Лекція 10

1. Модуль УАПП

1.1 Загальні відомості

Обмін інформацією між мікропроцесором/мікроконтролером та модемом може здійснюватися через інтерфейс RS–232 (рисунок 1), який включає:

– УАПП – універсальний асинхронний послідовний програмований приймач передавач;

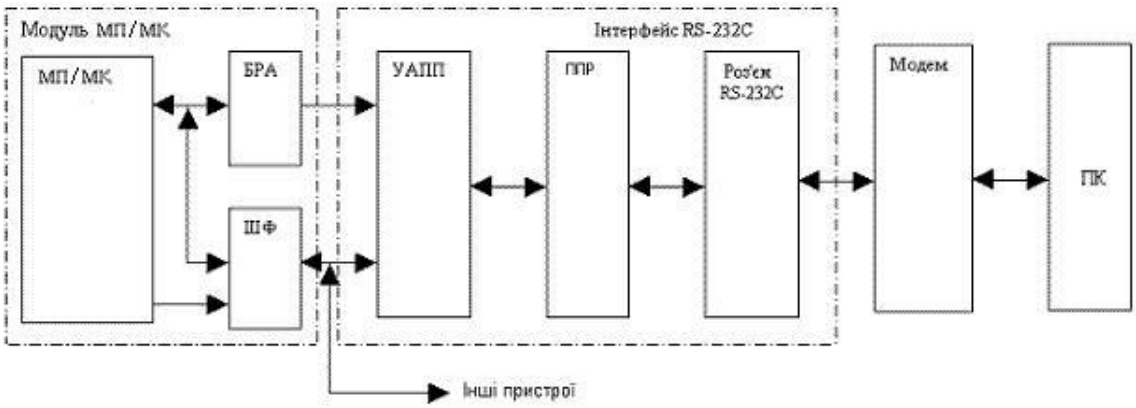
– ППР – пристрій перетворення рівнів;

– роз'єм RS–232C.

Крім інтерфейсу RS–232C схема сполучення містить:

– БРА – буферний регістр адреси;

– ШФ – шинний формувач.



МП - мікропроцесор; БРА – буферний регістр адреси
ШФ – шинний формувач; УАПП – універсальний асинхронний послідовний програмований приймач – передавач;
ППР – пристрій перетворення рівнів; ПК - персональний комп'ютер
МК- мікроконтролер

Рисунок 1 – Структурна схема сполучення МП/МК з модемом за допомогою інтерфейсу RS – 232C

1.2 Універсальний асинхронний послідовний програмований приймач–передавач (УАПП)

УАПП, наприклад, мікросхема TL16C450 (рисунок 2) перетворює дані з паралельного формату в послідовний при передачі (виведенні) з мікропроцесора з послідовного формату в паралельний при прийомі (введенні) у мікропроцесор.

Формат даних, які передаються у канал зв'язку в послідовному асинхронному форматі, представлений на рисунку 3.

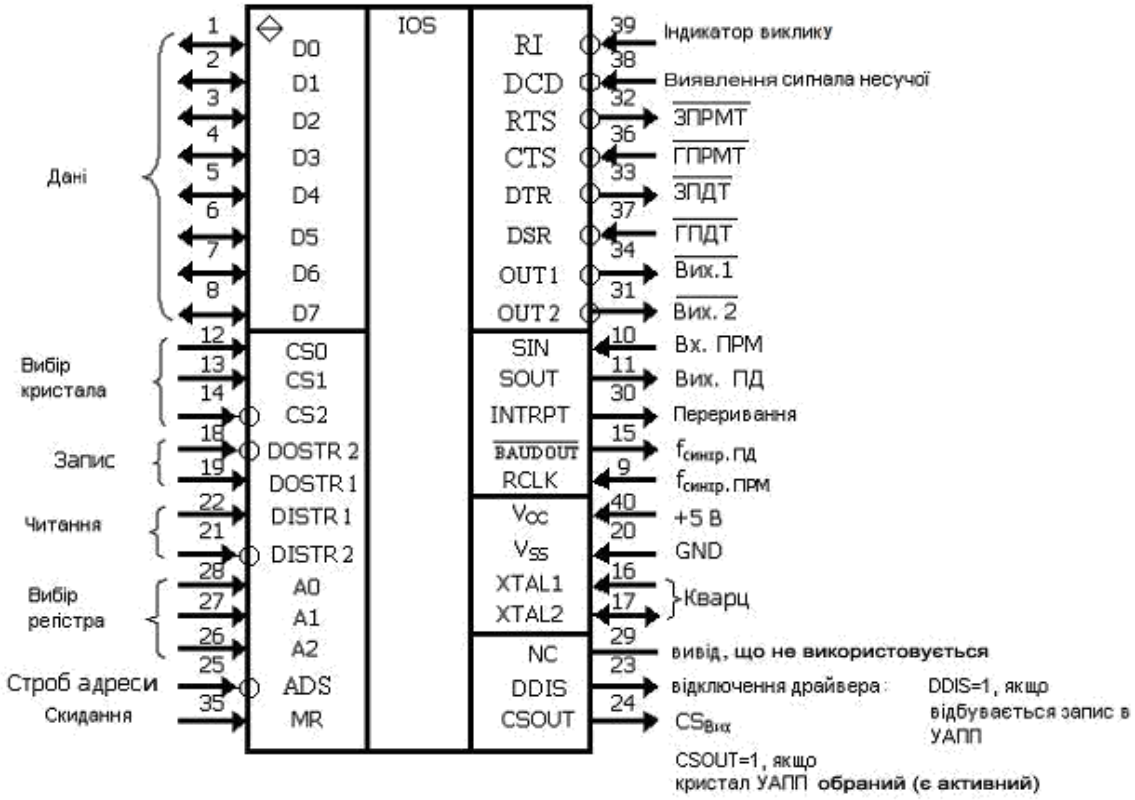


Рисунок 2 – Умовне позначення IMC TL16C450 (УАПП–UART)

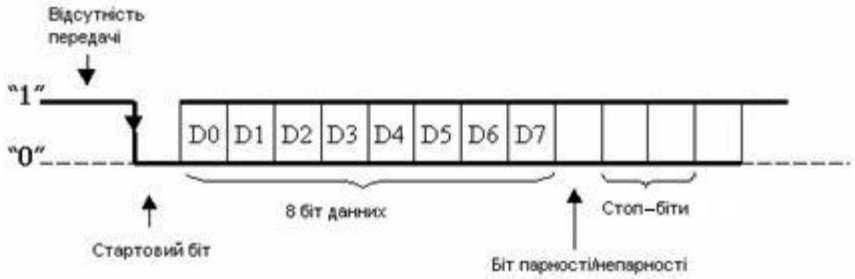


Рисунок 3 – Формат даних інтерфейсу RS–232C

Власне дані (5, 6, 7 чи 8 біт) супроводжуються стартовим бітом, бітом парності/непарності (якщо такий контроль програмно передбачений) і стоповим одиничним сигналом, що включає 1; 1,5 чи 2 стоп–біти. Одержавши стартовий біт, приймач вибирає з лінії біти даних через визначені інтервали часу. Дуже важливо, щоб тактові частоти приймача і передавача були однаковими (допустима розбіжність – не більше 10 %).

Швидкість передачі за RS–232C, наприклад, може вибиратися з ряду: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 біт/с (бод).

1.3 Пристрій перетворення рівнів (ППР)

Усі сигнали RS–232C передаються/приймаються спеціально обраними рівнями, що забезпечують високу завадостійкість зв'язку (рисунок 4). Слід зазначити, що дані передаються/приймаються в інверсному вигляді: логічній одиниці відповідає низький рівень, а логічному нулю – високий рівень.

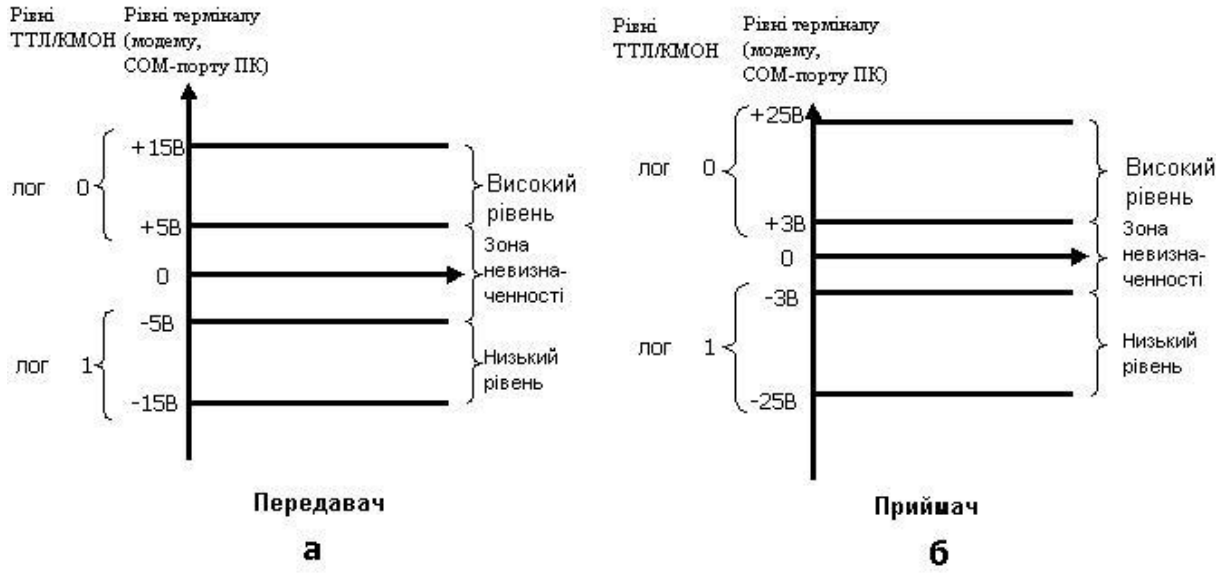


Рисунок 4 – Рівні сигналів RS – 232C: а – на передаючій, б – на приймальній кінцях лінії зв’язку

Як видно з рисунка 4 під час передачі логічного нуля на виході інтерфейсу формується високий рівень напруги в діапазоні: +5В…+15В, під час передачі логічної одиниці – низький рівень напруги в діапазоні: –5В…–15В.

Під час прийому на вхід інтерфейсу надходить високий рівень напруги в діапазоні: +3В…+25В, що несе інформацію про логічний 0, чи низький рівень напруги в діапазоні: –3В…–25В, що відображає логічну одиницю.

Таким чином, для узгодження ТТЛ/КМОН – рівнів сигналів, що діють у мікропроцесорній системі, з рівнями сигналів послідовного інтерфейсу, що передаються у лінію зв'язку або приймаються з лінії зв'язку використовують пристрої перетворення рівнів (ППР).

Існують різні варіанти схемної реалізації ППР, одним із яких є застосування мікросхеми фірми MAXIM: MAX232A. Дана мікросхема (рисунок 5) вимагає одне джерело живлення +5В та ряд додаткових елементів –

конденсаторів C1, C2, … , C5, що не є надмірною ціною за переваги її застосування.

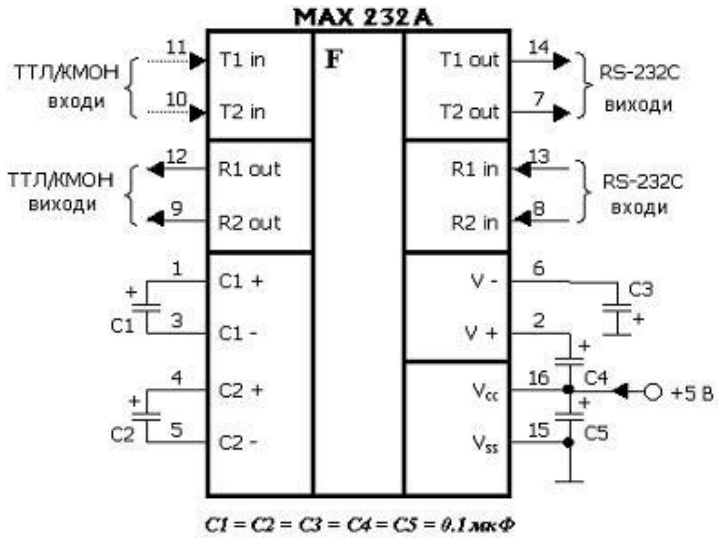


Рисунок 5 – Позначення і особливості підключення мікросхеми MAX232A

1.4 Роз’єм RS–232C

Для зв'язку інтерфейсу RS–232C із зовнішнім терміналом (модемом) або ПК може використовуватися 25– чи 9–контактні роз’єми (рисунок 6).

Роз’єм RS–232C

Рисунок 6 – 9-контактний роз'єм RS-232C

Призначення основних контактів наступне:

SG – сигнальне заземлення, нульовий провід;

TxD – дані, що передаються мікропроцесором у послідовному коді (від’ємна логіка)

RxD – дані, що приймаються мікропроцесором у послідовному коді (від’ємна логіка);

DCD – виявлення несучої даних (детектування сигналу, що приймається МП – м від модему);

DTR – запит передавача термінала (модему або ПК);

DSR – готовність передавача термінала (модему або ПК);

RTS – запит приймача термінала (модему або ПК);

CTS – готовність приймача термінала (модему або ПК);

RI – індикатор виклику. Говорить про прийом модемом сигналу виклику від телефонної мережі.

1.5 Модуль УАПП в мікроконтролерах сімейства LPC2300

1.5.1 Загальна характеристика

У даний час у більшості мікроконтролерах сімейства LPC2300 є по чотири модулі універсального асинхронного приймача–передавача (UART): UART0, UART1, UART2, UART3]. Всі модулі використовуються однаковим чином, якщо не рахувати того, що в модулі UART1 реалізовано повний модемний інтерфейс, а модуль UART3 підтримує передачу за протоколом IrDA.

Всі модулі відповідають специфікації «Промислового стандарту 550» та мають вбудований контролер швидкості передачі і 16 – байтні буфери (FIFO) прийому і передачі (рисунок 7). Крім цього, за допомогою модуля UART3 можна реалізувати не тільки провідний зв’язок за протоколом RS–232, а також безпровідний за стандартом IrDA.

Спрощену архітектуру модуля UART0 наведено на рисунку 8.

1.5.2 Налаштування швидкості обміну

Користувач може задавати швидкість обміну для любого з модулів UART безпосередньо в програмі чи використавши функцію автоматичного визначення швидкості. В останньому випадку модуль буде сконфігуровано автоматично.

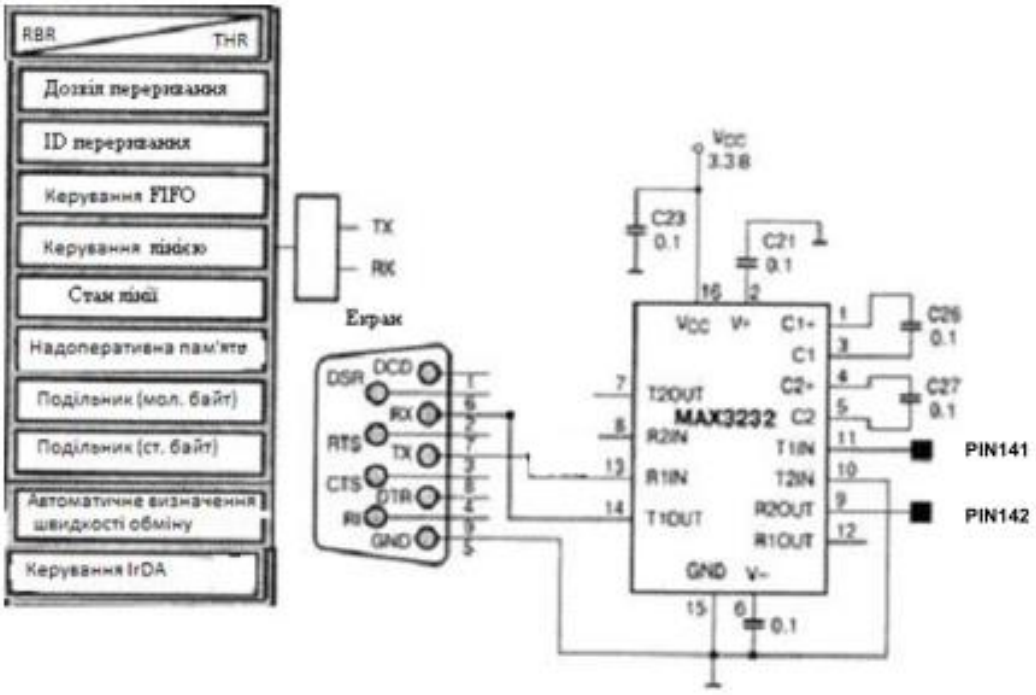


Рисунок 7 – Керуючі регістри модуля UART та зв’язок модуля з модемом

Ініціалізація модуля UART проводиться таким чином:

```
voidinit_serial (void) {          /* Ініціалізація послідовного інтерфейсу */
PINSEL = 0x00050000;              /*Дозволяємо RxD та TxD/
U0LCR = 0x00000083;               /*8 біт, без контролю парності, 1 стоп–біт*/
U0DLL = 0x00000002;               /*Швидкість обміну 9600 бод при частоті VPB 30
                                  МГц *//*
U0LCR = 0x00000003; }             /*DLAB=0*
```

Спочатку програмується блок керування функцією виводів для відключення виводів мікроконтролера від портів введення/виведення і підключення їх до модуля UART. Потім в регістрі керування лінією UART (LCR) задається формат передачі даних (рисунок 9).

У даному прикладі встановлюється наступний формат даних, які передаються: 8–бітове слово, один стоп–біт, контроль парності вимкнено. У регістрі LCR є додатковий біт (DLAB), який керує доступом до регістра–защіпки подільника контролера швидкості передачі. Для програмування контролера цей біт потрібно встановити в 1.

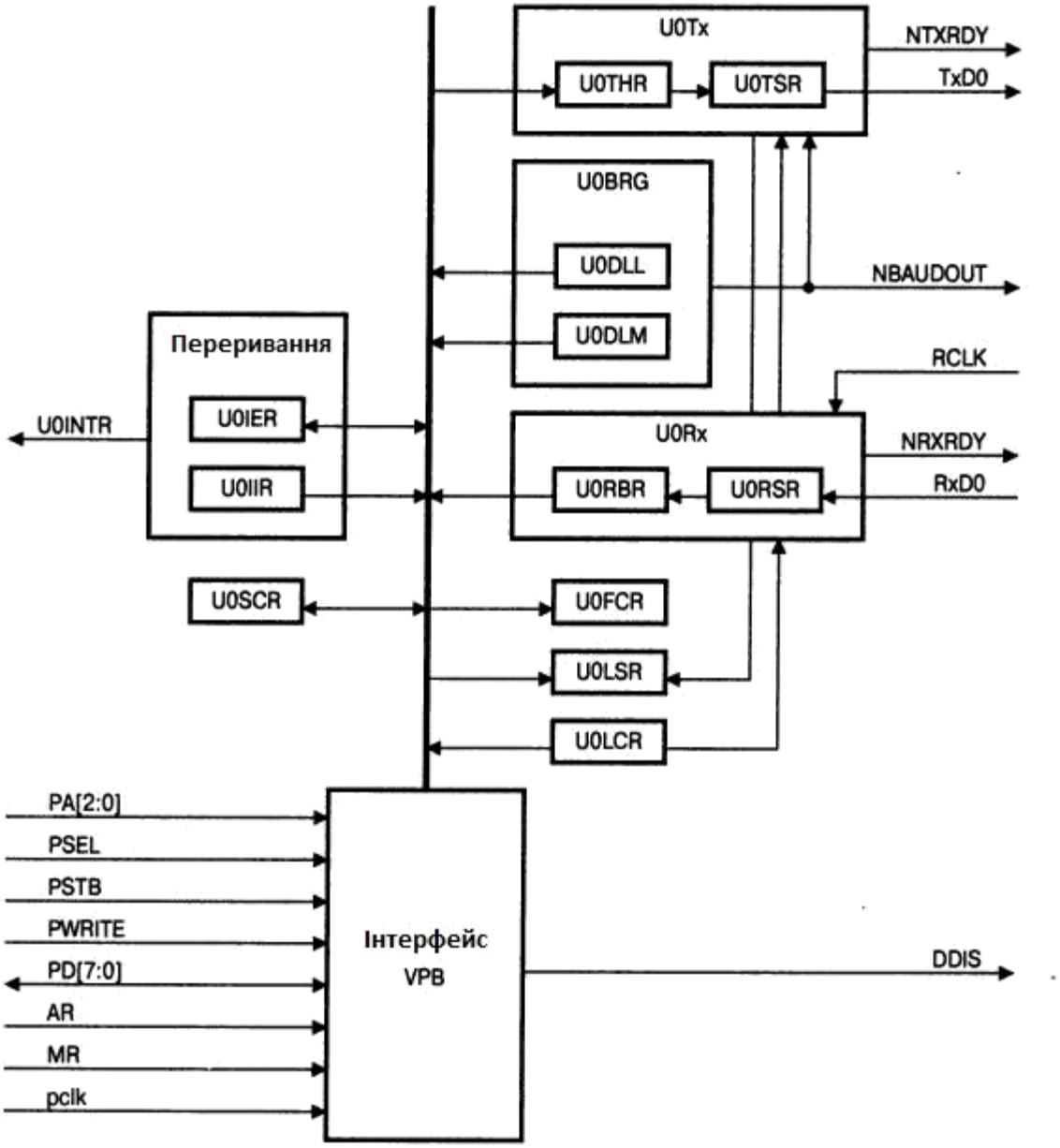


Рисунок 8 – Спрощена архітектура модуля UART0

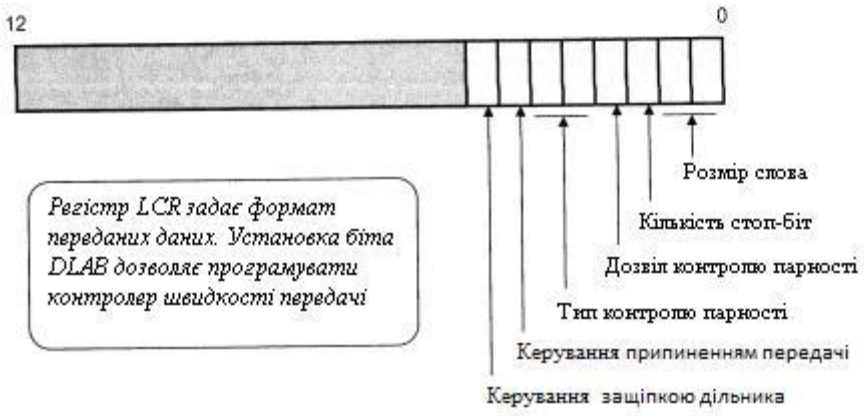


Рисунок 9 – Регістр керування лінією UART

Контролер швидкості передачі являє собою 16–бітний попередній подільник, що формує з сигналу Pclk тактовий сигнал модуля UART, частота якого повинна бути в 16 разів більше необхідної швидкості передачі. Таким чином, формула для обчислення коефіцієнта ділення має вигляд:

Кділ = Pclk/16*BAUD, де BAUD – необхідна швидкість передачі.

Наприклад, для програмування BAUD = 9600 біт/с при Pclk=30 М Гц

Кділ = 30000000/16 х 9600 ≈ 194, або 0xC2.

Дійсна швидкість передачі при такому значенні дільника дорівнює 9665 бод. Часто немає можливості встановити точне значення необхідної швидкості передачі, однак модуль UART буде працювати навіть при 5–відсотковому відхиленні тривалості бітового інтервалу від номінальної. Завдяки цьому у вас є певна свобода при конфігуруванні UART, якщо частоту Pclk необхідно підбирати для отримання точних часових характеристик інших периферійних пристроїв, наприклад для тактової синхронізації модуля CAN. Коефіцієнт ділення подільника зберігається в двох регістрах–защіпках: DLM і DLL. Перші вісім біт обох регістрів використовуються для зберігання відповідно старшого і молодшого байтів коефіцієнта ділення, як показано на рисунку 10.

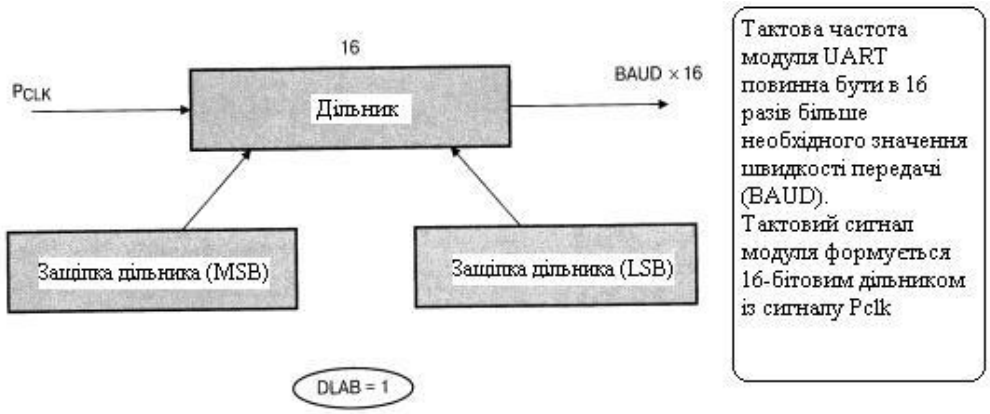


Рисунок 10 – Формування швидкості передачі даних UART

Програмування швидкості обміну виконується при одиничному значенні біта DLAB регістра LCR, після чого цей біт треба скинути в нуль, щоб захистити регістр U0DLL та U0DLM від випадкової зміни.

1.5.3 Автовизначення швидкості обміну

Будь–який з модулів UART мікроконтролера LPC23xx можна сконфігурувати для автоматичного визначення швидкості передачі пакету даних, який приймається. Після визначення швидкості в регістри–защіпки подільника заносяться коректні значення, які забезпечують роботу модуля на даній швидкості. Керування функцією автоматичного визначення швидкості здійснюється за допомогою однойменного регістра UxACR (рисунок 11).

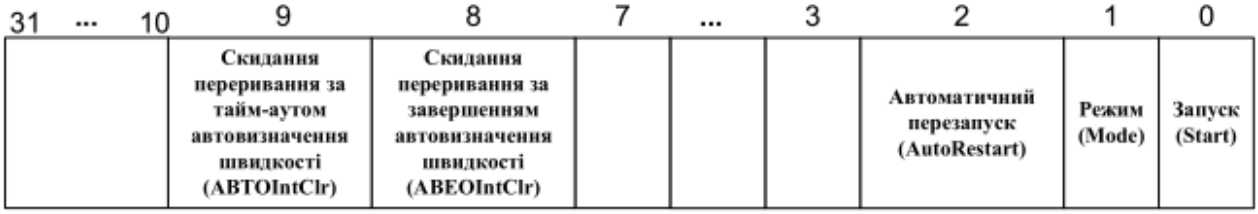


Рисунок 11 – Регістр автоматичного визначення швидкості обміну

Після встановлення біта Start регістра автоматичного визначення швидкості обміну модуль UART переходить до очікування символу «A» (0x16) чи «a» (0x41). Під час прийому символу за допомогою вбудованого таймера визначається інтервал між двома фронтами символу, на основі чого обчислюється швидкість обміну. Блок автоматичного визначення швидкості може працювати в 2 режимах. В режимі 1 (біт Mode встановлено) швидкість обміну визначається за тривалістю старт–біта, а режим 0 (біт Mode скинуто) – за загальною тривалістю старт–біта та 0–го біта слова даних. Обидва режими повністю взаємозамінні, тим не менше режим 0 більше підходить для роботи при високих швидкостях обміну, а режим 1 – при низьких.

Якщо потрібний символ не буде передано, то блок автоматичного визначення швидкості зафіксує тайм–аут. Якщо біт AutoRestart встановлено, то модуль UART буде старатись визначити швидкість обміну за наступним прийнятим символом. Якщо потрібно, модуль UART може згенерувати переривання при успішному визначені швидкості і при виявлені тайм–ауту.

1.5.4 Передача даних

Після ініціалізації модуля UART дані можна передавати, записуючи їх в регістр зберігання передавача (THR). Аналогічно, дані можна приймати, зчитуючи їх з регістра буфера приймача (RBR). Насправді обидва регістри мають однакову адресу в пам'яті. Запис за цією адресою записує символ в FIFO–буфер передачі, а читання за цією адресою завантажує символ з FIFO–буфера прийому. Нижче наведено приклади підпрограм, які здійснюють передачу і прийом символів в/з UART.

```
Int putchar (int ch)                    /* Записує символ в послідовний порт */
{
if {ch == '\n')
{
while (!(U0LSR & 0x20));
U0THR = CR; /* Виводимо CR */
}
while (!(U0LSQ & 0x20)) ;
return {U0THR = ch);
}
Int getchar (void)                      /* Читає символ з послідовного порту */
while (!(U0LSR & 0x01));
return (U0RBR);
}
```

Функції putchar() и getchar() призначено для читання/запису одного символу з/в UART. Ці низькорівневі функції використовуються функціями бібліотеки STDIO компілятора Keil, такими як printf () та scanf (). Так що, якщо ви хочете перенаправити стандартне введення/виведення з UART, скажімо, на клавіатуру і РК – дисплей, перепишіть ці функції для підтримки прийому і передачі одиночних символів з/в необхідного пристрою введення/виведення. Обидві функції putchar () та igetchar () зчитують регістр стану лінії (LSR) для

виявлення помилок обміну за UART, а також для контролю стану FIFO–буферів прийому і передачі (рисунок 12).

Регістр LSR містить набір прапорців, які відображають стан UART. Їх стан можна відслідковувати в циклі або ж зчитувати у підпрограмі обробки переривання від UART.

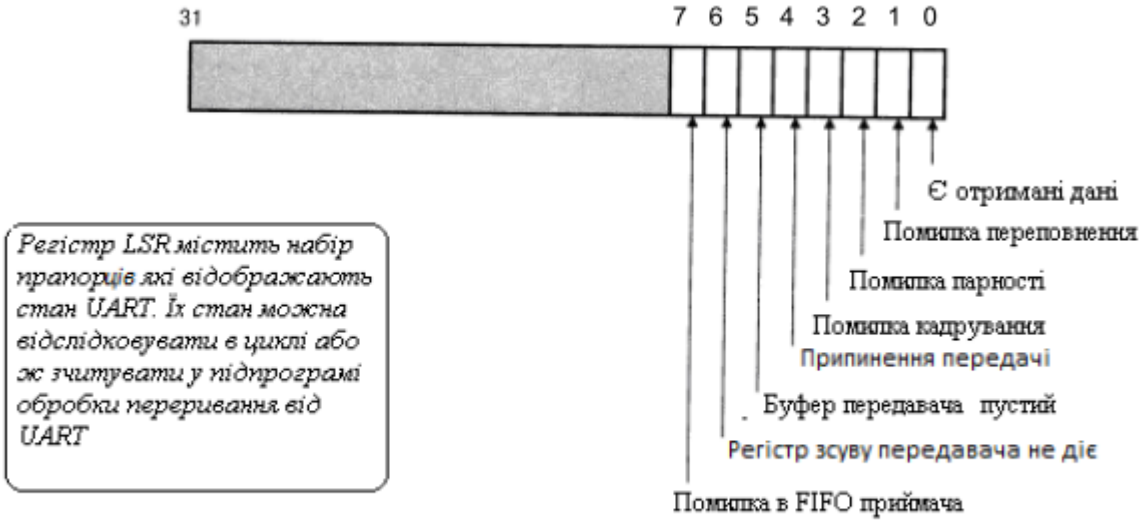


Рисунок 12 – Регістр стану лінії UART

У контролері переривань модулю UART виділено один канал, при цьому переривання може генеруватися трьома джерелами. По–перше, переривання UART може генеруватися при зміні стану каналу прийому. Тобто, переривання буде згенеровано при виникненні будь–якої помилки, для визначення якої можна прочитати регістр LSR. Решта два джерела переривань – це переривання приймача і передавача. Переривання за прийомом ініціюється символами, які поміщаються в FIFO–буфер прийому (рисунок 13). Глибина буфера, при якій буде генеруватися переривання, задається в регістрі керування FIFO.

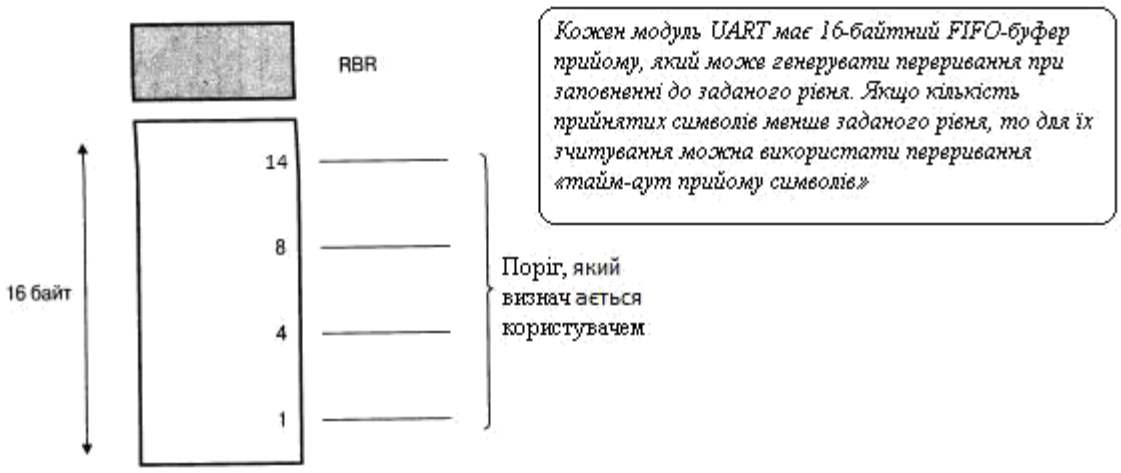


Рисунок 13 – FIF0–буфер прийому

Кожен модуль UART має 16–байтний FIFO–буфер прийому, який може генерувати переривання при заповненні до заданого рівня. Якщо кількість прийнятих символів менше заданого рівня, то для їх зчитування можна використати переривання «тайм–аут прийому символів».

Переривання за прийомом може генеруватися після прийому 1, 4, 8 або 14 символів. Нехай задана генерація переривання при появі в буфері восьми символів. Тоді при прийомі 34–байтного пакета буде згенеровано чотири переривання, і два символи залишаться в FIFO–буфері. Ці символи викличуть переривання «тайм–аут прийому символів» (CTI).

Переривання CTI виникає, якщо в FIFO–буфері є один або більше символів і стан буфера не змінюється протягом інтервалу, який дорівнює часу передачі 3,5 ... 4,5 символів.

FIFO–буфер передачі (рисунок 14) генерує переривання при спустошенні регістра зберігання передавача, а також при спустошенні регістра зсуву передавача.

Як і FIFO–буфер прийому, FIFO–буфер передачі має глибину 16 байт. FIFO–буфер передачі може генерувати переривання при спустошенні та при завершенні передачі.

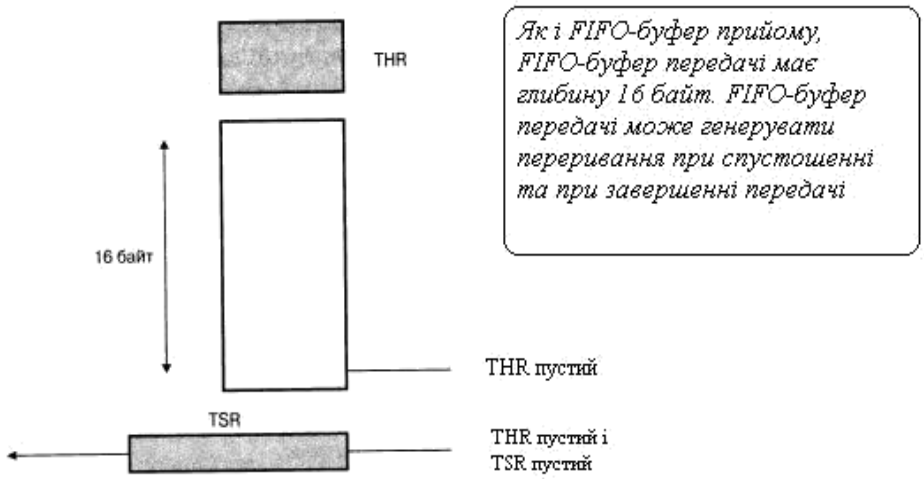


Рисунок 14– FIFO–буфер передачі

Модуль UART1 має таку ж базову структуру, що і UART0, однак до того ж він дозволяє керувати модемом (рисунок 15).

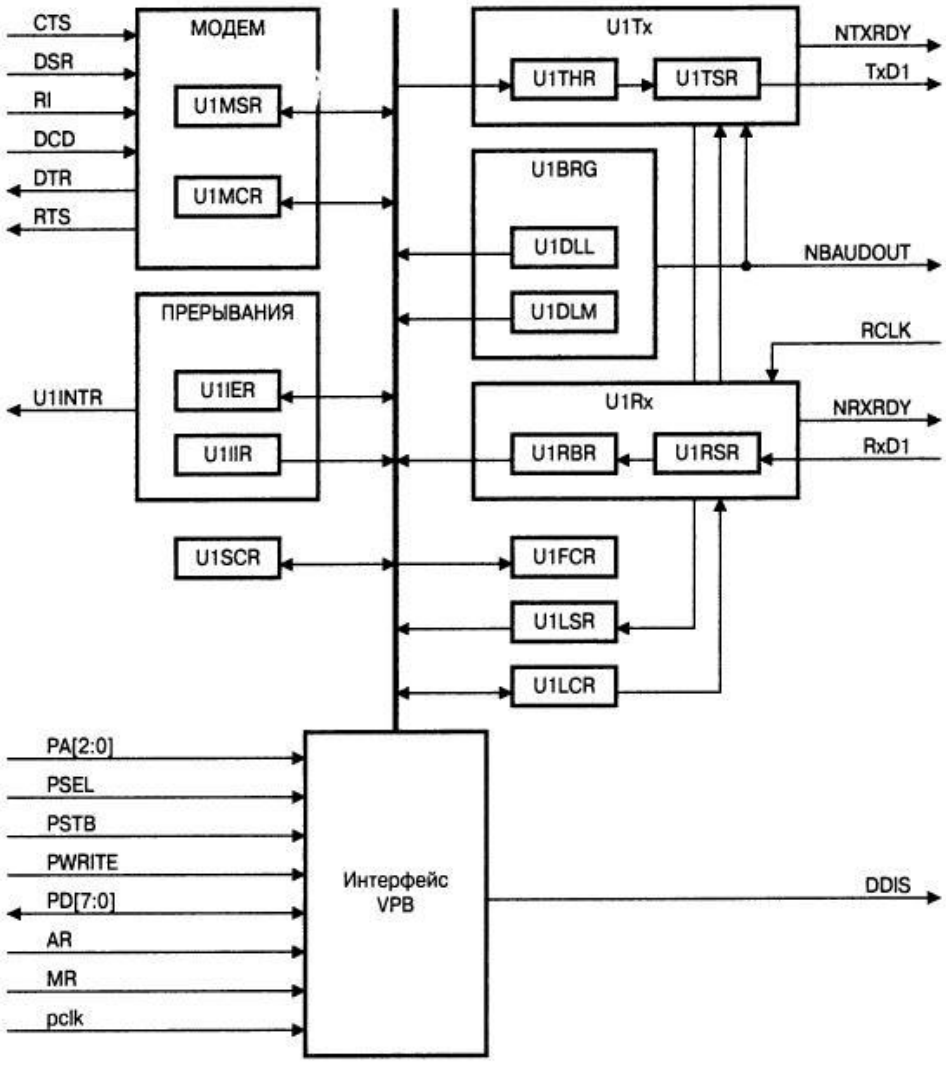


Рисунок 15 – Спрощена архітектура модуля UART1

Для підтримки повного модемного інтерфейсу (CTS, DCD, DSR, DTR, RI, RTS) використовуються додаткові виводи мікроконтролера, а в модулі є два додаткових регістри (регістр керування модемом і регістр стану модему) і додаткове джерело переривання, що генерує переривання при зміні стану модему (рисунок 16).

Ці додаткові можливості забезпечують оптимальне підключення до модему з генерацією переривання при кожній зміні регістра стану модему.

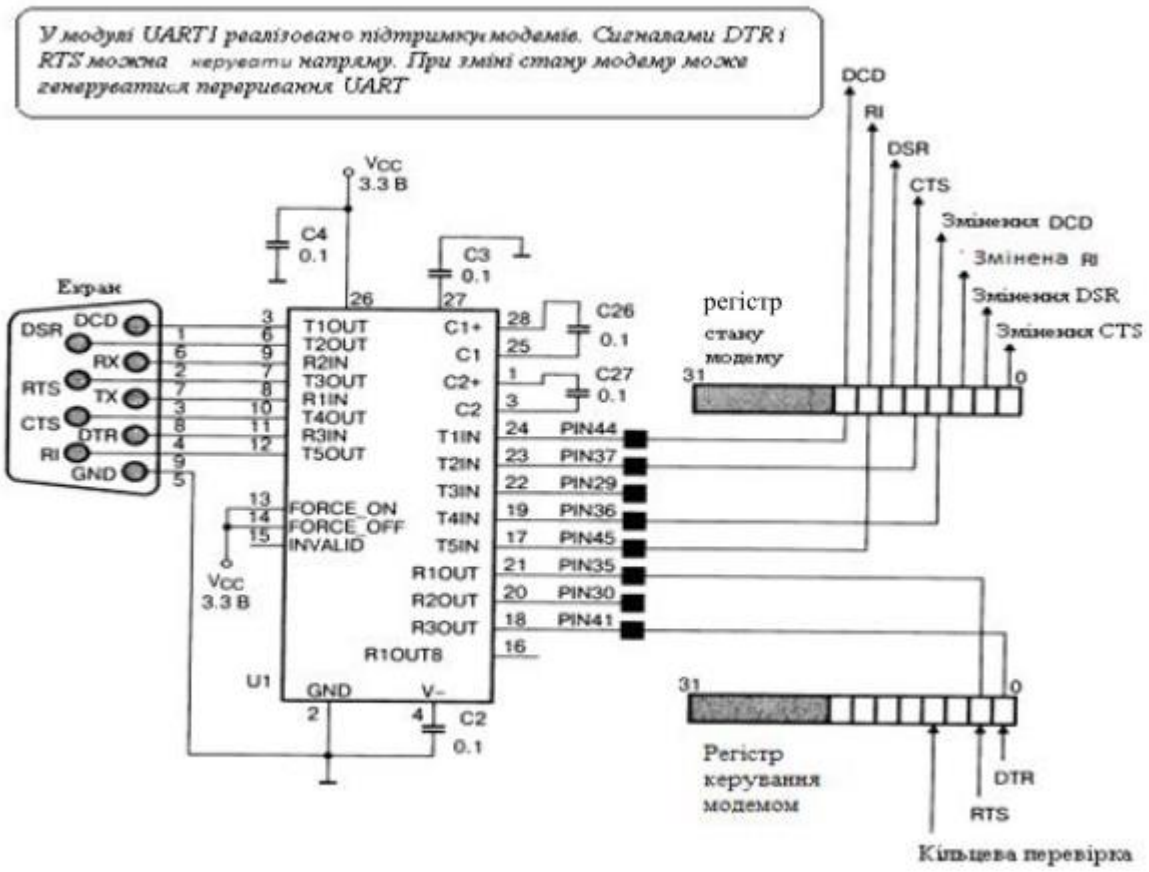


Рисунок 16 – Керування модемом модуля UART1

1.5.5 Організація обміну за протоколом IrDA

В модулі UART3 присутній додатковий керуючий регістр, який дозволяє керувати формуванням імпульсів на виході передавача в відповідності з інфрачервоним стандартом IrDA (рисунок 17).

Регістр керування IrDA використовується для керування модулем UART3 в відповідності до стандартів IrDA. При цьому є можливість інвертування імпульсів, за допомогою яких передаються дані. Тривалість імпульсів можна задавати, як стандартну (3/швидкість обміну), або кратну періоду сигнал Pclk.

Рисунок 17 – Регістр керування IrDA

1.5.6 Опис керуючих регістрів UART

Карту регістрів UART0 представлено в таблиці 1.

Таблиця 1 – Карта регістрів UART0

Назва	Опис	Доступ	Значення після ски-дання	Адреса
U0RBR	Регістр буфера прийому	RO	NA	0xE000C000 DLAB = 0
U0THR	Регістр буфера передачі	WO	NA	0xE000C000 DLAB = 0
U0IER	Регістр дозволу переривання	R/W	0	0xE000C004 DLAB = 0
U0IIR	Регістр ідентифікації пере-ривання	RO	0x01	0xE000C008
U0FCR	Регістр керування буфером FIFO	WO	0	0xE000C008
U0LCR	Регістр керування лінією	R/W	0	0xE000C00C
U0LSR	Регістр стану лінії	RO	0x06	0xE000C014
U0SCR	Робочий регістр клавіатури	R/W	0	0xE000C01C
U0DDL	Молодший байт коефіцієнта подільника	R/W	0x01	0xE000C000 DLAB = 1
U0DLM	Старший байт коефіцієнта подільника	R/W	0	0xE000C004 DLAB = 1

1.5.7 Регістр U0RBR

UART0_RBR

Опис	Доступ	Скинуте значення	Адреса
Приймальний буферний регістр	RO	невизначене	0×E000C000

U0RBR є вищим байтом UART0 Rx FIFO. Вищий байт з Rx FIFO містить "найстаріший" отриманий символ і може бути прочитаний через інтерфейс шини. LSB (біт 0) представляє "найстаріший" отриманий біт. Якщо отриманий символ менше 8 біт, то не використовувані MSB заповнюються нулями.

1.5.8 Регістр U0DLL

UART0_ DLL

Опис	Доступ	Скинуте значення	Адреса
Защіпка подільника (LSB)	R/W	0×01	0×E000C000

Опис бітів регістра U0DLL представлено в таблиці 2

Таблиця 2– Опис бітів регістра U0DLL

U0DLL	Функції	Опис	Скинуте значення
7:0	Регістр LSB подільника	Регістр, що містить молодший байт подільника, який разом з регістром U0DLM задає швидкість обміну через UART0	0x01

1.5.9 Регістр U0DLM

UART0_DLM

Опис	Доступ	Скинуте значення	Адреса
Защіпка подільника MSB	R/W	0	0×E000C004

Опис бітів регістра U0DLM представлено в таблиці 3

Таблиця 3 – Опис бітів регістра U0DLM

U0DLM	Функції	Опис	Скинуте значення
7:0	Регістр MSB подільника	Регістр, що містить старший байт подільника, який разом з регістром U0DLL задає швидкість обміну через UART0	0

16–розрядний подільник є частиною генератора–контролера швидкості передачі UART і зберігає значення, яке використовується для формування тактового сигналу модуля UART з сигналу Pclk, який повинен бути в 16 разів більше потрібної швидкості обміну BAUD. Регістри U0DLL і U0DLM утворюють 16 бітний подільник, де U0DLL містить нижні 8 біт, а U0DLM верхні 8 біт дільника.

1.5.10 Регістр U0IER

Опис	Доступ	Скинуте значення	Адреса
Регістр дозволу переривання	R/W	0	0×E000C004

U0IER дозволяє переривання UART0. Опис бітів регістра U0IER описано в таблиці 4.

Таблиця 4 – Опис бітів регістра U0IER

U0IER	Функції	Опис	Скинуте значення
0	Дозвіл переривання "прийняті дані доступні" (RBR)	0: заборонити переривання. 1: дозволити переривання за наявністю прийнятих даних	0
1	Дозвіл переривання "буфер регістра передавача порожній"	0: заборонити переривання. 1: дозволити переривання, коли регістр зберігання передавача порожній	0
2	Дозвіл переривання "за станом лінії прийому Rx"	0: заборонити переривання за станом лінії прийому. 1: дозволити переривання за станом лінії прийому	0
	Зарезервовані	Програмне забезпечення користувача не повинно записувати данні у зарезервовані біти	N/A

1.5.11 Регістр U0IIR

UART0_IIR

Опис	Доступ	Скинуте значення	Адреса
Регістр ідентифікації переривання	RO	0x01	0×E000C008

U0IIR ідентифікує переривання UART0. Опис бітів регістра U0IIR представлено в таблиці 5.

Таблиця 5 – Опис бітів регістра U0IIR

U0IIR	Функції	Опис	Скинуте значення
0	Наявність перери-вання	0: наявність переривання. 1: відсутність переривання.	1
3:1	ID пере-ривання	011: переривання за станом лінії прийому. 010: прийняті дані доступні". 110: тайм–аут індикації прийнятого сим-волу(CTI) 001: буфер передавача порожній. Всі інші варіанти зарезервовані.	0
5:4	Зарезерво-вані	Програмне забезпечення користувача не повинно записувати дані у зарезервовані біти	N/A
7:6	Дозвіл бу-фера FIFO	Ці біти еквівалентні біту U0FCR[0]	0

1.5.12 Регістр U0FCR

UART0_FCR

Опис	Доступ	Скинуте значен-ня	Адреса
FIFO регістр ке-рування	R/W	0	0×E000C008

U0FCR керує діями UART0 Rx та Tx FIFO. Опис бітів регістра U0FCR представлено в таблиці 6.

Таблиця 6 – Опис бітів регістра U0FCR

U0FCR	Функції	Опис	Скинуте значення
0	Дозвіл буфера FIFO	0: забороняє приймальний та передавальний буфери FIFO. 1: дозволяє приймальний та передавальний буфери FIFO.	0
1	Скидання приймального буфера FIFO	Запис логічної 1 в цей біт очищає всі байти в приймальному буфері FIFO UART0 і скидає логічну схему вказівника	0
2	Скидання передавального буфера FIFO	Запис логічної 1 в цей біт очищає всі байти в передавальному буфері FIFO UART0 і скидає логічну схему вказівника	0
5:3	Зарезервовані	Програмне забезпечення користувача не повинно записувати данні у зарезервовані біти	N/A
7:6	Вибір рівня спрацьовування тригера приймаль ного буфера FIFO	00: рівень тригера 0 (за замовчуванням прийнято 1 символ). 01: рівень тригера 1 (за замовчуванням прийнято 4 символи). 10: рівень тригера 2 (за замовчуванням прийнято 8 символів). 11: рівень тригера 3 (за замовчуванням прийнято 14 символів). Ці біти визначають, скільки символів буде записано в приймальний буфер FIFO UART0 перед тим, я буде активовано переривання.	0

1.5.13 Регістр U0LCR

UART0_LCR

Опис	Доступ	Скинуте значення	Адреса
Регістр керування лінією	R/W	0	0×E000C00C

U0LCR визначає формат інформаційних символів, які передаються або отримуються. Опис бітів регістра U0LCR представлено в таблиці 71.

Таблиця 7 – Опис бітів регістра U0LCR

UOLCR	Функції	Опис	Скинуте значення
1:0	Вибір довжини слова	00: довжина слова – 5 біт. 01: довжина слова – 6 біт. 10: довжина слова – 7 біт. 11: довжина слова – 8 біт.	0
2	Вибір кількості стопових бітів	0: 1 стоповий біт 1: 2 стопових біта (1,5, якщо U0LCR[1:0] = 00)	0
3	Дозвіл контролю парності	0: відключення генерації контролю парності та його перевірки 1: дозвіл генерації контролю парності та його перевірки	0
5:4	Вибір режиму контролю парності	00: доповнення та перевірка на непарність 01: доповнення та перевірка на парність 10: доповнення до непарності перевірка на парність 11: доповнення до парності перевірка на непарності (див. таблицю 4.22)	0
6	Керування припиненням передачі	0: заборонити припинення передачі, 1: дозволити припинення передачі	0
7	Доступ до регістра подільника	0: заборонити доступ до регістра подільника 1: дозволити доступ до регістра подільника	0

Вплив розрядів D5, D5 на вид перевірки при прийомі описано в таблиці 8.

Таблиця 8 – Вплив розрядів D5, D5 на вид перевірки при прийомі

D3	D4	D5	Доповнення при передачі до:	Перевірка при прийомі на:
0	*	*	Ні	Ні
1	0	0	Непарного числа	Непарне число
1	1	0	Парного числа	Парне число
1	0	1	Непарного числа	Парне число
1	1	1	Парного числа	Непарне число

Коли біт припинення передачі дорівнює логічній 1, то послідовний вихід (SOUT) встановлюється в стан пропуску (логічного 0) і, незалежно від інших сигналів керування передачею, знаходиться в цьому стані до скидання біта 6 в початкове нульове значення. Ця особливість дозволяє МП – ру керувати процесом підключення терміналу (модему) до системи зв'язку.

1.5.14 Регістр U0LSR

UART0_LSR

Опис	Доступ	Скинуте значення	Адреса
Регістр стану лінії	R0	0× 60	0× E000C014

UOLSR – регістр використовується тільки для читання та забезпечує інформацію про стан UART0 Rx та Tx блоків. Біти регістра U0LSR описано в таблиці 9.

Модуль UART2 та UART3 мікроконтролера LPC2378 програмуються аналогічно.

Таблиця 9 – Опис бітів регістра U0LSR

UOLSR	Функції	Опис	Скинуте значення
0	Дані приймача готові	0: U0RBR пустий. 1: U0RBR містить достовірні дані. Скидання біта U0LSR[0] в стан логічного 0 проводиться або при читанні мікропроцесором даних з буферного регістра приймача, або при програмному запису логічного 0 в цей біт.	0
1	Помилка переповнення	0: стан помилки переповнення неактивний. 1: стан помилки переповнення активний. Біт U0LSR[1] встановлюється, якщо на приймач U0RSR UART0 поступив новий символ, а приймальний буфер UART0 повний. Біт скидається при зчитуванні регістра U0LSR. При цьому відбувається втрата попереднього прийнятого символу.	0
2	Помилка парності	0: стан помилки парності неактивний. 1: стан помилки парності активний. Біт U0LSR[2] встановлюється, якщо біт парності отриманого символу знаходиться в неправильному стані. Біт скидається при зчитуванні регістра U0LSR.	0
3	Помилка кадрової синхронізації	0: стан помилки кадрової синхронізації неактивний. 1: стан помилки кадрової синхронізації активний. Біт U0LSR[3] встановлюється, якщо стоп–біт отриманого символу дорівнює логічному 0. Біт скидається при зчитуванні регістра U0LSR.	0
4	Припинення передачі (BI)	0: стан припинення передачі неактивний. 1: стан припинення передачі активний. Біт U0LSR[4] встановлюється в стан логічної 1, якщо вхід даних, що приймаються, утримується в стані пропуску (логічний 0) впродовж часу, що перевищує передачу: старт–біта, бітів даних, паритету і стоп–біта/стоп–бітів. Біт скидається при зчитуванні регістра U0LSR.	0

Продовження таблиці 9

4	Припинення передачі (BI)	0: стан припинення передачі неактивний. 1: стан припинення передачі активний. Біт U0LSR[4] встановлюється в стан логічної 1, якщо вхід даних, що приймаються, утримується в стані пропуску (логічний 0) впродовж часу, що перевищує передачу: старт–біта, бітів даних, паритету і стоп–біта/стоп–бітів. Біт скидається при зчитуванні регістра U0LSR.	0
5	Буфер передавача пустий (THRE)	0: U0THR містить достовірні данні. 1: U0THR пустий. Біт U0LSR[5] встановлюється, якщо регістр U0THR UART0 пустий і скидається при записі даних в регістр U0THR .	1
6	Регістр зсуву передавача не діє (TEMT)	Цей біт є індикатором того, що регістр зсуву передавача не діє (не працює). Він скидається в стан логічного 0 при записі даних з регістра зберігання передавача в регістр зсуву передавача. D6 може бути доступний тільки при читанні. Біт скидається при записі даних в регістр U0THR	1
7	Помилка приймального буфера FIFO (RxFE)	0: U0RBR не містить символи з помилками прийому через UART0 або біт U0FCR[0] = 0. 1: UART0 U0RBR містить символи с помилками прийому через UART0. Біт U0LSR[7] встановлюється, якщо символ з помилкою прийому поміщається в регістр U0RBR . Біт скидається при зчитуванні регістра U0LSR.	0

Нижче розглядаються відмінності, які мають керуючі регістри модуля UART1. Карта регістра UART1 представлена в таблиці 10.

Таблиця 10 – Керуючі регістри модуля UART1

Назва	**Опис**	**Доступ**	**Значення після скидання**	**Адреса**
U1RBR	Регістр буфера прийому	RO	NA	0xE0010000 DLAB = 0
U1THR	Регістр буфера передачі	WO	NA	0xE0010000 DLAB = 0

Продовження таблиці 10

U1IER	Регістр дозволу переривання	R/W	0	0xE0010004 DLAB = 0
U1IIR	Регістр ідентифікації переривання	RO	0x01	0xE0010008
U1FCR	Регістр керування буфером FIFO	WO	0	0xE0010008
U1LCR	Регістр керування лінією	R/W	0	0xE001000C
U1MCR	Регістр керування модемом	R/W	0	0xE0010010
U1LSR	Регістр стану лінії	RO	0x06	0xE0010014
U1MSR	Регістр стану модему	RO	0	0xE0010018
U1SCR	Робочий регістр клавіатури	R/W	0	0xE001001C
U1DDL	Молодший байт коефіцієнту подільника	R/W	0x01	0xE0010000 DLAB = 1
U1DLM	Старший байт коефіцієнту подільника	R/W	0	0xE0010004 DLAB = 1

1.5.15 Регістр U1MCR

UART1_MCR

Опис	**Доступ**	**Скинуте значення**	**Адреса**
Регістр керування модемом	R/W	0	0xE0010010

U1MCR – регістр керування модемом UART1. Біти регістра U1MCR описано в таблиці 11

Таблиця 11 – Опис бітів регістра U1MCR

U1MCR	Функції	Опис
0	Керування станом виходу DTR (запит передавача терміналу – ЗПДТ)	Керує станом виходу DTR. Читання цього біта повертає значення логічного 0, якщо активовано режим модемного шлейфу (петлі), в якому відбувається взаємне “замикання” сигналів модемного інтерфейсу (виходи замикаються на виходи)
1	Керування станом виходу RTS (запит приймача терміналу – ЗПРМТ)	Керує станом виходу RTS. Читання цього біта повертає значення логічного 0, якщо активовано режим модемного шлейфу (петлі), в якому відбувається взаємне “замикання” сигналів модемного інтерфейсу (виходи замикаються на входи)
3:2	Зарезервовані	Програмне забезпечення користувача не повинно записувати данні у зарезервовані біти
4	Вибір режиму “шлейфу” (петлі)	0: заборонити режим модемного “шлейфу”. 1: дозволити режим модемного “шлейфу”. В режимі модемного “шлейфу” відбувається діагностичний тест логічних схем нтерфейсу. При цьому, дані з виходу передавача подаються в послідовному вигляді на вхід приймача. В цьому режимі вхід RxD1 не задіяний, а вихід TxD1 утримується в стані 1. Чотири керуючих сигнали від модема (CTS, DSR, RI, DCD) відключаються від виводів мікроконтролера. Керуючі сигнали для модема (RTS, DTR) підключаються до відповідних входів від модема. В результаті такої комутації, значення старших 4 бітів регістра U1MSR

Продовження таблиці 11

		визначається значеннями молодших чотирьох бітів U1MCR.
7:5	Зарезервовані	Програмне забезпечення користувача не повинно записувати дані у зарезервовані біти

1.5.16 Регістр U1MSR

UART1_MSR

Опис	Доступ	Скинуте значення	Адреса
Регістр стану модему	R0	0	0xE0010018

U1MSR – регістр тільки для читання, який забезпечує інформацію про стан модему UART1. Біти регістра U1MSR описано в таблиці 12

Таблиця 12 – Опис бітів регістра U1MSR

U1MSR	Функції	Опис	Скинуте значення
0	Зміна сигналу на вході CTS	0: на вході CTS від модему не було виявлено змін. 1: на вході CTS від модему було виявлено зміни. Біт скидається при читанні регістра U1MSR	0

Продовження таблиці 12

1	Зміна сигналу на вході DSR	0: на вході DSR від модему не було виявлено змін. 1: на вході DSR від модему було виявлено зміни. Біт скидається при читанні регістра U1MSR	0
2	Зміна сигналу на вході RI	0: на вході RI від модему не було виявлено змін. 1: на вході RI від модему було виявлено зміни. Біт скидається при читанні регістра U1MSR	0
3	Зміна сигналу на вході DCD	0: на вході DCD від модему не було виявлено змін. 1: на вході DCD від модему було виявлено зміни. Біт скидається при читанні регістра U1MSR	0
4	Готовність приймача терміналу (ГПРМТ) CTS	Відображає значення вхідного сигналу CTS. В режимі модемного шлейфу значення цього біта визначає стан біта U1MCR[1]	0
5	Готовність передавача терміналу (ГПДТ) DSR	Відображає значення вхідного сигналу DSR. В режимі модемного шлейфу значення цього біта визначає стан біта U1MCR[0]	0

Продовження таблиці 12

6	RI	Відображає значення вхідного сигналу RI. В режимі модемного шлейфу значення цього біта визначає стан біта U1MCR[2]	0
7	DCD	Відображає значення вхідного сигналу DCD. В режимі модемного шлейфу значення цього біта визначає стан біта U1MCR[3]	0

1.5.17 Налаштування швидкості обміну інформації між МК ARM7 та інтерфейсом УАПП

Розглянемо приклад написання мовою Асемблер фрагменту програми, який забезпечує потрібну швидкість обміну інформацією між МК – м ARM7 та інтерфейсом УАПП TL16C450/550; розрахувати значення коефіцієнта ділення подільника інтерфейсу та відносну похибку отриманої швидкості від заданої.

Вихідні дані: швидкість обміну Vпд = 9600 бод; частота тактового генератора шини VPB мікроконтролера: P_{CLK}=30 МГц.

Величина швидкості передачі даних Vпд визначається виразом:

$$Vпд = P_{CLK}/(k_{діл}*16),$$

де $k_{діл}$ – значення коефіцієнта ділення, яке повинно програмно завантажуватися у двобайтний регістр подільника УАПП, P_{CLK} – тактовий сигнал шини VPB мікроконтролера.

При Vпд = 9600 бод та P_{CLK} = 30 МГц $k_{діл}$=195=0x00C3,

тоді Vпд. реальна = 9615,39 бод.

Відносна похибка

$$\delta = \frac{V_{ПД\ реальна} - V_{ПД}}{V_{ПД}} \cdot 100\% = \frac{9615,39 - 9600}{9600} \cdot 100\$ = 0,16\%.$$

Для програмування заданої швидкості необхідно завантажити:

молодший байт $k_{діл}$ = C3h в молодший байт подільника (U0DLL), а старший байт $k_{діл}$ = 00h в старший байт подільника (U0DLM) модуля УАПП МК– ра.

Для написання фрагменту програми потрібно знати адресу регістра подільника УАПП, та керуючого регістра U0LCR, які дорівнюють:

0xE000C00C – для регістра U0LCR;

0xE000C000 – для регістра U0DLL;

0xE000C004 – для регістра U0DLM.

Команди мікроконтролера ARM7, які використовують безпосередній операнд, мають декілька особливостей. Якщо довжина операнда дорівнює не більше восьми біт, то цей операнд без проблем компілюється і займає молодший байт машинного коду команди. Довжина безпосереднього операнда може складати до 32–х розрядів, але інформативна частина цього операнда повинна займати дві сусідні тетради (1 байт). Цей 32–х розрядний операнд при компіляції займає 12 бітів машинного коду команди, з яких вісім молодших – це інформативна частина, а наступні 4 розряди – це значення циклічного зсуву (максимальне значення – 16). При виконанні такої команди процесор циклічно зсуває інформативний байт вправо на величину, яку записано у 4–х розрядний зсув та помножену на 2 (максимальне значення: 16·2=32).

В результаті отримується початковий 32–х розрядний безпосередній операнд.

Для звернення до регістрів подільника U0DLL та U0DLM треба встановити біт DLAB у 1. DLAB – це старший розряд регістра керування лінією U0LCR.

Для встановлення DLAB в 1 програма містить 6 команд:

```
MOV R1, #0xE0000000   ; R1 ← #0xE0000000,
MOV R2, #0x0000C000   ; R2 ← #0x0000C000,
ADD R1, R1, R2        ; R1 ← R1+R2,
ADD R1, R1, #0x0C     ; R1 ← R1+0x0C,
MOV R2, #0x80         ; R2 ← #0x80,
STRB R2, [R1]         ; M (R1)←R2.
```

В останній команді вміст регістра R2, який дорівнює 0x80, завантажується у пам’ять з адресою регістра U0LCR, яка дорівнює #0xE000C00C. Після цього наступні одинадцять команд завантажують в регістр подільника потрібний коефіцієнт ділення.

Завантаження молодшого байта:
регістр U0DLL ← МБ $k_{діл}$ = C3h;

```
MOV R1, #0xE0000000       ; R1 ← #0xE0000000,
MOV R2, #0x0000C000       ; R2 ← #0x0000C000,
ADR R1, R1, R2            ; R1← R1+R2,
MOV R2, #0xC3             ; R2 ← 0xC3,
STRB R2, [R1]             ; M(R1) ← R2.
```

Завантаження старшого байта:
регістр U0DLM ← СБ $k_{діл}$ = 00h;

```
MOV R1,#0xE0000000        ; R1 ← 0xE0000000,
MOV R2,#0x0000C000        ; R2 ← 0x0000C000,
ADD R1, R1, R2            ; R1 ← R1+R2,
ADD R1, R1, #0x04         ; R1 ← R1+0x04,
MOV R2, #0x00             ; R2 ← 0x00,
STRB R2, [R1]             ; M(R1)←R2.
```

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Опишіть структурну схему сполучення МП/МК з модемом за допомогою інтерфейсу RS – 232C.
2. Скільки модулів УАПП мають мікроконтролери сімейства LPC2300?
3. Які особливості мають модулі UART1 та UART3?
4. Назвіть основні керуючі регістри модуля UART.
5. Як виконується ініціалізація модуля UART?
6. Як формується тактовий сигнал модуля UART?
7. Як здійснюється керування функцією автоматичного визначення швидкості передачі пакету даних, який приймається?
8. В який регістр модуля UART записуються дані для передачі?
9. Прапорці якого регістра відображають стан модуля UART?
10. Як програмується потрібна швидкість обміну в модулі UART?
11. Які переривання може генерувати модуль UART?
12. Поясніть особливості організації обміну за протоколом IrDA.
13. Опишіть призначення бітів регістра U0RBR.
14. Опишіть призначення бітів регістрів U0DLL та U0DLM.
15. Опишіть призначення бітів регістрів U0IER та U0IIR.
16. Опишіть призначення бітів регістрів U0LCR та U0LSR.
17. Опишіть призначення бітів регістрів U1MCR та U1MSR.
18. Поясніть призначення основних контактів роз’єму RS–232C.
19. При якому значенні біта DLAB регістра LCR виконується програмування швидкості обміну?
20. Опишіть призначення та програмування FIF0–буферів передачі та прийому.
21. Які помилки прийому може виявляти модуль UART?
22. Як біти керуванням модемом можуть використовуватися для синхронізації обміну між мікропроцесором та терміналом (модемом)?