лічильник (наприклад GOTO), то для завершення команди потребується другий цикл.

Таблиця 5.11 – Мікроконтролери групи PIC16C5XX

Назва	Частота, МГц	Виводи	Порти	ПЗП	ОЗП
PIC16C52	4	18	12	384x12 ППЗУ	25x8
PIC16C54	20	18	12	512x12 ППЗУ	25x8
PIC16C54A	20	18	12	512x12 ППЗУ	25x8
PIC16CR54	20	18	12	512x12 ПЗП	25x8
PIC16CR54A	20	18	12	512x12 ПЗП	25x8
PIC16CR54B	20	18	12	512x12 ПЗП	25x8
PIC16C55	20	28	20	512x12 ППЗУ	25x8
PIC16C56	20	18	12	1024x12 ППЗУ	25x8
PIC16CR56	20	18	12	1024x12 ПЗП	25x8
PIC16C57	20	28	20	2048x12 ППЗУ	72x8
PIC16CR57A	20	28	20	2048x12 ПЗП	72x8
PIC16CR57B	20	28	20	2048x12 ПЗП	72x8
PIC16C58A	20	18	12	2040x12 ППЗУ	72x8
PIC16CR58A	20	18	12	2048x12 ПЗП	72x8
PIC16CR58B	20	18	12	2048x12 ПЗП	72x8

Синхронізація командних циклів і конвеєрна обробка команд. Тактова частота з входу OSC1 ділиться на чотири при формуванні чотирьох фаз синхронізації, які не перетинаються: такти Q1, Q2, Q3, Q4. Програмний лічильник (PC) збільшується на кожному такті Q1. Вибірка команди із пам'яті програм і запис її в регістр виконується на такті Q4. Команда декодується і виконується на протязі наступних чотирьох тактів Q1... Q4.

Командний цикл складається із чотирьох тактів. Вибірка і виконання команд проводиться у конвеєрний спосіб, тобто в одному командному циклі декодується і виконується одна команда і вибирається наступна. Конвеєрна обробка дозволяє виконувати кожну команду за один цикл. Якщо команда змінює програмний лічильник (наприклад, GOTO), то для завершення команди потрібен другий цикл. Цикл вибірки починається з інкрементування програмного лічильника на протязі такта Q1. В циклі виконання на протязі такта Q1, вибрана і записана команда знаходиться в регістрі команд (IR). Ця команда потім декодується і виконується на протязі тактів Q2, Q3 і Q4. Дані із

пам'яті зчитуються на протязі такта Q2 (читання операнда), а запис виконується під час такту Q4 (запис результату). Всі команди виконуються за один цикл, крім команд розгалуження програми, які виконуються за два цикли, так як вибрана команда забирається із конвеєра, а нова команда вибирається і потім виконується. У PIC16/17 програмна пам'ять розташована усередині кристала. Виконувана програма може знаходитися тільки у вбудованому ПЗП.

Основні технічні характеристики мікроконтролерів PIC16C5XX:

- 33 команди і всі команди виконуються за один цикл, окрім команд переходу (2 цикли);

- робоча частота 4 ... 20 МГц ;
- 12-бітові команди і 8-бітові дані;
- 7 (8) спеціальних апаратних регістрів SFR
- дворівневий апаратний стек;
- пряма, безпосередня, непряма і відносна адресація даних і команд;
- 12 (20) ліній вводу-виводу з індивідуальною настройкою;
- максимальні вхідний/вихідний струми – 10 мА;
- 8-бітовий таймер/лічильник RTCC з 8-бітовим програмованим попереднім дільником;

- автоматичне скидання при включенні;
- таймер включення при скиданні;
- сторожовий таймер (WDT) з власним вбудованим генератором, що забезпечує підвищену надійність;

- EPROM біт секретності для захисту коду;
- біти ідентифікації;
- економічний режим SLEEP;
- вибрані користувачем біти для установки режиму збудження вбудованого генератора;

- RC генератор (RC);
- звичайний кварцевий резонатор (XT);
- високочастотний кварцевий резонатор (HS);

- економічний низькочастотний кристал (LP);
- вбудований пристрій програмування EPROM пам'яті програм (використовуються тільки два виведення);
 - економічна високошвидкісна КМОП EPROM технологія;
 - статичний принцип в архітектурі;
 - широкий діапазон напруг живлення і температур;
 - комерційний: 2,0...6,0 В; 0...+70 °С;
 - промисловий: 2,0...6,0 В; -40...+70 °С;
 - низьке енергоспоживання;
 - 2 мА (типово для 5 В, 4 МГц),
 - 15 мкА (типово для 3 В, 32 кГц);
 - 1 мкА (у режимі очікування при 3 В);

Регістри пам'яті даних (ОЗП) розділяються на дві функціональні групи: спеціальні регістри і регістри загального призначення. Спеціальні регістри включають регістр таймера/лічильника реального часу (TMRO), лічильник команд (PC), регістр стану (STATUS), регістри вводу/виводу (PORT) , і регістр непрямої адресації (PSR). Крім того, спеціальні регістри управляють конфігурацією портів вводу/виводу і режимів попереднього дільника. Регістри загального призначення використовуються програмою для зберігання змінних по розсуду користувача. У мікроконтролерах сімейства PIC16C5X існує пряма і непряма адресація всіх регістрів і елементів пам'яті. Всі спеціальні регістри і лічильник команд також відображаються на пам'яті даних.

Мікроконтролери PIC16C5X мають ортогональну (симетричну) систему команд, що дозволяє виконувати будь-яку операцію з будь-яким регістром, використовуючи будь-який метод адресації. Це полегшує програмування для них, і значно зменшує час, необхідний на навчання роботи з ними.

У мікроконтролерах PIC16C5X є 8-розрядний арифметико-логічний пристрій (АЛП) і робочий регістр W. АЛП виконує складання, віднімання, зміщення, бітові і логічні операції. У командах, що мають два операнди, одним з операндів є робочий регістр W. Другий операнд може бути константою або

вмістом будь-якого регістра ОЗП. У командах з одним операндом операнд може бути вмістом робочого регістра або вмістом будь-якого регістра ОЗП. Для виконання всіх операцій АЛП використовується робочий регістр W, який не може бути прямо адресований. Залежно від результату виконання операції можуть змінитися значення бітів перенесення C, десяткового перенесення DC і нуля Z в регістрі стану STATUS. При відніманні біти C і DC працюють, як біти займу і десяткового займу, відповідно. Регістр STATUS доступний для будь-якої команди так само, як і будь-який інший регістр. У його склад входять так само біти T0 і PD, які встановлюються апаратно і не можуть бути змінені програмно. Дані розряди встановлюються у відповідні стани при включенні живлення, скиданні і переході в режим SLEEP. Проводячи їх програмне опитування, можна визначити спосіб запуску програми.

Сімейство мікроконтролерів PIC16C5X містить дворівневий апаратний стек. При виконанні команди звернення до підпрограми у вершину стека завантажується лічильник команд, заздалегідь збільшений на одиницю. Одночасно старе значення з вершини стека копіюється в стек рівня 2.

Однією з основних особливостей портів вводу/виводу є те, що програма може зчитувати і записувати дані в них аналогічно регістрам загального призначення. При читанні завжди прочитується дійсний стан виводів, незалежно від того, запрограмовані окремі розряди як входи або як виходи. Після скидання всі розряди програмуються як входи. Виходи портів є заціпки, і їх стан не міняється до наступного запису в порт. Установка режиму кожного розряду у всіх портах проводиться за допомогою установки відповідних розрядів в регістрах управління режимами портів TRIS (TRISA, TRISB або TRISC).

Модуль таймера (TMR0) в даних мікроконтролерах має наступні особливості:

- 8-розрядний таймер/лічильник, доступний по читанню і запису;
- 8-розрядний запрограмований попередній дільник, який може бути програмно підключений або до таймера, або до таймера WDT;

- внутрішнє або зовнішнє тактування, при цьому може бути вибраний фронт тактуючого імпульсу.

Таймер має два режими роботи: режим таймера і режим лічильника. У режимі таймера TMR0 збільшується в кожному командному циклі (якщо немає попереднього дільника). У режимі лічильника TMR0 збільшується по кожному перепаду 1/0 або 0/1 на виведенні TOCK1. Перепад, що збільшує значення TMR0, вибирається установкою відповідного біта в службовому регістрі OPTION, який програмно доступний по запису. У цьому ж регістрі проводиться установка режиму роботи попереднього дільника (TMR0/WDT) і значення коефіцієнта ділення.

Сімейство мікроконтролерів PIC16C5X має набір спеціальних функцій, призначених для розширення можливостей системи, мінімізації вартості, виключення навісних компонентів, забезпечення мінімального енергоспоживання і захисту коду програми від прочитування. До складу даних функцій входять:

- вибір типу генератора;
- таймер скидання (DRT);
- сторожовий таймер (WDT);
- режим зниженого енергоспоживання (SLEEP);
- захист коду програми від прочитування;
- біти ідентифікації.

Мікроконтролери сімейства PIC16C5X, окрім PIC16C52, мають вбудований сторожовий таймер WDT, який може бути вимкнений тільки через біт конфігурації, що задається при програмуванні. Для підвищення надійності він працює від власного RC- генератора. Таймер скидання ВКТ призначений для підтримки контролера в скинутому стані протягом 18 мс після включення живлення для стабілізації роботи генератора. Наявність цих таймерів дозволяє в багатьох застосуваннях відмовитися від схеми зовнішнього скидання.

Режим зниженого енергоспоживання SLEEP призначений для

забезпечення дуже малого струму споживання в режимі очікування (менше 1мкА при вимкненому сторожовому таймері). Вихід з режиму SLEEP можливий по зовнішньому сигналу скидання або по закінченню витримки сторожового таймера. Можливість вибору типу генератора дозволяє ефективно використовувати мікроконтролери сімейства в різних додатках. Використання RC-генератора дозволяє зменшити вартість системи, а LP-генератор на низькочастотному кварцевому резонаторі скорочує енергоспоживання. Захист коду програми і установка коду ідентифікації проводяться шляхом установки відповідних розрядів в слові конфігурації при програмуванні мікроконтролерів.

Для розробки і відладки програмних модулів вільно розповсюджується асемблер MPASM, емулятор MPSIM, інтегрована система відладки для Windows MPLAB, а також велике число добре документованих прикладів застосування мікроконтролерів PIC в різних областях з вихідними текстами.

Для апаратної підтримки режиму програмування мікросхем є різні типи програматорів, що підключаються до комп'ютерів типу IBM PC. Існують типи програматорів, які можна підключати безпосередньо до робочої плати контролера, що значно прискорює відладку. Програмування мікроконтролерів проводиться через послідовний канал, який використовує два розряди порту вводу-виводу. Режим програмування задається шляхом установки на виводі скидання мікроконтролера напруги +12 В.

МК сімейства PIC16CXX в порівнянні з іншими 8-розрядними мікроконтролерами такого ж класу дозволяють зменшити програму 2:1, збільшити швидкодію 4:1 і ідеально підходять для дешевих додатків, що вимагають аналоговий інтерфейс.

PIC16X7XX – сімейство дешевих, високоефективних, 8-розрядних КМОП мікроконтролерів з вбудованим аналого-цифровим (analog-to-digital (A/D)) перетворювачем. Серед мікроконтролерів PIC16CXX дане сімейство займає середнє положення.

Для того, щоб зменшити кількість зовнішніх компонентів і таким чином понизити вартість, підвищити надійність системи і зменшити споживання,

сімейство мікроконтролерів PIC16X7XX володіє додатковими можливостями.

У МК PIC16X7XX є чотири режими генератора:

- RC-генератор на одному контакті забезпечує дешеве рішення;
- LP-генератор забезпечує мінімальне споживання;
- XT-генератор – стандартне рішення;
- HS-генератор – для високочастотних додатків.

Режим остановки дозволяє різко зменшити споживання. Пробудження з режиму остановки може здійснюватися за допомогою зовнішніх і внутрішніх переривань і скидань.

Високонадійний сторожовий таймер з своїм RC-генератором забезпечує захист від зациклення програми.

Варіант мікросхем CERDIR з ультрафіолетовим стиранням ідеально підходить для розробки і відладки програми, тоді як одноразово програмовані варіанти рентабельні для випуску продукції в будь-якому об'ємі. Мікросхеми з FLASH-пам'яттю програм дозволяють здійснювати розробку і відладку програм на готовому пристрої і не вимагають заміни мікроконтролера після закінчення відладки. Сімейство мікроконтролерів PIC16X7XX пристосоване для застосування у віддалених пристроях захисту і датчиках, для приладів управління і автомобілів.

Технологія програмованого ПЗП робить настройку прикладних програм швидкою і надзвичайно зручною. Малогабаритні корпуси мікросхем роблять це сімейство мікроконтролерів досконалим для всіх додатків без обмежень.

Низька ціна, мала споживана потужність, висока ефективність, зручність при використанні і гнучкість I/O роблять PIC16X7XX універсальним навіть в областях, де використання мікроконтролерів раніше не розглядалося (наприклад, функції таймера, послідовний зв'язок, збір і порівняння даних, функції ШІМ і додатки із співпроцесором).

Сімейство мікроконтролерів PIC16X7XX є версією, розширеної архітектури МК PIC16C5X. Всі мікроконтролери сімейства PIC16CXX, що випускаються в однакових корпусах, сумісні по контактах.

Програма для PIC16C5X може бути легко перенесена в пристрої сімейства PIC16X7XX. Щоб перетворити програму, написану для PIC16C5X, в програму для PIC16X7XX, необхідно виконати наступні кроки:

1) видалити всі операції вибору сторінок пам'яті програми (операції над бітами RA2, RA1, PAO) для команд CALL і GOTO;

2) повторно проглянути всі обчислені операції для переходів (запис в PC, складання з PC і т.ін.), щоб переконатися, що біти сторінки встановлені вірно згідно новій структурі;

3) виключити всі перемикання сторінок пам'яті даних. Перевизначити змінні в даних, щоб перерозподілити їх;

4) перевірити всі записи в регістри STATUS, OPTION і SFR, оскільки вони змінилися;

5) замінити вектор скидання, оскільки він став 0000h.

Варіанти пристроїв PIC16X7XX

Мікроконтролери сімейства PIC16X7XX випускають для різних частотних діапазонів і в різних варіантах корпусів. Залежно від додатку і промислових вимог, користуючись табл. 5.12, можна вибрати необхідний варіант пристрою.

Для сімейства PIC16X7XX існує декілька типів позначення пристроїв в номері:

1. С – як, наприклад, PIC16C74. Ці пристрої мають програмований ПЗП пам'яті програм і функціонують в стандартному діапазоні напруг живлення.

2. LC – як, наприклад, PIC16LC74. Ці пристрої мають програмований ПЗП пам'яті програм і функціонують в розширеному діапазоні напруг живлення.

3. CR – як, наприклад, PIC16CR72. Ці пристрої мають масочне ПЗП пам'яті програм і функціонують в стандартному діапазоні напруг живлення.

4. F – як, наприклад, PIC16F73. Ці пристрої мають FLASH-пам'ять програм, що допускає до 100 циклів стирання і запису.

Таблиця 5.12 – Мікроконтролери сімейства PIC16X7XX.

Прилад	Частота МГц	Пам'ять програм			RAM, байт	Контакти I/O	Периферія						Особливості				Тип корпусу	Примітка
		PROM	ROM	FLASH			ADC	Vref	TMR	PWM	PCP	PORT	ICSP	BOR	PLVD	WDT		
PIC16C71	20	1Kx14	-	-	36	13	4(8біт)	-	1-8біт	-	-	-	+	-	-	+	18JW, 18P, 18SO	Струм по будь-якому I/O контакті
PIC16C710	20	512x14	-	-	36	13	4(8біт)	-	1-8біт	-	-	-	+	+	-	+	18JW, 18P, 18SO, 20SS	Струм по будь-якому I/O контакті 25 ма
PIC16C711	20	1Kx14	-	-	68	13	4(8біт)	-	1-8біт	-	-	-	+	+	-	+	18JW, 18P, 18SO, 20SS	Струм по будь-якому I/O контакті 25 ма
PIC16C712	20	1Kx14	-	-	128	13	4(8біт)	-	2-8біт, 1-16біт	1	1	-	+	+	-	+	18JW, 18P, 18SO, 20SS	Струм по будь-якому I/O контакті 25 ма
PIC16C715	20	2Kx14	-	-	128	13	4(8біт)	-	1-8біт	-	-	-	+	+	-	+	18JW, 18P, 18SO, 20SS	Струм по будь-якому I/O контакті 25 ма
PIC16C716	20	2Kx14	-	-	128	13	4(8біт)	-	2-8біт, 1-16біт	1	1	-	+	+	-	+	18JW, 18P, 18SO, 20SS	Струм по будь-якому I/O контакті 25 ма
PIC16C717	20	2Kx14	-	-	256	16	6 (10біт)	1	2-8біт, 1-16біт	1	ECCP	SPI/MI ² C	+	PBOR	+	+	18JW, 18P, 18SO, 20SS	Внутрішній генератор 4МГц
PIC16C72	20	2Kx14	-	-	128	22	5(8біт)	-	2-8біт, 1-16біт	1	1	SPI / I ² C	+	+	-	+	28JW, 28SP, 28SO, 28SS, 28ML	Струм по будь-якому I/O контакті 25 ма
PIC16CR72	20	-	2Kx14	-	128	22	5(8біт)	-	2-8біт, 1-16біт	1	1	SPI / I ² C	-	+	-	+	28SP, 28SO, 28SS	Струм по будь-якому I/O контакті 25 ма
PIC16C73	20	4Kx14	-	-	192	22	5(8біт)	-	2-8біт, 1-16біт	2	2	SPI / I ² C, USART	+	+	-	+	28JW, 28SP, 28SO, 28SS, 28ML	Струм по будь-якому I/O контакті 25 ма
PIC16C74	20	4Kx14	-	-	192	33	8(8біт)	-	2-8біт, 1-16біт	2	2	SPI / I ² C, USART, PSP	+	+	-	+	40JW, 40P, 44L, 44PQ, 44PT	Струм по будь-якому I/O контакті 25 ма
PIC16C745	24	8Kx14	-	-	256	22	5(8біт)	-	2-8біт, 1-16біт	2	2	USART, USB	+	+	-	+	28JW, 28SP, 28SO	Модуль USB, PLL на 4
PC16C765	24	8Kx14	-	-	256	33	8(8біт)	-	2-8біт, 1-16біт	2	2	USART, USB, PSP	+	+	-	+	40JW, 40P, 44L, 44PT	Модуль USB, PLL на 4
PIC16C76	20	8Kx14	-	-	368	22	5(8біт)	-	2-8біт, 1-16біт	2	2	SPI / I ² C, USART	+	+	-	+	28JW, 28SP, 28SO	Струм по будь-якому I/O контакті 25 ма

Продовження табл. 5.12

Прилад	Частота МГц	Пам'ять програм			RAM байт	Контакти I/O	Периферія						Особливості				Тип корпусу	Примітка
		PROM	ROM	FLASH			ADC	Vref	TMR	FWM	PCP	PORT	ICSP	BOR	PLVD	WDT		
PIC16C77	20	8Kx14	-	-	368	33	8(8біт)	-	2-8біт, 1-16біт	2	2	SPI/I ² C, USART,	+	+	-	+	40JW,40P,44L, 44PQ,44PT	Струм по будь-якому I/O контакті
PIC16C770	20	4Kx14	-	-	256	16	6 (12біт)	1	2-8біт, 1-16біт	1	ECCP	SPI/M ² C	+	PBOR	+	+	20JW, 20P, 20SO, 20SS	Внутрішній генератор 4МГц
PIC16C771	20	4Kx14	-	-	256	16	6 (12біт)	1	2-8біт, 1-16біт	1	ECCP	SPI/M ² C	+	PBOR	+	+	20JW 20P, 20SO, 20SS	Внутрішній генератор 4МГц
PIC16C773	20	4Kx14	-	-	256	22	6 (12біт)	1	2-8біт, 1-16 біт	2	2	SPI/M ² C, AUSART	+	PBOR	+	+	28JW, 28SP, 28SO, 28SS	Струм по будь-якому I/O контакті
PIC16C774	20	4Kx14	-	-	256	33	10 (12біт)	1	2-8 біт, 1-16біт	2	2	SPI/M ² C, AUSART,	+	PBOR	+	+	40JW, 40P,44L, 44PQ,44PT	Струм по будь-якому I/O контакті
PIC16C781	20	1Kx14	-	-	128	16	8(8біт)	1	2-8біт, 1-16біт	-	-	-	+	PBOR	+	+	20JW,20P,20SO, 20SS	2 компаратори, підсилювач, ЦАП
PIC16C782	20	2Kx14	-	-	128	16	8(8біт)	1	2-8біт, 1-16біт	-	-	-	+	PBOR	+	+	20JW,20P,20SO, 20SS	2 компаратори, підсилювач, ЦАП
PIC16F72	20	-	-	2Kx14	128	22	5(8біт)	-	2-8біт, 1-16біт	1	1	SPI / I ² C	+	+	-	+	28SP,28SO,28SS, 28ML	Струм по будь-якому I/O контакті
PIC16F73	20	-	-	4Kx14	192	22	5(8біт)	-	2-8біт, 1-16біт	2	2	SPI/I ² C, USART	+	+	-	+	28SP, 28SO, 28SS, 28ML	Читання пам'яті програм
PIC16F74	20	-	-	4Kx14	192	33	8(8біт)	-	2-8біт, 1-16біт	2	2	SPI/I ² C, USART,	+	+	-	+	40P, 44L, 44PT	Читання пам'яті програм
PIC16F76	20	-	-	8Kx14	368	22	5(8біт)	-	2-8біт, 1-16біт	2	2	SPI / I ² C, USART	+	+	-	+	28SP, 28SO, 28SS, 28ML	Читання пам'яті програм
PIC16F77	20	-	-	8Kx14	368	33	8(8біт)	-	2-8біт, 1-16біт	2	2	SPI/I ² C, USART,	+	+	-	+	40P,44L,44PT	Читання пам'яті програм

Умовні позначки:

PROM—програмувальна пам'ять програм
 ROM—масочне ПЗУ пам'яті програм
 RAM—пам'ять даних
 I/O — ввід-вивід
 ADC—аналого-цифровий перетворювач
 Vref—джерело опорної напруги
 TMR—таймери
 PWM—широкоімпульсний модулятор
 AUSART—адресований USART

CCP—модуль порівняння, накопичення й ШИМ
 ECCP—модернізований PCP
 PORT—порти вводу-виводу
 SPI — синхронний послідовний інтерфейс
 I²C -інтерфейс інтегральних схем
 M²C -ведучий I²C
 USART—універсальний синхронно-асинхронний приємопередавач

PSP—паралельний ведений порт
 USB—універсальна послідовна шина
 BOR—скидання при зниженні живлення
 PBOR—програмувальне скидання при зниженні живлення
 PLVD—програмувальний контроль напруги живлення.
 WDT—сторожовий таймер
 ICSP—внутрішнє послідовне програмування
 PLL—схема множення частоти генератора

Пристрої з ультрафіолетовим стиранням. Варіант пристрою з ультрафіолетовим стиранням, що випускається в корпусах CERDIP, оптимальний для розробки макету і відладки програмного забезпечення. Записаний в ПЗП варіант програми може бути знищений, і пристрій перепрограмований для будь-якого з режимів генератора.

Програматори PICSTART[®] Plus і PRO MATI[®] II здійснюють програмування всіх мікрокристалів PIC16X7XX.

Одноразово програмовані пристрої випускаються в пластмасових корпусах і дозволяють користувачу програмувати їх один раз. На додаток до пам'яті програми необхідно також програмувати слово конфігурації.

Пристрої з FLASH пам'яттю програм випускаються за ціною одноразово програмованих пристроїв. Їх доступність особливо корисна, коли необхідна гнучкість при частих модифікаціях програми і малих об'ємах виробів.

Пристрої з масковим ПЗП програмуються заводом-виготовником в процесі виробництва. Пристрої з масковим ПЗП виготовляються для користувачів, які випускають велику кількість продукції без додаткових операцій програмування, оскільки програма відпрацьована і не вимагає змін. Ці пристрої ідентичні одноразово програмованим, але вся пам'ять програм і слово конфігурації вже запрограмовані виготовником.

Структурна схема мікроконтролерів PIC16C710/71/711/715 представлена на рис. 5.13.

Пристрої PIC16C710/71 мають 36 байтів пам'яті даних (RAM), PIC16C711 – 68 байтів і PIC16C715 – 128 байтів. У кожного з цих пристроїв по 13 контактів I/O (PORTA і PORTB). Пристрої PIC16C712/716 мають: 13 контактів I/O (PORTA і PORTB); два 8 розрядних таймери (TMR0 і TMR2) і один 16-розрядний (TMR1); модуль порівняння/накопичення ШІМ (CCPI); 8-розрядний АЦП з 4-ма мультиплексованими вхідними каналами.

Крім того, до складу мікроконтролерів входять наступні периферійні пристрої:

- 8-розрядний лічильник/таймер (TMR0)
- 8-розрядний А/D перетворювач з 4 мультиплексованими вхідними каналами.

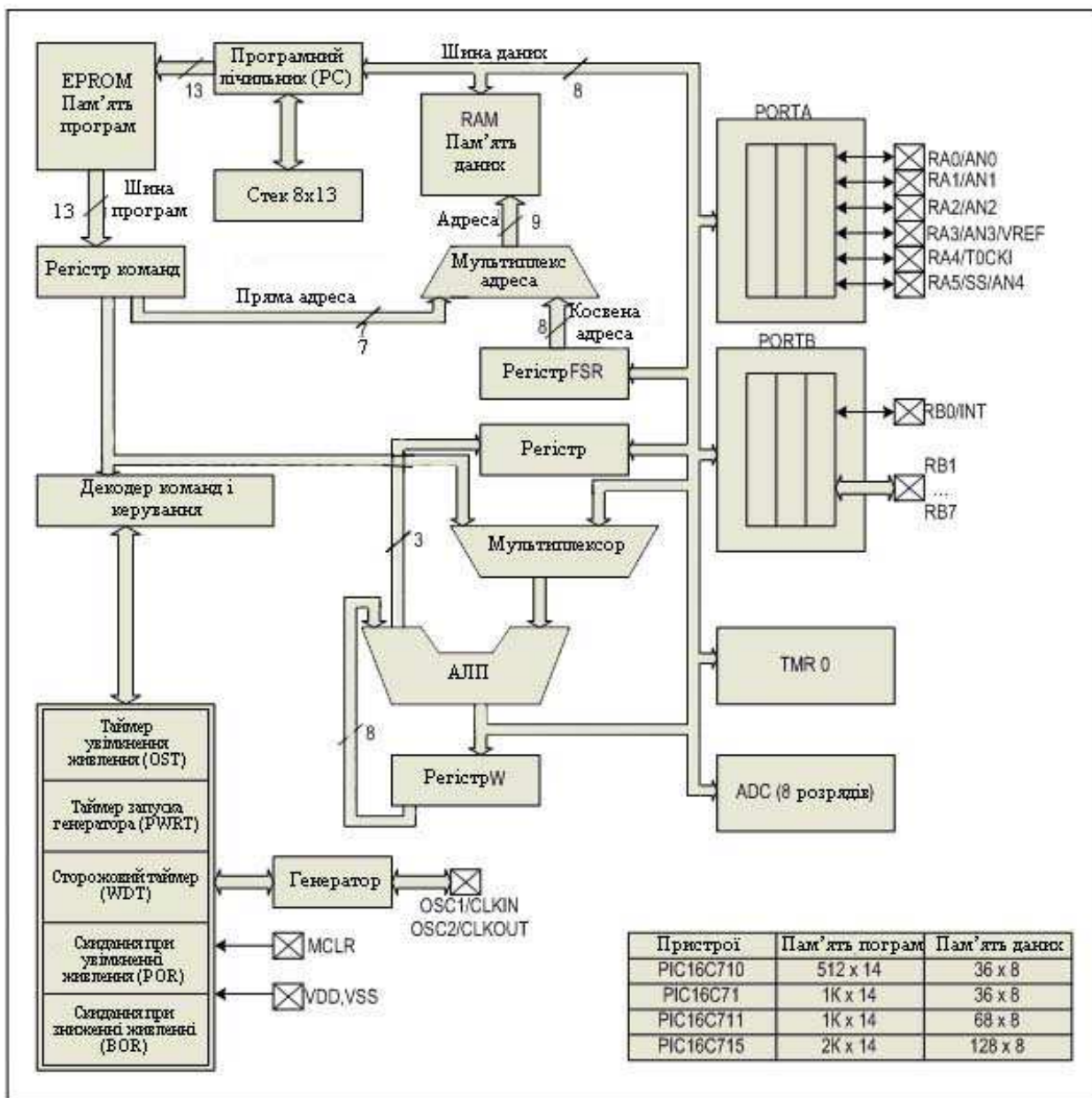


Рисунок 5.13 – Структурна схема PIC16C710/71/711/715

Пристрої PIC16C717/770/771 мають 256 байтів пам'яті даних, 16 контактів I/O (PORTA і PORTB). Вони містять наступні периферійні пристрої: два 8-розрядних (TMR0 і TMR2) і один 16-розрядний (TMR1) таймери; вдосконалений модуль порівняння/накопичення/ШИМ (ECCP), що дозволяє управляти силовими ключами, мостовою схемою (4 канали), напівмостовою схемою (2 канали) або одноканальною схемою з 10-розрядним дозволом на частоті 20 кГц;

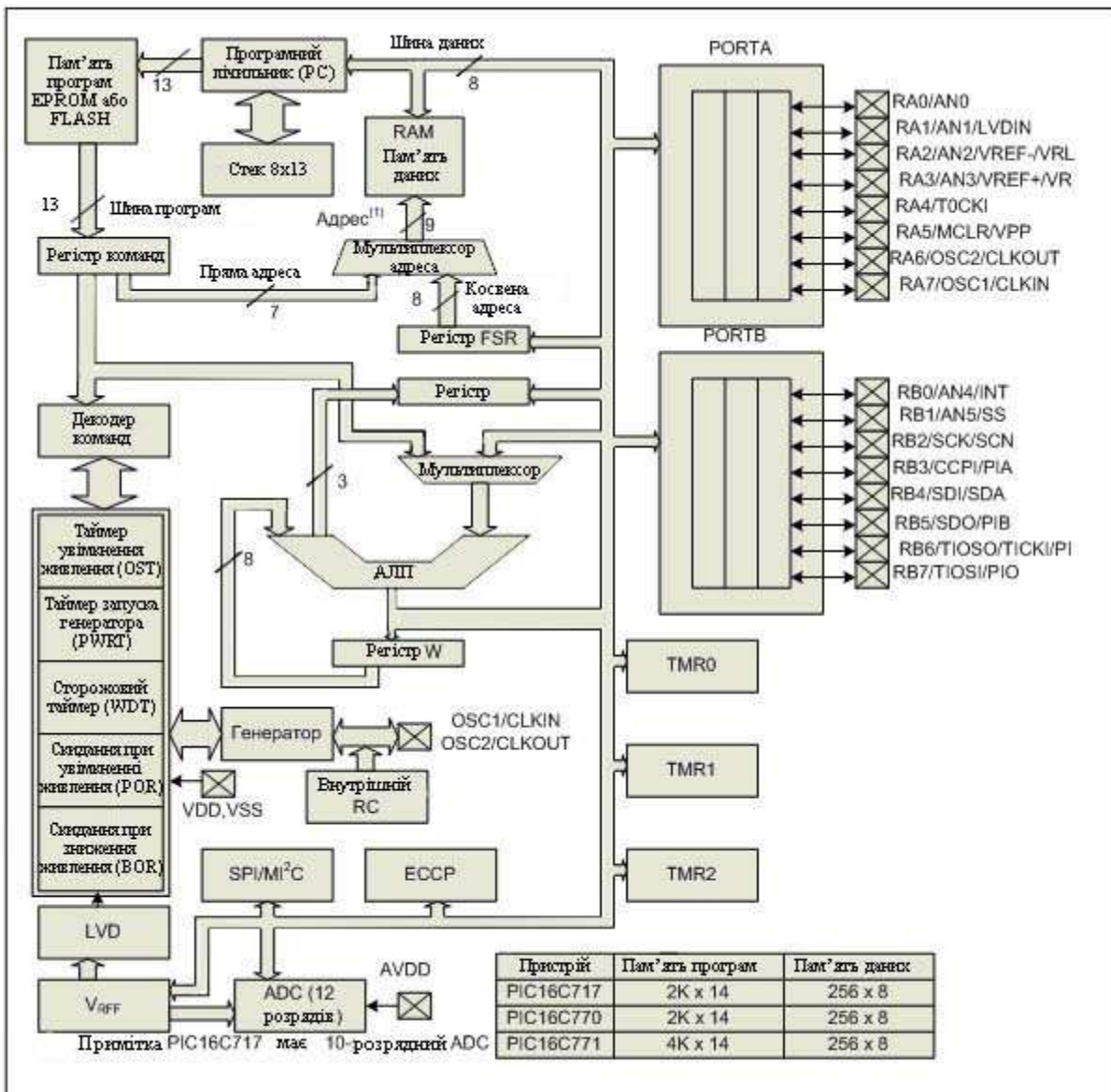


Рисунок 5.14 – Структурна схема PIC16C711/770/771

- синхронний, послідовний порт, який може функціонувати як трьохпровідний послідовний периферійний інтерфейс (SPI) або двопровідна шина I²C з апаратною реалізацією режиму ведучої шини (MI²C);
- швидкодіючий A/D-перетворювач з 6 мультиплексованими вхідними каналами і внутрішнім керованим джерелом опорної напруги, причому PIC16C717 має 10-розрядний ADC, а PIC16C770/771 — 12-розрядний;
- вбудований RC-генератор тактових імпульсів, частота якого може бути задана при програмуванні 4 МГц або 37 кГц;

Система команд

Кожна команда PIC16X7XX є 14-розрядне слово, розділене на код операції, який визначає тип команди, і один або більшу кількість операндів, над якими виконуються операції. Система команд високо ортогональна і згрупована в три базові категорії: байтові операції, операції над розрядами, операції з константами (літералами) і операції управління. У байтових операціях «f» – покажчик регістра, а «d» – покажчик адресата результату. Покажчик регістра визначає, який регістр повинен брати участь в операції. Покажчик результату адресата визначає, куди повинен бути поміщений результат операції. Якщо «d» = 0, результат поміщається в регістр W. Якщо «d»=1, результат поміщається в регістр, визначений в команді. У операціях над розрядами «b» – покажчик розряду, над яким виконується операція, «f» – покажчик регістра, в якому знаходиться операнд і куди поміщається результат.

У операціях з константами і командах управління «k» є восьми або одинадцятирозрядне постійне число або значення літерала (мітка). Всі команди виконуються за один командний цикл. Але якщо результат перевірки умови істинний або лічильник програм змінений в результаті виконання команди, то в цьому випадку команди виконуються за два цикли, другий цикл виконується як операція NOP. Один цикл команди складається з чотирьох періодів генератора.

Всі команди виконуються за один цикл, крім команд розгалуження програми, які виконуються за два цикли, тому що вибрана команда вивантажується із конвеєра, а нова команда вибирається, а потім виконується. Перелік команд та їх структура приведені у додатку 3.

Докладний опис структурних схем, пристроїв які в них входять, особливостей функціонування в різних режимах, системи команд і застосування МК сімейства PIC16X7XX приведено в [23].

5.5. Мікроконтролери фірми Motorola

Мікроконтролери фірми Motorola займають особливе місце серед аналогічної продукції інших фірм через низку обставин.

По-перше, фірма Motorola пропонує *найширшу в світі номенклатуру* мікроконтролерів, що охоплює практично всі області застосування і що включає близько 300 моделей – від простих дешевих до високопродуктивних 32-розрядних мікроконтролерів з RISC-ядром і потужною периферією [24]. Як наслідок, користувач має можливість вибрати для свого додатку оптимальну модель мікроконтролера як по набору вбудованих функцій, мінімізуючи число компонентів в системі, так і по економічних параметрах, відповідних об'єму і особливостям виробництва.

Другою важливою особливістю мікроконтролерів (і решти продукції) фірми Motorola є їх *висока якість і надійність*. Будучи традиційним постачальником військово-промислового і аерокосмічного комплексів, а також автомобільної промисловості США, що пред'являють підвищені вимоги до надійності компонентів, ця фірма виробила і продовжує розвивати спеціальну програму підвищення якості продукції, і заслуги фірми в цій області відмічені національною нагородою США «За вищу якість продукції», а також численними нагородами як кращому постачальнику від таких крупних компаній, як General Motors, Ford, Chrysler, Bosch, і ін. [24].

Ці дві важливі особливості, а також наявність широкого вибору програмних і апаратних засобів підтримки розробки, доступних як від фірми Motorola, так і від безлічі інших фірм, докладна довідкова література і інформація про застосування, зокрема доступна по мережі Інтернет, забезпечують можливість отримання безкоштовних технічних консультацій і програмного забезпечення. Ці та інші переваги використання продукції, що стала фактично промисловим стандартом, забезпечують фірмі Motorola стійке перше місце з продажу мікроконтролерів в світі протягом ряду останніх років.

На жаль, мікроконтролери фірми Motorola мають один важливий недолік – кількість різновидів одноразово програмованих кристалів, що

обмежує їх застосування в дрібносерійному виробництві.

Кожний з мікроконтролерів Motorola, що випускаються фірмою, відноситься до однієї з крупних груп, або сімейств, коротка характеристика яких приводиться нижче.

Сімейство HC05

Сімейство HC05 містить найбільшу кількість модифікацій мікроконтролерів (близько 180), оскільки це сімейство неабиякою мірою формувалося крупними споживачами фірми Motorola, що замовляли розробку мікроконтролерів потрібної конфігурації під свою конкретну продукцію, тому сімейство HC05 іноді називають сімейством «замовлених» мікроконтролерів.

Областями застосування мікроконтролерів сімейства HC05 є найрізноманітніші пристрої зв'язку, автомобільної і побутової електроніки, промислового управління, комп'ютерної периферії.

Всі мікроконтролери цього сімейства мають однакове *8-розрядне процесорне ядро*, засноване на популярній процесорній архітектурі 6800 і відрізняються набором периферійних функцій. Це означає, що застосування будь-якого мікроконтролера цього сімейства відкриває користувачу можливість використовувати придбаний досвід при створенні нових пристроїв як із застосуванням інших мікроконтролерів з обширного сімейства HC05, так і на основі продуктивнішого, але програмно сумісного сімейства HC08.

До складу мікроконтролерів сімейства HC05 входять: ПЗП всіх типів, ОЗП, таймери, АЦП, ШІМ, контролери ЖКІ і інших дисплеїв, послідовні інтерфейси і багато інших пристроїв. Всі представники сімейства HC05 мають версії із зниженим живленням і розширеним температурним діапазоном і випускаються в найрізноманітніших корпусах.

Сімейство HC08

Сімейство HC08 є наступним кроком в розвитку програми замовлених мікроконтролерів фірми Motorola для масових додатків і характеризується підвищеною в 5...10 разів продуктивністю процесорного ядра, сумісного за системою команд з ЦПУ HC05. Сімейство HC08 підтримує ефективні

додаткові команди і методи адресації, а також такі нові функції, як прямий доступ до пам'яті, технологія нечіткої логіки і елементи цифрової обробки сигналів.

При цьому повністю статичне процесорне ядро оптимізоване для роботи із зниженою напругою живлення і дозволяє гнучко управляти споживанням за допомогою вбудованого синтезатора тактової частоти. Сімейство HC08 є першим 8-розрядним сімейством з *визначуваною користувачем архітектурою* на базі набору стандартних модулів, що значно прискорює цикл розробки нового замовленого мікроконтролера.

Набір модулів в даний час включає різні типи ПЗП і ОЗП, таймери, послідовні інтерфейси, АЦП, контролер ЖКІ, контролер ПДП, силові і високовольтні ключі і т.ін. Перші представники цього сімейства з'явилися в 1994р., зараз до складу сімейства входять близько 20 моделей. Нова програма «Замовлений мікроконтролер за 7 днів», введена фірмою Motorola в 1996 р, дозволила радикально скоротити цикл розробки нових мікроконтролерів сімейства HC08, що, безумовно, приведе до його динамічного розвитку.

Сімейство HC11

На відміну від спеціалізованих мікроконтролерів «замовлених» сімейств, сімейство MC68HC11 містить близько 40 універсальних і високопродуктивних мікроконтролерів, орієнтованих як на масові ринки, так і на середнє і дрібне виробництво. *Процесорне ядро мікроконтролерів цього сімейства характеризується підвищеною продуктивністю, відрізняється від HC05 ефективнішою архітектурою, системою команд, наявністю додаткових методів адресації і можливістю адресувати більший об'єм зовнішньої пам'яті.* Мікроконтролери сімейства HC11 містять вбудовану пам'ять різних типів і конфігурацій.

Периферійні функції представлені багатofункціональними таймерами, АЦП (до 12 каналів і 10 розрядів), вбудованим співпроцесором, прискорюючим виконання множення і ділення на порядок, ШІМ і ЦАП; послідовними інтерфейсами, контролером ПДП, синтезатором тактової частоти і іншими функціями. Як і в інших сімействах, є велика

різноманітність корпусів, а також версії із зниженою напругою живлення і розширеним температурним діапазоном.

Сімейство HC05. Загальна характеристика і класифікація

Сімейство HC05 має в своєму складі близько 180 представників, що дає можливість використовувати найбільш широкий вибір варіацій вбудованої пам'яті і периферійних функцій. Це сімейство є основою програми фірми Motorola по створенню мікроконтролерів з характеристиками, визначуваними користувачами, або таких «замовлених» мікроконтролерів. Ця програма направлена на створення мікроконтролерів, що поєднують низьку вартість при серійному виробництві з широкими функціональними можливостями і вимагаючих використання мінімального числа додаткових елементів.

Позначення мікроконтролерів містить символ, наступний в назві мікроконтролера безпосередньо за MC68HC05 і що відносить його до однієї з підгруп, в межах сімейства (або до серії), які відрізняються один від одного функціональними особливостями. З іншого боку, більшість мікроконтролерів сімейства HC05 створювалися під певні додатки, тому класифікацію зручно провести з урахуванням цих двох чинників одночасно.

Мікроконтролери загального призначення

Серія C характерна широкою різноманітністю вбудованої пам'яті і ліній паралельного вводу-виводу. Асинхронний, послідовний інтерфейс (SCI) дозволяє організувати обмін даними із зовнішніми пристроями з швидкістю до 131 кГц. Високошвидкісний синхронний послідовний інтерфейс (SPI) зручний для управління дисплеями і зовнішніми периферійними пристроями по 4-дротяній лінії. Всі мікроконтролери серії C мають в своєму складі 16-бітовий програмований таймер з функціями «схожої фіксації» і «вихідного порівняння» для одночасного вимірювання часових параметрів зовнішніх імпульсів і генерації імпульсного сигналу. Найбільш популярним (і, як наслідок, широко доступним) представником серії C є мікроконтролер MC68HC705C8A з одноразово програмованою вбудованою пам'яттю, великою кількістю ліній вводу-виводу, наявністю версії з подвоєною тактовою частотою і захистом коду від читання. Мікроконтролер

MC68HC05C0 є єдиним представником сімейства HC05, що не має вбудованого ПЗП і що адресує зовнішню пам'ять до 64 К байт.

Серія J включає недорогі 20-вивідні мікроконтролери, що містять ПЗП, багатофункціональний таймер і функцію переривання реального часу. Найбільш яскравою моделлю цієї серії є мікроконтролер MC68HC705J1A, наявність в якому програмованої пам'яті із захистом від читання, виходів з високою здатністю навантаження, переривань від клавіатури, а також швидкісної версії і вельми недорогого комплекту налагоджувальних засобів, роблять цей новий мікроконтролер все більш популярним.

Серія K містить найдешевші з тих, що випускаються фірмою Motorola 16-вивідні мікроконтролери, що включають пам'ять, таймер переривання реального часу, лінії з підвищеною здатністю навантаження і програмовану користувачем «ідентифікаційну» область пам'яті (8 байт) навіть в масовому варіанті.

Мікроконтролер MC68HC805K3 з вбудованою пам'яттю EEPROM (Flash) призначений для макетування і невеликих виробничих серій.

Серія P характерна наявністю вбудованого АЦП, різноманітністю варіантів вбудованої пам'яті (включаючи EEPROM), наявністю простого послідовного порту, а також невеликим 28-вивідним корпусом і низькою ціною. Найбільш популярним представником цієї серії є мікроконтролер MC68HC705P9.

Мікроконтролери для телекомунікацій

Серія F була спеціально створена для побудови абонентських телефонних апаратів різних груп складності і терміналів. Головною особливістю мікроконтролерів цієї серії є наявність в них генератора DTMF для тонального набору номера і цифрової сигналізації. Ряд мікроконтролерів цієї серії додатково містять контролер ЖК-дисплея, а також великий об'єм ПЗП для реалізації складних алгоритмів сучасних ТА і незалежну пам'ять для зберігання номерів. Всі ці функції, разом з низьким споживанням, що дозволяє житися від телефонної лінії або батарей, роблять мікроконтролери серії P привабливими для створення як масових телефонних апаратів, так і складного абонентського устаткування.

Серія L. Головними особливостями цієї серії є: наявність вбудованих контролерів алфавітно-цифрових і графічних ЖК-дисплеїв (від 32 до 40000 сегментів), наявність тонального генератора, годинника реального часу і низьке споживання енергії. Вони також широко використовуються при створенні різноманітного, особливо портативного, зв'язного устаткування: бездротових телефонів, пристроїв персонального виклику (пейджерів), радіостанцій, цифрових блокнотів і т.ін.

Серія E включає мікроконтролери, що містять синтезатор тактової частоти з ФАПЧ для гнучкого управління споживанням, незалежну пам'ять, АЦП і інтерфейс І²С, і використовується в засобах зв'язку як мікроконтролери загального призначення.

Серія C. Мікроконтролери загального призначення, а також активно використовуються в комунікаційних пристроях, таких, як абонентські модулі АТС, системи цифрового ущільнення абонентських ліній і т.д.

Мікроконтролери для побутової електроніки

Серія L, що включає широкий вибір мікроконтролерів з контролерами ЖКІ, часто використовується в різноманітних побутових пристроях, що вимагають виводу на ЖКІ-дисплей.

Серія M містить вбудований контролер вакуум-флюоресцентного індикатора. Вбудований формувач забезпечує можливість управління по 24 лініях при напрузі 40 В. До складу мікросхем серії також входять 8-бітовий таймер, 6-канальний 8-бітовий АЦП, асинхронний, послідовний порт і великий об'єм вбудованого ПЗП.

Серія MS характеризується наявністю в 28-вивідному корпусі швидкісних каналів ШІМ і 6-канального АЦП та призначена для управління електродвигунами в «білій техніці»: холодильниках, пральних машинах, кухонних комбайнах і т.д.

Серії T спеціально призначена для застосування у відео- і телевізійній апаратурі, і містить драйвер кольорового екранного дисплея, що дозволяє відображати на екрані ЕПТ символи і текстову інформацію. Вбудований АЦП може використовуватися для управління настройкою на певний канал телебачення, канали ШІМ використовуються для управління гучністю звуку,

яскравістю зображення і т.д. Інтерфейс I2C дозволяє управляти іншими підсистемами ТБ приймача, наприклад відеопроцесором.

Серія СС є продовженням серії Т і має розширений драйвер екранного дисплея з можливістю секціонування даних.

Серії мікроконтролерів загального призначення, особливо найбільш дешеві серії з малою кількістю виводів (К, J і RC), використовують для вирішення простих завдань управління в різноманітних побутових пристроях (наприклад, пульти дистанційного керування).

Мікроконтролери для автомобільної електроніки

Серія Х включає мікроконтролери з вбудованим контролером локальної мережі (CAN-Controller Area Network) Європейського керуючого стандарту для побудови мультиплексної шини автомобіля. Мікроконтролери цієї серії використовують для локального управління збору даних в різних підсистемах автомобіля (приладова панель, склопідіймачі, підвіска, ABS, і т. ін.). Ряд моделей серії Х містять розширений таймер, незалежну пам'ять, АЦП, ШІМ і розширений послідовний порт.

Серія V призначена для виконання функцій, аналогічних серії Х, але орієнтована на інші стандарти побудови мультиплексної шини – MDLC (Message Data Link Control) або JL 850. Характерною особливістю мікроконтролерів цієї серії є поєднання декількох технологій: КМОП, високовольтної і силової, що дозволило створити «систему на кристалі», що містить, крім стандартних блоків мікроконтролера, високовольтний регулятор напруги, трансівери мультиплексної шини, EEPROM, АЦП, ШІМ і інші пристрої.

Серії К, J, P загального призначення часто використовують для реалізації окремих функцій автомобіля, наприклад охоронної сигналізації (електронний ключ, центральний замок).

Мікроконтролери для промислового управління

Серія В поєднує великий об'єм вбудованого програмованого ПЗП, EEPROM, АЦП і ШІМ, а також таймер і розширений послідовний порт. Така конфігурація дозволяє використовувати мікросхеми серії в найрізноманітніших індустриальних додатках.

Серія МС характеризується наявністю в 28-вивідному корпусі швидкісних каналів ШІМ і 6-канального АЦП і призначена для управління електродвигунами.

Серія Х включає мікроконтролери з контролером локальної мережі, що управляє, випускається в різних модифікаціях (ПЗП від 4К до 32К, корпус від 28 до 64 виводів, АЦП, ШІМ) і застосовується для побудови локальних вузлів збору/керування даних в розподілених системах управління технологічним устаткуванням. Мікроконтролери загального призначення серій К, J, Р також використовують при створенні розподілених систем управління/збору інформації, наприклад систем пожежної сигналізації, систем охорони доступу і т.ін. «Ідентифікаційна» програмована область пам'яті дозволяє будувати «адресні» датчики.

Мікроконтролери *серії L* з контролером ЖКІ також можуть використовуватися в таких системах як вузли збору/обробки даних з індикацією. Низьке споживання мікроконтролерів цих серій дозволяє створювати системи з одночасною передачею живлення і даних по одній лінії.

ЦПУ і система команд сімейств HC05

ЦПУ сімейства HC05, що відрізняється простотою і зручністю програмування, має стандартну внутрішню тактову частоту 2 МГц, для деяких мікроконтролерів існують версії з тактовою частотою 4 МГц (цикл команди 250 нс). Програмна модель ЦПУ містить 5 регістрів (регістри ЦПУ не є частиною карти пам'яті).

Акумулятор (АСС) – 8-бітовий регістр загального призначення, в якому зберігаються операнди, результати арифметичних операцій, а також дані, з якими проводяться які-небудь операції. Акумулятор також використовується і для логічних операцій.

Індексний регістр (Х) – використовується або при індексному режимі адресації, або як допоміжний акумулятор. Цей регістр може бути завантажений як безпосередньо, так і з пам'яті; може бути збережений в елементі пам'яті або порівняний з її вмістом.

Лічильник команд (РС) містить адресу команди, яка слідує за

виконуваною, або адресу операнда, що входить в код програми. Розрядність РС залежить від об'єму вбудованого ПЗП.

Показчик стека (SP) містить адресу наступної (вільної) комірки стека. Глибина стека мікроконтролерів сімейства HC05 складає 64 байти. Виклик підпрограми використовує 2 комірки стека, переривання – 5 комірок.

Регістр ознак (CC) містить 5 прапорів, що встановлюються залежно від результату виконання арифметичних і інших команд. Цими прапорами є: прапор напівперенесення (H), прапор негативного результату (S), прапор нульового результату (Z), біт маски переривань (I) і прапор перенесення (C).

Система команд мікроконтролерів включає 65 команд, що діляться на наступні групи:

- команди переміщення даних (LDA, STA, CLR, LDX, STX...);
- команди передачі управління (JMP, JSR, RTI, переходи за умовами і бітами...);
- арифметичні команди (ADD, SUB, MUL...);
- логічні команди (AND, OR, COM, NEG...);
- команди роботи з бітами (BSET, BCLR, зміщення...);
- спеціальні команди (WAIT, STOP, SWI...).

Команди MC68HC05 виконуються, як правило, за 2...5 циклів внутрішньої тактової частоти, що складає 1...2,5 мкс при стандартній внутрішній тактовій частоті 2 МГц. Мікроконтролери сімейства MC68HC05 використовують вісім режимів адресації: неявна, безпосередня, пряма, розширена, індексна без зсуву, індексна з 8-розрядним зсувом, індексна з 16-розрядним зсувом, відносна.

Вбудована пам'ять мікроконтролерів сімейства HC05 може включати ПЗП (маскове, одноразово програмоване, програмоване з УФ-стиранням, програмоване з електричним стиранням) і ОЗП об'ємом до 768 байтів. У карту пам'яті включені регістри портів паралельного вводу-виводу (причому, як правило, ці адреси однакові для всіх моделей сімейства), а також адреси регістрів, що управляють, і регістрів даних периферійних пристроїв (таймера, послідовного інтерфейсу і т.д.). Область завантажувального ПЗП є масочною і дозволяє здійснювати «самозавантаження» даних у вбудовану пам'ять

(включаючи програмування ППЗП) через паралельні або послідовні порти. Нарешті, вектори переривань, розташовані в ППЗП, визначають адреси переходів по перериваннях від периферійних підсистем (таймера, послідовних інтерфейсів, виведення зовнішнього переривання) і в інших випадках (RESET, програмне переривання).

Вбудовані підсистеми вводу-виводу мікроконтролерів сімейства HC05 зустрічаються в найбільшій кількості мікроконтролерів даного сімейства. Основу блоку таймера складає 16-бітовий лічильник з попереднім дільником, що має можливість формувати переривання по переповнюванню і що синхронізується внутрішньою тактовою частотою, що ділиться на 2. Більшість мікроконтролерів сімейства містять також пов'язані з лічильником підсистеми вхідної фіксації (IC) і вихідного порівняння (OC). Система IC служить для обробки часових параметрів зовнішніх сигналів і дозволяє записувати в регістр IC вміст лічильника при перепаді рівня зовнішнього сигналу з видачею відповідного переривання або установкою прапора. Система OC призначена для генерації імпульсного сигналу з програмованими часовими параметрами і дозволяє видавати в лінію порту «0» або «1» у момент рівності вмісту лічильника і вмісту регістра OC. У найбільш простих моделях сімейства функції IC і OC можуть бути відсутніми і замінюватися перериваннями реального часу (RTI) з програмованим інтервалом між перериваннями. Ще однією важливою системою, пов'язаною з таймером, є система стеження за виконанням програми (COP WatchDog). Ця система формує Reset, якщо періодично з певним проміжком часу не буде проведений запис в спеціальний регістр.

Послідовний інтерфейс зв'язку SCI є повнодуплексний асинхронний приймач-передавач і може бути використаний для зв'язку з терміналом, РС (наприклад, по RS-232) або іншими мікроконтролерами. Вбудований генератор частоти обміну дозволяє ділити внутрішню тактову частоту з отриманням 32 стандартних частот обміну – від 37,56 бод до 125 кбод. SCI, так само підтримує такі функції, як програмована довжина посилки, виходу з режиму очікування приймача по вільній лінії або адресному маркеру, окремий дозвіл приймача і передавача, виявлення помилки кадру і шуму в

лінії (з часовим дозволом 1/16 біта). SCI може формувати 5 видів переривань (або встановлювати 5 прапорів) за наступними умовами: «регістр даних передавача порожній», «передача завершена», «регістр даних приймача заповнений», «приймач переповнений» і «лінія прийому вільна».

Послідовний периферійний інтерфейс SPI використовується для синхронної передачі інформації в послідовному коді на менші відстані, але із значно більшою швидкістю і дозволяє МК взаємодіяти з різними периферійними пристроями, АЦП. SPI підтримує наступні функції: повний дуплекс; режим провідного і веденого; 4 програмованих тактових частоти до 1,05 МГц з програмованою полярністю і фазою; прапор переривання по закінченню передачі; захист від конфліктів на магістралі.

Аналого-цифровий перетворювач проводить перетворення зовнішньої напруги в діапазоні від V_{ss} (нижня опорна напруга, що підключається до загальної шини) до V_{rh} (верхня опорна напруга) в 8-розрядний код від \$00 до \$FF відповідно. АЦП використовує метод послідовних наближень, процес перетворення займає 32 цикли внутрішньої тактової частоти (16 мкс при 2 МГц). При тактовій частоті, менше 1 МГц використовується вбудований RC-генератор частоти для АЦП (1,5 МГц). Вбудований мультиплексор дозволяє проводити перетворення по одному із зовнішніх аналогових входів (до 8), а також вимірювати V_{ss} , V_{rh} , $(V_{rh}+V_{ss})/2$ для проведення контролю і юстирування. Підсистема АЦП містить регістр управління (задає режим роботи і запуск перетворення), регістр статусу (містить прапор закінчення перетворення) і регістр даних (результат перетворення). Характеристики МК представників сімейства HC05 приведені у табл. 5.13, в якій використовуються наступні умовні позначення:

Ch – канал (channel);

EEPROM - електрично стираний програмований ПЗП;

i – вхід;

I/O – ввід-вивід;

IC – вхідна фіксація (Input Capture), підсистема таймера для вимірювання тимчасових параметрів;

MFT – багатофункціональний таймер (Multifunction Timer);

о – вихід;

OC – вихідне порівняння (Output Capture), підсистема таймера для генерації імпульсу сигналу;

RTI – система переривань реального часу (Real Time Interrupt);

SCI – послідовний асинхронний інтерфейс зв'язку (Serial Communication Interface);

Таблиця 5.13 – Характеристики МК сімейства HC05

Назва	ПЗП	EEPROM	Таймер	I/O	Послідовний інтерфейс	АЦП	ШІМ	Корпус	ОЗП	Примітка
68HC705B5	6К	—	16bit 2IC, 20C	94 8i	SCI+	8ch	2ch 8bit	56-B	176	Загальне призначення АЦП+ШІМ+
68HC705B16	15К	256	16bit 2IC, 20C	32 20	SCI+	8ch •	2ch 8bit	52-FN 64-FU	352	Вбуд. накачка захисту запису ЕСПЗУ, версія 4МГц
68HC705BD3	7,75К	—	MFT, PTI	24	1 ² C	—	16ch 8bit	40-P 42-B	256	Процесор горизонтального і вертикального розгорнення
68HC705MC4	3,5К	—	16bit 2IC, 20C MFT, RTI	22	SCI	6ch 8bit	2ch 8bit	28-P 28-DW 28-S	176	Швидкісний ШІМ для керування двигуном
68HC705P6A	4К	—	16bit 1 IC, 1 OC	20	SIOP	4ch 8bit	-	28-P 28-DW	176	8КВІ, 2 виходи по 15 мА

SCI+ - послідовний інтерфейс SCI, який може працювати також як SPI;

SIOP – послідовний порт вводу-виводу (Simple serial input/output port);

Короткий опис деяких представників сімейств HC05

МК загального призначення MC68HC805K3, блок-схема якого представлена на рис. 5.15, призначений для роботи в системах, що вимагають використання малогабаритного МК з мікроспоживанням і низькою вартістю: у охоронних системах, датчиках, побутових пристроях, портативних засобах зв'язку, пультах дистанційного керування і т.ін.

Коротка характеристика:

- ЦПУ HC05, множення 8x8;
- живлення від 1.8 В (запис EEPROM від 3 В), низьке споживання;



Рисунок 5.15 – Блок-схема мікроконтролера MC68HC805K3

- 920+16 байтів EEPROM;
- переривання реального часу;
- переривання від клавіатури по 8 лініях (*);
- здатність навантаження 8 мА по чотирьох лініях (**);
- режими зниженого споживання STOP, WAIT;
- корпуси DIP-16, SOIC-16.

Мікроконтролер загального призначення MC68HC705P6A, блок-схема якого представлена на рис. 5.16, знаходить широке застосування в різноманітних пристроях, що вимагають обробки аналогового сигналу і обмежень до габаритів і вартості вживаного МК, наприклад, в пристроях контролю температури, в електронних вагах, локальних вузлах управління, пристроях сигналізації, побутовій техніці і т.ін.

Коротка характеристика:

- ЦПУ HCS05, множення 8x8, АЦП (4 канали, 8 розрядів);
- синхронний послідовний порт для підключення периферійних пристроїв, 16-розрядний таймер з функціями вхідного захоплення і вихідної фіксації, 4,6 кбайта програмованого ПЗП і 176 байтів ОЗП;
- 21 лінія вводу-виводу, переривання від 8 ліній, 2 виходи із струмом 15 мА;
- режими зниженого споживання STOP, WAIT, HALT, збереження даних в ОЗП, 28-вивідні корпуси DIP, SOIC.

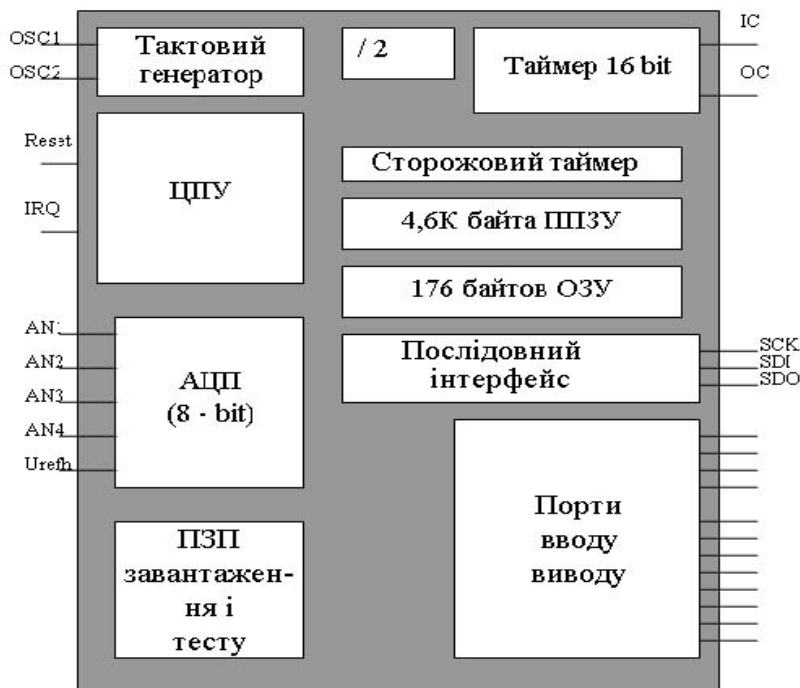


Рисунок 5.16 – Блок-схема мікроконтролера MC68HC705P6A

Мікроконтролер MC68HC705L1 приведений на рис. 5.17. Головними його особливостями є наявність вбудованого контролера ЖКІ на 64 сегменти і 6-канального 8-розрядного АЦП, а також робота від зниженої напруги живлення. Він широко застосовується в різних портативних вимірювальних приладах, засобах зв'язку, CD плеєрах, радіоприймачах і т.ін.

Коротка характеристика:

- контролер ЖКІ до 64 сегментів (організація 3x12, 3x16 4x12, 4x16);

- АЦП (6 каналів, 8 розрядів);
- 16-бітовий таймер з подвійними функціями IC і OC;
- лінія вводу-виводу: 17 двонапр., 10 входів, 2 виходи;
- версії з живленням від 2,7В; корпус 56 –SDIP, 64 – QFP;
- режими зниженого споживання STOP, WAIT.

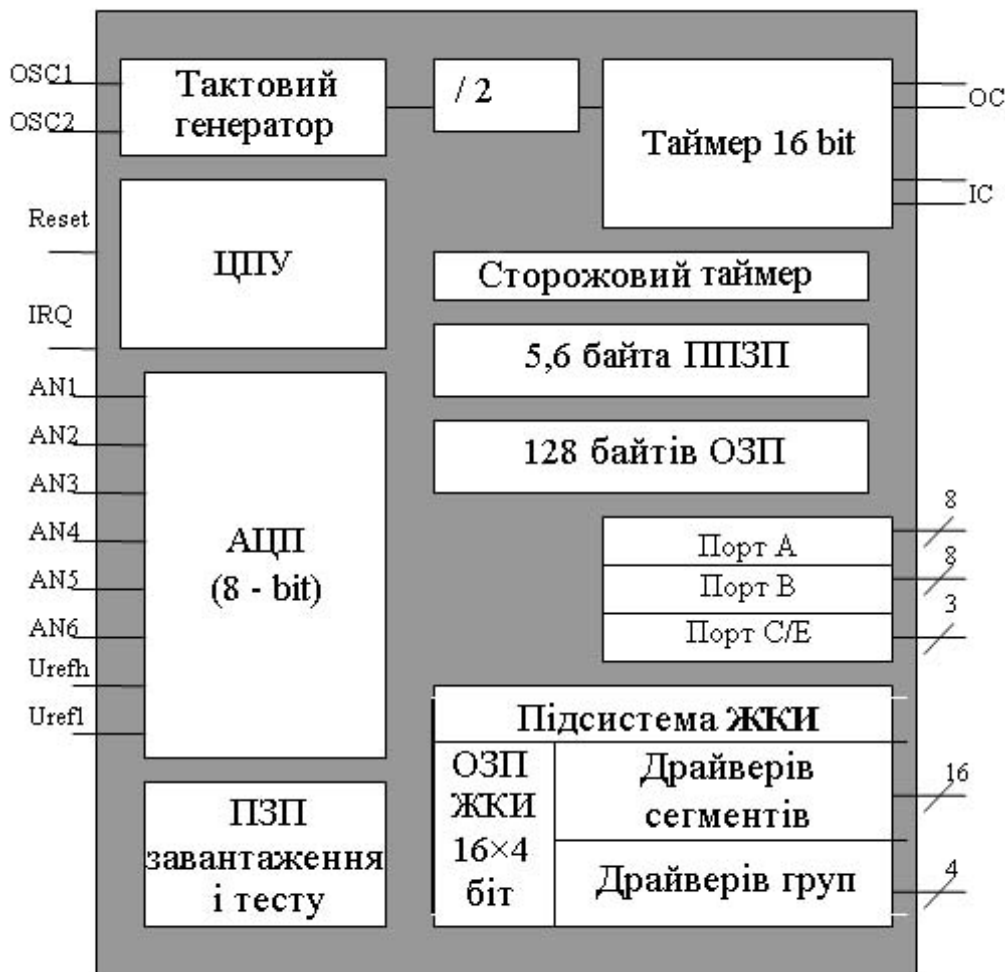


Рисунок 5.17 – Блок-схема мікроконтролера MC68HC705L1

Мікроконтролер MC68HC705MC4, блок-схема якого представлена на рис. 5.18, оптимізований для управління електродвигунами з регульованою швидкістю обертання, які використовуються в побутовій техніці: компресорах холодильників, вентиляторах, пральних машинах, кухонних комбайнах та ін. Цей МК застосовується також для управління ключами в джерелах живлення, а

також в інших пристроях.

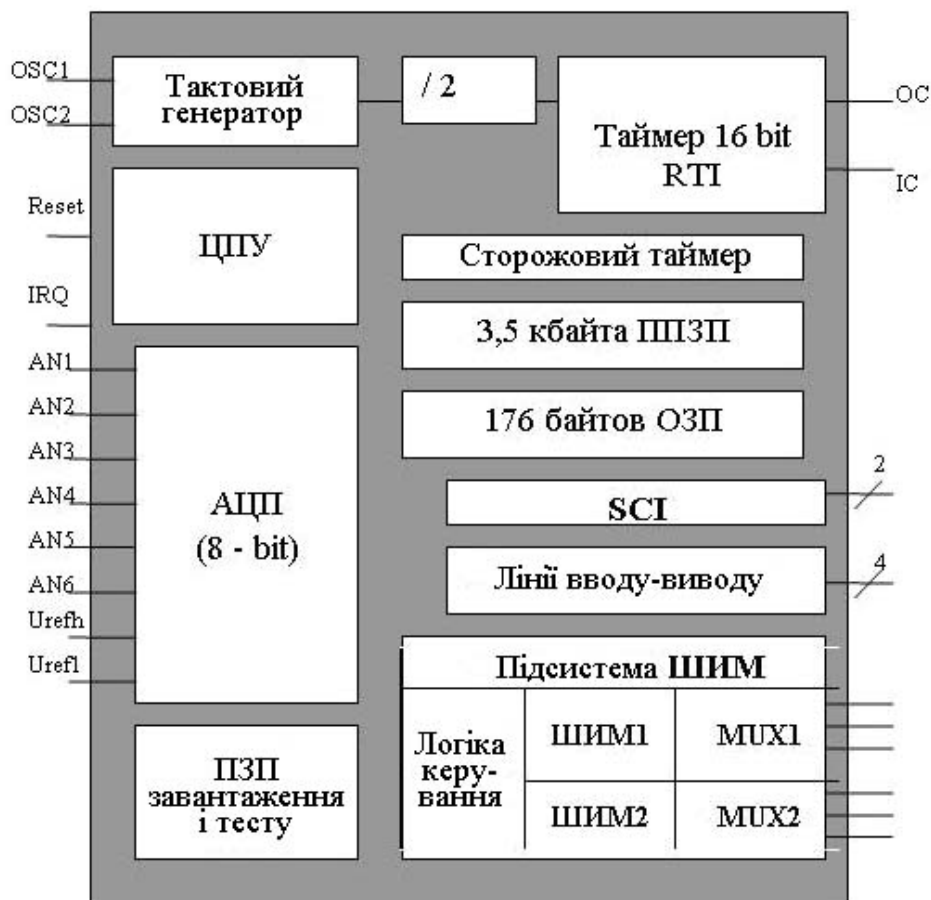


Рисунок 5.18 – Блок схема мікроконтролера MC68HC705MC4

Коротка характеристика:

- два 8-бітові канали ШИМ, незалежні або зв'язані;
- незалежні лічильники каналів;
- частота до 23 кГц;
- заповнення від 0 до 100 %, програмована полярність;
- програмоване перекриття;
- механізм одночасного оновлення регістрів ШИМ;
- механізм мультиплексування: 3 виходи з високою здатністю навантаження на кожен канал ШИМ;
- АЦП (6 каналів, 8 розрядів);

- 3,5 кбайта ПЗУ, 176 байтів ОЗП, 16-бітовий таймер, RTI;
- асинхронний послідовний інтерфейс, 28-вивідні корпуси DIP, SOIC.

Сімейство 16-розрядних МК HC16 широко використовується в різних системах управління автомобілями, в телекомунікаційному устаткуванні (сотових телефонах, телефонних комутаторах), побутовій електроніці (відеокамерах, телевізорах, цифрових аудіосистемах), офісній техніці (факсах, модемах, копіювальній техніці), медичному устаткуванні, робототехніці.

Ефективному використанню цих МК сприяє висока продуктивність 16-бітового CPU16 з частотою 25 МГц і потужна периферія.

Сімейство 32-розрядних МК 68300 є найбільш поширеним з високопродуктивних МК фірми Motorola. У ньому можна виділити *3 основних групи*, що принципово відрізняються по функціональному призначенню: *комунікаційну* (що містить комунікаційний співпроцесор), *промислового управління* (містить таймерний співпроцесор і застосовується в індустріальних системах управління, автомобільних контроллерах і т.д.), *загального призначення* (містить крім центрального процесора найбільш поширену універсальну периферію).

Докладніший опис архітектури (ЦПУ, системи команд, вбудованої пам'яті і підсистем вводу-виводу), зведених таблиць і структурних елементів і схем представників сімейств HC05, HC08, HC11, 16-розрядних HC12 і HC16, 32-розрядних 68300 мікроконтролерів фірми Motorola наведено у [24].

Контрольні питання і завдання

1. Призначення, схема і принцип дії однокристальних МК.
2. Розкажіть про призначення і роботу основних елементів МК51.
3. Пристрій і режими роботи портів вводу-виводу МК51.
4. Призначення, пристрій і режими роботи послідовного порту УАПП МК51.
5. Назвіть найбільш важливі особливості архітектури ОМК.
6. Де зберігаються програма і дані в МК51?
7. Викладіть основні функції, що виконуються АЛП в МК51.

8. Регістри спеціальних функцій МК51, їх структура і призначення.
9. Призначення, устрій і режими роботи таймера/лічильника МК51 і порядок його програмування.
10. Особливості роботи МК51 в режимах скидання, покрокового виконання команд, холостого ходу і зниженого енергоспоживання.
11. Призначення і функціонування пристрою управління і синхронізації.
12. Викладіть загальну характеристику системи команд МК51.
13. Приведіть команди передачі даних і приклади їх застосування.
14. Команди арифметичних і логічних операцій та приклади їх застосування.
15. Назвіть команди операцій з бітами і дайте приклади їх застосування.
16. Приведіть команди передачі управління і приклади їх застосування.
17. Приведіть приклади запису програм на мові Асемблер МК51.
18. Призначення і функціонування відладчика програм FD-51.
19. Призначення асемблюючої програми ASM-51.
20. Назвіть основні етапи і порядок відладки програм.
21. Склад програмного забезпечення крос-засобів відладки програм.
22. Приведіть структурну схему, особливості архітектури і системи команд МК MCS251.
23. Особливості архітектури МК фірми Atmel серій AT89 і AT90.
24. Архітектура МК фірми Microchip сімейств PIC16/17 і PIC16X7XX.
25. Особливості архітектури і застосування МК фірми Motorola.
26. Структура МК сімейства HC05 і його основні характеристики.
27. Назвіть область застосування МК фірми Atmel.
28. Назвіть область застосування МК фірми Microchip.
29. Назвіть область застосування МК фірми Motorola.
30. Приведіть блок-схему МК сімейства HC05.
31. Призначення і устрій послідовного асинхронного інтерфейсу зв'язку SCI.
32. Особливості роботи і призначення багатофункціонального таймера MFT.

33. Призначення системи переривань реального часу RTI.
34. Приведіть блок-схему МК MC68HC705MC4 і назвіть області його застосування.
35. Приведіть приклади можливого застосування МК AT89 і AT90 в електроапаратобудуванні (ЕА) і електропобутовій техніці (ЕПТ).
36. Приведіть приклади застосування МК HC05 для захисту в ЕА і ЕПТ.
37. Назвіть області застосування 16-ти розрядних МК HC16 і 32-х розрядних МК68300.
38. Приведіть приклад застосування МК PIC16 у ЕПТ і ЕА .
39. Призначення та устрій сторожового таймера і температурного сенсора в МК.
40. Охарактеризуйте систему команд МК PIC16/17 і приведіть приклади застосування команд в управлінні ЕА і ЕПТ.
41. Викладіть особливості конвеєрної системи обробки команд у мікроконтролерах PIC16/17.
42. Призначення і устрій високошвидкісного синхронного послідовного інтерфейсу SPI.
43. Призначення і функціонування системи переривань реального часу RTI (Real Time Interrupt).
44. Призначення і функціонування підсистеми таймера входної фіксації ІС і вихідного порівняння ОС.
45. Призначення і характеристики широтно-імпульсного модулятора.
46. Назвіть серії мікроконтролерів сімейства HC05 які призначені для:
 - управління електродвигунами в ЕПТ: холодильниках, пральних машинах, мікрохвильових печах, кухонних комбайнах і т.ін.
 - в системах управління автомобіля (приладова панель, склопідйомники, підвіска і т.ін.).
 - в різноманітних побутових пристроях, пультах керування.
 - в відео- і телевізійній апаратурі для управління кольоровими екранними дисплеями, яскравістю зображення, гучністю звуку, вибором певного каналу і т.ін.

6. ЛАБОРАТОРНИЙ ПРАКТИКУМ І СТЕНД МК51

Останніми роками позначилася тенденція широкого впровадження в системи управління електромагнітними і електроенергетичними системами і їх елементами однокристальних мікроконтролерів серії МК51, як найбільш популярних і корисних, в зв'язку з низькою їх вартістю і доступністю. Тому і в основу лабораторного стенду для вивчення однокристальних мікроЕОМ покладений МК такого ж типу – 1816BE51 (i8051), що є “de facto” всесвітнім промисловим стандартом.

Особливістю сімейства МК51 є: високий рівень уніфікації елементів, можливість перепрограмування на реалізацію тих або інших функцій без заміни складу комплексу технічних засобів, низькі витрати на обслуговування, контроль і діагностування, програмна сумісність з високопродуктивними (25 MIPS) МК фірми Cygnal.

Розроблені на кафедрі “Електричні апарати” НТУ “ХПІ” стенди МК51 і лабораторний практикум (див. нижче) дозволяють глибше вивчити структуру і особливості роботи МК51 в різних режимах, набути практичних навичок в роботі з мікроконтролерами цього сімейства, в складанні і відладці програм за допомогою крос-засобів.

6.1. Лабораторна робота №1. Дослідження лабораторного стенду мікроконтролера МК51

Мета роботи – ознайомлення із структурою стенду мікроконтролера серії МК51, командами, режимами роботи, складанням і виконанням програм.

Порядок підготовки до роботи

Вивчити:

- рекомендовану літературу [20, 21, 25, 26];
- опис лабораторної установки;
- методичні вказівки до виконання лабораторної роботи.

Опис лабораторного стенду. Лабораторний стенд включає:

- макет мікроконтролера МК51 (рис.6.1), який за допомогою послідовного порту RS232, що має розв'язку на оптронах, сполучений з ПЕОМ ІВМ;
- джерело живлення +5 В (15 В) виконане на окремій платі і таке, що підключається до електричної мережі напругою 220 В і частотою 50 Гц;
- двопробеневий осцилограф С1-55, який використовується для дослідження різних режимів роботи МК51 шляхом підключення його до тих або інших пристроїв за допомогою гнізд, відповідно до функціональної схеми лабораторного стенду, приведеної на рис.6.1.

Електрична схема і комутаційні гнізда розташовані на лицьовій панелі макету, яка виконана з органічного скла. Тут же знаходяться перемикачі і кнопки, які використовуються для подання на входи різних пристроїв логічної інформації, а також 24 світлодіоди, які призначені для індикації стану шини адреси і даних, порту P1.

Основні відомості про лабораторний стенд МК51

Лабораторний стенд (ЛС), виконаний на базі однокристалного мікроконтролера МК51, відноситься до мікропроцесорних пристроїв обчислювальної техніки і може використовуватись в системах управління технологічним устаткуванням у складі випробувального, науково-дослідного, контрольно-вимірювального і іншого устаткування. Призначення виводів мікросхеми і структурні схеми МК приведені на рис. 5.1, 5.2 і 6.8.

Лабораторний стенд дозволяє проводити дослідження:

- структурної схеми МК51, арифметико-логічного пристрою, резидентної пам'яті, пристрою управління і синхронізації;
- портів вводу-виводу інформації і доступу до зовнішньої пам'яті;
- таймера-лічильника, послідовного інтерфейсу, системи переривань і режимів читання і запису даних;
- особливих режимів роботи, завантаження і верифікації прикладних програм, покрокового виконання команд, скидання, режимів холостого ходу і зниженого енергоспоживання;
- системи команд, вводу, редагування, трансляції і налагодження

- прикладних програм;
- організації взаємодії мікроконтролера з об'єктом управління, вводу інформації з датчиків, виводу сигналів, що управляють, з МК51, масштабування, перетворення кодів;
 - аналого-цифрових (АЦП) і цифро-аналогових (ЦАП) перетворень, вводу інформації з клавіатури, виводу і відображення інформації;
 - послідовного порту МК51 для зв'язку з інтерфейсом RS232.

Структура функціональної схеми.

Функціональна схема лабораторного станду, призначеного для вивчення мікроконтролерів сімейства МК51, представлена на рис. 6.1.

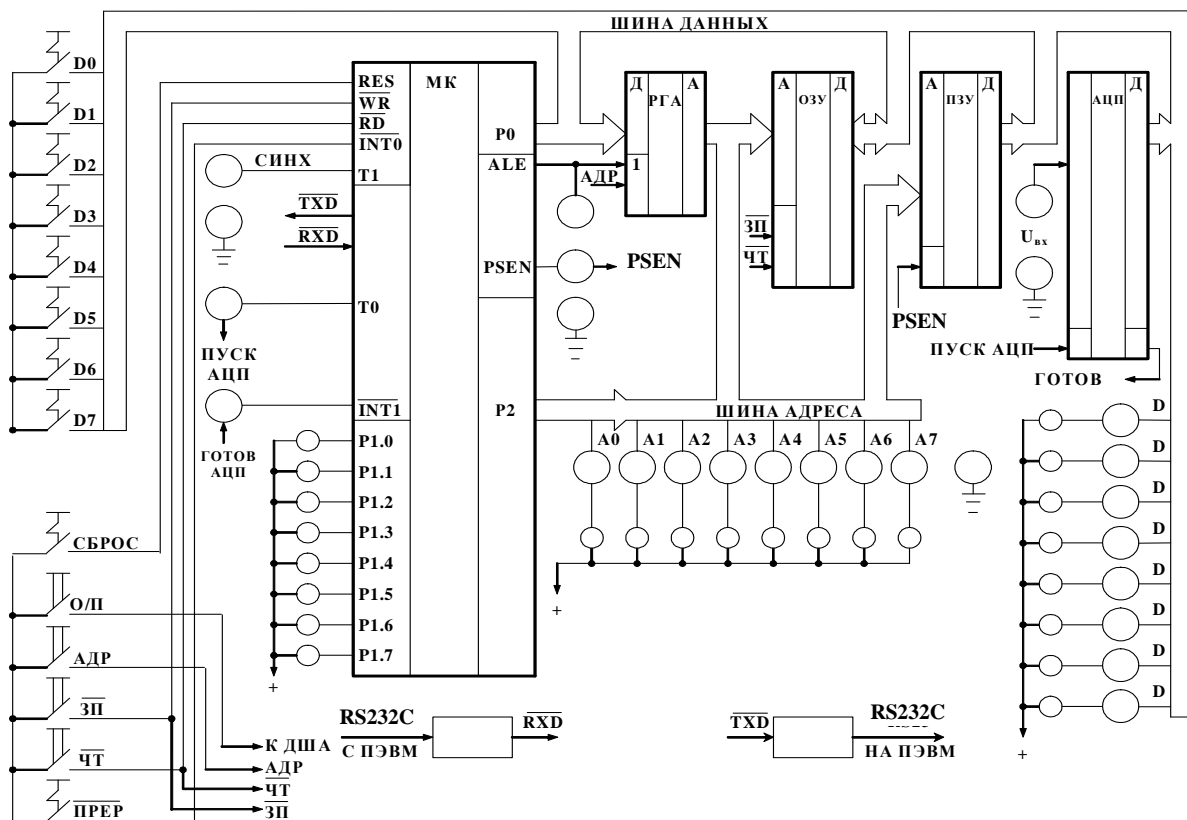


Рисунок 6.1 – Функціональна схема лабораторного станду МК51

Схема включає: мікроконтролер типу КР1830ВЕ31, ВІС ОЗП - КР537РУ10, ВІС ПЗП - К537РФ2, АЦП - К1113ПВ1, мультиплексор на 8 каналів 590КН6, 8-розрядний регістр адреси, 8-розрядний регістр даних (індикація), 8-розрядний регістр даних, 8-розрядний регістр аналого-

цифрового перетворювача (АЦП), 8-розрядний регістр вводу, клавіші для вводу інформації, логіку для управління доступом до ОЗП/ПЗП і регістрів, 8 кнопок Д0-Д7 з фіксацією для вводу адреси і даних для програмування стенду, кнопку “СКИДАННЯ” з фіксацією, кнопку ПРЕР (переривання) з фіксацією, кнопки АДР (адреса), ЗП (запис), ЧТ (читання) – без фіксації, кнопку О/П (ОЗП/ПЗП, англ. Start ROM-RAM), 24 світлодіоди АЛ307Б для індикації стану шини даних, шини адреси і порту P1, 30 контактних гнізд для підключення вимірювальних приладів і імітаційних сигналів, інтерфейсні мікросхеми для зв'язку стенду з ПЕОМ, роз'єм для підключення ПЕОМ через порт СОМ2, блок живлення.

Призначення елементів і функціонування схеми лабораторного стенду

Контактні гнізда “СИНХ”, “ПУСК АЦП”, “ГОТОВИЙ АЦП”, “ALE”, “PSEN”, “Увх” служать відповідно для виконання студентами необхідних підключень в схемі. Для підключення зовнішнього генератора подій служить контактне гніздо “СИНХ”. До нього може бути підключений генератор тактової частоти і генератор зовнішніх одиночних тактових імпульсів, які підраховуватиме внутрішній лічильник мікроЕОМ відповідно до заданої програми. Гнізда “ПУСК АЦП” і “ГОТОВИЙ АЦП” служать для підключення керуючих сигналів аналого-цифрового перетворювача стенду. При цьому вихід T0 мікроЕОМ виступає як джерело сигналів запуску на вимірювання, а вхід INT1 – для прийому сигналів готовності чергового вимірювання АЦП.

Контактні гнізда “ALE”, “PSEN”, служать для підключення осцилографа з метою спостереження часових діаграм роботи мікроЕОМ або для синхронізації роботи зовнішніх пристроїв, підключених до стенду.

Гніздо “Увх” призначене для подачі на вхід АЦП вимірюваної напруги.

Кнопки Д0-Д7 мають фіксацію і призначені для набору кодів даних і адреси, які необхідно занести в оперативну пам'ять контролера.

Світлодіодні індикатори Д0-Д7 забезпечують відображення кодів даних, які задаються відповідними кнопками.

Світлодіодні індикатори А0-А7 призначені для відображення стану шини адреси контролера.

Кнопка “СКИДАННЯ” (рос. “СБРОС”) служить для початкового встановлення однокристалльної мікроЕОМ. При її натисненні мікроЕОМ встановлює на шині даних адресу команди 0000H.

Кнопка “АДР” служить для запису коду даних, набраного за допомогою кнопок Д0-Д7 в реєстр адреси контролера РГА.

Кнопка “ЗП” призначена для запису коду даних, набраних за допомогою кнопок Д0-Д7, в заданий елемент оперативної пам'яті контролера.

Кнопка “ЧТ” служить для читання коду даних, який записаний в елементі пам'яті, адреса якої зберігається в реєстрі РГА і відображення його – на світлодіодних індикаторах Д0-Д7.

Кнопка “ПРЕР” служить для подачі на вхід переривання мікроЕОМ команди, яка забезпечує виконання її програми обслуговування переривання.

Контактне гніздо “Т0” призначене для подачі сигналу на виконання циклу вимірювань. Цей сигнал поступає на відповідний вхід мікроЕОМ, що забезпечує виконання підпрограми обслуговування АЦП.

Світлодіодні індикатори, які підключені до ліній порту Р1, служать для виведення логічних сигналів, які повідомляють про виконання деяких програмних подій. Наприклад, цей порт може бути використаний для виведення восьмирозрядних двійкових кодів результатів обчислень. Оптронні розв'язки ОР1 і ОР2 служать для електричного розділення ланцюга каналу зв'язку з ПЕОМ.

Лабораторний стенд побудований з функціонально закінчених модулів, які сполучені між собою за магістральним принципом. Системна магістраль (канал) дозволяє адресувати 64К 8-розрядних слів пам'яті команд і пам'яті даних. Активним пристроєм в каналі є МК, крім випадку, коли МК знаходиться в режимі скидання.

У лабораторному стенді передбачені два основні режими роботи: “ПДП” і “РОБОТА”. У режимі “ПДП” ввод в ОЗП з клавіатури програми або даних виконується при натиснутій кнопці “СКИДАННЯ” (рос. “СБРОС”).

У режимі “РОБОТА” кнопка “СКИДАННЯ” віджата, мікроконтролер виконує програму, яка знаходиться або в ПЗП, або раніше введена в ОЗП.

При включенні стенду мікроконтролер починає виконувати програму

за нульовою адресою. Для роботи разом з ЕОМ ІВМ РС повинна бути запущена програма, яка розташована в ПЗП, а для автономної роботи – програма, яка вводиться користувачем в ОЗП. Для вибору джерела початку виконання програми (ОЗП/ПЗП) служить кнопка S10 – О/П. Коли кнопка віджата за адресою 0–32768 знаходиться ПЗП, а ОЗП знаходиться в кінці адресного простору (32768–40960). При натиснутій кнопці за адресою 0–8192 знаходиться ОЗП, а за адресою 8193-32768 – ПЗП. Головним елементом електричної, принципової схеми мікроконтролерного стенду є мікроконтролер КР1816ВЕ51, частота роботи якого задається зовнішнім кварцем. Виводи порту P0 утворюють двонаправлену 8-бітову шину даних, яка з'єднує контролер з ОЗП, ПЗП, портами вводу-виводу, зібраними на регістрах.

Система ПЗП (ДД8) включає програми режимів початкового пуску, резистентного перевіряючого тесту, збору і обробки інформації з АЦП і портами вводу-виводу, програми для зв'язку з комп'ютером. Молодша частина шини адреси A0-A7 фіксується в регістрі ДД7 по спаду сигналу, який приходить на вхід С. Залежно від режимів роботи стенду на вхід С регістра ДД7 можуть поступати, або сигнал ALE з мікроконтролера, або сигнал ADR з кнопки “АДР”. Функцію перемикача залежно від режиму роботи виконує мультиплексор, зібраний на мікросхемах ДД 4.4 і ДД 5.4. Управління мультиплексором проходить по сигналу RES. Старша частина адреси A8-A15 виходить з порту P2 МК. У сумі це складає 16 адресних ліній, які забезпечують доступ до 65535 байтів зовнішньої пам'яті.

Схема дозволу для доступу до ОЗП/ПЗП побудована на елементах ДД12.4/ДД3.3. Якщо МК встановлює на A15 логічну одиницю, тобто адреса 32768, то включається доступ до ОЗП.

При натиснутій кнопці О/П доступ до ПЗП включається, якщо A13 або A14=1, тобто при адресі 8192-32768. У інших діапазонах пам'яті працює ОЗП.

Сигнал RD RAM (читання пам'яті) формується на мікросхемах ДД2.1, ДД2.4. На вхід елемента “Г” подається RD, PSEN. RDC1. Таким чином, вибірка даних з ОЗП відбуватиметься при читанні МК з пам'яті, запуску програми з ОЗП і читанні з клавіатури в режимі “ПДП”.

Сигнал WR RAM формується також на елементі ДД2.2. Запис в ОЗП можливий як в режимі “РОБОТА”, так і з клавіатури в режимі “ПДП”.

На елементі ДД2.3 зібрана схема усунення дребезгу контактів кнопок “ПЕРЕР” і “СКИДАННЯ”. Елементи “Г” сполучені так, що утворюють нетактований RS-тригер. Схема управління регістром ДД1 (кнопка даних) виконана так, що вона пропускає дані на вихід тільки в режимі “ПДП”, якщо натиснута кнопка “АДР” або “ЗП”.

Аналого-цифрова частина зібрана на мультиплексорі аналогових каналів КР560КН6 (ДА1), 10-бітовому АЦП К1113ПВ1 (ДА2) і карті читання даних з АЦП, зібраній на регістрі КР1533ИР33 (ДД). АЦП може працювати в двох режимах з різною точністю – 8 і 10 біт. При розподільній здатності 8 біт, якщо ведеться передача перетвореної інформації через послідовний порт, швидкість буде трохи вища.

Процес перетворення аналогових сигналів в цифрову форму протікає таким чином. Мікроконтролер записує в порт P1 номер каналу для перетворення (0-8), потім необхідно провести запуск АЦП послідовного наближення К1113 ПВ1 (ДА2). Це виконується шляхом установки біта 5 порту P3 в 1. Після закінчення перетворення на виходах АЦП встановлюється перетворений цифровий код сигналу, який подається на вхід вибраного каналу мультиплексора. Одночасно з видачею даних АЦП тестує переривання мікроконтролера, після чого викликається відповідна процедура обробки, яка через регістр ДД16 прочитає цей код. Для перетворення всіх восьми каналів необхідно в тому ж порядку повторити послідовність операцій, які описані вище. Таким чином, швидкість перетворення аналогової інформації знаходиться в прямій залежності від числа каналів вводу, які обробляються. Для одного каналу вводу час буде мінімальним і дорівнюватиме 30 мкс, для N каналів воно складатиме $30 \text{ мкс} \times N$.

Інтерфейс зв'язку лабораторного стенду з комп'ютером типу IBM зібраний відповідно до вимог, встановлених стандартом RS232C. Крім того, на оптронах, зібраних в ДІГ8 корпус (АОТ101), зроблена гальванічна розв'язка від решти частини схеми.

Схемою передавача є перетворювач рівнів TTL в напругу $-5 \text{ В} +5 \text{ В}$.

Приймач проводить зворотне перетворення. За логічну одиницю вважається рівень напруги $-3...-5$ В, а за логічний нуль $+3...+5$ В.

Порядок виконання роботи

Для підготовки лабораторного стенду необхідно:

- підключити стенд до мережі змінного струму 220В;
- вимикач живлення на задній панелі стенду перевести у верхнє положення (“Включити”); при цьому на індикаторах стенду з'явиться якась випадкова інформація;
- перевести мікроконтролер в режим ручного управління (без ПЗП); для цього клавіша “О/П” повинна бути натиснута;
- натиснути клавішу “СКИДАННЯ” (фіксується при натисненні).

Задача 1. Виконати запис даних в пам'ять згідно варіантам (табл. 6.1).

Для виконання цієї операції необхідно спочатку встановити адресу заданої комірки пам'яті в двійковому коді, а після цього записати в комірку необхідні дані (також в двійковому коді).

Встановлення необхідної адреси виконується таким чином:

1. На клавішах з фіксацією D_0 - D_7 набирають адресу в двійковому коді. При цьому треба звернути увагу на те, що D_0 – це молодший розряд, а D_7 – старший.

Наприклад, щоб встановити адресу комірки пам'яті з номером 5, потрібно натиснути клавіші D_2 і D_0 , що відповідає двійковому коду 00000101 (інші клавіші повинні бути не натиснуті).

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
□	□	□	□	□	■	□	■

2. Для запису адреси, встановленої на клавішах, в регістр адреси потрібно натиснути клавішу “АДР” (без фіксації). Після цього на індикаторах адреси A_0 - A_7 з'явиться встановлена адреса в двійковому коді.

У нашому прикладі для адреси 5 індикатори матимуть такий вигляд:

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
○	○	○	○	○	●	○	●

3. Для запису даних в комірку пам'яті за встановленою адресою необхідно на клавішах D_0 - D_7 набрати відповідне число в двійковому коді,

після чого натиснути клавішу “ЗП”. При цьому на індикаторах D0-D7 повинне з'явитися набране число в двійковому коді. Таким чином, дані будуть записані за встановленою адресою (адреса відображається на індикаторах A0-A7). Наприклад, щоб записати число 7, потрібно натиснути клавіші D2, D1, D0, що відповідає двійковому коду 00000111:

D₇ D₆ D₅ D₄ D₃ D₂ D₁ D₀
 □ □ □ □ □ ■ ■ ■

Після цього потрібно натиснути клавішу “ЗП”. При цьому на індикаторах D₀-D₇ з'явиться записане за встановленою адресою в комірку пам'яті число 7 в двійковому коді:

D₇ D₆ D₅ D₄ D₃ D₂ D₁ D₀
 ○ ○ ○ ○ ○ ● ● ●

Примітка: У разі запису даних в декілька комірок пам'яті, адресу кожної з них перед записом, треба встановлювати окремо.

Варіанти даних для виконання завдання приведені в таблиці 6.1.

Таблиця 6.1 – Варіанти даних

№ Варіанту.	Початкова адреса N	Дані для вводу в комірки, починаючи з адреси N			№ варіанту	Початкова адреса N	Дані для вводу в комірки, починаючи з адреси N		
		N	N+1	N+2			N	N+1	N+2
1	00	37	54	34	14	39	58	23	11
2	03	69	44	46	15	42	54	123	03
3	06	33	50	22	16	45	12	64	10
4	09	07	00	255	17	48	15	54	45
5	12	04	214	23	18	51	08	68	25
6	15	23	16	56	19	54	09	82	124
7	18	69	61	04	20	57	35	28	02
8	21	38	69	36	21	60	01	78	255
9	24	57	68	12	22	63	23	54	12
10	27	68	46	69	23	66	58	12	02
11	30	89	56	98	24	69	54	84	200
12	33	125	91	245	25	72	98	58	145
13	36	45	27	56	26	75	65	05	204

Завдання 2. Дослідити зміст пам'яті.

1. Аналогічно п. 1 попереднього завдання встановлюємо клавішами D0-D7 потрібну адресу комірки пам'яті згідно варіанту.

2. Натискаємо клавішу "ЧТ". Після цього на індикаторах повинен з'явитися в двійковому коді вміст комірки пам'яті за встановленою адресою.

3. Одержані результати звести в табл.6.2.

Таблиця 6.2 – Одержані результати

АДРЕСА	Початкові дані	Прочитані дані
N	Двійковий код встановлений в комірці N	Двійковий код прочитаний з комірки N
N+1	Двійковий код встановлений в комірці N+1	Двійковий код прочитаний з комірки N+1
N+2	Двійковий код встановлений в комірці N+2	Двійковий код прочитаний з комірки N+2

Зміст звіту:

- мета роботи, структурна схема стенду і одержані результати (табл.6.2).

Контрольні питання

1. Структура мікроконтролера і призначення виводів мікросхеми МК51.
2. Структура лабораторного стенду і система команд МК51.
3. Основні функції і правила користування стендом.
4. Режими роботи лабораторного стенду.

6.2. Лабораторна робота № 2. Програмування і ввід програм з клавіатури стенду МК51

Мета роботи – вивчити команди і особливості організації програм на мові Асемблер МК51, їх вводу з клавіатури стенду та виконання.

Порядок підготовки до роботи:

- вивчити рекомендовану літературу [20,21,25, 26];
- ознайомитися з системою команд мікроконтролера і основними їх функціями (див. табл. 5.7); звернути увагу на особливості складання і

виконання програм на мові Асемблер МК51.

Використовуване устаткування – лабораторний стенд МК51.

Основні відомості

Аналогічно з регістром ознак процесора серії КР580 в мікроконтролері МК51 є регістр, який містить в собі інформацію про результати виконання обчислень: ознаки перенесення (С), додаткового перенесення (АС), переповнення (OV), паритету (Р) і прапор користувача (F0). Але на відміну від регістра ознак, цей регістр ще містить в собі інформацію про те, який з чотирьох можливих банків регістрів використовується програмою. Цей регістр називається регістром СЛОВА СТАНУ (PSW – англ.).

Для встановлення відповідного банку регістрів, який використовуватиметься програмою, необхідно на початку програми встановити відповідне значення біт RS0 і RS1 у регістрі «слова стану» (PSW).

Формат регістра «слова стану» програми ССП (PSW) приведений нижче на рис. 6.2.

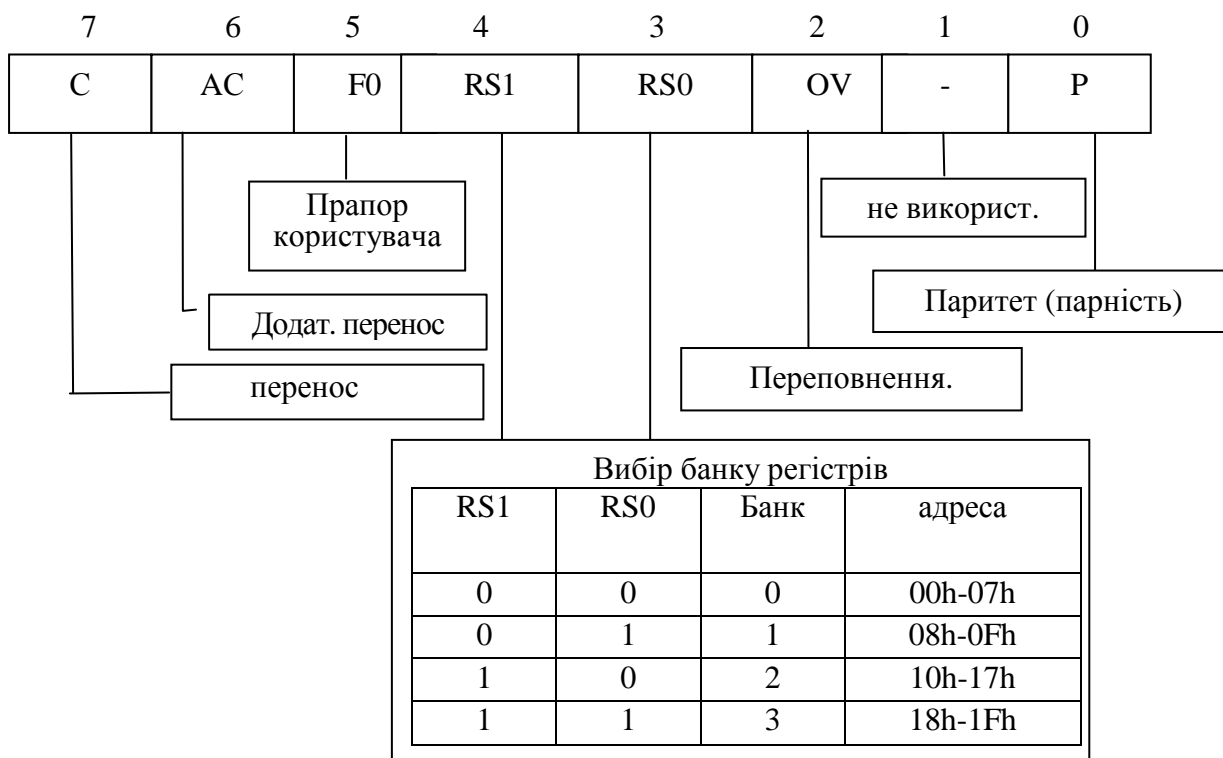


Рисунок 6.2 – Формат регістра «слова стану» (PSW)

Для цього використовуються команди встановлення і скидання біт відповідно SETB bit і CLR bit. Адреса регістра «слова стану» – 0Dh. Формат 2-байтної команди: SETB xxxx xxxx, або CLR xxxx xxxx.

Перший байт зліва – двійковий код команди SETB (11010010) або CLR (11000010). Старші чотири біти в другому байті команд – це адреси слова стану PSW (1101 в двійковому коді), а молодші чотири біти – номер біта в регістрі PSW, який встановлюється або скидається (RS0 (0011), RS1 (0100)).

Наприклад, щоб встановити нульовий банк регістрів треба виконати дві команди скидання біт RS0 і RS1 в регістрі слова стану (PSW):

```
CLR PSW. 3    11000010  1101 0011
CLR PSW. 4    11000010  1101 0100
```

Для того, щоб встановити перший банк регістрів, треба виконати одну команду скидання і одну встановлення бітів:

```
SETB PSW. 3   11010010  11010011
CLR PSW. 4    11000010  11010100
```

Коди операцій SETB і CLR: SETB – 11010010 CLR 11000010.

Для того, щоб результат розрахунків програми можна було перевірити візуально, в кінці програми розрахунків треба поставити команду виведення змісту акумулятора в порт P1, до якого підключені світлодіоди. Таким чином, судячи за станом світлодіодів ми зможемо прочитувати зміст акумулятора. При цьому треба врахувати одну особливість – дані в порт виводяться в інвертованому вигляді, тобто перш ніж записувати акумулятор в порт, його треба проінвертувати. Команда інвертування змісту акумулятора – CPL A 11110100.

Команда запису в порт вмісту акумулятора така:

```
MOV P1, A    1111010 10010000.
```

В кінці програми треба поставити оператор відносного переходу, щоб програма зупинилася на циклі на останньому операторі.

Це пояснюється тим, що в системі команд МК51 у зв'язку із специфікою роботи не передбачена команда зупинки програми користувача, а тому в кінці програми замість оператора зупинки використовується оператор переходу на відповідну кількість кроків назад. У мнемонічному

кодi це повинно виглядати так: Мітка1: NOP холоста команда;

SJMP Мітка1: команда відносного переходу

У двійковому кодi для нашого випадку це виглядає таким чином:

NOP 00000000 SJMP rel 10000000 11111101

Порядок виконання роботи

Завдання 1. Скласти алгоритм і програму арифметичного складання (або віднімання) двох чисел – N1 і N2 і логічної операції з числом N3 згідно відповідним виразам (1–4). До складу програми повинні увійти команди встановлення активного банку регістрів (у регістрі PSW), команди занесення даних в регістри Rn і акумулятор, команди арифметичних і логічних операцій, інвертування змісту акумулятора, запису вмісту акумулятора в порт, а також команда відносного переходу в кінці програми. Варіанти завдання приведені нижче в табл. 6.3.

$(N1+N2) \vee N3$ (1) $(N1-N2) \wedge N3$ (2) $(N1+N2) \nabla N3$ (3) $(N1-N2) \nabla N3$ (4)

Таблиця 6.3 - Варіанти даних в десятковому кодi

Номер за списком	Номер виразу	N1	N2	N3	Банк робочих регістрів	Номери регістрів, Rn
1	1	3	9	6	0	R1,R2
2	2	6	4	10	1	R3,R4
3	3	10	2	56	2	R5,R6
4	4	64	48	12	3	R1,R2
5	1	4	12	16	2	R3,R4
6	2	18	2	11	1	R5,R6
7	3	40	8	1	0	R1,R2
8	4	56	6	25	1	R3,R4
9	1	10	38	6	2	R5,R6
10	2	128	64	2	3	R1,R2
11	3	2	2	2	2	R3,R4
12	4	4	2	2	1	R5,R6
13	1	1	3	4	0	R1,R2
14	2	26	2	12	1	R3,R4
15	3	10	10	1	2	R5,R6
16	4	100	96	2	3	R1,R2
17	1	4	12	6	2	R3,R4
18	2	7	5	10	1	R5,R6
19	3	14	2	56	0	R1,R2
20	4	64	48	18	1	R3,R4

Продовження табл. 6.3

21	1	4	12	16	2	R5,R6
22	2	18	2	10	3	R1,R2
23	3	40	8	2	2	R3,R4
24	4	56	6	28	1	R5,R6
25	1	1	5	3	0	R1,R2

Регістри для виконання програми використовуються згідно варіанту завдання. При виконанні завдання можна користуватися командами, які приведені у табл. 6.4.

Таблиця 6.4 – Назви команд і коди

Мнемокод команди	Двійковий код команди	Коментар
S JMP rel	10000000 + rel	PC←PC+2+rel. Короткий відносний перехід всередині сторінки в 256 байт, де rel – 8-бітний відносний адрес передачі управління (-127 ÷ +128) Rn номер регістра (Rn) в двійковому коді (n=0 ÷ 7)
MOV Rn, #d	01111rrr # d	Rn←#d. Завантаження в регістр безпосереднього операнда константи #d.
MOV A, Rn	11101rrr	A←Rn. Пересилка з регістра в акумулятор. rrr – трьохбітове поле, яке визначає номер регістра (Rn) в двійковому коді (n=0 ÷ 7) (000 – R0, 001 – R1 ... 111 – R7).
MOV A, #d	01110100 #d	A← #d. Завантаження в акумулятор константи.
MOV P1, A	11110101 10010000	P1←A. Записати в порт P1 вміст акумулятора.
ADD A, Rn	00101rrr	A←A+Rn. Додавання акумулятора з регістром (n=0 ÷ 7).
SUBB A, Rn	10011rrr	A←A-C-Rn. Віднімання з акумулятора регістра і зайому.
RL A	00100011	Циклічний зсув акумулятора вліво.
RR A	00000011	Циклічний зсув акумулятора вправо.
ANL A,Rn	01011rrr	A←A∧Rn. Логічне «І» акумулятора і регістра.
ORL A, Rn	01001rrr	A←A∨Rn. Логічне «АБО» акумулятора і регістра.
XRL A,Rn	01101rrr	A←A⊕Rn. Логічне “Виключаюче АБО” акумулятора і регістра.
MOVX @DPTR, A	11110000	((DPTR)) ← (A). Пересилка байта даних з акумулятора в комірки розширеної зовнішньої пам'яті даних.
CPL A	11110100	(A) ← (\overline{A}). Інвертування вмісту акумулятора.
NOP	00000000	Порожня команда.

Завдання 2. Виконати розрахунки відповідно з розробленою програмою на лабораторному стенді і вручну згідно з варіантом табл. 6.3. Одержані результати порівняти. Приклад програми для варіанту 16 наведено нижче.

Мітка	Команда	1-й байт	2-й байт
	SETB PSW.3	11010010	11010011
	SETB PSW.4	11010010	11010100
	MOV A, #100	01110100	01100100
	MOV R1, #96	01111001	01100000
	SUBB A, R1	10011001	
	MOV R2, #2	01111010	00000010
	XRL A, R2	01101010	
	CPL A	11110100	
	MOV P1, A	11110101	10010000
LOOP:	NOP	00000000	
	SJMP LOOP	10000000	(адр. LOOP:)

Примітка: Введення програми або даних з клавіатури ЛС виконується при натиснутій кнопці "СКИД", а "О/П" віджатій. Для виконання програми необхідно встановити початкову адресу програми і віджати кнопку "СКИД".

Зміст звіту:

- текст програми в мнемонічному і двійковому коді;
- результати розрахунків за програмою і вручну.

Контрольні питання

1. Поясніть програму і команди, які використовуються в ній.
2. Як здійснюється вибір активного банку регістрів і виконання роботи в цілому?
3. Приведіть приклади команд роботи з бітами і їх формати.
4. Поясніть команди арифметичних і логічних операцій.
5. Викладіть команди переходів і виклику підпрограм.
6. Приведіть команди запису в порт і інвертування змісту акумулятора.
7. Приведіть команди пересилки і циклічного зсуву акумулятора.
8. Призначення і формат регістра СЛОВА СТАНУ (PSW).
9. Особливості команд відносного, довгого, абсолютного і непрямого переходів.
10. Поясніть команди SETB PSW.3 і CLR PSW.3 і запишіть їх двійковий код.

6.3. Лабораторна робота № 3. Складання і відладка програм

Мета роботи – навчитися складати, компілювати і відладжувати програми на мові Асемблер МК51 крос-засобами, використовуючи персональний комп'ютер.

Порядок підготовки до роботи:

- вивчити рекомендовану літературу [20, 21, 25, 26];
- ознайомитися з системою команд МК51 і правилами роботи з програмним забезпеченням крос-засобів та відладчиком FD-51;
- ознайомитися із структурою карти резидентної (внутрішньої) пам'яті даних МК51.

Опис устаткування:

- персональний комп'ютер сумісний з IBM;
- програмне забезпечення крос-засобів;
- лабораторний стенд МК51.

Основні відомості. До мінімального складу програмного забезпечення крос-засобів відносяться:

системна програма для вводу початкового тексту програми, його редагування і запису на зовнішній носій (так званий редактор тексту або символний редактор);

програма-транслятор, яка перетворює початковий текст програми на об'єктний модуль ASM-51, з редактором зовнішніх зв'язків, що дозволяє включити в програму модулі, розроблені незалежно один від одного;

програма відладчик – емулятор, що забезпечує переклад переміщуваних програмних модулів на абсолютні адреси.

До засобів розробки і відладки відносяться програми асемблера, компоновки і програми відладчики – емулятори.

Програма-асемблер – це програма, яка транслює початкову програму в об'єктні коди. Вона виконує привласнення дійсних адрес, перетворення чисел, привласнення дійсних значень символним змінним.

Програма на асемблері складається з *команд, директив і керуючих параметрів.*

Команди асемблера – мнемонічне кодування дій, які повинен виконати процесор.

Директиви – псевдокоманди, які при трансляції не перетворюються на машинні коди команд і використовуються для визначення змінних, структури програми, задання констант і т.ін.

Керуючі параметри указують транслятору на виконання певних дій.

Програма асемблера ASM-51 дозволяє використовувати модульний принцип, тобто обробляти окремі частини складної програми - модулі.

Після відладки всіх модулів окремо вони можуть бути зібрані в один файл спеціальною програмою компоновки RL51.

Початкова програма (файл name.a51) створюється за допомогою редактора текстових файлів і містить команди на мові Асемблер, директиви, що управляють, параметри асемблера і коментарі. Після створення початкової програми вона обробляється викликаною програмою асемблера ASM51.EXE (транслятором) за допомогою команди ASM51 name.a51.

Результатом роботи програми буде:

об'єктний файл, представлений в машинних кодах (форма початкових даних асемблера, які виконуються, що є абсолютний формат шістнадцятирічного коду і файл, який може бути запрограмований на МК51 і включає інформацію, необхідну для складання і відладки);

лістинговий файл (запис початкової програми і об'єктного коду, в який асемблер вносить повідомлення про всі помилки кодування і службову інформацію).

Асемблер має декілька директив, які дозволяють користувачу встановити символні імена (директива EQU), зарезервувати і ініціалізувати місце в пам'яті, керувати розміщенням програми.

Директиви управління: ORG – управління лічильником адреси; (задає асемблеру адресу елементу пам'яті, в якій повинна розміщуватися наступна за нею команда прикладної програми); END – кінець програми (указує на закінчення трансляції).

Відладчик асемблерних програм дозволяє:

завантажити для відладки HEX файл, що виробляється трансляторами з мови Асемблера (крос-засоби), а також файли чистого двійкового коду, зчитані з ПЗП;

переглянути на екрані дизасембльований текст завантаженої програми, адреси і коди команд, область імітованого ОЗП даних, зовнішньої пам'яті, пам'яті програм, вміст всіх регістрів;

виконати завантажену програму покроково з прогляданням результатів після кожного кроку, внести зміни в програму;

внести зміни у вміст регістрів, прапорів і пам'яті і т.ін.

Функціональні клавіші, які використовуються в даній роботі:

F1 – виконати поточну інструкцію завантаженої програми (поточною є підкреслена інструкція у вікні тексту програми (рис. 6.3));

F3 – дозволяє надавати числову інформацію в десятковому, двійковому, шістнадцятиричному кодах для регістрів;

F4 – перемикає вікно пам'яті даних з внутрішньої пам'яті на зовнішню;

F6 – перемикає форму представлення пам'яті у вікні в двійкову і шістнадцятиричну.

Bank 0	Bank 1	Bank 2	Bank 3	Special Function Registers		
R0=00->00	R0=00->00	R0=00->00	R0=00->00	TH0= 00	TL0= 00	
R1=00->00	R1=00->00	R1=00->00	R1=00->00	TH1= 00	TL1= 00	
R2=00	R2=00	R2=00	R2=00	P0= FF	P1= FF	
R3=00	R3=00	R3=00	R3=00	P2= FF	P3= FF	
R4=00	R4=00	R4=00	R4=00	DPH= 00	DPL= 00	
R5=00	R5=00	R5=00	R5=00	SP= 07	IP= 10100000	
R6=00	R6=00	R6=00	R6=00	TMOD=00000000	IE= 01000000	
R7=00	R7=00	R7=00	R7=00	TCON=00000000	SCON=00000000	
A=00	B=00		PC=0000	SBUF=00	PSW= 02	
			INT RAM		P S W	
			0000 00 00 00 00 00 00		C AC FO S1 S0 OV ** P	
			0006 00 00 00 00 00 00		0 0 0 0 0 0 1 0	
			000C 00 00 00 00 00 00		PGM ROM	
			0012 00 00 00 00 00 00		0000 00 00 00 00	
			0018 00 00 00 00 00 00		0004 00 00 00 00 07 00	
			001E 00 00 00 00 00 00		0008 00 00 00 00 06 00	
			0024 00 00 00 00 00 00		000C 00 00 00 00 05 00	
			002A 00 00 00 00 00 00		0010 00 00 00 00 04 00	
			0030 00 00 00 00 00 00		0014 00 00 00 00 03 00	
			0036 00 00 00 00 00 00		0018 00 00 00 00 02 00	
			CMD >_		s ms ncs 01 00	
					000 000 000 00 00	
1 Step 2 Proc 3Reg:DEC4 In/Ex 5SetBrk 6Mem:BIN7 I/E ↑ 8 Pgm ↑ 9 I/E ↓ 0 Pgm ↓						

Рисунок 6.3 – Вид вікна відладчика емулятора FD51

Команди відладчика:

H – для отримання довідки.

L [<тип пам'яті><початкова адреса>]<файл. HEX>[/A] – завантажити файл в пам'ять. Тип пам'яті: I (внутрішня), E (зовнішня) і P (програмна). Початкова адреса і тип пам'яті вказується тільки при завантаженні двійкового файлу;

/A – ключ тільки при завантаженні програми ISIS – II Macro Assembler;

R <номер регістра> = <число> – занести число в регістр поточного банку. Наприклад R4 = FF;

<ім'я регістра> = <число> – занести число в регістр спеціального призначення: A, B TH0, TH1, TL0, TL1, DPL, DPH, DPTR, SP, IP, IE, TMOD, TCON, SCON, SBUF, PC. Наприклад TH0=FF;

<ім'я прапора> = <число> – встановити або зняти прапор (біт) в PSW. Наприклад S1=0;

RST – імітується скидання процесора;

QUIT – вихід в DOS;

N – очищення змісту відладчика.

Правила запису програм на мові Асемблер для МК51

Початковий текст програми на мові Асемблер має певний формат.

Кожна команда (і псевдокоманда) є рядок чотирьох-ланкової конструкції, яка приводиться нижче:

Мітка	Операція	Операнд	Коментар
-------	----------	---------	----------

Ланки відділяються одна від іншої довільним числом пропусків.

Мітка. У полі мітки розміщується символічне ім'я елемента пам'яті, в якій зберігається відмічена команда або операнд. Мітка є букво – цифровою комбінацією, яка починається з букви. Використовуються тільки букви латинського алфавіту. Асемблер МК51 допускає використання в мітках символу підкреслення (_). Довжина мітки не повинна перевищувати 31 символ для МК51. Мітка завжди завершується двокрапкою.

Псевдокоманди асемблера не перетворюються на двійкові коди, а тому

не можуть мати міток. Винятки становлять псевдокоманди резервування пам'яті і визначення даних (DS, DB, DW). У псевдокоманд, які здійснюють визначення символічних імен, в полі мітки записується обумовлене символічне ім'я, після якого двокрапка не ставиться.

Як символічні імена і мітки не можуть бути використані мнемокоди команд, псевдокоманд і операторів асемблера, а також мнемонічні позначення регістрів і інших внутрішніх блоків МК.

Операція. У полі операції записується мнемонічне позначення команди МК або псевдокоманди асемблера, що є скороченням (аббревіатурою) повного англійського найменування виконуваної дії. Наприклад: MOV – move – перемістити JMP – jump – перейти, DB – define byte – визначити байт.

Для МК51 використовується строго визначений і обмежений набір мнемонічних кодів. Будь-який інший набір символів, розміщений в полі операції, сприймається асемблером як помилковий.

Операнди. У цьому полі визначаються операнди (або операнд), які беруть участь в операції. Команди асемблера можуть бути без -, одно -, або двооперандними. Операнди розділяються комою (,). Операнд може бути заданий безпосередньо або у вигляді його адреси (прямої або непрямої). Безпосередній операнд представляється числом (MOV A, #18) символічним або ім'ям (ADDC A, # OPER 1) з обов'язковою вказівкою префікса безпосереднього операнда (#). Пряма адреса операнда може бути задана мнемонічним позначенням (IN A, P1), числом (INC 50), символічним ім'ям (MOV A, MEMORY). Непряма адресація указується префіксом @. У командах передачі управління операндом може бути число (LCALL 0135H), мітка (JMP LABEL), непряма адреса (JMP @ A) або вираз (JMP π - 2, де π – поточний вміст лічильника команд). Використовувані, як команди, символічні імена і мітки повинні бути визначені, а числа представлені з вказівкою системи числення, для чого використовується суфікс (літера, яка стоїть після числа):

В – для двійкової Q – для вісімкової, D – для десяткової і H – для шістнадцятиричної. Число без суфікса за умовчанням вважається десятковим.

Обробка виразів в процесі трансляції

Асемблер МК51 допускає використання виразів в полі операндів,

значення яких обчислюються в процесі трансляції. Вираз є сукупність символічних імен і чисел, зв'язаних оператором асемблера. Оператори асемблера забезпечують виконання арифметичних (“+” – складання, “-” – віднімання “*” – множення, “/” – цілочисельне ділення, MOD – ділення по модулю) і логічних (OR – АБО, AND – І, XOR – виключаюче АБО, NOT – НІ) операцій у форматі двобайтових слів. Наприклад, запис ADD A, # (NOT 13+1) еквівалентна запису ADD A, # 0F3H і забезпечує складання вмісту акумулятора з числом 13, представленим в додатковому коді. Використовуються також оператори LOW і HIGH, які дозволяють виділити молодший і старший байти 2-байтного операнда.

Коментар. Поле для коментаря може бути використано програмістом для текстового або символного пояснення логічної організації прикладної програми. Поле для коментаря не компілюється асемблером, а тому в ньому допустимо використовувати будь-які символи. За правилами мови асемблера поле для коментаря починається після крапки з комою.

Псевдокоманди асемблера. Асемблююча програма транслює початкову програму в об'єктні коди. Хоча програма бере на себе багато що з рутинних завдань програміста, таких як привласнення дійсних адрес, перетворення чисел, привласнення дійсних значень символічним змінним і т.д., але програміст все-таки повинен вказати їй деякі параметри: наприклад, початкова адреса прикладної програми, кінець програми, яка асемблюється, формати даних і т.ін. Всю цю інформацію програміст вставляє в початковий текст своєї прикладної програми у вигляді псевдокоманд (директив) асемблера, які тільки керують процесом трансляції і не перетворюються на коди об'єктної програми.

Наприклад, псевдокоманда ORG 10H задає асемблеру адресу елементу пам'яті (10H), в якій повинна бути розташована наступна за нею команда прикладної програми.

Псевдокомандою EQU можна будь-якому символічному імені, яке використовується в програмі, поставити у відповідність певний операнд. Наприклад, запис SEM EQU 15 призводить до того, що в процесі компіляції усюди, де зустрічається символічне ім'я SEM, його буде замінено числом 15.

Символічні імена операндів, переобумовлених в процесі виконання

програми, визначаються псевдокомандою SET:

```
ALFA SET 3
...
...
...
ALFA SET ALFA +1
```

Асемблер МК51 дозволяє визначити символічне ім'я як адреси внутрішніх (псевдокоманда DATA), зовнішніх (XDATA) даних або адресу біта (псевдокоманда BIT). Наприклад, директива ERROR...FLAG BIT 25H3 визначає символічне ім'я ERROR_FLAG як третій комірки осередку ОЗП з адресою 25H.

Псевдокоманда DB забезпечує занесення в ПП константи, яка є байтом.

Псевдокомандою END програміст дає асемблеру вказівку про закінчення трансляції.

В результаті трансляції повинна бути одержана карта пам'яті програм, де кожному елементу пам'яті буде поставлений у відповідність код, який зберігається в ній. Відповідно формату команд, для представлення їх об'єктних кодів виділяється один, два або три елементи пам'яті програм. У першій комірці завжди розташовується код операції, в другій (а для МК51 і в третій) – безпосередній операнд, адреса операнда, який прямоадресується, або зсув (для команд передачі управління МК51). Для команд LCALL і LJMP в другому і третьому байтах об'єктного коду вказується адреса передачі управління (у другому – старша частина, в третьому – молодша).

Послідовний інтерфейс

При використанні МК необхідно здійснювати передачу і прийом інформації, представлену послідовним кодом (молодшими бітами вперед) в повному дуплексному режимі обміну. Для цього використовується універсальний асинхронний приймач-передавач (УАПП). У його склад входить спеціальний буфер-регістр (SBUF). Запис байта в буфер призводить до автоматичного перепису байта в зсуваючий регістр передавача і ініціює початок передачі байта. Наявність буфера-регістра приймача дозволяє суміщати операцію читання раніше прийнятого байта з прийомом чергового байта.

Послідовний порт МК51 може працювати в чотирьох режимах.

Режим 0. У цьому режимі інформація приймається і передається через зовнішнє виведення входу приймача (RXD). Приймаються або передаються 8 біт даних. Через зовнішні вихідні виводи передавача (TXD) видаються імпульси зсуву, які супроводжують кожен біт. Частота передачі біта інформації дорівнює $1/12$ частоти резонатора.

Режим 1. У цьому режимі передаються через TXD або приймаються з RXD 10 біт інформації: старт-біт (0), 8 біт даних і стоп біт (1). Швидкість прийому/передачі – величина змінна і задається таймером.

Режим 2. Дані передаються через TXD або приймаються з RXD по 11 біт: старт-біт, 8 біт даних і стоп-біт. При передачі даних дев'ятий біт може приймати значення “0” або “1”. Наприклад, для підвищення надійності передачі шляхом контролю по парності в ньому може бути поміщено значення ознаки паритету і слова стану програми (PSW.0). Частота прийомів/передач вибирається програмно і може дорівнювати $1/32$ або $1/64$ частоти резонатора залежно від біта SMOD, що управляє.

Режим 3. Співпадає з режимом 2 у всіх деталях за винятком частоти прийому передачі, що є величиною змінною і задається таймером.

Управління режимом роботи УАПП здійснюється через спеціальний регістр (SCON). Цей регістр містить не тільки біти, що управляють, які визначають режим роботи послідовного порту, але і дев'ятий біт прийнятих або переданих даних (RB8 і TB8), а також біти переривання приймачів-передавачів RI, TI.

Швидкість прийому передачі в режимі “0” залежить тільки від резонансної частоти кварцевого резонатора: $f_0 = f_p/12$. Призначення біт регістрів управління відображені в табл.6.5

За один машинний цикл послідовний порт передає один біт інформації. У режимах 1,2,3 швидкість прийому передачі залежить від значення біта SMOD, що управляє, в регістрі спеціальних функцій (PUM).

Програма “Монітор” для ПЕОМ надає користувачу ряд можливостей для забезпечення зручного зв'язку із лабораторним стендом МК51, які приведені нижче:

- прийом (передача) даних з (в) внутрішнього ОЗП контролера;
- прийом (передача) даних з (в) ПЕОМ;
- запуск і відладка програми користувача.

Таблиця 6.5 – Призначення біт регістра управління УАПП (SCON)

Симв.	Позиція	Ім'я і призначення
SM0 SM1	SCON.7 SCON.6	Біти управління режимом УАПП. Встановлюються і скидаються програмно (див. примітку).
SM2	SCON.5	Біт управління режимом УАПП. Встановлюються програмно для заборони прийому повідомлень, в яких дев'ятий біт дорівнює "0"
REN	SCON.4	Біт дозволу прийому. Встановлюється і скидаються програмно для дозволу/заборони прийому послідовних даних.
TB8	SCON.3	Передача біта 8. Встановлюється/скидається програмно. Для задання дев'ятого переданого біта в режимі УАПП – 9 біт
RB8	SCON.2	Прийом біта 8. Встановлюється/скидається апаратно. Для фіксації дев'ятого прийнятого біта в режимі УАПП – 9 біт
TI	SCON.1	Прапор переривання передавача встановлюється апаратно при закінченні передачі байта. Скидається програмно після обслуговування переривання
RI	SCON.0	Прапор переривання приймача встановлюється апаратно при прийомі байта. Скидається програмно після обслуговування переривання

Примітка: Коды режимів роботи УАПП

SM0	SM1	Режим роботи УАПП
0	0	Зсуваючий регістр розширення вводу-виводу
0	1	УАПП 8 біт – Змінна швидкість передачі
1	0	УАПП 9 біт – Фіксована швидкість передачі
1	1	УАПП 9 біт – Змінна швидкість передачі.

Наявність зрозумілого інтерфейсу і набір директив управління дозволяє проводити відладку програми користувача і контроль результатів роботи безпосередньо на стенді. Програма "Монітор" написана на мові Паскаль, і для забезпечення її роботи потрібен ПВМ, сумісна з IBM, ОЗП не менше 2 Мб, і наявність не менше ніж 20 кб на диску. Зв'язок із стендом здійснюється через послідовний порт COM2. "Монітор" поставляється у вигляді одного виконуваного файлу monitor.exe і не потребує додаткових

налагоджень. Запуск програми здійснюється з командного рядка DOS командою > monitor.exe (або будь-яким іншим чином).

Примітка 1. Перед запуском програми переконайтеся, що стенд включений і підключений до ПЕОМ.

Після запуску програми “Монітор” на екрані буде виведене вікно програми, яка складається з 7 вікон (рис.6.4).

Ввод директив здійснюється у вікні команд (місце для нової директиви позначається запитом DOS). Результати роботи директиви будуть відображені у відповідному вікні. У разі виникнення помилки повідомлення про неї буде виведено у вікні “Повідомлення про помилки”.

Примітка 2. Якщо директива містить цифрові параметри (адреса або дані), то їх треба вводити в 16-річному коді.

Примітка 3. Якщо після запуску програми “Монітор” виникає непоправна помилка, подібна до “Відсутній зв'язок з контролером”, то вихід з програми здійснюється клавішею “Esc”. Під час роботи вихід з програми здійснюється шляхом вводу директиви Q.

Директиви «Монітора» приведені в табл. 6.6.

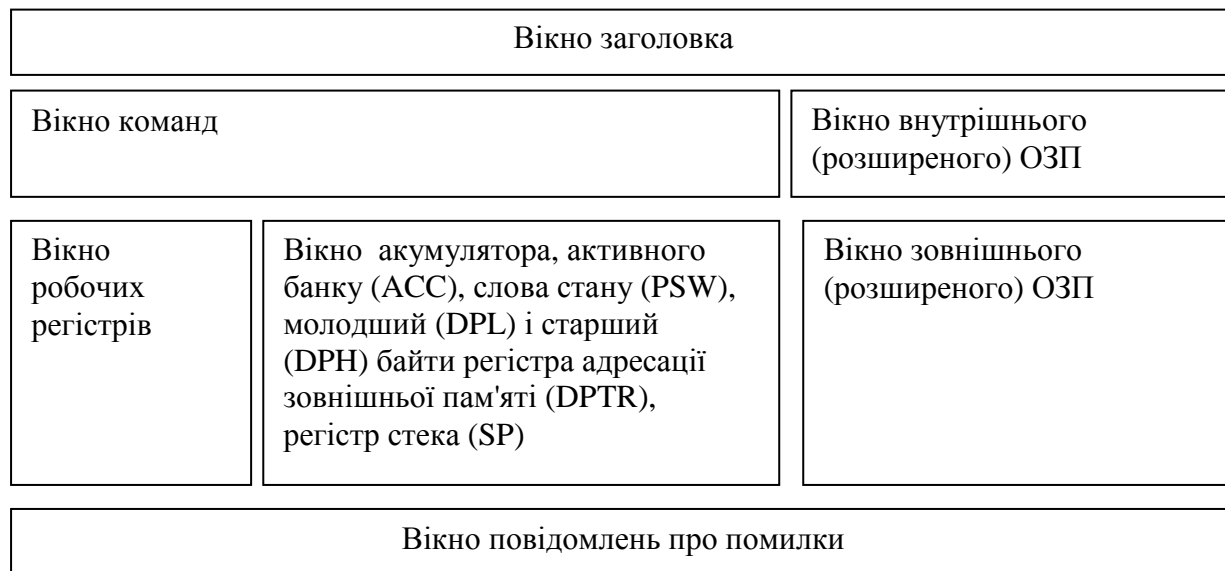


Рисунок 6.4 – Вікно програми “Монітор”

Програма «Монітор» для стенду МК51. Монітор є ядром програмного забезпечення стенду. У основу інтерфейсу обміну мікропроцесорного контролера (МК) стенду і ПЕОМ покладений протокол, який передбачає передачу спеціальної серії інформаційних повідомлень, необхідних для успішного функціонування робочої програми користувача. Алгоритм складається з наступних кроків:

Крок 1. Виконується ініціалізація контролера стенду, яка складається з наступних операцій:

- формування заборони всіх переривань;
- призначення стека монітора;
- заповнення ОЗП константою FF (очищення пам'яті);
- установка робочого банку даних (банк 3-18H..1FH);
- очищення ОЗП стану програми користувача (запис значення 0 в комірці ACCBuf, PSWBuf, DPTR);
- призначення стека користувача (DPTR:=#Stek1);
- установка адреси процедури збереження стану програми користувача (DPTR:=#Return);
- початкова установка коду в точці зупинки ([Kod]:=#02 код команди LJMP) і адреси повернення ([Kod+1]:=Return);
- ініціалізація порту послідовного обміну.

Алгоритм функціонування монітора представлений на рис. 6.5.

Таблиця 6.6 – Директиви програми “Монітор”

Код	Формат	Призначення	Примітки
L	LFName.hex	Ввод масиву даних з файлу	FName.hex – ім'я 16-річного файлу
G	GAAAA BBBB	Перехід до програми користувача із зупинкою	AAAA – стартова адреса BBBB – адреса зупинки
J	JAAAA	Запуск програми користувача без зупинки	AAAAA – стартова адреса
P	PAAAA	Продовження виконання програми від точки зупинки	AAAAA – адреса повторного запуску програми користувача
D	DAAAA BBBB	Читання масиву даних з ОЗП в ПЕОМ	AAAA – початкова адреса BBBB – кінцева адреса

Продовження табл. 6.6

K	KAAAA BBBB	Знаходження контрольної суми масиву даних	AAAA – початкова адреса BBBB – кінцева адреса
F	FAAAA BBBB CC	Заповнення масиву константою	AAAA – початкова адреса BBBB – кінцева адреса CC – константа
T	TAAAA BBBB CCCC	Копіювання масиву даних в ОЗП	AAAA – початкова адреса BBBB – кінцева адреса CCCC – початкова адреса копії масиву
M	MAAAA	Читання байта даних із зовнішнього ОЗП	AAAA – адреса байта даних
R	RAA	Читання байта даних з внутрішнього ОЗП	AA – адреса байта даних
Y	YDD	Запис байта даних за поточною адресою	DD – байт даних R0 – адреса призначення
A	A <масив>	Запис 32 байтів у внутрішнє ОЗП	60h – початкова адреса масиву
B	B	Читання 32 байтів з внутрішнього ОЗП	60h – початкова адреса масиву – джерела
Q	Q	Вихід з програми	

Крок 2. Виконуються запит обміну з ПЕОМ шляхом посилки в канал обміну коду запиту 00H і вводу коду директиви монітора;

Крок 3. Виконується аналіз коду директиви і відповідно до табл. 6.6. одна з директив монітора. Якщо одержано невірний код директиви, виконується перехід до кроку 2.

Виконання роботи. Частина 1. Відладка програми крос-засобами

Завдання. Записати в мнемонічному коді скореговану програму в роботі №2 у файл name.a51. Зробити трансляцію файлу в об'єктний і шістнадцятирічний коди. Перевірити роботу програми в середовищі програми-відладчика FD51. Занести в звіт зміст акумулятора.

Підготовка до роботи

1. Включити стенд. Всі клавіші стенду перевести у віджате положення, а потім включити комп'ютер.

2. Натиснути клавішу “Esc” (виконується завантаження комп'ютера Norton Commander (NC)). Ми працюємо в лівій панелі NC. Для цього, утримуючи клавішу <ALT>, натискаємо клавішу F1.

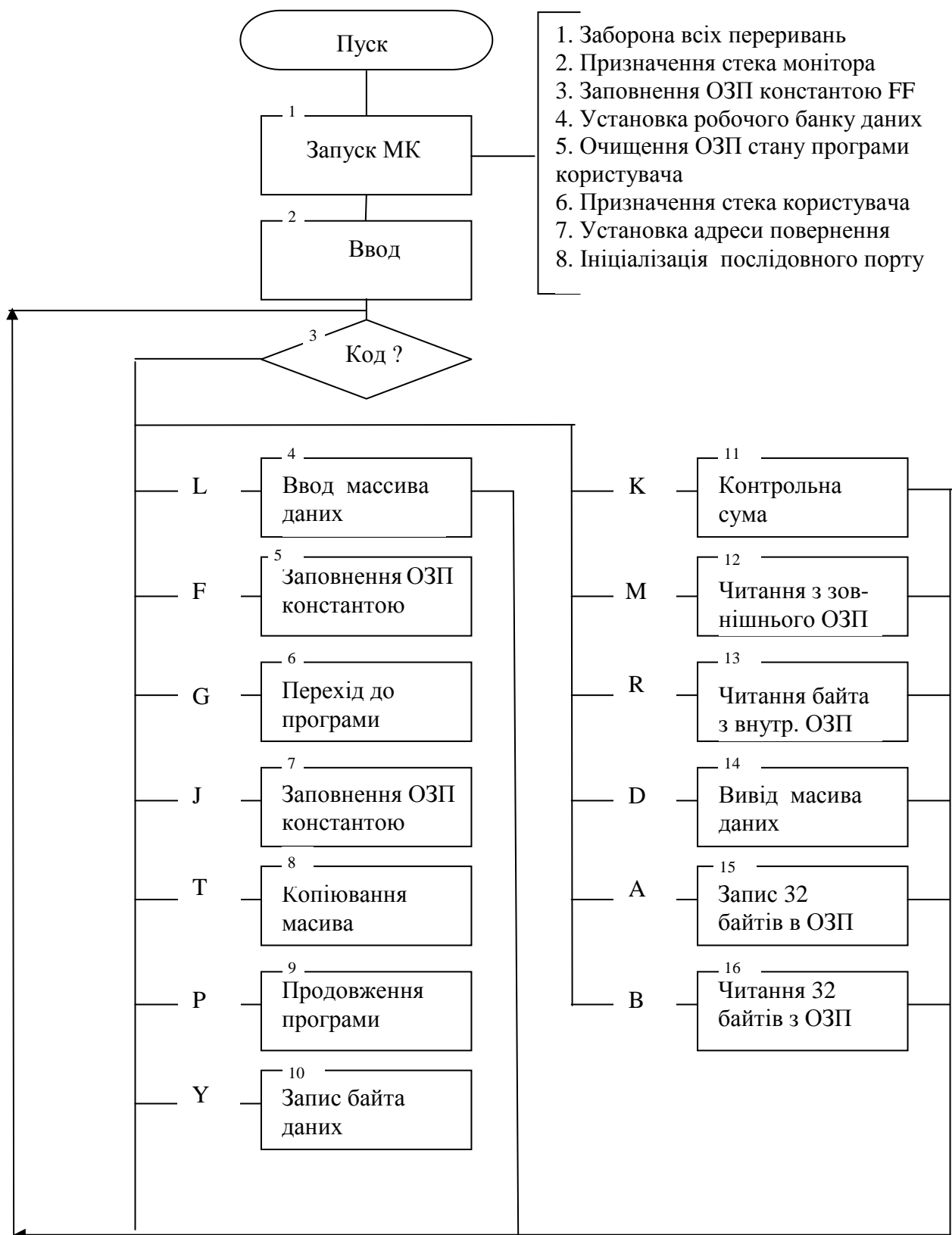


Рисунок 6.5 – Алгоритм функціонування монітора лабораторного стенду

У вікні дисків вибираємо клавішами управління курсором (<←> уліво, <→> управо) диск D (жовтий колір на D), а на деяких комп'ютерах диски C або E.

3. На вибраному диску, наприклад D, знаходимо директорію Lab_3 (D:\) (клавішами <↑>,<↓> наводимо курсор на директорію Lab_3 і, натиснувши клавішу <Enter>, потрапляємо в директорію D:\Lab_3.

4. Знаходячись в директорії D:\Lab_3, створимо новий файл. Для цього натискаємо комбінацію клавіш <Shift>+<F4> і у вікні редактора, яке з'явилося на екрані, введемо ім'я нового файлу name.a51 (де name – прізвище студента до 7 символів); натискаємо <Enter> і у відкритому вікні нового файлу набираємо програму в мнемонічному коді, побудовану в лабораторній роботі №2, скореговану таким чином. На початку програми записуємо директиву ORG 00H, потім команду SJMP 30H, за нею директиву ORG 30H, далі програму, яка використовувалася в лаб. № 2. Блок команд в кінці програми, який раніше використовувався для зациклення (NOP-SJMP), замінюється одним оператором повернення з підпрограми – RET. Це пояснюється тим, що будь-яка програма, завантажена з ПЕОМ по директиві “L”, сприймається “монітором” стенду як підпрограма. Як відомо, повернення з підпрограми виконується командою RET. Для збереження змін у файлі використовуйте клавішу < F2 >.

Виконання програми

1. Створену нову програму (файл name.a51) обробляємо програмою асемблер (файл ASM51.EXE). Для цього введемо в командній строчці CMD> команду ASM51 name.a51. Результатом виконання програми буде об'єктний файл (name.obj), представлений в машинних кодах, а також лістинговий файл name.lst, який містить текст початкової програми, її шістнадцятирічний код, відомості про помилки (якщо вони є) і службову інформацію.

2. Одержаний файл name.obj необхідно перекодувати в шістнадцятирічний файл, для чого, використовуючи команду OH.EXE, введемо в командну строчку OH name.obj і натиснемо <ENTER>. Результатом виконання цієї команди буде файл в шістнадцятирічному коді name.hex. Надалі виконання роботи ведеться саме з цим файлом (name.hex).

3. У директорії D:\LAB_3 знаходимо файл відладчика програм

FD51.EXE і, натиснувши клавішу <ENTER>, запускаємо його.

4. У відкритому вікні відладчика у вікні команд CMD> вводимо команду L name.hex і натискаємо <ENTER>. У вікні програми буде виведено дизасембльовану програму.

5. Натискаючи клавішу <F1>, крок за кроком виконаємо програму. Дійшовши до команди RET, зафіксуємо в звіті зміст акумулятора.

6. Вихід з програми відладчика проводиться командою QUIT, яка вводиться у вікні команд CMD>.

7. Результати роботи (файли з назвою name і з розширеннями a51, hex, lst, obj) зберегти в директорії D:\LAB_3.

Частина 2. Виконання програми і перевірка результатів

Підготовка до роботи. Перед виконанням ч. 2 необхідно скорегувати програму, складену в частині 1. Для цього досить замінити на початку програми в директивах ORG адреси з 00H на 8000H, а 30H на 8030H і в команді SJMP з 30H на 8030H. Подальше виконання роботи проводиться згідно п. 1-6 ч. 2.

Порядок виконання роботи

1. Користуючись програмою “Монітор” ПЕОМ, за допомогою директиви “F” заповнити адресний простір з 8000 h до 8050 h константою FF.

2. За допомогою директиви “L” завантажити hex-файл програми в ОЗП стенду за адресою 8000 h.

3. За допомогою директиви “D” з урахуванням початкової і кінцевої адрес вивести на екран текст завантаженої програми.

4. Користуючись директивою “J” з початковою адресою програми як параметром, запустити програму на виконання.

5. Перевірити індикацію порту P1 лабораторного стенду МК51.

6. Перевірити двійковий код вмісту порту P1, що висвітився на індикаторах (P0-P7). Він повинен відповідати інвертованому змісту акумулятора, який був зафіксований в звіті в першій частині роботи.

Зміст регістра A (відображається в 16-річному коді) і зміст порту P1 (відображається в двійковому коді) повинні бути інверсними.

Зміст звіту. Повний текст скорегованої програми з директивами і коментарями і результати розрахунку.

Контрольні питання

1. Структура програми на асемблері.
2. Функції відладчика.
3. Структура вікна і команди відладчика FD51.
4. Функції і структура вікна програми “Монітор”.
5. Директиви програми “Монітор”.
6. Алгоритм функціонування монітора лабораторного стенду.
7. Призначення програми ASM-51.EXE і результати її виконання.
8. Устрій, принцип дії і режими роботи УАПП.
9. Зміст файлів name.obj і name.lst.
10. Призначення команди OH name.obj і результати її виконання.
11. Який файл використовується при роботі з відладчиком програм?
12. Як запустити файл відладчика програм?

6.4. Лабораторна робота №4. Дослідження аналого-цифрового перетворювача

Мета роботи – ознайомитися з принципом дії аналогово-цифрового перетворювача (АЦП) і навчитися використовувати його можливості при складанні програм, що управляють, на асемблері МК51.

Підготовка до роботи: вивчити рекомендовану літературу [20,21,26], ознайомитися з принципом дії і правилами роботи з АЦП.

Використовуване устаткування: ПЕОМ, лабораторний стенд МК51, джерело живлення напругою 5В, вольтметр, потенціометр.

Основні відомості про аналогово-цифровий перетворювач (АЦП) і цифро-аналоговий перетворювач (ЦАП)

При використанні мікроконтролера для вирішення деяких завдань управління виникає необхідність проведення перетворення аналогових сигналів в цифрові і навпаки – цифрового коду в аналоговий сигнал. Для цього використовують АЦП і ЦАП. При виборі цих пристроїв враховують

розрядність, швидкодію і вартість. Перетворення аналогового сигналу в цифровий код, який може бути оброблений МК, можна здійснити декількома способами, що приводяться нижче.

1. *Апаратний*. Реалізований на основі ВІС АЦП, що підключається до порту МК. В цьому випадку МК тільки ініціює АЦП і через задані періоди дискретизації прочитує з нього цифровий код. Даний спосіб характеризується найвищою швидкістю, але вимагає використання ВІС АЦП, що не завжди виправдано економічно.

2. *Апаратно-програмний*. Реалізується на основі ВІС ЦАП і програми зважування біт (послідовних наближень побітового урівноваження). Даний спосіб характеризується високою швидкістю і можливістю використання простих і дешевих схем ЦАП і операційного підсилювача.

3. *Програмно-апаратний* – побудований на основі методу подвійної інтеграції. Найбільш дешевий і повільний. Може забезпечити високу точність перетворення. З додаткового устаткування вимагає два операційних підсилювача і аналоговий мультиплексор на два входи.

4. *Апаратно-програмний*. Реалізується на основі перетворювача напруга–частота і програми вимірювання періодів сигналу.

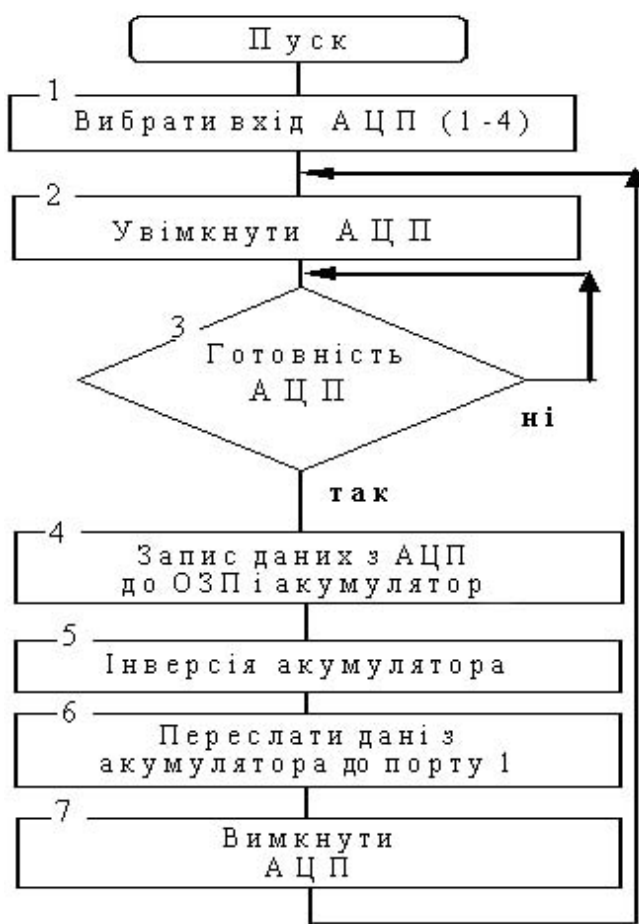
Після обробки одержаного сигналу МК видає сигнали управління в двійковому коді, які необхідно перед подачею на пристрій управління об'єктом перетворити на аналоговий сигнал. Для цієї мети використовують ЦАП. ВІС ЦАП підключають до одного з портів МК. Видача аналогового керуючого сигналу здійснюється по команді (OUTL P1, A). При цьому на виході ЦАП з'явиться напруга (струм), пропорційна двійковому коду, завантаженому в порт 1.

Ряд об'єктів управління може потребувати безперервного сигналу, що управляє, в складній формі. Для реалізації такого сигналу в МК використовуються цифрові методи інтеграції. На кожному інтервалі часу функція, яка інтегрується, замінюється її дискретним значенням, а програма реалізується з використанням процедур видачі коду і часової затримки заданої тривалості.

У цій роботі використовується апаратний спосіб перетворення аналогового сигналу в двійковий код. Аналого-цифрова частина зібрана на

мультиплексорі аналогових каналів КР560КН6 (ДА1), десятибітового АЦП К1113ПВ1 (ДА2), і карти читання даних з АЦП, зібраної на регістрі КР1533ИР33 (ДД). АЦП може працювати в двох режимах з різною точністю – 8 і 10 біт. При роздільній здатності 8 біт, якщо ведеться передача перетвореної інформації через послідовний порт, швидкість буде трохи вища. Перетворені в двійковий код дані поступають в порт P1 мікроконтролера.

Алгоритм прийому даних мікроконтролером від АЦП приведений на рис. 6.6, а текст цієї програми представлений нижче.



Програма прийому даних
МК від АЦП

```

AJMP AZP
.ORG 030H
AZP:
P1 .EQU 090H
P3.3 .EQU 0B3H
P3.4 .EQU 0B4H
LOOP:
CLR P3.4
WAIT:
JB P3.3,WAIT
MOV R0,#C000H
MOVX A,@R0
MOV P1,A
SETB P3.4
SJMP LOOP
END
  
```

Рисунок 6.6 – Алгоритм прийому даних мікроконтролером від АЦП

Для вибору номера входу треба занести код входу в регістр, який керує номером входу АЦП. Адреса регістра – A000h.

Коди входів АЦП:

Вхід 1 (0) 00В 00Н Вхід 3 (2) 10В 02Н

Вхід 2 (1) 01В 01Н Вхід 4 (3) 11В 03Н

Порядок виконання роботи. Для виконання завдання необхідно зібрати схему, приведену на рис. 6.7, в якій подаватиметься сигнал (постійна напруга від 0 до 5 В) на вхід АЦП.

Схема складається з джерела живлення напругою 5 В, потенціометра і вольтметра, який підключений паралельно до входу АЦП. Вольтметр використовується для еталонного контролю напруги сигналу.

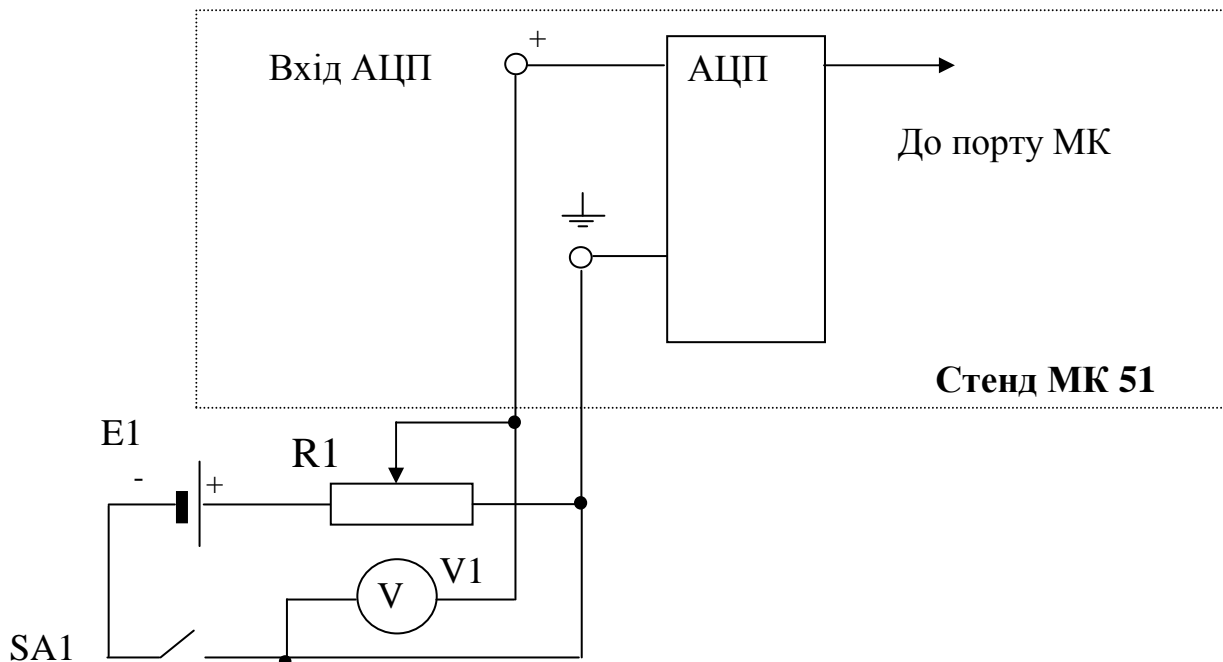


Рисунок 6.7 – Схема емулятора вхідного сигналу для АЦП

Після того, як схема підключена, в ОЗП стенду з персонального комп'ютера необхідно завантажити програму для роботи з АЦП. Програма в шістнадцятиричному коді знаходиться у файлі `azp.hex`, і після завантаження в ОЗП повинна читати сигнал з входу АЦП і передавати його в порт P1. При цьому відповідно до зміни напруги на вході АЦП (змінюється потенціометром) змінюватиметься і зміст порту P1, що відобразиться на світлодіодах P0–P7. Для завантаження програми треба перевести стенд в режим роботи з ПЗП, а далі діяти по схемі лабораторної роботи №3 (ч.2): запустити програму “monitor” і за допомогою директиви “L” завантажити файл `azp.hex`. Після того, як програма завантажена в ОЗП стенду, його треба

перевести в режим роботи без ПЗП і вимкнути режим скидання.

За допомогою директиви J8030 програма запускається на виконання, після чого на індикаторах порту P0–P7 у режимі реального часу починає відображатися стан входу АЦП.

Завдання 1. Розрахувати розрядність цифрової індикації порту P1.

Розв'язання. Для вирішення цього завдання необхідно, 5–10 вибраних значень напруги записати в таблицю стану світлодіодів (зміст порту P1) і розрахувати вагомість розряду порту P1 по формулі:

$$C=U/P1,$$

де C – відношення, яке характеризує величину зміни напруги на вході АЦП, що викликає зміну сигналу на виході (у порту P1) на одну одиницю;

U – напруга на вході АЦП, показана вольтметром V1;

$P1$ – зміст порту P1 у десятковому коді.

Зміст звіту:

- схема підключення джерела сигналу до входу АЦП (рис.6.7);
- таблиця залежності сигналів в порту P1 від сигналів на АЦП;
- розрахунок вартості розряду індикації порту.

Контрольні питання

1. Викладіть принцип дії схеми (рис. 6.7) і порядок виконання лабораторної роботи.
2. На чому засновані принципи дії АЦП і ЦАП ?
3. Поясніть алгоритм програми прийому даних мікроконтролером з АЦП.
4. Складіть програму прийому даних з АЦП.
5. Вкажіть способи перетворення аналогового сигналу в цифровий код і назад.
6. Поясніть структурну схему МК51, що приводиться нижче на рисунку 6.8 і її роботу при зборі інформації з аналогових датчиків.
7. Назвіть призначення виводів мікросхеми МК51, приведену на рисунку 5.1, і поясніть схему підключення АЦП до МК51.
8. Поясніть структуру карти адресації бітів в резидентній пам'яті даних, представлену нижче на рис. 6.9.

9. Поясніть структуру карти адресації бітів в блоці реєстрів спеціальних функцій, приведену нижче на рис. 6.10 і призначення реєстрів спеціальних функцій.

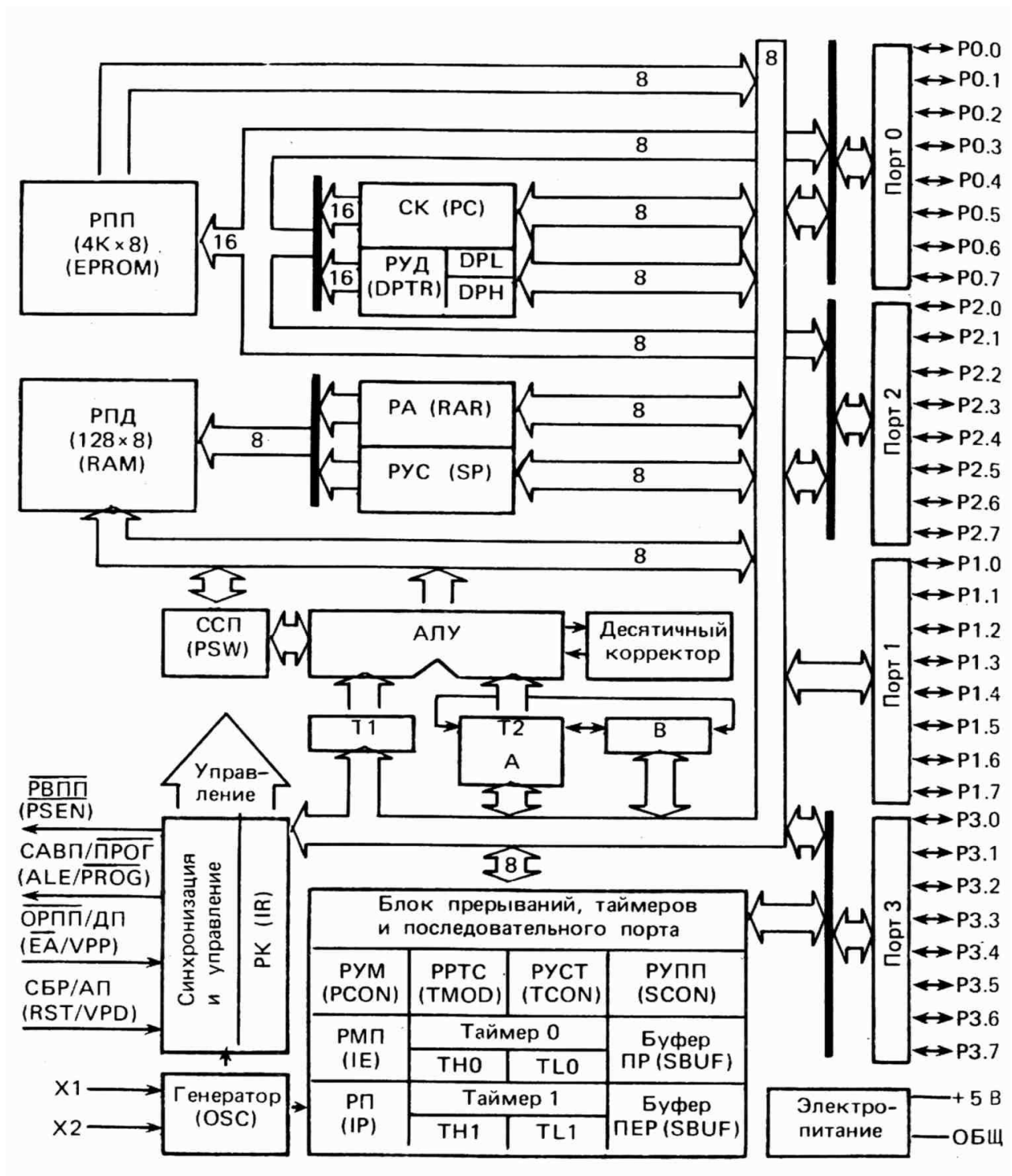


Рисунок 6.8 – Структурна схема МК51

Карти адресації біт

Адрес РПД (D7)									(D0)
7FH									
2FH	7F	7E	7D	7C	7B	7A	79	78	
2EH	77	76	75	74	73	72	71	70	
2DH	6F	6E	6D	6C	6B	6A	69	68	
2CH	67	66	65	64	63	62	61	60	
2BH	5F	5E	5D	5C	5B	5A	59	58	
2AH	57	56	55	54	53	52	51	50	
29H	4F	4E	4D	4C	4B	4A	49	48	
28H	47	46	45	44	43	42	41	40	
27H	3F	3E	3D	3C	3B	3A	39	38	
26H	37	36	35	34	33	32	31	30	
25H	2F	2E	2D	2C	2B	2A	29	28	
24H	27	26	25	24	23	22	21	20	
23H	1F	1E	1D	1C	1B	1A	19	18	
22H	17	16	15	14	13	12	11	10	
21H	0F	0E	0D	0C	0B	0A	09	08	
20H	07	06	05	04	03	02	01	00	
1FH	Банк 3								
18H	Банк 2								
17H	Банк 1								
10H	Банк 0								
0FH	Банк 0								
08H	Банк 0								
07H	Банк 0								
00H	Банк 0								

Рисунок 6.9 – Карта адресації бітів у резидентній пам'яті даних (РПД)

Прямий адрес біта (D7)									(D0)	Имя регистра
0FFH										
0F0H	F7	F6	F5	F4	F3	F2	F1	F0	B	
0E0H	E7	E6	E5	E4	E3	E2	E1	E0	A	
0D0H	D7	D6	D5	D4	D3	D2	D1	D0	PSW	
0B8H	-	-	-	BC	B8	BA	B9	B8	IP	
0B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3	
0A8H	AF	-	-	AC	AB	AA	A9	A8	IE	
0A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2	
98H	9F	9E	9D	9C	9B	9A	99	98	SCON	
90H	97	96	95	94	93	92	91	90	P1	
88H	8F	8E	8D	8C	8B	8A	89	88	TCON	
80H	87	86	85	84	83	82	81	80	P0	

Рисунок 6.10 – Карта адресації бітів у блоці регістрів спеціальних функцій

7. ЗАСТОСУВАННЯ МІКРОКОНТРОЛЕРІВ

В даний час мікропроцесорні пристрої широко застосовуються для вирішення самих різних завдань практично у всіх областях діяльності людини. Можливі області застосування МК пов'язані з їх архітектурою і розрядністю:

- 8-розрядні мікроконтролери сімейства МК51 застосовуються для управління різними технологічними процесами у виробництві, випробуваннях і дослідженнях устаткування, в пристроях управління транспортом, в службовій, комерційній, торговій і побутовій апаратурі, пристроях автоматичного управління вимірювальними приладами;
- 16-розрядні МП використовуються в системах зв'язку, збору і обробки інформації, контрольно-розподільних системах, в системах навігації, аналого-цифрових перетворювачах, в мікроЕОМ широкого призначення;
- 32-розрядні застосовуються в мікро- і міні-ЕОМ широкого призначення, МПС цільового призначення, спеціалізованих процесорах, цифрових фільтрах, автокореляторах.
- 64-розрядні - в мультимедійних, в 2D і 3D графічних системах.

7.1. Застосування мікроконтролерів в електромеханічних системах

В даний час МК широко використовуються в управлінні різними електромеханічними системами, використовуваними на транспорті, в енергетиці, різних галузях промисловості, електропобутовій техніці і т.ін.

У [21] приведені класифікація і опис завдань електромеханіки і транспорту, які розв'язуються із застосуванням мікроконтролерних пристроїв. Там же приведені приклади застосування мікроконтролерів на транспорті. Так, для управління потужними транспортними дизельними установками від 250 до 3000 кВт і більше використовується електронний регулятор фірми –«Хайнцман». Там же вказані найбільш відомі фірми-виробники таких регуляторів, як, «Бош», «Хайнцман», «Барбет», «Вудвард» і

«Колман». Основою регуляторів такого класу є швидкодіючий, потужний МП. Власна програма регулятора зберігається в постійній FLASH пам'яті.

Регулятор визначає робочий стан дизеля по сигналах, які поступають від різних датчиків. У двигуна можуть контролюватися температура охолоджуючої рідини, тиск масла, число оборотів і інші параметри [21].

У [27] приведений приклад і функціональна схема з використанням мікроконтролера в структурі системи *управління процесом рекуперативного гальмування*, побудованої для електрорухомого складу постійного струму, експлуатованого на залізницях України. Ця система дозволяє істотно зменшити споживання енергії.

Приведена [21] система управління технологічним процесом без вводу ЕОМ в контур регулювання, в якій керуюча програма знаходиться в ЕОМ, забезпечує задану послідовність виконання технологічних операцій, задану тривалість їх виконання в часі, темп прискорення і уповільнення, обробку сигналів від технологічних датчиків, а також датчиків аварійних ситуацій з відповідним відключенням. Виконавчим органом є електродвигун з його власною системою управління.

Там же приведені приклади застосування МК в системах управління технологічними процесами з частковим і повним включенням ЕОМ в контури регулювання технологічних параметрів і електроприводу.

Для реалізації системи векторного управління асинхронними і синхронними електроприводами, такими фірмами як «SIEMENS», «ABB», «MITSUBISHI» і ін. використовуються спеціалізовані мікросистеми фірм «TEXAS INSTRUMENT» або аналог «DEVICES» [28].

Для цифрового управління електродвигунами і для вирішення ряду інших завдань фірма «TEXAS INSTRUMENT» розробила сімейство цифрових сигнальних процесорів (ЦСП) TMS320C20x (платформа C2000) з розвиненою периферією і невисокою вартістю. Розроблені фірмою три платформи – C2000, C5000, C6000 за запатентованою технологією виробництва кремнію TimeLine™ з роздільною здатністю 0,18 мкм дозволяють забезпечити весь діапазон можливих застосувань ЦСП. Цим забезпечується можливість найширшого вибору процесора по критерію “продуктивність/вартість/споживана потужність”. Ці мікросистеми

дозволяють реалізувати складні обчислювальні алгоритми з необхідною швидкістю. Інтервал повторення основних обчислень складає 25 мкс. Платформа C5000 орієнтована на застосування в портативних пристроях і в мобільному зв'язку. Використання 0,18 мкм технології дозволяє досягти продуктивності (до 800 MIPS) при зниженні енергоспоживання до 0,05 мВт/MIPS і вартості окремих ЦСП до 5\$.

Платформа C6000 покликана забезпечити максимальну продуктивність в системах управління, що вимагають граничних швидкостей обчислень як з фіксованою, так і плаваючою крапками. Вона включає дві гілки 32-розрядних ЦСП з фіксованою і плаваючою крапками. До сімейств процесорів з фіксованою крапкою відносяться C62x з продуктивністю від 1200 до 2400MIPS і нові процесори сімейства C64x з продуктивністю від 4800 до 8800 MIPS. Процесори сімейства C67x відносяться до пристроїв з плаваючою крапкою і продуктивністю до одного мільярда операцій з плаваючою крапкою в секунду (1 GFLOPS) при тактовій частоті 167 МГц. Висока продуктивність процесорів цієї платформи досягається за рахунок нової архітектури Velocity™ з “дуже довгим командним словом” (VLIW) і з новітніми апаратними рішеннями. На виконання видається вісім 32-розрядних команд, кожна з яких виконується одним з восьми незалежних функціональних пристроїв, згрупованих в 2 блоки. Процес виконання команд конвеєризований і розпадається на етапи вибірки, розпаковування, декодування і виконання. C6000 застосовується в модемних пулах, базових станціях і відеосистемах.

7.2. Застосування мікроконтролерів в електроапаратобудуванні

МП пристрої використовуються в управлінні електричними апаратами, комплектними розподільними пристроями (КРП), технологічними процесами виробництва, випробувань і досліджень, гнучкими системами релейного захисту (ГСРЗ) і т.ін. [29, 30, 31, 32]. Нижче наводяться приклади використання МП в комплектних розподільних пристроях з високовольтними електричними апаратами [31], в системах автоматичного управління

процесами випробувань і досліджень електричних апаратів [29, 30] і в ГСРЗ [32].

У [31] приведено КРП з мікропроцесорним управлінням і релейним захистом, що випускає АТ “Рівненський завод високовольтних апаратів” (РЗВА). У шафах КРП серії КУ-10Ц як основну комплектуючу апаратуру застосовують:

- високовольтний вакуумний вимикач ВВКЭ-10 (VMIS);
- трансформатор струму ТЛК-10,ТВЛ;
- трансформатори струму нульового захисту ТЗЛМ;
- трансформатори напруги ЗНОЛ-06, НОЛ-08, НАМИ;
- обмежувачі перенапруження ОПНС, розрядники РВО;
- запобіжники силові ПКНТ;
- запобіжники трансформаторів напруги ПКН.

КРП серії НКАИ670049.003 з мікропроцесорним управлінням мають порівняно з КРП інших серій аналогічного призначення ряд переваг, а саме:

1. Наочність процесу роботи КРП за рахунок більшої кількості вимірювань і сигналізації, а також показу інформації на динамічних екранах, які дають можливість оператору своєчасно реагувати для запобігання аварії;

2. Дистанційне керування, як терміналами релейного захисту, так і первинним устаткуванням підстанцій (порівняно з місцевим управлінням у разі використання традиційного устаткування);

3. Постійну діагностику устаткування, що дозволяє проводити передаварійну профілактику устаткування (порівняно з поставарійним або періодичним технічним обслуговуванням традиційного устаткування);

4. Можливість покрокового нарощування системи, як релейного захисту, так і систем вимірювання і управління, зміни їх функцій шляхом перепрограмування;

5. Можливість реєстрації і збереження всіх величин, контрольованих параметрів в передаварійних і аварійних режимах роботи, що дозволяє провести точний поставарійний комп'ютерний аналіз причин аварії (така можливість повністю відсутня в традиційному устаткуванні);

6. Можливість реалізації ряду допоміжних функцій управління і контролю.

Приведені вище переваги, забезпечуються застосуванням в новому КРП мікропроцесора типу « SPAC» фірми АББ – Чебоксари (Росія) або REF-542 АБВ. Докладніший виклад особливостей застосування МП « SPAC» і REF-542 АБВ в КРП приведений у [31].

Застосування МП в гнучких системах релейного захисту (ГСПЗ) забезпечує ефективне запобігання і (або) локалізацію аварій [32].

Кінцевою метою функціонування релейного захисту (РЗ) є забезпечення безаварійності об'єктів захисту (ОЗ) (електричних станцій, ліній електропередачі, електроенергетичних установок і т.п.), тобто можливості системи РЗ шляхом відключення ОЗ своєчасно запобігати розвитку аварійних ситуацій, небезпечних для устаткування і обслуговуючого персоналу. Попереднє покоління пристроїв РЗ було створено на базі електромеханічних реле, напівпровідникових елементів і аналогових інтегральних мікросхем (ІМС). Створені на основі таких непрограмованих елементів, вони функціонально є кінцеві автомати другого роду з незмінною (жорсткою) архітектурою і знаходять застосування в даний час при реалізації простих алгоритмів виявлення пошкоджень [32]. На відміну від них *ГСПЗ володіють можливістю* перепрограмування на реалізацію тих або інших функцій без зміни складу комплексу технічних засобів і реалізації алгоритмів виявлення пошкоджень підвищеної складності з використанням принципів адаптації і автоматизації процесів діагностики і настройки апаратури. Це дозволяє понизити збиток від пошкодження ОЗ, підвищити якість електроенергії і скоротити витрати на обслуговування, контроль, розробку і проектування РЗ. Наприклад, застосування МП релейного захисту на базі МП серії К589 дозволяє забезпечити комплексний захист генератора (КЗГ) автономної електростанції [32]. Вказаний захист здійснює виявлення перевантажень первинного двигуна (наприклад, турбіни) генератора, зовнішніх і внутрішніх коротких замикань (КЗ), зниження напруги, переходу генератора в руховий режим. Перевантаження генератора, що викликає перегрів його обмоток, виявляється по параметру “квадрат діючого значення струму”. КЗГ контролює струми всіх трьох фаз. Допустима тривалість

перевантаження визначається фазою з максимальним значенням струму. Цей же інтегральний параметр, використовується для захисту від зовнішніх КЗ. Перевантаження приводного двигуна виявляється по двох параметрах – частоті напруги на виведеннях генератора і активної потужності, що віддається ним. Якщо зниження частоти не перевищує заданого значення, допустима тривалість перевантаження визначається тільки активною потужністю. Для забезпечення можливості збереження генератора в роботі у разі перевантажень в комплексному захисті генератора (КЗГ) передбачені два ступені розвантаження (шляхом відключення невідповідальних споживачів). Зниження напруги виявляється по параметру “квадрат діючого значення напруги”, а перехід генератора в режим двигуна (що є неприпустимим навантаженням для паралельно працюючих з ним генераторів), по параметру “зворотна активна потужність”. Детальніше питання застосування МП і МК в ГСРЗ викладені в [32].

7.3. Застосування МК51 при випробуваннях електричних апаратів захисту

Аналіз вимог, що пред'являються до електричних апаратів захисту (автоматичних вимикачів (АВ) і швидкодіючих запобіжників (ШЗ) і ін.) і методів їх випробувань, указує на вельми широкий перелік параметрів, які повинні перевірятися і досліджуватися при проведенні різних випробувань і досліджень. До таких параметрів відносяться: номінальний струм, струм перевантаження, струм короткого замикання, напруга на дузі, Джоулевий інтеграл, час відключення, температура на виводах, температура в центрі плавкого елемента, температура контактів, швидкість руху дуги в дугогасних решітках і т.ін. Все це указує на необхідність використання вельми широкого спектру, відповідних датчиків, що дозволяють з необхідною точністю відстежувати зміну цих параметрів в процесі досліджень [29].

При проведенні комутаційних досліджень на постійному струмі таких електричних апаратів, як швидкодіючі запобіжники, автоматичні вимикачі та інші використовуються експериментальні установки, які включають головний ланцюг і ланцюг управління. Схема однієї із таких установок

представлена на рис. 7.1. Головний ланцюг установки складають ударний генератор (УГ) ($U_n = 880 \text{ В}$, $I_{уд} = 70 \text{ кА}$), регульовані реактори L , регульований опір R_a , захисний вимикач (ЗВ), вмикаючий апарат (ВА), макет апарату (МА). Проведення досліджень здійснюється за допомогою пульта електронного управління (ПЕУ) і електромеханічного або електронного осцилографа (ЕО).

Вимірювання струмів проводиться за допомогою шунта з опором $R_{ш} = 0,7 \cdot 10^{-5} \text{ Ом}$. Напряга на дузі вимірюється по схемі дільника напруги. Криві струму і напруги в стандартних експериментах записувалися на світлочутливий папір за допомогою світлопроменевого осцилографа. В цьому випадку обробка осцилограм проводиться графічним методом, що вимагає великих трудових витрат і часу і знижує точність вимірювань.

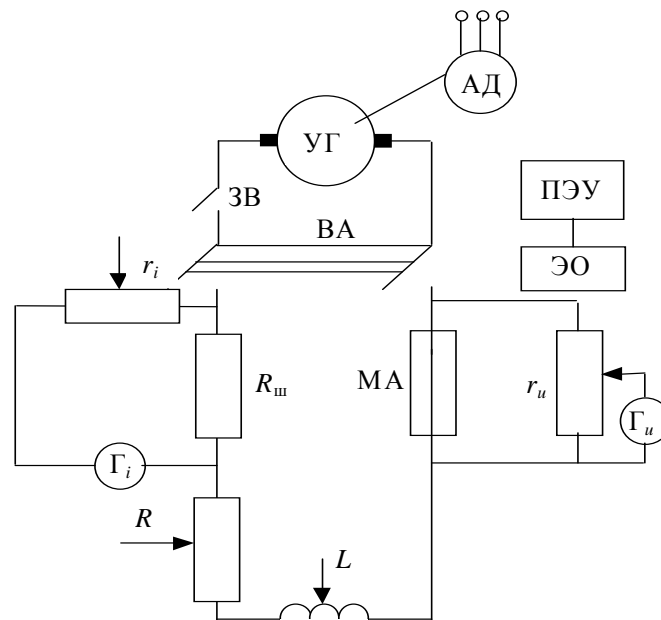


Рисунок 7.1 – Схема експериментального стенду для комутаційних досліджень електричних апаратів

При дослідженні маловивчених процесів для того, щоб забезпечити реєстрацію можливих гострих піків перенапруження, для запису кривих струму і напруги необхідно використовувати електронний осцилограф і здійснювати фотографування за допомогою фотоприставки. У випадках, коли потрібна підвищена точність, обробка осцилограми проводиться за допомогою вимірювального мікроскопа. Все це також приводить до

додаткових матеріальних, тимчасових і трудових витрат. Скоротити терміни проведення комутаційних досліджень, підвищити точність вимірювань, понизити їх вартість можна, застосувавши розроблену і викладену в [29] автоматизовану систему управління технологічним процесом досліджень (АСУ ТПВ) із застосуванням однокристального мікроконтролера.

Структурна схема АСУ ТПВ, представлена на рисунку 7.2, виконана на базі мікроконтролера серії МК1816ВЕ51 [29]. Схема включає:

- датчики контрольованих параметрів (струму, напруги, температури, Джоулевого інтеграла) Д1–Д4 (первинні перетворювачі);
- нормуючі підсилювачі У1–У4;
- чотирьохканальний комутатор аналогових сигналів типу КР590КИ6;
- аналого-цифровий перетворювач типу К1113ПВ1;
- мікроконтролер, що містить вбудований генератор тактових сигналів, пам'ять команд, ОЗП, вбудовані 3 порти і послідовний канал зв'язку;
- компаратори К1–К4 типу К554 СА3, виходи яких по «АБО» об'єднані з вихідними сигналами мікроконтролера, що управляють;
- пристрої узгодження і обміну УСО1-УСО4, які включають виконавчі пристрої силової установки, які задають режим випробувань або досліджень.

Через послідовний інтерфейс RS232C АСУТПВ пов'язана з ПЕОМ, яка може змінювати режими випробувань або досліджень, а також приймати, запам'ятовувати, відображати і документувати результати випробувань або досліджень.

До об'єкту дослідження підключені відповідні датчики. Датчики контрольованих параметрів Д1–Д4 є первинними перетворювачами струму, напруги, температури, Джоулевого інтеграла в напругу. Нормуючі підсилювачі погоджують вихідну напругу датчиків з необхідним вхідним сигналом АЦП 0–10 В і забезпечують низький вихідний опір.

Комутатор аналогових сигналів перемикає один з входів на вихід залежно від керуючого коду, що поступив від мікроконтролера.

АЦП є швидкодіючим десятирозрядним перетворювачем вхідної

напруги в паралельний двійковий код. Запуск перетворювача проводиться мікроконтролером, закінчення перетворення викликає сигнал готовності, який є командою для прочитування даних.

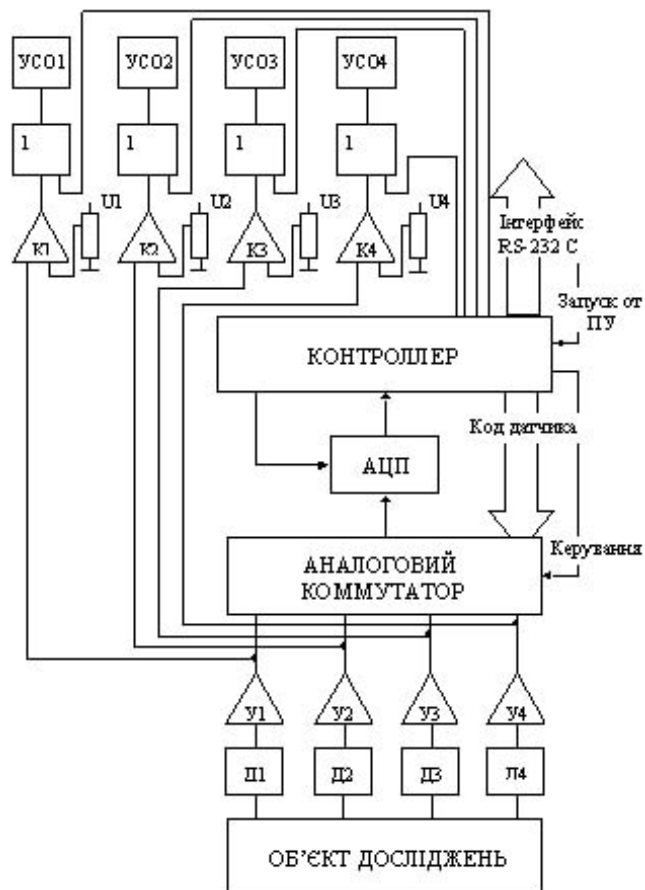


Рисунок 7.2 – Структурна схема автоматизованої системи управління технологічним процесом досліджень електричних апаратів

Мікроконтролер як мікропроцесорний пристрій, відповідно до записаної в пам'ять програми, управляє процесом досліджень або випробувань шляхом опиту із заданою періодичністю датчиків Д1–Д4 відповідно до алгоритму управління. Вихідні сигнали датчиків унаслідок їх різної фізичної природи можуть потребувати посилення і проміжного перетворення на АЦП або на схемах формувачів сигналів (ФС), які найчастіше виконують функції гальванічної розв'язки і формування рівнів двійкових сигналів стандарту ТТЛ. Мікроконтролер з необхідною періодичністю оновлює слова, що управляють, на своїх вихідних портах.

Деяка частина слова, що управляє, може інтерпретуватися як сукупність прямих двійкових сигналів управління (СУ), які через схеми формувачів сигналів, (підсилювачі потужності, реле, оптрони і т.ін.) або пристрої зв'язку з об'єктом (УСО1–УСО4) поступають на виконавчі механізми (ВМ). Компаратори К1–К4 є паралельним апаратним контуром для захисту від аварійних режимів. УСО1–УСО4 є підсилювачі потужності, які управляють виконавчими пристроями ВМ силової установки.

Обґрунтування вибору мікропроцесорної системи приведене на прикладі для досліджень швидкодіючих запобіжників.

Мікропроцесор для описаної вище системи вибирається виходячи з характеру досліджуваних процесів і умов досліджень:

- швидкості протікання процесів;
- кількості досліджуваних параметрів і частоти опиту датчиків;
- завдань по переробці інформації;
- умов експлуатації і вимог по надійності.

Аналіз вихідних даних цього завдання показує, що його рішення може бути здійснено на базі мікроконтролера серії МК51.

Система на базі цього мікроконтролера здатна опитувати датчики з частотою 100 мкс, тобто за час відключення запобіжника $t_{відк} \leq 10$ мкс система встигне опитати датчики 100 разів, чого цілком достатньо для зняття і побудови характеристик запобіжника з необхідною точністю.

Найбільш прийнятним для дослідження характеристик швидкодіючих запобіжників і автоматичних вимикачів є мікроконтролер типу МК1816ВЕ51, що має наступні технічні показники:

- тип – паралельний;
- розрядність паралельно оброблюваної інформації – 8 двійкових розрядів;
- форма представлення чисел – двійковий додатковий код;
- методи адресації – регістрова, пряма, непряма-регістрова безпосередня;
- одиниця, що адресується, – байт;
- кількість команд – 111, включаючи команди арифметичних і

логічних операцій, стекових операцій, складання слів двійкової довжини, операції управління;

- формат команд – однобайтна, двобайтова, трьохбайтна;
- час виконання команд – 1–4 мкс;
- 32 РОН і набір регістрів спеціальних функцій;
- 128 визначуваних користувачем програмно-керованих прапорів;
- послідовний інтерфейс;
- чотири 8-розрядні програмовані канали вводу-виводу;
- два 16-бітові багаторежимні таймери/лічильники;
- система переривання з п'ятьма векторами і двома рівнями з

програмною установкою пріоритету;

- місткість внутрішнього ОЗП – 128 байтів, ПЗП – 4 кбайта.

Діалог з МК здійснюється за допомогою послідовного інтерфейсу RS-232C через ПЕОМ або пульт управління.

Важливою особливістю арифметико-логічного пристрою (АЛП) мікроконтролера сімейства МК51 є його здатність оперувати не тільки байтами, але і бітами. Окремі програмно-доступні біти можуть бути встановлені, скинуті, інвертовані, передані, перевірені і використані в логічних операціях. Це дозволяє при керуванні об'єктами часто застосовувати алгоритми, що містять операції над вхідними і вихідними булевими змінними. АЛП являє собою паралельний 8-розрядний пристрій, що забезпечує виконання арифметичних і логічних операцій, а також операцій зсуву, обнуління тощо. АЛП може оперувати чотирма типами інформаційних об'єктів: булевими (біт), цифровими (4 біта), байтними (8 біт) і адресними (16 біт). У АЛП виконується 51 різна операція пересилки або перетворення цих даних. Оскільки використовується 11 режимів адресації (7 для даних і 4 для адрес), то шляхом комбінування «операція/режим адресація» базове число команд 111 розширюється до 255. В АЛП реалізується механізм каскадового виконання простих операцій для реалізації складних команд, наприклад таких як команда умовної передачі управління за результатами порівняння.

Робота схеми здійснюється по розробленому алгоритму і програмі. Алгоритм роботи схеми представлений на рис. 7.3.

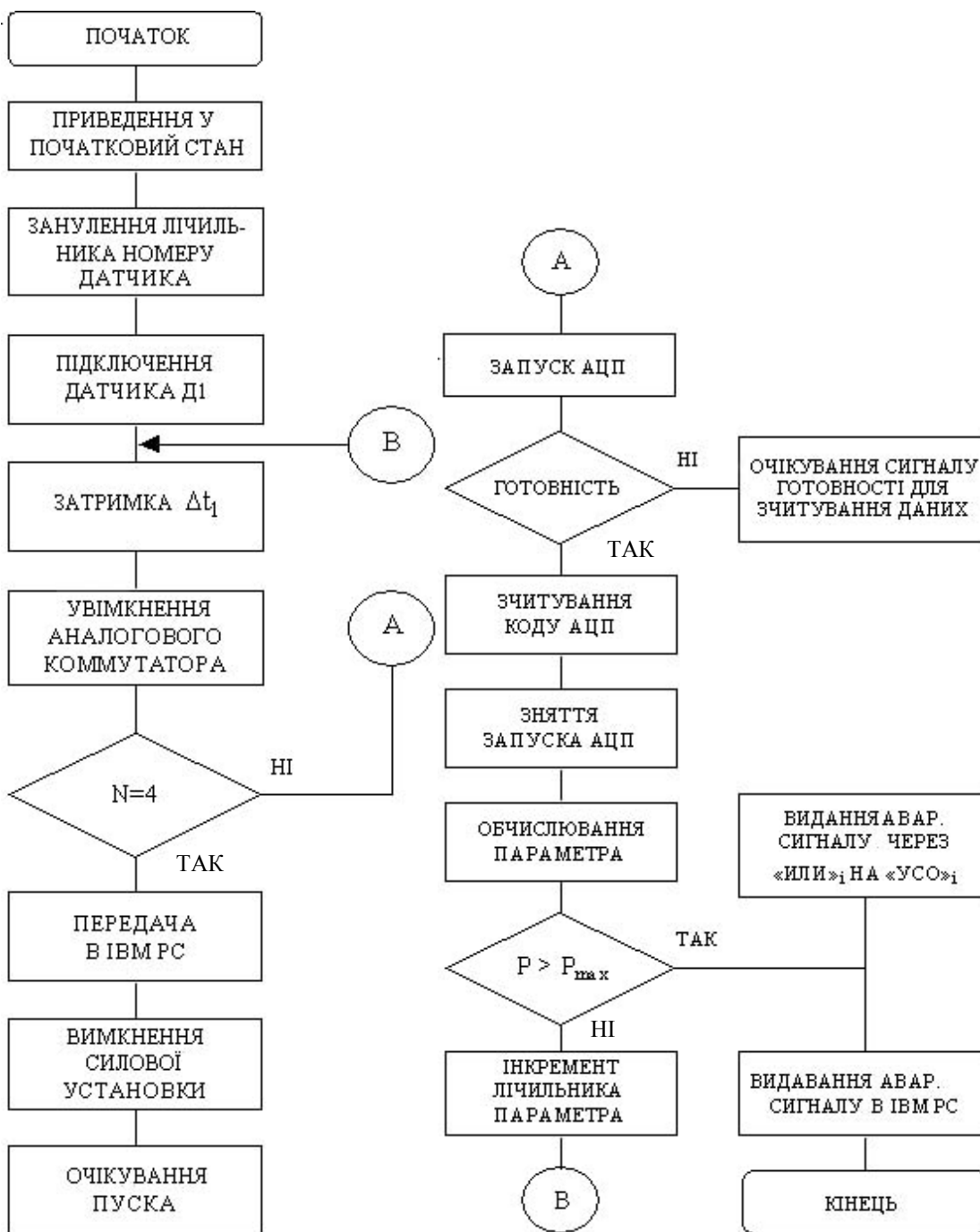


Рисунок 7.3 – Алгоритм роботи схеми автоматизованої системи управління технологічним процесом дослідження швидкодіючих запобіжників

Розроблена АСУ ТПВ дозволяє скоротити терміни проведення досліджень, підвищити достовірність і економічну ефективність.

7.4. Структурна схема АСУ ТПВ з паралельними АЦП

При дослідженні багатьох різних за природою фізичних процесів (горіння дуги, нагрів, електродинамічні зусилля, електромагнітне поле і ін.), що протікають при комутації в електричних апаратах захисту виникає необхідність у визначенні їх характеристик і параметрів. Ці процеси при відключенні аварійних струмів вельми короткочасні і мають тривалість від 1 до 10 мс. Для дослідження таких процесів можуть бути використані установки, приведені в [29, 33, 34], схема однієї з яких описана в п. 7.2 і приведена на рис. 7.1. Управління і проведення досліджень в таких установках здійснюється за допомогою пульта електронного управління і електромеханічного або електронного осцилографа, що надалі при графічній обробці осцилограм приводить до погрішностей, додатковим матеріальним, часовим і трудовим витратам і стають малоефективними. Запропонована вище в п. 7.3 (рис.7.2) структурна схема АСУ ТПВ на базі МК51 з послідовним АЦП у ряді випадків також не забезпечує ефективного рішення цих задач зважаючи на низьку швидкодію МК, велике число датчиків і частоти їх опиту. Це завдання можна вирішити, використовуючи схему АСУ ТПВ з паралельними АЦП, включивши до неї швидкодіючий МК серії MCS251 [30].

Для захисту електроустановок в аварійних режимах найширше використовуються такі електричні апарати захисту, як автоматичні вимикачі і швидкодіючі запобіжники. Тому розробку структурної схеми АСУ ТПВ доцільно виконати для одного з цих апаратів захисту. АСУ ТПВ для інших апаратів може відрізнятися тільки кількістю і найменуваннями контрольованих параметрів і відповідними датчиками, які будуть підключені до досліджуваного апарату.

Розглянемо на прикладі технічного завдання на створення мікроконтролерної системи управління стендом для випробувань швидкодіючих запобіжників з нижче наступними початковими даними: кількість контрольованих параметрів (датчиків) – 6, зокрема:

- струм (захист);

- напруга (захист);
- температура в центрі плавкого елемента (захист);
- температура на виводах;
- Джоулевий інтеграл і інтеграл горіння дуги;
- тривалість одного опиту – 4 мкс;
- кількість розрядів перетвореної інформації – 8;
- зовнішній інтерфейс обміну – RS232C;
- кількість опитувань датчиків (не менше) – 1000.

Вирішити це завдання і скоротити терміни проведення комутаційних досліджень, підвищити точність вимірювань, понизити їх вартість дозволяє АСУ ТПВ з паралельними АЦП, яка розроблена на базі високопродуктивного МК сімейства MCS251 8XC251SB і представлена на рис. 7.4. Схема АСУ ТПВ включає:

- датчики контрольованих параметрів (струму, напруги, температури в центрі і на виводах, Джоулевого інтеграла, інтеграла горіння дуги) Д1-Д6;
- нормуючі підсилювачі У1-У6;
- 8-канальний комутатор аналогових сигналів;
- АЦП типу К1108ПВ1 (А, Б);
- МК, що містить вбудований генератор тактових сигналів, пам'ять команд, ОЗП, вбудовані 4 порти і послідовний канал зв'язку;
- компаратори К1–К3 типу КР554СА3, виходи яких по «АБО» об'єднані з вихідними сигналами мікроконтролера, що управляють;
- пристрої зв'язку з об'єктом У, які включають виконавчі пристрої силової установки, які задають режим випробувань або досліджень.

Через послідовний інтерфейс RS232C, АСУ ТПВ пов'язана з ЕОМ, яка може змінювати режими випробувань або досліджень, а також приймати, запам'ятовувати, відображати і документувати результати випробувань або досліджень. Для виходу на послідовний інтерфейс необхідно вирішити такі проблеми: узгодження рівнів сигналів RS232 і МК-51 (TTL); підтримання стандартної швидкості прийому передачі; підтримання стандартних форматів посилки; підтримання стандартних протоколів обміну.

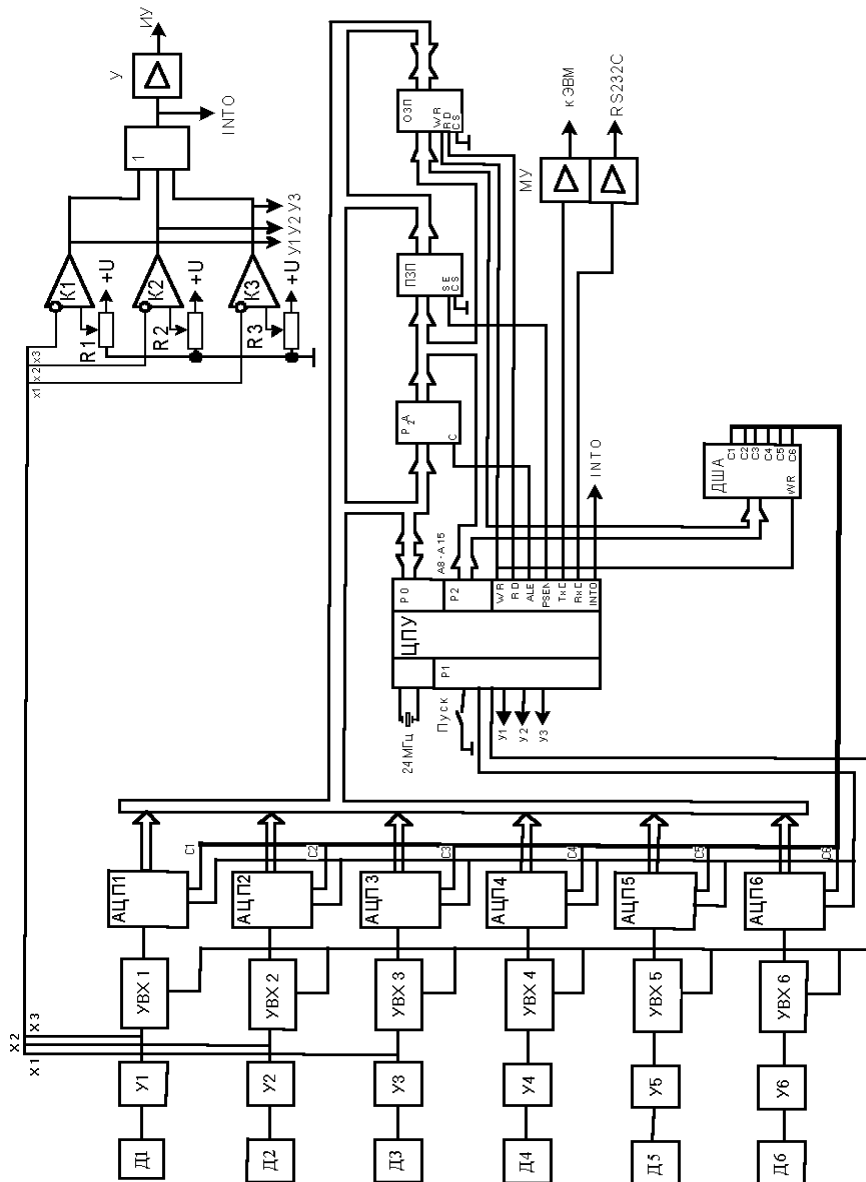


Рисунок 7.4 – Структурна схема АСУТПІВ з паралельними АЦП

Вибір мікроконтролера

Основним елементом системи управління є мікроконтролер 80C251SB фірми Intel. Даний мікроконтролер вибраний виходячи з таких умов: мікропроцесор цього типу (251) є подальшою розробкою широко відомого в світі мікропроцесора серії MCS51, програмно сумісний зверху, але із значно вищою швидкістю (одна операція виконується за 100 нс).

Мікроконтролер вибирають виходячи з умови забезпечення тривалості циклу АЦП – перетворення вхідних сигналів і запису їх в пристрій (ОЗП)

мікроконтролером, що оперативно запам'ятовує. Загальна кількість операцій дорівнює $5 \cdot 6 + 5 = 35$ команд. Для забезпечення необхідної швидкодії схеми вимірювання (відповідно до завдання 4 мкс) тривалість однієї команди розраховується як $t_k/35 = 114$ нс. Таку швидкодію може забезпечити МК 80C251SB.

Опис архітектури мікроконтролера

8XC251SB – перший МК в сімействі MCS251 компанії Intel. Нове сімейство 8-бітових мікроконтролерів підвищує функціональність і продуктивність широко поширених мікроконтролерів MCS51 при збереженні сумісності на рівні двійкових кодів. Завдяки сумісності по контактах з 8XC51FX МК 8XC251SB може служити засобом підвищення продуктивності існуючих апаратно-програмних систем. До типових областей застосування 8XC251SB можна віднести системи управління.

Всім МК сімейства MCS251 властиві такі загальні особливості:

- 24-бітова лінійна адресація до 16 Мбайт пам'яті;
- ЦПУ регістрової архітектури з регістрами, що адресуються як байти, слова і подвійні слова;
- сторінковий режим, прискорюючий вибірку команд із зовнішньої пам'яті;
- конвеєр команд;
- розширена система команд, що включає 16-бітові арифметичні і логічні команди;
- 64-кбайтовий зовнішній стек;
- мінімальний час виконання команд (2 такти в порівнянні з 12 тактами у МК MCS51);
- двійкова сумісність з МК MCS51;

Перерахуємо деякі переваги, пов'язані з цими особливостями:

- збереження програм, написаних для МК MCS51;
- значно вища швидкість обробки, чим у МК MCS51 при тій же тактовій частоті;
- підтримка програм і даних великого розміру;
- підвищена продуктивність програм на мові С.

Функціональна блок-схема МК 8XC251SB, представлена нижче на рис. 7.5, має наступну структуру. Ядро процесора, загальне для всіх мікроконтролерів MCS251. Окремі контролери сімейства відрізняються по набору периферійних блоків на кристалі, портів вводу-виводу, по зовнішній системній шині, розміру ОЗП на кристалі, а також по типу і об'єму внутрішньої пам'яті програм. До складу периферійних пристроїв 8XC251SB входять виділений сторожовий таймер, таймер-лічильник, матриця програмованих лічильників і порт послідовного вводу-виводу. МК 8XC251SB має чотири 8-бітові порти вводу-виводу, P0–P3. Кожну лінію порту вводу-виводу можна окремо запрограмувати або як сигнал загального призначення, або як сигнал спеціальної функції, який підтримує або зовнішню шину, або вбудований периферійний пристрій. Порти P0 і P2 утворюють зовнішню шину, що має 16 ліній для мультиплексування 16-бітової адреси і 8-бітових даних. (МК 8XC251SB дозволяє також конфігурувати 17-й біт зовнішньої адреси). Порти P1 і P3 утворюють сигнали управління шиною і периферією.

МК 8XC251SB, функціональна блок-схема якого представлена на рис. 7.5, має два режими зниженого споживання. У холостому режимі тактові сигнали ЦПУ зупиняються, але підтримується синхронізація периферії. У режимі мікроспоживання внутрішній тактовий генератор зупиняється, і весь кристал переходить в статичний стан. По дозволеному перериванню або апаратному скиданню кристал може вийти з холостого режиму або режиму низького споживання і повернутися в нормальний режим.

Мікроконтролери MCS251 мають розширену систему команд, доповнену новими операціями, режимами адресації і операндами.

Багато команд можуть працювати з 8-, 16- і 32-бітовими операндами що забезпечують зручне і ефективно програмування на мовах високого рівня типу С. Включені такі нові можливості, як команда TRAP, новий режим адресації із зсувом і ряд команд умовного переходу. Аналіз опису системи команд і її порівняння з системою команд мікроконтролерів MCS51 показує, що 8XC251SB можна конфігурувати для роботи в двійковому або початковому режимах. У будь-якому режимі 8XC251SB може виконувати всі команди архітектури MCS51 і MCS251.

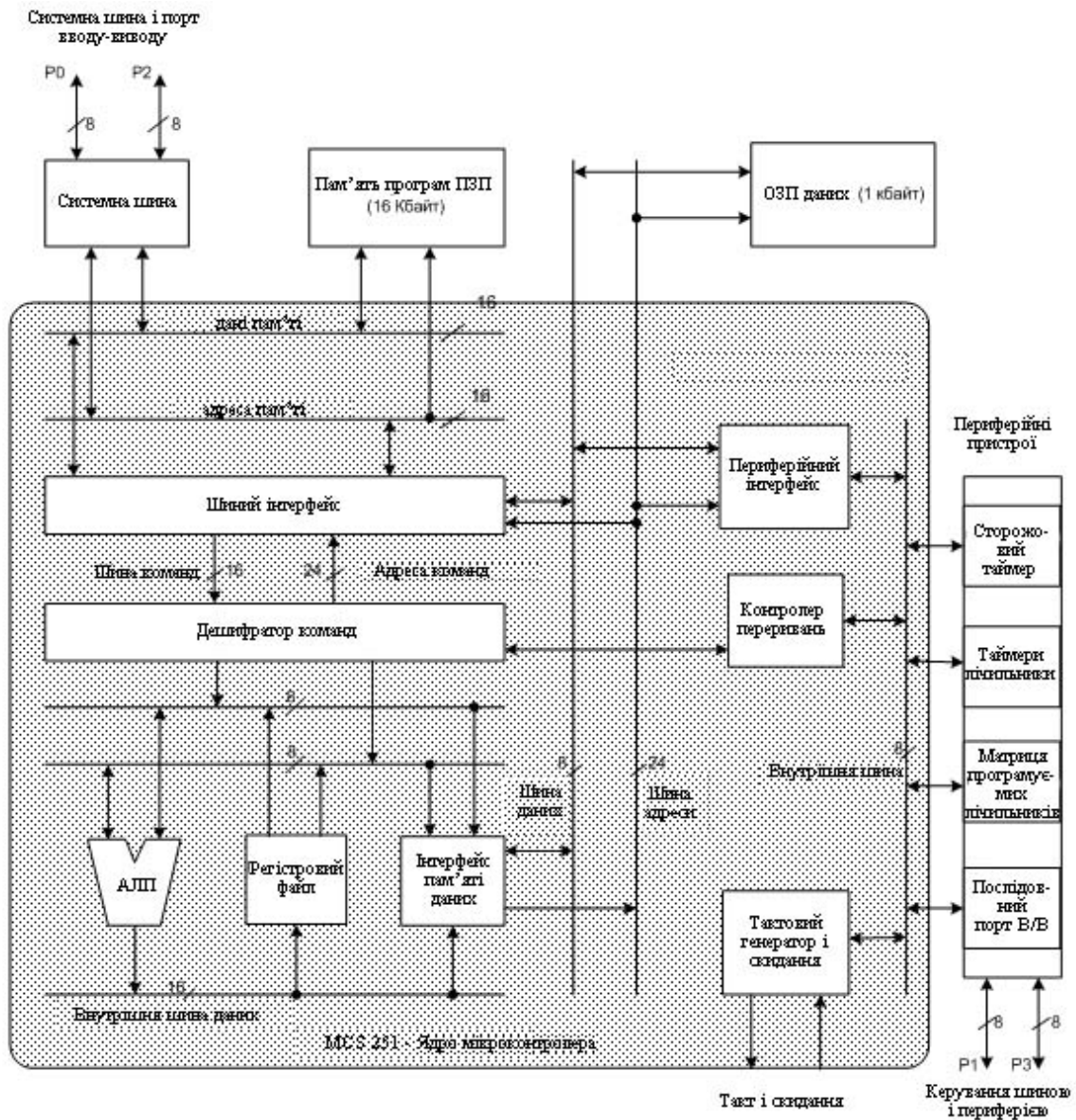


Рисунок 7.5 – Функціональна блок-схема 8XC251SB

Проте початковий режим найбільш ефективний для виконання команд архітектури MCS251, а двійковий – для команд архітектури MCS51.

У двійковому режимі об'єктні коди MCS51 можна виконувати на мікроконтролері 8XC251SB без перекомпіляції.

Якщо система спочатку була спроектована для MCS51, а новий МК 8XC251SB виконуватиме код, написаний для MCS51, то продуктивність буде вища, якщо 8XC251SB працює в двійковому режимі. Об'єктний код, написаний для MCS51, працює швидше на 8XC251SB. Проте, якщо велику

частину коду переписати в новій системі команд, то продуктивність буде вища, якщо 8XC251SB працює в початковому режимі. В цьому випадку 8XC251SB може працювати значно швидше, ніж мікроконтролер MCS51.

Мікроконтролери MCS251 для зберігання команд і даних використовують єдину, лінійну 16-Мбайтову область адресації пам'яті. 8XC251SB може адресувати до 128 кбайт фізичної зовнішньої пам'яті. Регістри спеціальних функцій і регістровий файл мають окремі області адресації.

Вибір АЦП

У схемі, що розробляється, застосовується 8-бітове перетворення. Кількість розрядів АЦП вибрана виходячи з приведеної погрішності аналогових датчиків Д1–Д6, що не перевищує 0,5 %. Погрішність перетворень АЦП Δ визначається значенням молодшого розряду одержуваного коду, і рівна $1/2^n$, де n – кількість розрядів. Вона не повинна перевищувати погрішності вхідного аналогового сигналу. При $n=8$, $\Delta=1/256=0,004=0,4\%$, що цілком прийнятно для нашого випадку. Вибирати АЦП з великою розрядністю не має сенсу, оскільки молодші розряди не будуть достовірними. Крім того, швидкодіючі АЦП (у нашому випадку час перетворення менше 1 мкс) достатньо дорогі, і із збільшенням розрядності ціна їх різко зростає. У даній схемі потрібно шість АЦП для забезпечення одночасного зняття аналогового сигналу з подальшим послідовним прочитуванням результатів і записом їх в ОЗП. Даним вимогам задовольняє АЦП К1108ПВ1 (А, Б). Мікросхема 10-розрядного швидкодіючого функціонально закінченого АЦП послідовного наближення К1108ПВ1 (А, Б) призначена для перетворення аналогового сигналу в двійковий паралельний цифровий код.

До складу функціональної схеми перетворювача входять: джерело опорної напруги (ДОН), генератор тактових імпульсів (ГТІ), вихідний регістр з трьома логічними станами і функцією зберігання інформації протягом одного циклу перетворення, вихідний регістр (ВРг) регістр послідовного перетворення (РПП), цифро-аналоговий перетворювач (ЦАП), багатовхідний компаратор напруг (КН) з вхідним віднімаючим пристроєм, дешифратор рівнів струму і ін.

Мікросхема розрахована на перетворення однополярної вхідної напруги в діапазоні від 0 до 3 В, що подається на вхід через зовнішній операційний підсилювач (ОУ) або пристрій вибірки і зберігання (УВХ) при максимальній частоті перетворення 1,1 МГц для 10-розрядного режиму і 1,33 МГц для 8-розрядного режиму.

Для роботи АЦП К1108ПВ1 потрібно декілька зовнішніх керамічних конденсаторів і джерело напруги $U_{cc1}=5 \text{ В} \pm 5\%$ і $U_{cc2}=5,2 \text{ В} \pm 5\%$. Номінальне значення напруги внутрішнього ДОН складає 2,5 В. Потужність, споживана від джерела живлення, не перевищує 0,85 Вт.

Опис роботи схеми АСУ ТПВ, приведеної на рис. 7.4

Сигнали, що поступають з аналогових датчиків Д1–Д6, нормуються підсилювачами У1–У6, виконаними на базі прецизійних операційних підсилювачів КР140УД26Б. Після цього вони поступають на пристрої вибірки і зберігання УВХ1-УВХ6, виконані на базі мікросхем КР1100СК2. По команді МК УВХ1-УВХ6 одночасно запам'ятовують аналогові сигнали від відповідних підсилювачів У1–У6 на якийсь час, необхідний для роботи ВІС АЦП. Наступна команда МК запускає аналого-цифрові перетворювачі АЦП1-АЦП6, відбувається паралельне в часі перетворення вхідних аналогових сигналів в 8-розрядний двійковий код. Потім знімаються сигнали управління УВХ1-УВХ6 і АЦП1-АЦП6. Перетворена інформація зберігається у вихідних регістрах АЦП1-АЦП6. У циклах адресного читання МК послідовно прочитує перетворену інформацію з АЦП1-АЦП6. Перетворений сигнал, записується в ОЗП статичного типу, виконаного на базі мікросхеми IDT7164S20T місткістю 8кx8 біт фірми NSC, час доступу 20 нс. Сигнали дозволу роботи УВХ і АЦП видаються мікроконтролером. АЦП стробується імпульсами, виданими дешифратором адреси (ДША). У якості ДША застосовується демультіплексор 3 на 8 із сторобуванням, мікросхема КР1533ИД7 або КР1554ИД7. Тактова частота мікроконтролера в 24 МГц забезпечується вбудованим в МК генератором при підключенні до контролера зовнішнього кварцевого резонатора. Програма роботи пристрою розміщена в програмованому постійному запам'ятовуючому пристрої (ППЗП), з ультрафіолетовим стиранням типу D27C64A1 фірми Intel місткістю 8кx8 біт, час вибірки 20 нс. ВІС ППЗП встановлюється на сокеті,

що дозволяє багато разів міняти роботу пристрою, пристосовувавши його до різних типів випробовуваних запобіжників, тим самим, забезпечуючи гнучкість побудови системи автоматичного управління. Після закінчення одного циклу випробувань МК проводить настройку зв'язку з IBM PC, після чого відбувається передача масиву даних результатів випробувань, що містяться в ОЗП, для подальшої обробки і аналізу. Передача проводиться в "старт-стопному" режимі для RS232C з контролем на парність або непарність кожної послідовності і перевірки за допомогою контрольної суми масиву.

При виникненні аварійної ситуації на входах 1–3 аварійний сигнал, посилений компараторами К1–К3, подається на елемент АБО, який в свою чергу видає імпульс на відключення всієї установки. У якості компараторів К1–К3 вибираємо широко використовувані інтегральні схеми КР554СА3 з відкритим колектором, що полегшує узгодження з логічними рівнями мікроконтролера і цифрових інтегральних схем. Установка схеми захисту виставляється резисторами R1–R3. При аварії МК переходить в режим переривання, виконання основної програми припиняється і МК видає повідомлення про аварійну ситуацію в ПЕОМ.

Основний алгоритм роботи схеми АСУ ТПВ [30]

На початку алгоритму проводиться установка початкового стану всіх керуючих сигналів: управління УВХ, управління АЦП, програмуються всі параметри для настройки послідовного порту вводу-виводу RS232C. Потім проводиться очікування команди "пуск" (2) від зовнішньої схеми. У головному циклі алгоритму (3–14) проводиться включення УВХ1–УВХ6 (3), включення АЦП1–АЦП6 (4), встановлюється лічильник параметрів $i=1$. Виключення УВХ1–УВХ6 відбувається через час затримки, рівний 0,8 мкс, необхідний для перетворення аналогового сигналу в цифровий код, після чого знімається сигнал дозволу роботи АЦП (7). Внутрішній цикл знімання інформації і запису в ОЗП виконується 6 разів (8-12), інкрементується номер вибраного АЦП і адреса ОЗП. Блок 13 резервує дві адреси ОЗП для збільшення (у разі потреби) кількості датчиків до восьми, блок 14 аналізує стан, коли проведено 1000 опитувань і вихід "Г" означає закінчення процесу опитування всіх параметрів.

Підраховується контрольна сума всього масиву (15) по модулю 256 і

записується у відповідну комірку ОЗП. Потім відбувається настройка зв'язку з ІВМ РС (17–18), відбувається передача масиву з контролем норми передачі (19–20). Якщо інформація передана достовірно, то відбувається вихід з алгоритму, у разі недостовірності переданої інформації відбувається повторна передача масиву (21) [30].

Алгоритм переривання

Переривання по аварії відбувається по входу INT0. У блоці 1 підпрограма мікроконтролера прочитує сигнали, які вказують на причину аварії. Ознака причини аварії передається через послідовний порт вводу-виводу RS232C у ПЕОМ аналогічно блокам 17–21 основного алгоритму [30].

Робота схеми АСУ ТПВ з паралельними АЦП здійснюється по викладених вище алгоритмам, що дозволяють набагато скоротити терміни проведення досліджень комутаційних електричних апаратів захисту, підвищити достовірність результатів досліджень і їх економічну ефективність. Розроблені АСУ ТПВ і алгоритми можуть використовуватися при проведенні випробувань або досліджень і інших електричних апаратів і пристроїв електропобутової техніки.

Контрольні питання і завдання

1. Назвіть області застосування МК і завдання, які розв'язуються з їх допомогою. Приведіть приклади для електромеханічних систем.
2. Викладіть приклади застосування МК в електроапаратобудуванні і електропобутовій техніці.
3. Складіть структурну схему АСУ ТПВ для опиту 5 датчиків 20 разів в період ($f=50$ Гц) на базі МП КР580 і алгоритм її роботи і програму.
4. Складіть структурну схему АСУ ТПВ для опиту 6 датчиків 25 разів в період ($f=60$ Гц) на базі МК51 і приведіть алгоритм.
5. Приведіть приклади застосування МК в управлінні технологічним процесом з включенням ЕОМ в контур регулювання технологічного параметра.
6. Приведіть приклади застосування МК в управлінні технологічним процесом без включення ЕОМ в контур регулювання.

7. Викладіть мету і особливості застосування МК в комплектних розподільних пристроях (КРП).
8. За якими показниками і як здійснюється вибір мікроконтролера?
9. Складіть структурну схему АСУ ТПВ з паралельними АЦП і приведіть приклад розрахунку її швидкодії.
10. Викладіть порядок розрахунку швидкодії структурної схеми АСУ ТПВ з послідовним АЦП при опитуванні 6-ти датчиків з частотою опитування 200 разів за період ($f=50$ Гц).
11. Назовіть області використання 8-, 16-, 32- і 64-розрядних мікропроцесорів та мікроконтролерів та наведіть приклади.
12. Приведіть приклади застосування МК в управлінні електромеханічними системами транспорту.
13. Для чого застосовують МК в системах управління технологічними процесами з частковим і повним включенням ЕОМ в контур регулювання при проведенні випробувань і досліджень ЕА і ЕБТ?
14. Викладіть призначення і дайте характеристику сімейства цифрових сигнальних процесорів (ЦПС) TMS320C20X (платформа C2000), розроблених фірмою "TEXAS INSTRUMENT".
15. Де використовуються і які мають особливості ЦПС з платформами C5000 і C6000, яка їх продуктивність і за рахунок чого вона досягається?
16. Яка основна комплектуюча апаратура застосовується в КРП серії КУ-10Ц з мікропроцесорним керуванням?
17. Особливості і переваги застосування МП і МК в гнучких системах релейного захисту в порівнянні з аналоговими елементами і мікросхемами.
18. Призначення і вибір датчиків (первинних перетворювачів), нормуючих підсилювачів (нормувачів), мультиплексорів (аналогових електронних комутаторів або цифрових), компараторів, АЦП і ЦАП, пристроїв вибірки і зберігання, пристроїв зв'язку з об'єктом керування.
19. Призначення і функціонування послідовного інтерфейсу RS232C.
20. Як проводиться розрахунок швидкодії АСУ ТПВ на базі МП або МК і які шляхи її підвищення?
21. Наведіть приклади застосування МП і МК в електричних апаратах.

СПИСОК ЛІТЕРАТУРИ

1. *Паскалев Ж.* Первые шаги в вычислительной технике: Пер. с болг. – М.: Радио и связь, 1987. – 146 с.
2. Микропроцессорные БИС и микроЭВМ / *А. А. Васенков, Н. М. Воробьев, В. Л. Дихунян и др.* / Под ред. *А. А. Васенков.* – М.: Сов. радио, 1980. – 280 с.
3. Краткий терминологический словарь по микропроцессорной технике. – М.: Международный центр научной и технической информации, 1984. – 104 с.
4. *Напрасник М. В.* Микропроцессоры и микроЭВМ: Учебное пособие. – М.: Высш. школа, 1989. – 192 с.
5. *Гилмор Ч.* Введение в микропроцессорную технику: Пер. с англ. – М.: Мир, 1984. – 334 с.
6. *Акушский И. Я., Юдицкий Д. И.* Машинная арифметика в остаточных классах. – М.: Сов. радио, 1986. – 440 с.
7. *Лысинов Б. Г.* Арифметические и логические основы автоматов: Учебник для вузов. – 2-е изд., перераб. и доп. – Мн.; Высш. школа, 1980. – 336 с.
8. Микропроцессоры: В 3 т. – Т. 1.: Архитектура и проектирование микроЭВМ. Организация вычислительных процессов / *П. В. Нестеров, В. Ф. Шаньшин, В. Л. Горбунов и др.* / Под ред. *Л. Н. Преснухина* – М.: Высш. школа, 1986. – 495 с.
9. Микропроцессоры: В 3 т. – Т. 2: Средства сопряжения. Контролирующие и информационно управляющие системы / *В. Д. Вернер, Н. В. Воробьев, А. В. Горячев и др.* / Под ред. *Л. Н. Преснухина* – М.: Высш. школа, 1986. – 383 с.
10. *Алексенко А. Г., Галицин А. А., Иванников А. Д.* Проектирование радиоэлектронной аппаратуры на микропроцессорах. – М.: Радио и связь, 1984 – 272 с.
11. Методичні вказівки до лабораторних робіт з курсу “Мікропроцесорні пристрої” для студентів спеціальності 092206 “Електричні машини та апарати” / Уклад. *Ю. С. Гришук, Т. П. Павленко.* –

Харків: ХДПУ, 1999. – 32 с.

12. Современный компьютер: Сб. научно–популярных статей: Пер с англ. / Под ред. Курочкина В. М. – М.: Мир, 1986. – 212 с.

13. Балашов Е. П., Пузанков Д. В. Микропроцессоры и микропроцессорные системы. – М.: Радио и связь, 1981. – 328 с.

14. Горбунов Л. П., Панфилов Д. И., Преснухин Д. Л. Микропроцессоры. Основы построения микроЭВМ. – М.: Высш. школа, 1984. – 144 с.

15. Щербаков О. А. Физические основы записи информации в ПЗУ // Микропроцессорные средства и системы, 1985. – № 3. С. 72–75.

16. Лебедев О. Н. Микросхемы памяти и их применение. – М.: Радио и связь, 1990. – 234 с.

17. Расширенный микропроцессорный комплект БИС серии K588/ В. А. Бобков, Б. Н. Чернуха, В. С. Свиридович, В. П. Ключников // Микропроцессорные средства и системы, 1987. – № 1. – С. 6–10.

18. Гитис Э. И., Пискунов Е. А. Аналого–цифровые преобразователи. – М.: Энергоиздат, 1981. – 360 с.

19. Мікропроцесорна техніка: Підручник / Ю. І. Якименко, Т. О. Терещенко, Є. І. Сокол та ін. / За ред. Т. О. Терещенко – К.: Політехнік, 2003. – 440 с.

20. Сташин В. В., Урусов А. В., Мологонцева О. Ф. Проектирование цифровых устройств на однокристалльных микроконтроллерах. – М.: Энергоатомиздат, 1990. – 224 с.

21. Алексієв О. П., Богаєвський О. Б., Волков В. П. Мікроконтролери для транспортних і промислових застосувань.: архітектура та програмування: Навч. посібник. – Харків: ХНАДУ, 2004. – 156 с.

22. Встраиваемый микроконтроллер 8XC251SB: Руководство пользователя. – К.: “Квазар – Микро”, 1995. – 379 с.

23. Ульрих В. А. Микроконтроллеры PIC16X7XX.– Изд. 2–е, перераб. и доп. / Под ред. С. Л. Корякина–Черняка – СПб.: Наука и техника, 2002. – 320 с.

24. Костин Г. Ю. Микроконтроллеры фирмы Motorola. – К.: КТЦ–МК, 1995. – 37 с.

25. Методические указания к изучению курса «Микропроцессорные устройства» для студентов специальности 092206 «Электрические машины и аппараты» / Сост. *Ю.С. Грищук*. – Харьков: НТУ «ХПИ», 2001. – 24 с.

26. Методичні вказівки до лабораторних робіт з курсу «Мікропроцесорні пристрої» Ч.2. «Однокристалльні мікроконтролери» для студентів спеціальностей 092206 «Електричні машини та апарати» і 092205 «Електропобутова техніка» усіх форм навчання. / Уклад. *Ю. С. Грищук*. – Харків: НТУ «ХПИ», 2003. – 43 с.

27. *Устименко Д. В.* Применение микроконтроллеров в схемах электроподвижного состава. // «Вестник НТУ «ХПИ». Сб. науч. трудов. Вып. 11. – Харьков: НТУ «ХПИ». 2003. – С. 126–128.

28. Процессоры цифровой обработки сигналов фирмы «TEXAS INSTRUMENTS». – М.: ЗАО СКАНТИ-РУС, 2001. – 35 с.

29. *Грищук Ю. С., Ржевский А. Н., Грищук С. Ю.* Автоматизированная система управления для коммутационных исследований и испытаний электрических аппаратов// «Вестник НТУ «ХПИ». Сб. науч. трудов. Вып. 17. – Харьков: НТУ «ХПИ». 2001. – С. 48–50.

30. *Грищук Ю. С., Кузнецов А. И., Ржевский А. Н., Грищук С. Ю.* Применение микроконтроллеров в схемах автоматизированного управления испытаниями электрических аппаратов. // «Вісник НТУ «ХПИ». Зб. наук. праць. – Харків: НТУ «ХПИ», 2005. – Вип. 35 – С. 63-68.

31. Комплектное распределительное устройство серии КУ–10Ц:

Техническая информация НКАИ. 670049.003 и 670049.007. – Ровно: СП РЗВА, 1998. – 24 с.

32. Микропроцессорные гибкие системы релейной защиты. / *В. В. Михайлов, Е. В. Кириевский, Е. М. Ульяницкий и др.* / Под ред. *В. П. Морозкина*. – М.: Энергоатомиздат, 1988. – 240 с.

33. *Дзержицкий С.* Испытания электрических аппаратов. Л.: Энергия, 1975. – 204 с.

34. *Петинов О. В., Щербаков Е.Ф.* Испытания электрических аппаратов. – М.: Высш. школа, 1985. – 215 с.

ДОДАТОК 1

Таблиця Д 1 - Система команд мікропроцесора КР580

Мнемокод	Кільк. байтів	Код	Алгоритм	Коментар
1. Команди пересилання (не виробляють ознаки)				
<i>MOV rd,rs</i>	1	01DDDSSS	$(rd) \leftarrow (rs)$	Уміст <i>rs</i> пересилається в <i>rd</i>
<i>MOV M,rs</i>	1	01110SSS	$(M) \leftarrow (rs)$	Уміст <i>rs</i> пересилається за адресою, що знаходиться в <i>M</i>
<i>MOV rd,M</i>	1	01DDD110	$(rd) \leftarrow (M)$	Уміст комірки, адреса якої в <i>M</i> , пересилається в <i>rd</i>
<i>MVI r, <b2></i>	2	00DDD110	$(r) \leftarrow \langle b2 \rangle$	Другий байт команди пересилається в <i>r</i>
<i>MVI M, <b2></i>	2	36H	$(M) \leftarrow \langle b2 \rangle$	Другий байт команди пересилається в комірку за адресою що знаходиться в <i>M</i>
<i>LXI B, <b2><b3></i>	3	01H	$(C) \leftarrow \langle b2 \rangle$ $(B) \leftarrow \langle b3 \rangle$	Уміст $\langle b2 \rangle$ пересилається в (C) . Уміст $\langle b3 \rangle$ пересилається в (B)
<i>LXI D, <b2><b3></i>	3	11H	$(E) \leftarrow \langle b2 \rangle$ $(D) \leftarrow \langle b3 \rangle$	Уміст $\langle b2 \rangle$ пересилається в (E) . Уміст $\langle b3 \rangle$ пересилається в (D)
<i>LXI H, <b2><b3></i>	3	21H	$(L) \leftarrow \langle b2 \rangle$ $(H) \leftarrow \langle b3 \rangle$	Уміст $\langle b2 \rangle$ пересилається в (L) . Уміст $\langle b3 \rangle$ пересилається в (H)
<i>LXI SP, <b2><b3></i>	1	31H	$(SP)M/L \leftarrow \langle b2 \rangle$ $(SP)CT \leftarrow \langle b3 \rangle$	Уміст $\langle b2 \rangle$ пересилається в молодший розряд покажчика стека (SP) . Уміст $\langle b3 \rangle$ пересилається в старші розряди покажчика стека (SP)
<i>LDAX B</i>	1	0AH	$(A) \leftarrow [(B), (C)]$	Уміст $(B)(C)$ записати в акумулятор
<i>LDAX D</i>	1	1AH	$(A) \leftarrow [(D), (E)]$	Уміст $(D)(E)$ записати в акумулятор
<i>STAX B</i>	1	02H	$[(B), (C)] \leftarrow (A)$	Вміст акумулятора переслати за адресою що знаходиться в $(B)(C)$
<i>STAX D</i>	1	12H	$[(D), (E)] \leftarrow (A)$	Вміст акумулятора переслати за адресою що знаходиться в $(D)(E)$
<i>LHLD <b2><b3></i>	3	2AH	$L \leftarrow \langle b3 \rangle \langle b2 \rangle$ $H \leftarrow \langle b3 \rangle \langle b2 \rangle$	Уміст комірок пам'яті з адресою $\langle b3 \rangle \langle b2 \rangle$ пересилається в $(H)(L)$
<i>SHLD <b2><b3></i>	3	22H	$[\langle b3 \rangle \langle b2 \rangle] \leftarrow (L)$ $[\langle b3 \rangle \langle b2 \rangle] \leftarrow (H)$	Уміст $(H)(L)$ пересилається в комірку пам'яті, адреса якої в $\langle b3 \rangle \langle b2 \rangle$
<i>XCHG</i>	1	EBH	$(H) \leftrightarrow (D) \quad (L) \leftrightarrow (E)$	Обмін умістом між $(H)(L)$ і $(D)(E)$
<i>XTHL</i>	1	E3H	$SP \leftrightarrow (L) \quad [SP+1] \leftrightarrow (H)$	Обмін умістом між $(H)(L)$ і $[SP+1], SP$
<i>SPHL</i>	1	F9H	$(SP) \leftarrow (H)(L)$	16-байтне число $(H)(L)$ пересилається в (SP)
<i>PCHL</i>	1	E9H	$(PC) \leftarrow (H)(L)$	Уміст $(H)(L)$ пересилається в лічильник команди
<i>STA <b2><b3></i>	3	32H	$[\langle b3 \rangle \langle b2 \rangle] \leftarrow (A)$	Уміст акумулятора пересилається в комірку пам'яті, адреса якої в $\langle b3 \rangle \langle b2 \rangle$
<i>LDA <b2><b3></i>	3	3AH	$(A) \leftarrow [\langle b3 \rangle \langle b2 \rangle]$	Уміст комірок пам'яті з адресою $\langle b3 \rangle \langle b2 \rangle$ пересилається в акумулятор

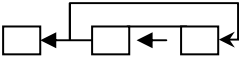
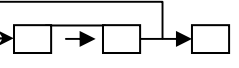
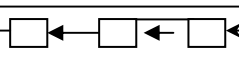
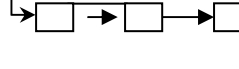
Продовження табл. Д 1

2. Команди розгалуження				
Мнемокод	Кільк. байтів	Код	Алгоритм	Коментар
<i>JMP</i> < <i>b2</i> >< <i>b3</i> >	3	C3	<i>без умов</i> $PC \leftarrow \langle b3 \rangle \langle b2 \rangle$	Безумовний перехід до команди за адресою < <i>b3</i> > < <i>b2</i> >
<i>JC</i> < <i>b2</i> >< <i>b3</i> >	3	DA	$/C=1$	Якщо (C)=0, виконується наступна команда
<i>JNC</i> < <i>b2</i> >< <i>b3</i> >	3	D2	$C=0$	Якщо (C)=1, виконується наступна команда
<i>JZ</i> < <i>b2</i> >< <i>b3</i> >	3	CA	$Z=1$	Якщо (Z)=0, виконується наступна команда
<i>JNZ</i> < <i>b2</i> >< <i>b3</i> >	3	C2	$Z=0$	Якщо (Z)=1, виконується наступна команда
<i>JP</i> < <i>b2</i> >< <i>b3</i> >	3	F2	$S=0$	Якщо (S)=1, виконується наступна команда
<i>JM</i> < <i>b2</i> >< <i>b3</i> >	3	FA	$S=1$	Якщо (S)=0, виконується наступна команда
<i>JPE</i> < <i>b2</i> >< <i>b3</i> >	3	EA	$P=1$	Якщо (P)=0, виконується наступна команда
<i>JPO</i> < <i>b2</i> >< <i>b3</i> >	3	E2	$P=0$	Якщо (P)=1, виконується наступна команда
<i>CALL</i> < <i>b2</i> >< <i>b3</i> >	3	CD	<i>без умов</i> $PC \leftarrow \langle b3 \rangle \langle b2 \rangle$	Виклик підпрограми (ПП)
<i>CC</i> < <i>b2</i> >< <i>b3</i> >	3	DC	$PC \leftarrow \langle b3 \rangle \langle b2 \rangle$ $C=1$	Якщо (C)=1, перехід на ПП за адресою, зазначеною в < <i>b3</i> > < <i>b2</i> >
<i>CNC</i> < <i>b2</i> >< <i>b3</i> >	3	DA	$C=0$ $PC \leftarrow \langle b3 \rangle \langle b2 \rangle$	Якщо (C)=0, перехід на ПП за адресою, зазначеною в < <i>b3</i> > < <i>b2</i> >
<i>CZ</i> < <i>b2</i> >< <i>b3</i> >	3	CC	$Z=1$	Якщо (Z)=1, перехід на ПП за адресою, зазначеною в < <i>b3</i> > < <i>b2</i> >
<i>CNZ</i> < <i>b2</i> >< <i>b3</i> >	3	C4	$Z=0$	Якщо (Z)=0, перехід на ПП за адресою, зазначеною в < <i>b3</i> > < <i>b2</i> >
<i>CM</i> < <i>b2</i> >< <i>b3</i> >	3	FC	$S=1$	Якщо (S)=1, перехід на ПП за адресою, зазначеною в < <i>b3</i> > < <i>b2</i> >
<i>CP</i> < <i>b2</i> >< <i>b3</i> >	3	F4	$S=0$	Якщо (S)=0, перехід на ПП за адресою, зазначеною в < <i>b3</i> > < <i>b2</i> >
<i>CPE</i> < <i>b2</i> >< <i>b3</i> >	3	EC	$P=1$	Якщо (P)=1, перехід на ПП за адресою, зазначеною в < <i>b3</i> > < <i>b2</i> >
<i>CP0</i> < <i>b2</i> >< <i>b3</i> >	3	E4	$P=0$	Якщо (P)=0, перехід на ПП за адресою, зазначеною в < <i>b3</i> > < <i>b2</i> >
<i>RET</i>	1	C9	$(PC) \leftarrow [SP] [SP+1],$ $(SP) = (SP) + 2$	Безумовне повернення із ПП
<i>RC</i>	1	D8	$C=1 (PC) \leftarrow [SP] [SP+1]$	C=1, повернення із ПП на команду, адреса якої записана у стеці
<i>RNC</i>	1	D0	$C=0$	Повернення із ПП при (C)=0
<i>RZ</i>	1	C8	$Z=1$	Повернення із ПП при (Z)=1
<i>RNZ</i>	1	C0	$Z=0$	(Z)=0, повернення із ПП на команду
<i>RM</i>	1	F8	$S=1$	(S)=1, повернення із ПП на команду
<i>RP</i>	1	F0	$S=0$	(S)=0, повернення із ПП на команду
<i>RPE</i>	1	E8	$P=1$	(P)=1, повернення із ПП на команду
<i>RPO</i>	1	E0	$P=0$	(P)=0, повернення із ПП на команду
<i>RST</i>	1			ПОВТОРНИЙ ЗАПУСК

Продовження табл. Д 1

3. Арифметико-логічні команди									
Мнемокод	Кільк. байтів	Код	Ознаки					Алгоритм	Коментар
			C	Z	S	P	C'		
<i>ADD rs</i>	1	10000SSS	+	+	+	+	+	$(A) \leftarrow (A) + (rs)$	Уміст акумулятора складається з <i>rs</i>
<i>ADC rs</i>	1	100001SSS	+	+	+	+	+	$(A) \leftarrow (A) + (rs) + (C)$	Уміст акумулятора складається з <i>rs</i> і бітом <i>C</i>
<i>SUB r</i>	1	10010SSS	+	+	+	+	+	$(A) \leftarrow (A) - (rs)$	Із умісту акумулятора віднімається вміст <i>rs</i>
<i>SBB r_s</i>	1	10011SSS	+	+	+	+	+	$(A) \leftarrow (A) - (rs) - (C)$	Із умісту акумулятора віднімається вміст <i>rs</i> та біт <i>C</i>
<i>ANA rs</i>	1	10100SSS	0	+	+	+	-	$A \leftarrow (A) \wedge (rs)$	Порозрядна кон'юнкція
<i>XRA rs</i>	1	10101SSS	0	+	+	+	-	$A \leftarrow (A) \nabla (rs)$	Порозрядне заперечення рівнозначності
<i>ORA rs</i>	1	10110SSS	0	+	+	+	-	$A \leftarrow (A) \vee (rs)$	Порозрядна диз'юнкція
<i>CMP rs</i>	1	10111SSS	+	1	+	+	+	$(A) - (rs)$	Порівняння за допомогою внутрішнього вирахування
<i>ADD M</i>	1	86H	+	+	+	+	+	$(A) \leftarrow (A) + (M)$	Уміст акумулятора складається із умістом комірки за адресою в <i>M</i>
<i>ADC M</i>	1	8EH	+	+	+	+	+	$(A) \leftarrow (A) + (M) + (C)$	Уміст акумулятора складається із умістом комірки за адресою в <i>M</i> і бітом <i>C</i>
<i>SUB M</i>	1	96H	+	+	+	+	+	$(A) \leftarrow (A) - (M)$	Із умісту акумулятора віднімається вміст комірки за адресою в <i>M</i>
<i>SBB M</i>	1	9EH	+	+	+	+	+	$(A) \leftarrow (A) - (M) - (C)$	Із умісту акумулятора віднімається вміст комірки за адресою в <i>M</i> і біт <i>C</i>
<i>ANA M</i>	1	A6H	0	+	+	+	-	$A \leftarrow (A) \wedge (M)$	Порозрядна кон'юнкція
<i>XRA M</i>	1	AEH	0	+	+	+	-	$A \leftarrow (A) \nabla (M)$	Порозрядне заперечення рівнозначності
<i>ORA M</i>	1	B6H	0	+	+	+	-	$A \leftarrow (A) \vee (M)$	Порозрядна диз'юнкція
<i>CMP M</i>	1	BEH	+	+	+	+	+	$(A) - (M)$	Порівняння за допомогою внутрішнього вирахування
<i>ADI <b2></i>	2	C6H	+	+	+	+	+	$A \leftarrow (A) + \langle b2 \rangle$	Уміст акумулятора складається із умістом 2-го байта команди
<i>ACI <b2></i>	2	CEH	+	+	+	+	+	$(A) \leftarrow (A) + \langle b2 \rangle + (C)$	Уміст акумулятора складається із умістом 2-го байта команди і бітом <i>C</i>
<i>SUI <b2></i>	2	D6H	+	+	+	+	+	$(A) \leftarrow (A) - \langle b2 \rangle$	Із умісту акумулятора віднімається вміст 2-го байта команди
<i>SBI <b2></i>	2	DEH	+	+	+	+	+	$(A) \leftarrow (A) - \langle b2 \rangle - (C)$	Із умісту акумулятора віднімається вміст 2-го байта команди й біт <i>C</i>

Продовження табл. Д 1

Мнемокод	Кільк. байтів	Код	Ознаки					Алгоритм	Коментар
			C	Z	S	P	C'		
<i>ANI</i> < <i>b2</i> >	2	E6H	0	+	+	+	-	$(A) \leftarrow (A) \wedge \langle b2 \rangle$	Порозрядна кон'юнкція
<i>XRI</i> < <i>b2</i> >	2	EEH	0	+	+	+	-	$(A) \leftarrow (A) \vee \langle b2 \rangle$	Порозрядне заперечення рівнозначності
<i>ORI</i> < <i>b2</i> >	2	F6H	0	+	+	+	-	$(A) \leftarrow (A) \vee \langle b2 \rangle$	Порозрядна диз'юнкція
<i>CPI</i> < <i>b2</i> >	2	FEH	+	+	+	+	+	$(A) - \langle b2 \rangle$	Порівняння за допомогою внутрішнього вирахування
<i>INR rd</i>	1	00DD100	-	+	+	+	+	$(rd) \leftarrow (rd) + 1$	Уміст <i>rd</i> збільшується на 1
<i>DCR rd</i>	1	00DDD101	-	+	+	+	+	$(rd) \leftarrow (rd) - 1$	Уміст <i>rd</i> зменшується на 1
<i>INX B</i>	1	03H	-	-	-	-	-	$(B)(C) \leftarrow (B)(C) + 1$	Уміст $(B)(C)$ збільшується на 1
<i>INX D</i>	1	13H	-	-	-	-	-	$(D)(E) \leftarrow (D)(E) + 1$	Уміст $(D)(E)$ збільшується на 1
<i>INX H</i>	1	23H	-	-	-	-	-	$(H)(L) \leftarrow (H)(L) + 1$	Уміст $(H)(L)$ збільшується на 1
<i>INX SP</i>	1	33H	-	-	-	-	-	$(SP) \leftarrow (SP) + 1$	Уміст (SP) збільшується на 1
<i>INR M</i>	1	34H	-	+	+	+	+	$(M) \leftarrow (M) + 1$	Уміст комірки за адресою в <i>M</i> збільшується на 1
<i>DCX B</i>	1	0BH	-	-	-	-	-	$(B)(C) \leftarrow (B)(C) - 1$	Уміст $(B)(C)$ зменшується на 1
<i>DCX D</i>	1	1BH	-	-	-	-	-	$(D)(E) \leftarrow (D)(E) - 1$	Уміст $(D)(E)$ зменшується на 1
<i>DCX H</i>	1	2BH	-	-	-	-	-	$(H)(L) \leftarrow (H)(L) - 1$	Уміст $(H)(L)$ зменшується на 1
<i>DCX SP</i>	1	3BH	-	-	-	-	-	$(SP) \leftarrow (SP) - 1$	Уміст (SP) зменшується на 1
<i>DCR M</i>	1	35H	-	+	+	+	+	$(M) \leftarrow (M) - 1$	Уміст комірки за адресою в <i>M</i> зменшується на 1
<i>RLC</i>	1	07H	+	-	-	-	-	$A_{m+1} \leftarrow A_m$ $A_0 \leftarrow A_7$ $(C) \leftarrow A_7$	
<i>RRC</i>	1	0FH	+	-	-	-	-	$A_m \leftarrow A_{m+1}$ $A_7 \leftarrow A_0$ $(C) \leftarrow A_0$	
<i>RAL</i>	1	17H	+	-	-	-	-	$A_{m+1} \leftarrow A_m$ $(C) \leftarrow A_7$ $A_0 \leftarrow (C)$	
<i>RAR</i>	1	1FH	+	-	-	-	-	$A_m \leftarrow A_{m+1}$ $A_7 \leftarrow (C)$ $(C) \leftarrow A_0$	
<i>CMA</i>	1	2FH	-	-	-	-	-	$(A) \leftarrow \overline{(A)}$	Уміст акумулятора інвертується
<i>CMC</i>	1	3EH	+	-	-	-	-	$(C) \leftarrow \overline{(C)}$	Уміст (C) інвертується
<i>STC</i>	1	37H	+	-	-	-	-	$(C) \leftarrow 1$	Тригер переносу встановлений в 1

Продовження табл. Д 1

Мнемокод	Кільк. байтів	Код	Ознаки					Алгоритм	Коментар
			C	Z	S	P	C'		
<i>DAD B</i>	1	09	+	-	-	-	-	$(H)(L) \leftarrow (H)(L) + (B)(C)$	Подвійне додавання $(H)(L)$ з $(B)(C)$
<i>DAD D</i>	1	19	+	-	-	-	-	$(H)(L) \leftarrow (H)(L) + (D)(E)$	Подвійне додавання $(H)(L)$ з $(B)(C)$
<i>DAD H</i>	1	29	+	-	-	-	-	$(H)(L) \leftarrow (H)(L) + (H)(L)$	Подвійне додавання $(D)(E)$ з $(H)(L)$
<i>DAD SP</i>	1	39	+	-	-	-	-	$(H)(L) \leftarrow (H)(L) + (SP)$	Подвійне додавання (SP) з $(H)(L)$
<i>DAA</i>	1	27	+	+	+	+	+	Десяткова корекція акумулятор	
4. Команди вводу-виводу. Звертання до стеку й керування МП									
<i>IN<b2></i>	2	B	-	-	-	-	-	$(A) \leftarrow (Дані)$	Тут <b2> є адресою пристрою уведення
<i>OUT<b2></i>	2	D3	-	-	-	-	-	$(Дані) \leftarrow (A)$	Тут <b2> є адресою пристрою виводу
<i>PUSH B</i>	1	C5	-	-	-	-	-	$[SP-1] \leftarrow (B)$ $[SP-2] \leftarrow (C)$ $(SP) \leftarrow (SP)-2$	Уміст (B, C) пересилається в стек, показчик стека зменшується на 2
<i>PUSH D</i>	1	D5	-	-	-	-	-	$[SP-1] \leftarrow (D)$ $[SP-2] \leftarrow (E)$ $(SP) \leftarrow (SP)-2$	Уміст (D, E) пересилається в стек, показчик стека зменшується на 2
<i>PUSH H</i>	1	E5	-	-	-	-	-	$[SP-1] \leftarrow (H)$ $[SP-2] \leftarrow (L)$ $(SP) \leftarrow (SP)-2$	Уміст (H, L) пересилається в стек, показчик стека зменшується на 2
<i>PUSH PSW</i>	1	F5	-	-	-	-	-	$[SP-1] \leftarrow (A)$ $[SP-2] \leftarrow (F)$ $(SP) \leftarrow (SP)-2$	Уміст (A, F) пересилається в стек, показчик стека зменшується на 2
<i>POP B</i>	1	C1	-	-	-	-	-	$(C) \leftarrow [SP]$ $(B) \leftarrow [SP-1]$ $[SP] \leftarrow [SP]+2$	Уміст (B, C) пересилається в стек, показчик стека збільшується на 2
<i>POP D</i>	1	D1	-	-	-	-	-	$(E) \leftarrow [SP]$ $(D) \leftarrow [SP+1]$ $[SP] \leftarrow [SP]+2$	Уміст (D, E) пересилається в стек, показчик стека збільшується на 2
<i>POP PSW</i>	1	F1	+	+	+	+	+	$(F) \leftarrow [SP]$ $(A) \leftarrow [SP+1]$ $[SP] \leftarrow [SP]+2$	Уміст (A, F) пересилається в стек, показчик стека збільшується на 2
<i>POP H</i>	1	E1	-	-	-	-	-	$(L) \leftarrow [SP]$ $(H) \leftarrow [SP+1]$ $[SP] \leftarrow [SP]+2$	Уміст (H, L) пересилається в стек, показчик стека збільшується на 2
<i>XTHL</i>	1	E3	-	-	-	-	-	$SP \leftrightarrow (L)$ $[SP+1] \leftrightarrow (H)$	Обмін умістом $(H)(L)$ і $[SP+1], SP$
<i>XCHG</i>	1	EB	-	-	-	-	-	$(H) \leftrightarrow (D)$ $(L) \leftrightarrow (E)$	Обмін умістом $(H)(L)$ і $(D)(E)$
<i>EI</i>	1	FB	-	-	-	-	-		Переривання дозволене
<i>DI</i>	1	F3	-	-	-	-	-		Переривання заблоковане
<i>NOP</i>	1	0	-	-	-	-	-		Операція не виконується
<i>HLT</i>	1	76	-	-	-	-	-		Відбувається зупинка можливості наступного запуску при сприйнятті запитів переривання

ДОДАТОК 2

Приклади виконання завдань

Завдання 1. Запрограмувати паралельний інтерфейс КР580ВВ55: режим 0, канал А – ввід, канал В – вивід, канал С – розряди 0...3 – вивід, 4...7 – ввід.

Рішення. Виходячи з умови завдання по формату керуючого слова (КС) [8, 11] складаємо двійковий код керуючого слова (КСВ), 1001 1000 і переводимо його в 16-річний код, що відповідає 98h.

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	0	0
9				8			

Заносимо програмно КС 98h в регістр керуючого слова (РКС).

адреса	код	команда	Коментар
8000	3E	MVI A, 98h	(A) ← 98h
8001	98		
8002	D3	OUT адр.РКС	(адр.РКС) ← (A)
8003	Адр. РКС		
8004	робоча програма		

Завдання 2. Запрограмувати таймер КР580ВІ53: канал 2, режим каналу 4, лічильник 2-10, режим читання ЧТ або запису ЗП – на льоту, N=5.

Рішення. По умові завдання і формату КС таймера [8, 11] складається код КС - 89h, який заноситься в РКС таймера (адреса E3) командами MVI A, 89h і OUT E3. Після цього заноситься інформація (число N=5) у канал лічильника командами MVI A, N і OUT E2, де E2 адреса каналу лічильника E2. Адреса каналу лічильника 0 - E0, лічильника 1- E1.

Варіанти завдань для програмування паралельного інтерфейсу КР580ВВ55 і таймера КР580ВІ53 приведені в [25].

ДОДАТОК 3
СИСТЕМА КОМАНД PIC16X7XX

Таблиця Д 3 – Опис полів кода операції

Поле	Опис
F	Адреса регістра (0x00...0x7f)
W	Робочий регістр (акумулятор)
B	Адреса розряду всередині 8-розрядного регістру
K	Постійне число або значення літерала (мітка)
X	Невизначене значення (= 0 або 1) асемблер генерує код $x = 0$, рекомендовано для сумісності з усіма інструментальними засобами програмного забезпечення
D	Показчик підсумку; $d = 0$: підсумок зберігається у W, $D = 1$: підсумок зберігається в регістрі f. За замовчуванням значення $d = 1$
label	Ім'я мітки
TOS	Верхня комірка стеку
PC	Програмний лічильник
PCLATH	Регістр старших розрядів програмного лічильника
GIE	Біт загального дозволу переривань
WDT	Сторожовий таймер
TO	Біт закінчення лічення сторожового таймера
PD	Біт скидання при увімкненні живлення
dest	Підсумок, регістр W або початковий регістр
[]	Додаткові параметри
()	Вміст
→	Призначення
< >	Показчик розряду
∈	Належить множині
<i>Курсив</i>	Термін, визначений користувачем (шрифт- <i>Курсив</i>)

Таблиця Д 3 – Система команд PIC16X7XX

Мнемоніка команди	Операція	Ци- кли	Код операції		Прапо- рці	При- мітка
			MSb	LSb		
БАЙТОВІ ОПЕРАЦІЇ З РЕГІСТРАМИ						
ADDWF t, d	Складання (W + f → d)	1	00 0111	dfff ffff	C,DC,Z	1,2
ANDWF t, d	Логічне «І» (W and f → d)	1	00 0101	dfff ffff	Z	1,2
CLRF f	Обнулити f	1	00 0001	1fff ffff	Z	2
CLRW -	Обнулити W	1	00 0001	0xxx xxxx	Z	
COMF t, d	Доповнення f (Логічне «НІ»)	1	00 1001	dfff ffff	Z	1,2
DECF t, d	Декрементf	1	00 0011	dfff ffff	Z	1,2
DECFSZ t, d	Декремент f, пропуск якщо «0»	1(2)	00 1011	dfff ffff		1,2,3
INCF t, d	Інкремент f	1	00 1010	dfff ffff	Z	1,2
INCFSZ t, d	Інкремент f, пропуск якщо «0»	1(2)	00 1111	dfff ffff		1,2,3

Продовження табл. Д 3

IORWF	t, d	Логічне «АБО» ($W \text{ or } f \rightarrow d$)	1	00 0100	dfff ffff	Z	1,2
MOVF	t, d	Пересилання ($f \rightarrow W$)	1	00 1000	dfff ffff	Z	1,2
MOVWF	f	Пересилання ($W \rightarrow f$)	1	00 0000	1fff ffff		
NOP	-	Пуста операція	1	00 0000	0xx0 0000		
RLF	t, d	Зсув вліво через перенесення	1	00 1101	dfff ffff	C	1,2
RRF	t, d	Зсув вправо через перенесення	1	00 1100	dfff ffff	C	1,2
SUBWF	t, d	Віднімання ($f - W \rightarrow d$)	1	00 0010	dfff ffff	C,CD,Z	1,2
SWAPF	t, d	Обміняти напівбайти f	1	00 1110	dfff ffff		1,2
XORWF	t, d	Виключаюче «АБО» ($W \text{ or } f \rightarrow d$)	1	00 0110	dfff ffff	Z	1,2
БІТОВІ ОПЕРАЦІЇ З РЕГІСТРОМ							
BCF	t, d	Обнулити біт b в f	1	01 00bb	dfff ffff		1,2
BSF	t, d	Встановити біт b в f	1	01 01bb	dfff ffff		1,2
BTFS	t, d	Тест біта b в f, пропуск якщо «0»	1(2)	01 10bb	dfff ffff		3
BTFS	t, d	Тест біта b в f пропуск якщо «1»	1(2)	01 11bb	dfff ffff		3
ОПЕРАЦІЇ З КОНСТАНТАМИ І КОМАНДАМИ КЕРУВАННЯ							
ADDLW	k	Складання ($k + W \rightarrow W$)	1	11 111x	kkkk kkkk	C,DC,Z	
ANDLW	k	Логічне «І» ($k \text{ and } W \rightarrow W$)	1	11 1001	kkkk kkkk	Z	
CALL	k	Виклик підпрограми k	2	10 0kkk	kkkk kkkk		
CLRWDТ	-	Обнулити WDT	1	00 0000	0110 0100		
GOTO	k	Перехід за адресою k	2	10 1kkk	kkkk kkkk		
IORLW	k	Логічне «АБО» ($k \text{ or } W \rightarrow W$)	1	11 1000	kkkk kkkk	Z	
MOVLW	k	Пересилання ($k \rightarrow W$)	1	11 00xx	kkkk kkkk		
RETFIE	-	Повернення з переривання	2	00 0000	0000 1001		
RETLW	k	Повернення з константою	2	11 01xx	kkkk kkkk		
RETURN	-	Повернення з підпрограми	2	00 0000	0000 1000		
SLEEP	-	Останов	1	00 0000	0110 0011	TO,PD	
SUBLW	k	Віднімання ($k - W \rightarrow W$)	1	11 110x	kkkk kkkk	C,DCY	
XORLW	k	Виключаюче «АБО» ($k \text{ xor}$	1	11 1010	kkkk kkkk	Z	

Примітка

- Усі команди виконуються як «читання-модифікація-запис». Якщо змінюється регістр I/O, то вихідним значенням буде величина, прочитана безпосередньо з контактів. Наприклад, якщо в регістрі даних '1', а контакти сконфігуровані як входи і зовнішній пристрій встановить низькій рівень, то до регістру даних будуть записані '0'.
- Якщо команда виконана над регістром TMR0 (коли $d = 1$, результат записується до регістру таймера 0), то предділитель буде обнулений.
- Якщо лічильник програм (PC) змінюється або результат перевірки умови істинний, то команда виконується за два цикли. Другий цикл виконується як команда NOP.

ДОДАТОК 4

Таблиця Д 4 – Команди Асемблера МК51

Назва команди	Група команд передачі даних						Операція
	Мнемокод	КОП	Т	Б	Ц		
Пересилання в акумулятор з регістра ($n = 0 \div 7$)	MOV A, Rn	11101rrr	1	1	1	$(A) \leftarrow (Rn)$	
Пересилання в акумулятор прямоадресованого байта	MOV A, ad	11100101	3	2	1	$(A) \leftarrow (ad)$	
Пересилання в акумулятор байта із РПД ($i = 0,1$)	MOV A, @Ri	1110011i	1	1	1	$(A) \leftarrow ((Ri))$	
Завантаження в акумулятор константи	MOV A, #d	01110100	2	2	1	$(A) \leftarrow \#d$	
Пересилання в регістр із акумулятора	MOV Rn, A	11111rrr	1	1	1	$(Rn) \leftarrow (A)$	
Пересилання в регістр прямоадресованого байта	MOV Rn, ad	10101rrr	3	2	2	$(Rn) \leftarrow (ad)$	
Завантаження в регістр константи	MOV Rn, #d	01111rrr	2	2	1	$(Rn) \leftarrow \#d$	
Пересилання по прямій адресі акумулятора	MOV ad, A	11110101	3	2	1	$(ad) \leftarrow (A)$	
Пересилання по прямій адресі регістра	MOV ad, Rn	10001rrr	3	2	2	$(ad) \leftarrow (Rn)$	
Пересилання прямоадресованого байта по прямій адресі	MOV add, ads	10000101	9	3	2	$(add) \leftarrow (ads)$	
Пересилання байта із РПД по прямій адресі	MOV ad, @Ri	1000011i	3	2	2	$(ad) \leftarrow ((Ri))$	
Пересилання по прямій адресі константи	MOV ad, #d	01110101	7	3	2	$(ad) \leftarrow \#d$	
Пересилання в РПД із акумулятора	MOV @Ri, A	1111011i	1	1	1	$((Ri)) \leftarrow (A)$	
Пересилання в РПД прямоадресованого байта	MOV @Ri, ad	0110011i	3	2	2	$((Ri)) \leftarrow (ad)$	
Пересилання в РПД константи	MOV @Ri, #d	0111011i	2	2	1	$((Ri)) \leftarrow \#d$	
Завантаження покажчика даних	MOV DPTR, #d16	10010000	13	3	2	$(DPTR) \leftarrow \#d16$	
Пересилання в акумулятор байта із ПП	MOVC A, @A+DPTR	10010011	1	1	2	$(A) \leftarrow ((A) + (DPTR))$	
Пересилання в акумулятор байта із ПП	MOVC A, @A+PC	10000011	1	1	2	$(PC) \leftarrow (PC) + 1$ $(A) \leftarrow ((A) + (PC))$	
Пересилання в акумулятор байта із ЗПД	MOVX A, @Ri	1110001i	1	1	2	$(A) \leftarrow ((Ri))$	
Пересилання в акумулятор байта з розширеної ЗПД	MOVX A, @DPTR	11100000	1	1	2	$(A) \leftarrow ((DPTR))$	
Пересилання у ЗПД із акумулятора	MOVX @Ri, A	1111001i	1	1	2	$((Ri)) \leftarrow (A)$	
Пересилання в розширену ЗПД із акумулятора	MOVX @DPTR, A	11110000	1	1	2	$((DPTR)) \leftarrow (A)$	
Завантаження в стек	PUSH ad	11000000	3	2	2	$(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (ad)$	

Продовження табл. Д4

Витяг зі стека	POP ad	11010000	3	2	2	$(ad) \leftarrow (SP) \quad (SP) \leftarrow (SP) - 1$
Обмін акумулятора з регістром	XCH A, Rn	11001rrr	1	1	1	$(A) \leftrightarrow (Rn)$
Обмін акумулятора із прямоадресованим байтом	XCH A, ad	11000101	3	2	1	$(A) \leftrightarrow (ad)$
Обмін акумулятора з байтом із РПД	XCH A, @Ri	1100011i	1	1	1	$(A) \leftrightarrow ((Ri))$
Обмін молодшої тетради акумулятора з молодшою тетрадою байта РПД	XCHD A, @Ri	1101011i	1	1	1	$(A_{0-3}) \leftrightarrow ((Ri)_{0-3})$

Група команд арифметичних операцій

Назва команди	Мнемокод	КОП	T	Б	Ц	Операція
Додавання акумулятора з регістром ($n = 0 \div 7$)	ADD A, Rn	00101rrr	1	1	1	$(A) \leftarrow (A) + (Rn)$
Додавання акумулятора із прямоадресованим байтом	ADD A, ad	00100101	3	2	1	$(A) \leftarrow (A) + (ad)$
Додавання акумулятора з байтом із РПД ($i = 0,1$)	ADD A, @Ri	0010011i	1	1	1	$(A) \leftarrow (A) + ((Ri))$
Додавання акумулятора з константою	ADD A, #d	00100100	2	2	1	$(A) \leftarrow (A) + \#d$
Додавання акумулятора з регістром і переносом	ADDC A, Rn	00111rrr	1	1	1	$(A) \leftarrow (A) + (Rn) + (C)$
Додавання акумулятора із прямоадресованим байтом і переносом	ADDC A, ad	00110101	3	2	1	$(A) \leftarrow (A) + (ad) + (C)$
Додавання акумулятора з байтом із РПД і переносом	ADDC A, @Ri	0011011i	1	1	1	$(A) \leftarrow (A) + ((Ri)) + (C)$
Додавання акумулятора з константою й переносом	ADDC A, #d	00110100	2	2	1	$(A) \leftarrow (A) + \#d + (C)$
Десяткова корекція акумулятора	DA A	11010100	1	1	1	Якщо $(A_{0-3}) > 9V((AC) = 1)$, то $(A_{0-3}) \leftarrow (A_{0-3}) + 6$, потім якщо $(A_{4-7}) > 9V((C \neq 1)$, то $(A_{4-7}) \leftarrow (A_{4-7}) + 6$
Віднімання з акумулятора регістра і позички	SUBB A, Rn	10011rrr	1	1	1	$(A) \leftarrow (A) - (C) - (Rn)$
Віднімання з акумулятора прямоадресованого байта й позички	SUBB A, ad	10010101	3	2	1	$(A) \leftarrow (A) - (C) - ((ad))$
Віднімання з акумулятора байта РПД і позички	SUBB A, @Ri	1001011i	1	1	1	$(A) \leftarrow (A) - (C) - ((Ri))$
Віднімання з акумулятора константи й позички	SUBB A, d	10010100	2	2	1	$(A) \leftarrow (A) - (C) - \#d$
Інкремент акумулятора	INC A	00000100	1	1	1	$(A) \leftarrow (A) + 1$
Інкремент регістра	INC Rn	00001rrr	1	1	1	$(Rn) \leftarrow (Rn) + 1$
Інкремент прямоадресованого байта	INC ad	00000101	3	2	1	$(ad) \leftarrow (ad) + 1$

Продовження табл. Д4

Інкремент байта в РПД	INC	@Ri	0000011i	1	1	1	$((Ri)) \leftarrow ((Ri)) + 1$
Інкремент покажчика даних	INC	DPTR	10100011	1	1	2	$(DPTR) \leftarrow (DPTR) + 1$
Декремент акумулятора	DEC	A	00010100	1	1	1	$(A) \leftarrow (A) - 1$
Декремент регістра	DEC	Rn	00011rrr	1	1	1	$(Rn) \leftarrow (Rn) - 1$
Декремент прямоадресуемого байта	DEC	ad	00010101	3	2	1	$(ad) \leftarrow (ad) - 1$
Декремент байта в РПД	DEC	@Ri	0001011i	1	1	1	$((Ri)) \leftarrow ((Ri)) - 1$
Множення акумулятора на регістр В	MUL	AB	10100100	1	1	4	$(B)(A) \leftarrow (A) \times (B)$
Ділення акумулятора на регістр В	DIV	AB	10000100	1	1	4	$(A).(B) \leftarrow (A)/(B)$

Група команд логічних операцій

Назва команди	Мнемокод	КОП	Т	Б	Ц	Операція
Логічне І акумулятора й регістра	ANL A, Rn	01011rrr	1	1	1	$(A) \leftarrow (A) \wedge (Rn)$
Логічне І акумулятора й прямоадресованого байта	ANL A, ad	01010101	3	2	1	$(A) \leftarrow (A) \wedge (ad)$
Логічне І акумулятора й байта із РПД	ANL A, @Ri	0101011i	1	1	1	$(A) \leftarrow (A) \wedge ((Ri))$
Логічне І акумулятора й константи	ANL A, #d	01010100	2	2	1	$(A) \leftarrow (A) \wedge \#d$
Логічне І прямоадресованого байта й акумулятора	ANL ad, A	01010010	3	2	1	$(ad) \leftarrow (ad) \wedge (A)$
Логічне І прямоадресованого байта й константи	ANL ad, #d	01010011	7	3	2	$(ad) \leftarrow (ad) \wedge \#d$
Логічне АБО акумулятора й регістра	ORL A, Rn	01001rrr	1	1	1	$(A) \leftarrow (A) \vee (Rn)$
Логічне АБО акумулятора й прямоадресованого байта	ORL A, ad	01000101	3	2	1	$(A) \leftarrow (A) \vee (ad)$
Логічне АБО акумулятора й байта із РПД	ORL A, @Ri	0100011i	1	1	1	$(A) \leftarrow (A) \vee ((Ri))$
Логічне АБО акумулятора й константи	ORL A, #d	01000100	2	2	1	$(A) \leftarrow (A) \vee \#d$
Логічне АБО прямоадресованого байта й акумулятора	ORL ad, A	01000010	3	2	1	$(ad) \leftarrow (ad) \vee (A)$
Логічне АБО прямоадресованого байта й константи	ORL ad, #d	01000011	7	3	2	$(ad) \leftarrow (ad) \vee \#d$
Виключаюче АБО акумулятора й регістра	XRL A, Rn	01101rrr	1	1	1	$(A) \leftarrow (A) \vee (Rn)$
Виключаюче АБО акумулятора й прямоадресованого байта	XRL A, ad	01100101	3	2	1	$(A) \leftarrow (A) \vee (ad)$
Виключаюче АБО акумулятора й байта із РПД	XRL A, @Ri	0110011i	1	1	1	$(A) \leftarrow (A) \vee ((Ri))$

Продовження табл. Д4

Виключаюче АБО акумулятора й константи	XRL A, #d	01100100	2	2	1	$(A) \leftarrow (A) \forall \#d$
Виключаюче АБО прямоадресованого байта і акумулятора	XRL ad, A	01100010	3	2	1	$(ad) \leftarrow (ad) \forall (A)$
Виключаюче АБО прямоадресованого байта й константи	XRL ad, #d	01100011	7	3	2	$(ad) \leftarrow (ad) \forall \#d$
Скидання акумулятора	CLR A	11100100	1	1	1	$(A) \leftarrow 0$
Інверсія акумулятора	CPL A	11110100	1	1	1	$(A) \leftarrow (\bar{A})$
Зсув акумулятора вліво циклічне	RL A	00100011	1	1	1	$(A_{n+1}) \leftarrow (A_n), n=0-6, (A_0) \leftarrow (A_7)$
Зсув акумулятора вліво через перенос	RLC A	00110011	1	1	1	$(A_{n+1}) \leftarrow (A_n), n=0-6, (A_0) \leftarrow (C), (C) \leftarrow (A_7)$
Зсув акумулятора вправо циклічне	RR A	00000011	1	1	1	$(A_n) \leftarrow (A_{n+1}), n=0-6, (A_7) \leftarrow (A_0)$
Зсув акумулятора вправо через перенос	RRC A	00010011	1	1	1	$(A_n) \leftarrow (A_{n+1}), n=0-6, (A_7) \leftarrow (C), (C) \leftarrow (A_0)$
Обмін місцями тетрад в акумуляторі	SWAP A	11000100	1	1	1	$(A_{0-3}) \leftrightarrow (A_{4-7})$

Група команд операцій з бітами

277

Назва команди	Мнемокод	КОП	Т	Б	Ц	Операція
Скидання переносу	CLR C	11000011	1	1	1	$(C) \leftarrow 0$
Скидання біта	CLR bit	11000010	4	2	1	$(b) \leftarrow 0$
Установка переносу	SETB C	11010011	1	1	1	$(C) \leftarrow 1$
Установка біта	SETB bit	11010010	4	2	1	$(b) \leftarrow 1$
Інверсія переносу	CPL C	10110011	1	1	1	$(C) \leftarrow (C)$
Інверсія біта	CPL bit	10110010	4	2	1	$(b) \leftarrow (b)$
Логічне І біта й переносу	ANL C, bit	10000010	4	2	2	$(C) \leftarrow (C) \wedge (b)$
Логічне І інверсії біта й переносу	ANL C, /bit	10110000	4	2	2	$(C) \leftarrow (C) \wedge (\bar{b})$
Логічне АБО біта й переносу	ORL C, bit	01110010	4	2	2	$(C) \leftarrow (C) \vee (b)$
Логічне АБО інверсії біта й переносу	ORL C, /bit	10100000	4	2	2	$(C) \leftarrow (C) \vee (\bar{b})$
Пересилання біта в перенос	MOV C, bit	10100010	4	2	1	$(C) \leftarrow (b)$

Продовження табл. Д4

Пересилання переносу в біт	MOV bit, C	10010010	4	2	2	(b) ← (C)
Група команд передачі керування						
Довгий перехід у повному обсязі пам'яті програм	LJMP ad 16	00000010	12	3	2	(PC) ← ad 16
Абсолютний перехід усередині сторінки в 2 Кбайта	AJMP ad 11	a ₁₀ a ₉ a ₈ 00001	6	2	2	(PC) ← (PC)+2, (PC ₀₋₁₀) ← ad 11
Короткий відносний перехід усередині сторінки в 256 байт	SJMP rel	10000000	5	2	2	(PC) ← (PC)+2 (PC) ← (PC)+rel
Непрямий відносний перехід	JMP @A+DPTR	01110011	1	1	2	(PC) ← (A)+(DPTR)
Перехід, якщо акумулятор дорівнює нулю	JZ rel	01100000	5	2	2	(PC) ← (PC)+2, якщо (A)=0, то (PC) ← (PC)+rel
Перехід, якщо акумулятор не дорівнює нулю	JNZ rel	01110000	5	2	2	(PC) ← (PC)+2, якщо (A) ≠ 0, то (PC) ← (PC)+rel
Перехід, якщо перенос дорівнює одиниці	JC rel	01000000	5	2	2	(PC) ← (PC)+2, якщо (C)=1, то (PC) ← (PC)+rel
Перехід, якщо перенос дорівнює нулю	JNC rel	01010000	5	2	2	(PC) ← (PC)+2, якщо (C)=0, то (PC) ← (PC)+rel
Перехід, якщо біт дорівнює одиниці	JB bit, rel	00100000	11	3	2	(PC) ← (PC)+3, якщо (b)=1, то (PC) ← (PC)+rel
Перехід, якщо біт дорівнює нулю	JNB bit, rel	00110000	11	3	2	(PC) ← (PC)+3, якщо (b)=0, то (PC) ← (PC)+rel
Перехід, якщо біт установлений, з наступним скиданням біта	JBC bit, rel	00010000	11	3	2	(PC) ← (PC)+3, якщо (b)=1, то (PC) ← (PC)+rel і (b) ← (0)
Декремент регістра й перехід, якщо не нуль	DJNZ Rn, rel	11011rrr	5	2	2	(PC) ← (PC)+2, (Rn) ← (Rn)-1 якщо (Rn) ≠ 0, то (PC) ← (PC)+rel
Декремент прямоадресованого байта й перехід, якщо не нуль	DJNZ ad, rel	11010101	8	3	2	(PC) ← (PC)+2, (ad) ← (ad)-1 якщо (ad) ≠ 0, то (PC) ← (PC)+rel

Продовження табл. Д4

Порівняння акумулятора із прямоадресованим байтом і перехід, якщо не дорівнює	CJNE A, ad, rel	10110101	8	3	2	(PC) \leftarrow (PC)+3, якщо (A) \neq (ad), то (PC) \leftarrow (PC)+rel, якщо (A)<(ad), то (C) \leftarrow 1, інакше (C) \leftarrow 0
Порівняння акумулятора з константою й перехід, якщо не дорівнює	CJNE A, #d, rel	10110100	10	3	2	(PC) \leftarrow (PC)+3, якщо (A) \neq #d, то (PC) \leftarrow (PC)+rel, якщо (A)<#d, то (C) \leftarrow 1, інакше (C) \leftarrow 0
Порівняння регістра з константою й перехід, якщо не дорівнює	CJNE Rn, #d, rel	10111rrr	10	3	2	(PC) \leftarrow (PC)+3, якщо (Rn) \neq #d, то (PC) \leftarrow (PC)+rel, якщо (Rn)<#d, то (C) \leftarrow 1, інакше (C) \leftarrow 0
Порівняння байта в РПД із константою й перехід, якщо не дорівнює	CJNE @Ri, #d, rel	1011011i	10	3	2	(PC) \leftarrow (PC)+3, якщо ((Ri))=#d, то (PC) \leftarrow (PC)+rel, якщо ((Ri))<#d, то (C) \leftarrow 1, інакше (C) \leftarrow 0
Довгий виклик підпрограми	LCALL ad 16	00010010	12	3	2	(PC) \leftarrow (PC)+3, (SP) \leftarrow (SP)+1, ((SP)) \leftarrow (PC ₀₋₇), (SP) \leftarrow (SP)+1, ((SP)) \leftarrow (PC ₈₋₁₅), (PC) \leftarrow ad16
Абсолютний виклик підпрограми в межах сторінки в 2 Кбайта	ACALL ad 11	a ₁₀ a ₉ a ₈ 10001	6	2	2	(PC) \leftarrow (PC)+2, (SP) \leftarrow (SP)+1, ((SP)) \leftarrow (PC ₀₋₇), (SP) \leftarrow (SP)+1, ((SP)) \leftarrow (PC ₈₋₁₅), (PC ₀₋₁₀) \leftarrow ad11
Повернення з підпрограми	RET	00100010	1	1	2	(PC ₈₋₁₅) \leftarrow ((SP)), (SP) \leftarrow (SP)-1, (PC ₀₋₇) \leftarrow ((SP)), (SP) \leftarrow (SP)-1
Повернення з підпрограми обробки переривання	RETI	00110010	1	1	2	(PC ₈₋₁₅) \leftarrow ((SP)), (SP) \leftarrow (SP)-1, (PC ₀₋₇) \leftarrow ((SP)), (SP) \leftarrow (SP)-1
Холоста команда	NOP	00000000	1	1	1	(PC) \leftarrow (PC) + 1

Навчальне видання

ГРИЦУК Юрій Степанович

МІКРОПРОЦЕСОРНІ ПРИСТРОЇ

НАВЧАЛЬНИЙ ПОСІБНИК

для студентів електромеханічних спеціальностей

Роботу до видання рекомендувала О.Г.Глебова

Редактор В.М. Баранов

План 2005р., поз. 56/92- 06

Підп. до друку

Формат 60x84 1/16. Папір офсетний.

Друк.– ризографія. Гарнітура Times New Roman. Ум. друк. арк. 11,7

Обл. – вид. арк.14,2. Наклад 300 прим. Зам. № Ціна договірна.

Видавничий центр НТУ „ХПІ”.

Свідоцтво про державну реєстрацію ДК № 116 від 10.07.2004р.

61002, Харків, вул. Фрунзе, 21

Друкарня НТУ „ХПІ”, 61002 Харків, вул. Фрунзе, 21