

## Лекція 4

### СИСТЕМА КОМАНД МІКРОКОНТРОЛЕРІВ AVR

Система команд AVR містить п'ять груп: умовного розгалуження, безумовного розгалуження, арифметичних і логічних операцій, команди пересилки даних, команди роботи з бітами. У версіях AVR мікроконтролерів сімейства Mega реалізована функція апаратного множення. Кількість команд навіть у межах одного сімейства AVR мікроконтролерів може змінюватися. Наприклад, для мікроконтролера Atmega103 до складу системи команд входить 121 інструкція.

Прості моделі AVR не мають деяких команд. Основна відмінність полягає в тому, що ті мікроконтролери (AT90S1200, Attiny10/11), у яких відсутня SRAM, не містять і відповідних команд роботи з оперативною пам'яттю. Крім того, AT90S1200 не має команд ADIW, SBIW, IJMP, ICALL, LPM, а Attiny10/11 – команд ADIW, SBIW, IJMP, ICALL, LPM, а ATtiny10/11 - команд ADIW, SBIW, IJMP, ICALL.

Мікроконтролери AVR сімейства Mega мають двобайтові, виконувані за три такти, команди абсолютних переходів JMP і CALL. В сімействах AVR мікроконтролерів Tiny і Classic ці команди не використовуються, тому що весь адресний простір обсягом до 4К слів досяжний за допомогою команд відносних переходів RJMP, RCALL. Також команда ELPM сторінкового читання FLASH пам'яті використовується лише в мікроконтролерах з великим обсягом пам'яті. Наприклад, вона входить до набору інструкцій мікроконтролера Atmega103 об'єм пам'яті програм якого має 128 Кбайт.

Спеціальна директива асемблера **device** <тип AVR> забезпечує контроль відповідності команд, використовуваних у тексті програми, типу зазначеного процесора.

### *Конфігурація пам'яті*

На рис. 1 показано як організована SRAM в мікроконтролерах AT90S2333/4433.

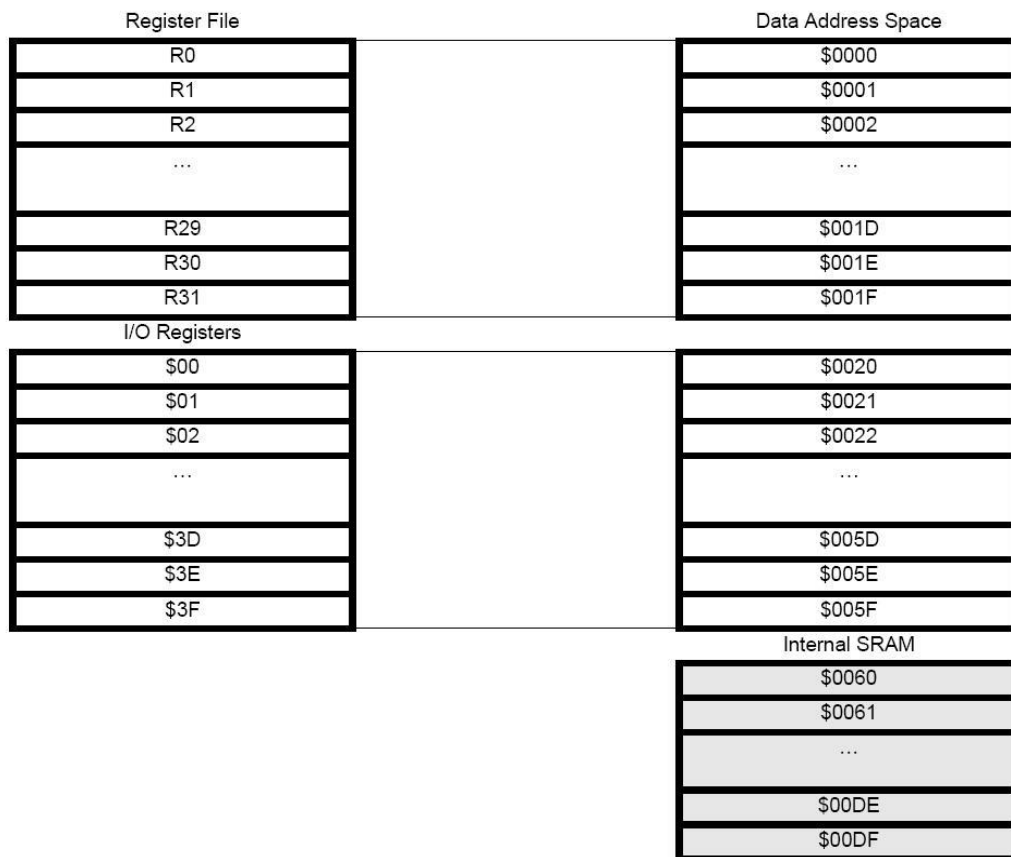


Рис. 1. Конфігурація пам'яті

На перших 224 адресах пам'яті даних розміщуються регістровий файл, простір пам'яті I/O і вбудована SRAM даних. Із них перші 96 адрес

займають регістровий файл і простір пам'яті I/O, в наступних 128 адресах розміщується вбудована SRAM.

При адресації пам'яті даних використовуються п'ять режимів адресації: безпосередня адресація, непряма зі зміщенням, непряма, непряма з переддекрементом і непряма з постдекрементом. Регістри з R26 по R31 регістрового файлу працюють як X, Y і Z регістри покажчика непрямої адресації.

Пряма адресація покриває всю пам'ять. Непряма адресація зі зміщенням доступних 63 адрес відносно базових адрес, які знаходяться в регістрах Y або Z. При використанні непрямої адресації з автоматичним переддекрементом і постдекрементом автоматично декрементуються і інкрементуються адреси, записані в регістри X, Y і Z. Всіма цими режимами перекривається весь адресний простір даних, включаючи 32 регістри загального призначення і 64 регістри I/O.

### **Режими адресації пам'яті програм і даних**

При зверненні до Flash пам'яті програм і пам'яті даних (SRAM, регістрового файлу і пам'яті I/O) AVR Enhanced RISC мікроконтролерами AT90S2333/4433 використовуються потужні і ефективні режими адресації. В даному розділі описуються режими адресації, які підтримуються AVR архітектурою. На рис. 2 OP означає частину слова команди, яка відповідає операційному коду.

### Безпосередня адресація, одиночний регістр Rd

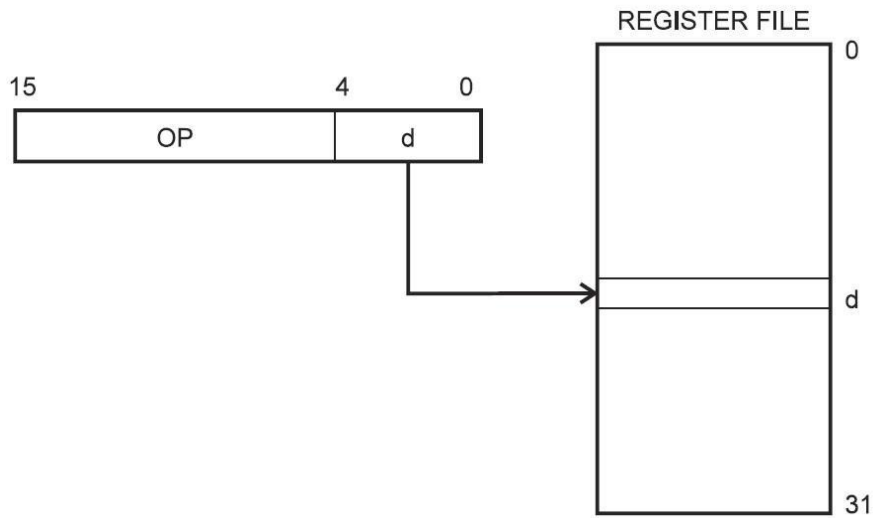


Рис. 2. Безпосередня адресація одного регістра

Операнд міститься в регістрі d (Rd).

### Безпосередня регістрова адресація, два регістра Rd і Rr

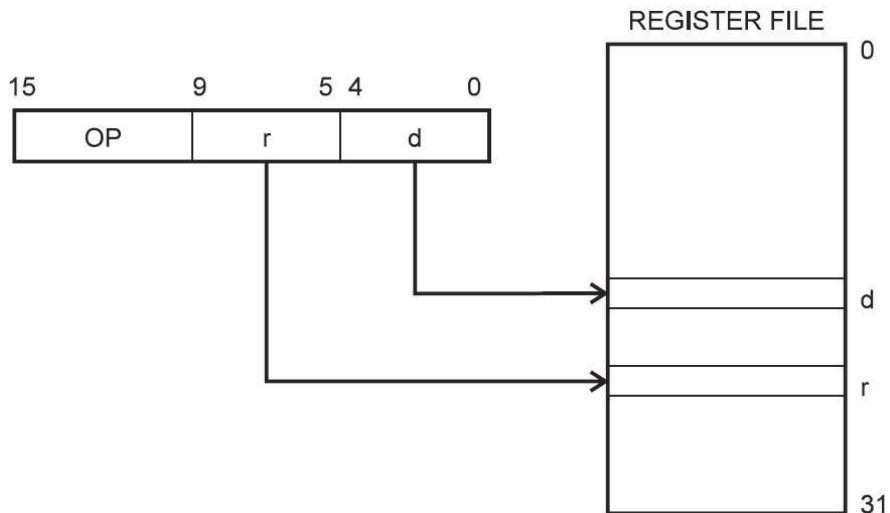


Рис. 3. Безпосередня регістрова адресація двох регістрів

Операнди містяться в регістрах r (Rr) і d (Rd).

Результат зберігається в регістрі d (Rd)

## Безпосередня адресація I/O

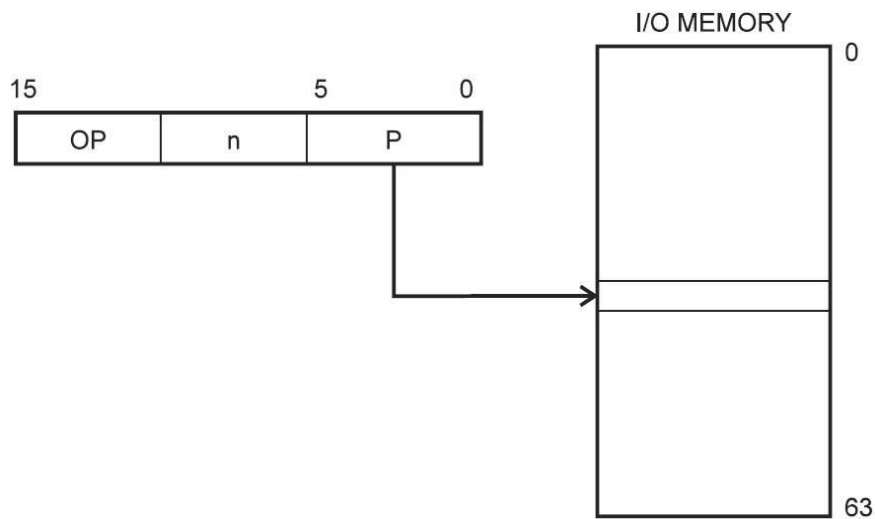


Рис. 4. Безпосередня адресація I/O

Адреса операнда міститься в 6 бітах слова команди. Величина  $n$  визначає адресу регістру джерела або регістру призначення.

## Безпосередня адресація даних

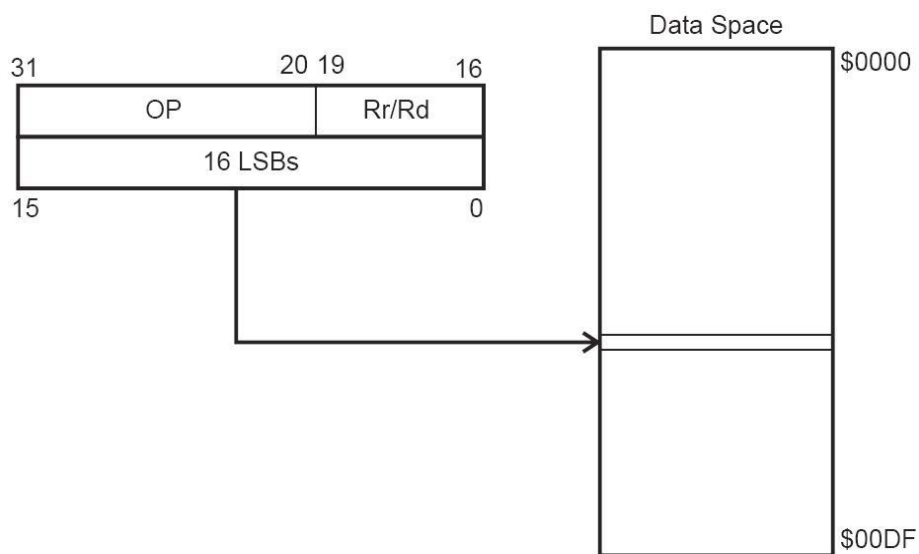


Рис. 5. Безпосередня адресація даних

16-розрядна адреса даних міститься в 16 молодших розрядах 32-розрядної команди. Rd/Rr визначають регістр джерело або регістр призначення.

### Непряма адресація даних зі зміщенням

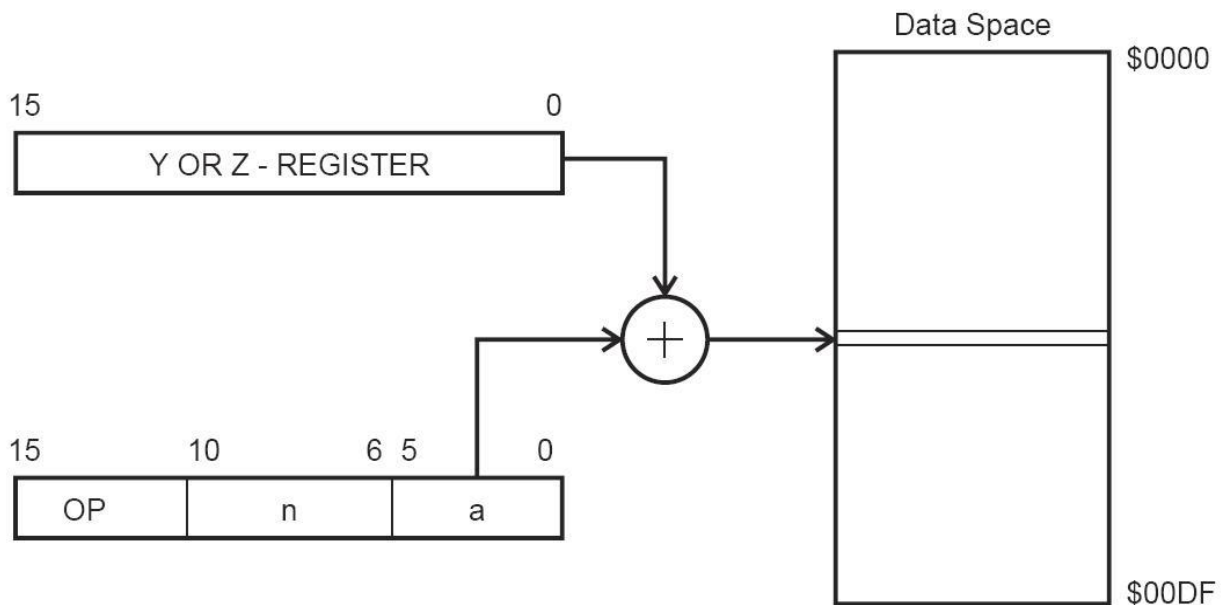


Рис. 6. Непряма адресація даних зі зміщенням

Адреса операнда вираховується підсумовуванням вмісту регістра Y або Z з 6 бітами адреси, які містяться в слові команди.

## Непряма адресація даних

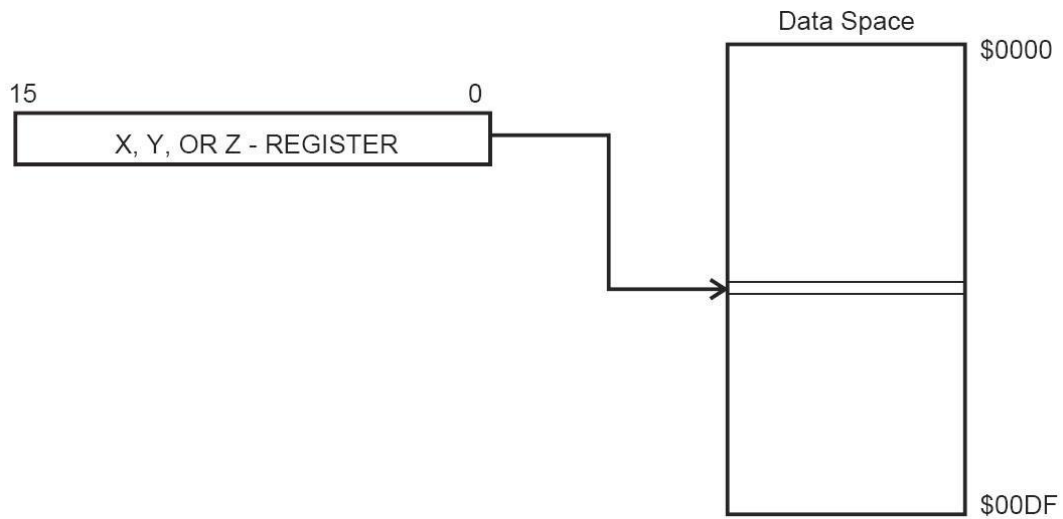


Рис. 7. Непряма адресація даних

Адреса операнда міститься в реєстрі X, Y або Z.

## Непряма адресація даних з переддекрементом

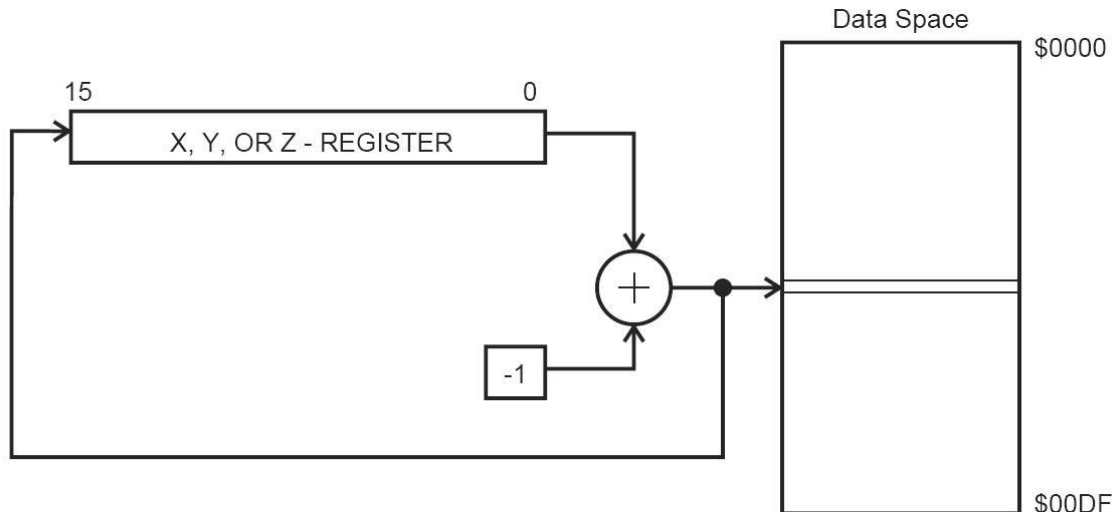


Рис. 8. Непряма адресація даних з переддекрементом

Перед виконанням операції реєстр X, Y або Z декрементується. Декрементований вміст реєстру X, Y або Z є адресою операнда.

## Непряма адресація даних з постінкрементом

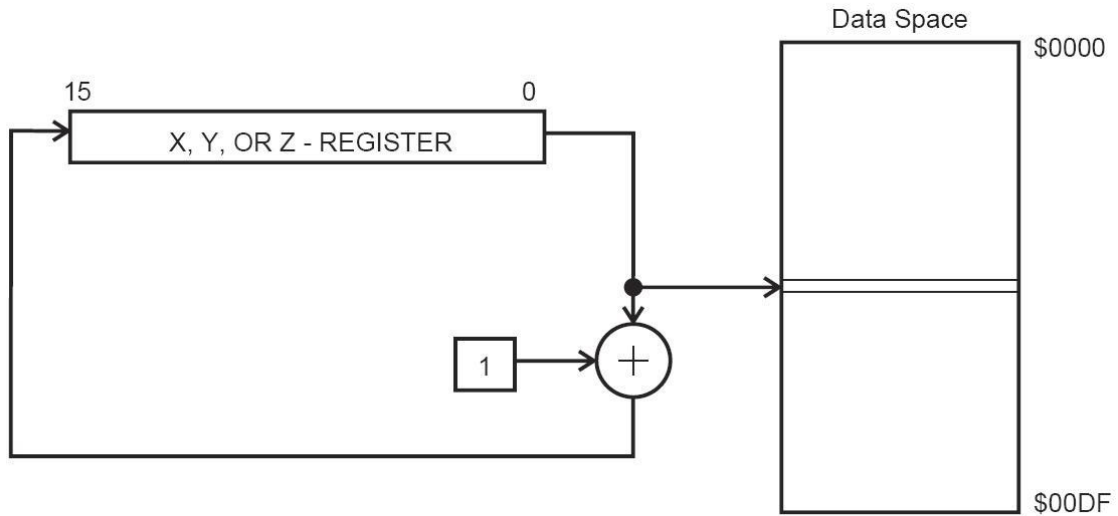


Рис. 9. Непряма адресація даних з постінкрементом

Після виконання операції реєстр X, Y або Z інкрементується. Адресою операнда є вміст X, Y або Z реєстра перед інкрементуванням.

## Адресація константи з використанням команд LPM

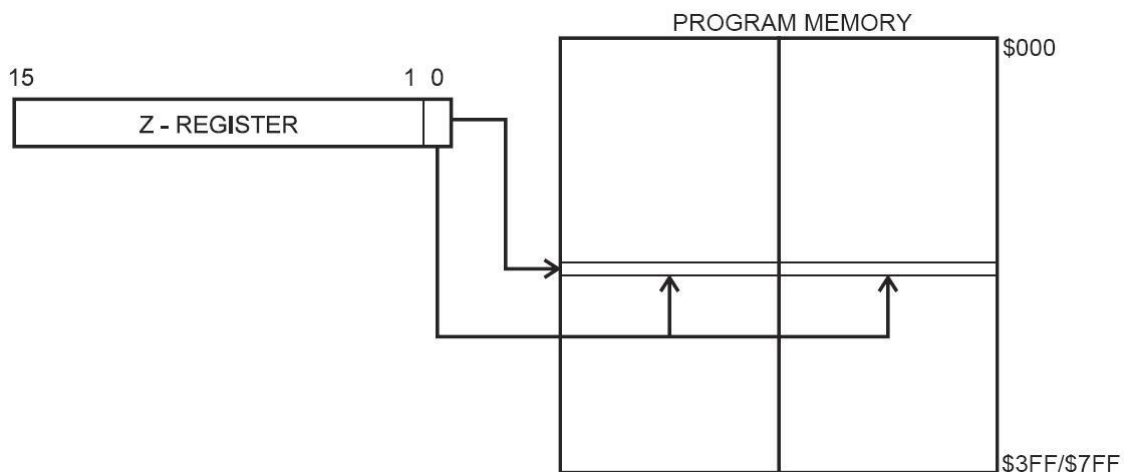


Рис. 10. Адресація константи коду пам'яті



Адреса байта константи визначається вмістом регістра Z. Старші 15 бітів визначають слово адреси (от 0 до 32К). Стан молодшого біта визначає вибір молодшого байта (LSB = 0) або старшого байта (LSB = 1).

### Непряма адресація пам'яті програм, команди IJMP і ICALL

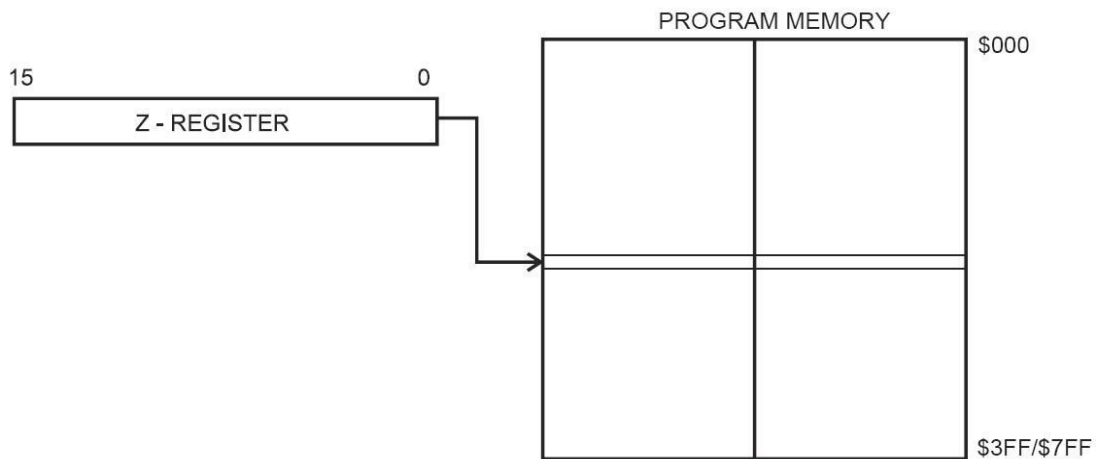


Рис. 11. Непряма адресація пам'яті програм

Виконання програми продовжується з адреси, яка міститься в регістрі Z (тобто лічильник команд загрузається вмістом регістру Z).

### Непряма адресація пам'яті програм, команди RJMP і RCALL

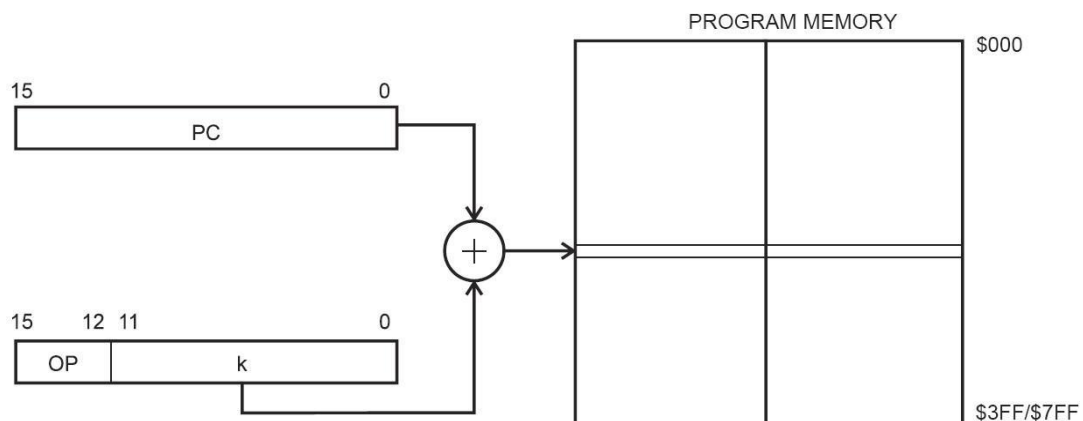


Рис. 12. Відносна адресація пам'яті програм

Виконання програми продовжується з адреси  $PC + k + 1$ . Значення відносної адреси може бути від -2048 до 2047.

Система команд МК AVR налічує до 133 різних команд. Розрізняють наступні групи команд AVR: команди логічних операцій, арифметичних операцій, команди операцій з розрядами, команди порівняння, команди зсуву, команди пересилання даних, команди керування МК, команди умовного та безумовного передавання керування. За кількістю реалізованих команд МК AVR більше схожі на CISC, ніж на RISC-мікроконтролери.

Систему команд МК AVR наведено у табл. 1, де використано такі позначення:

SREG - реєстр статусу;

RЗП - реєстр загального призначення;

PВВ - реєстр введення-виведення;

Z, C, N, V, H, T, I, S - прапорці реєстру статусу;

Rd - реєстр призначення у реєстровому файлі;

Rr - реєстр-джерело у реєстровому файлі;

K - байт даних (константа);

k - величина зсуву відносно стану лічильника команд при відносних переходах;

b - біт в реєстровому файлі або у реєстрі введення-виведення;

s - біт в реєстрі статусу;

X, Y, Z - реєстри непрямої адресації ( $X=R27:R26$ ,  $Y=R29:R28$ ,  $Z=R31:R30$ ).

В табл. 1 наведена кількість циклів, на протязі яких виконується кожна команда, а також показаний вплив команд на прапорці реєстра статусу. Якщо команда впливає на стан конкретного прапорця, його ім'я наведено у стовпці «Прапорці» табл. 1.

## Система команд мікроконтролерів AVR

Таблиця 1

Назва команди	Мнемокод	Операція	Цикли	Прапорці
<b>Команди логічних операцій</b>				
Логічне І двох РЗП	AND Rd, Rr	$Rd \leftarrow Rd \wedge Rr$	1	Z, N, V
Логічне І РЗП і константи	ANDI Rd, K	$Rd \leftarrow Rd \wedge K$	1	Z, N, V
Виключальне АБО двох РЗП	EOR Rd, Rr	$Rd \leftarrow Rd \oplus Rr$	1	Z, N, V
Логічне АБО двох РЗП	OR Rd, Rr	$Rd \leftarrow Rd \vee Rr$	1	Z, N, V
Логічне АБО РЗП і константи	ORI Rd, K	$Rd \leftarrow Rd \vee K$	1	Z, N, V
Перетворення у зворотний код вмісту РЗП	COM Rd	$Rd \leftarrow 0FFH - Rd$	1	Z, C, N, V
Перетворення у додатковий код вмісту РЗП	NEG Rd	$Rd \leftarrow 00H - Rd$	1	Z, C, N, V, H
Скидання розрядів РЗП	CLR Rd	$Rd \leftarrow 00H$	1	Z, N, V
Встановлення розрядів РЗП	SER Rd	$Rd \leftarrow 0FFH$	1	-
Перевірка РЗП на від'ємне (нульове) значення	TST Rd	$Rd \leftarrow Rd \wedge Rd$	1	Z, N, V
<b>Команди арифметичних операцій</b>				
Додавання двох РЗП	ADD Rd, Rr	$Rd \leftarrow Rd + Rr$	1	Z, C, N, V, H
Додавання двох РЗП з перенесенням	ADC Rd, Rr	$Rd \leftarrow Rd + Rr + C$	1	Z, C, N, V, H
Додавання регістрової пари і константи	ADIW Rd, K	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	2	Z, C, N, V, S
Віднімання двох РЗП	SUB Rd, Rr	$Rd \leftarrow Rd - Rr$	1	Z, C, N, V, H
Віднімання константи від РЗП	SUBI Rd, K	$Rd \leftarrow Rd - K$	1	Z, C, N, V, H
Віднімання двох РЗП із позикою	SBC Rd, Rr	$Rd \leftarrow Rd - Rr - C$	1	Z, C, N, V, H
Віднімання з РЗП константи із позикою	SBCI Rd, K	$Rd \leftarrow Rd - K - C$	1	Z, C, N, V, H
Віднімання константи від регістрової пари	SBIW Rd, K	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	2	Z, C, N, V, S
Декремент РЗП	DEC Rd	$Rd \leftarrow Rd - 1$	1	Z, N, V
Інкремент РЗП	INC Rd	$Rd \leftarrow Rd + 1$	1	Z, N, V
<b>Команди операцій з розрядами</b>				
Скидання розряду(ів) РЗП	CBR Rd, K	$Rd \leftarrow Rd \wedge (0FFH - K)$	1	Z, N, V
Встановлення розряду(ів) РЗП	SBR Rd, K	$Rd \leftarrow Rd \wedge K$	1	Z, N, V
Скидання розряду РВВ	CBI A, b	$A.b \leftarrow 0$	2	-
Встановлення розряду РВВ	SBI A, b	$A.b \leftarrow 1$	2	-

Продовж. табл.1

Скидання прапорця	BCLR s	SREG.s ← 0	1	SREG.s
Встановлення прапорця	BSET s	SREG.s ← 1	1	SREG.s
Завантаження розряду РЗП із прапорця T	BLD Rd, b	Rd.b ← T	1	-
Запис розряду РЗП у прапорець T (SREG)	BST Rr, b	T ← Rd.b	1	T
Скидання прапорця перенесення	CLC	C ← 0	1	C
Встановлення прапорця перенесення	SEC	C ← 1	1	C
Скидання прапорця від'ємного числа	CLN	N ← 0	1	N
Встановлення прапорця від'ємного числа	SEN	N ← 1	1	N
Скидання прапорця нуля	CLZ	Z ← 0	1	Z
Встановлення прапорця нуля	SEZ	Z ← 1	1	Z
Загальна заборона переривань	CLI	I ← 0	1	I
Загальний дозвіл переривань	SEI	I ← 1	1	I
Скидання прапорця знака	CLS	S ← 0	1	S
Установка прапорця знака	SES	S ← 1	1	S
Скидання прапорця переповнення додаткового коду	CLV	V ← 0	1	V
Встановлення прапорця переповнення додаткового коду	SEV	V ← 1	1	V
Скидання прапорця користувача T	CLT	T ← 0	1	T
Встановлення прапорця користувача T	SET	T ← 1	1	T
Скидання прапорця половинного перенесення	CLH	H ← 0	1	H
Встановлення прапорця половинного перенесення	SEH	H ← 1	1	H
<b>Команди порівняння</b>				
Порівняння двох РЗП	CP Rd, Rr	?(Rd - Rr)	1	Z, N, V, C, H
Порівняння РЗП з урахуванням переносу	CPC Rd, Rr	?(Rd - Rr - C)	1	Z, N, V, C, H
Порівняння РЗП з константою	CPI Rd, K	? (Rd - K)	1	Z, N, V, C, H

<b>Команди зсуву</b>				
Арифметичний зсув праворуч	ASR Rd	Rd7→ d6→Rd5→ Rd4→Rd3→Rd2 →Rd1→Rd0	1	Z, C, N, V
Логічний зсув ліворуч	LSL Rd	C ← Rd7← Rd6← Rd5← Rd4← Rd3 ← Rd2← Rd1← Rd0← 0	1	Z, C, N, V
Логічний зсув праворуч	LSR Rd	0→Rd7→ Rd6→ Rd5→Rd4→ Rd3 →Rd2→Rd1→ Rd0→C	1	Z, C, N, V
Зсув ліворуч через перенесення	ROL Rd	C ← Rd7← Rd6← Rd5← Rd4← Rd3 ← Rd2← Rd1← Rd0← C	1	Z, C, N, V
Зсув праворуч через перенесення	ROR Rd	C→Rd7→ Rd6→ Rd5→Rd4→ Rd3 →Rd2→Rd1→ Rd0→C	1	Z, C, N, V
Обмін місцями тетрад	SWAP Rd	Rd(3-0)↔ Rd(7-4)	1	-
<b>Команди пересилання даних</b>				
Пересилання між РЗП	MOV Rd,Rr	Rd← Rr	1	-
Пересилання між парами регістрів	MOVW Rd, Rr	Rd+1:Rd← Rr+ 1:Rr	1	-
Завантаження константи в РЗП	LDI Rd, K	Rd← K	1	-
Непряме читання	LD Rd, X	Rd ← [X]	2	-
Непряме читання з постінкрементом	LD Rd, X+	Rd ← [X], X← X+1	2	-
Непряме читання із переддекрементом	LD Rd, -X	X← X-1, Rd← [X]	2	-
Непряме читання	LD Rd, Y	Rd←-[Y]	2	-
Непряме читання з постінкрементом	LD Rd, Y+	Rd← [Y], Y← Y+1	2	-
Непряме читання із переддекрементом	LD Rd, -Y	Y← Y-1, Rd← [Y]	2	-
Непряме відносне читання	LD Rd, Y+q	Rd ← [Y+q]	2	-
Непряме читання	LD Rd, Z	Rd ← [Z]	2	-

Непряме читання з постінкрементом	LD Rd, Z+	Rd← [Z], Z←Z+1	2	-
Непряме читання із переддекрементом	LD Rd, -Z	Z←Z-1, Rd←[Z]	2	-
Непряме відносне читання	LD Rd, Z+q	Rd← [Z+q]	2	-
Безпосереднє читання з ОЗП	LDS Rd, k	Rd ← [K]	2	-
Непрямий запис	ST X, Rr	[X] ← Rr	2	-
Непрямий запис із постикрементом	STX+, Rr	[X] ← Rr, X←X+1	2	-
Непрямий запис із переддекрементом	ST -X, Rr	X←X-1, [X]←Rr	2	-
Непрямий запис	ST Y, Rr	[Y] ← Rr	2	-
Непрямий запис із постикрементом	ST Y+, Rr	[Y]←Rr, Y←Y+1	2	-
Непрямий запис із переддекрементом	ST -Y, Rr	Y←Y-1, [Y]←Rr	2	-
Непрямий відносний запис	STY+q, Rr	[Y+q] ← Rr	2	-
Непрямий запис	STZ, Rr	[Z] ← Rr	2	-
Непрямий запис із постикрементом	STZ+, Rr	[Z] ← Rr, Z←Z+1	2	-
Непрямий запис із переддекрементом	ST -Z, Rr	Z←Z-1, [Z]←Rr	2	-
Непрямий відносний запис	STZ+q, Rr	[Z+q] ← Rr	2	-
Безпосередній запис в ОЗУ	STS k, Rr	[k] ← Rr	2	-
Завантаження даних з пам'яті програм	LPM	R0 ← {Z}	3	-
Завантаження даних з пам'яті програм	LPM Rd, Z	Rd ← {Z}	3	-
Завантаження даних з пам'яті програм і постдекремент Z	LPM Rd, Z+	Rd←{Z}, Z←Z+1	3	-
Запис у програмну пам'ять	SPM	{Z}←R1:R0	-	-
Пересилання із РВВ у РЗП	IN Rd, P	Rd←-P	1	-
Пересилання з РЗП у РВВ	OUT P, Rr	P←Rr	1	-
Завантаження байта у стек	PUSH Rr	STACK ← Rr	2	-
Витягування байта зі стеку	POP Rr	Rd ← STACK	2	-
<b>Команди керування МК</b>				
Немає операції	NOP	-	1	-

Перехід в «сплячий» режим	SLEEP	-	3	-
Скидання сторожового таймера	WDR	-	1	-
Припинення програми	BREAK	Застосовується для відладки	-	-
<b>Команди передавання керування (безумовне передавання керування)</b>				
Відносний безумовний перехід	RJMP	$PC \leftarrow PC + k + 1$	2	-
Непрямий безумовний перехід	IJMP	$PC \leftarrow Z$	2	-
Відносний виклик підпрограми	RCALL	$PC \leftarrow PC + k + 1$	3	-
Непрямий виклик підпрограми	ICALL	$PC \leftarrow Z$	3	-
Повернення з підпрограми	RET	$PC \leftarrow STACK$	4	-
Повернення з підпрограми обробки переривань	RETI	$PC \leftarrow STACK$	4	I
<b>Команди передавання керування (пропускання наступної команди за умовою)</b>				
Порівняння й пропускання наступної команди при рівності	CPSE Rd, Rr	Якщо $Rd = Rr$	1/2/3	-
Пропускання наступної команди, якщо розряд РЗП скинутий	SBRC Rr, b	Якщо $Rr.b = 0$	1/2/3	-
Пропускання наступної команди, якщо розряд РЗП встановлений	SBRS Rr, b	Якщо $Rr.b = 1$	1/2/3	-
Пропускання наступної команди, якщо розряд РВВ скинутий	SBIC A, b	Якщо $A.b = 0$	1/2/3	-
Пропускання наступної команди, якщо розряд РВВ встановлений	SBIS A, b	Якщо $A.b = 1$	1/2/3	-
<b>Команди передавання керування за умовою</b>				
Перехід, якщо прапорець s регістра SREG скинутий	BRBC s, k	Якщо $SREG.s = 0$	1/2	-
Перехід, якщо прапорець s регістра SREG встановлений	BRBS s, k	Якщо $SREG.s = 1$	1/2	-
Перехід по перенесенню	BRCS k	Якщо $C = 1$	1/2	-

Перехід, якщо немає перенесення	BRCC k	Якщо $C = 0$	1/2	-
Перехід за умовою «дорівнює»	BREQ k	Якщо $Z = 1$	1/2	-
Перехід за умовою «нерівно»	BRNE k	Якщо $Z = 0$	1/2	-
Перехід за умовою «більше або дорівнює»	BRSN k	Якщо $C = 0$	1/2	-
Перехід за умовою «менше»	BRLO k	Якщо $C = 1$	1/2	-
Перехід за умовою «від'ємне значення»	BRMI k	Якщо $N = 1$	1/2	-
Перехід за умовою «додатне значення»	BRPL k	Якщо $N = 0$	1/2	-
Перехід за умовою «більше або дорівнює» (зі знаком)	BRGE k	Якщо $(N \vee V) = 0$	1/2	-
Перехід за умовою «менше» (зі знаком)	BRLT k	Якщо $(N \vee V) = 1$	1/2	-
Перехід по половинному перенесенню	BRHS k	Якщо $H = 1$	1/2	-
Перехід, якщо немає половинного перенесення	BRHC k	Якщо $H = 0$	1/2	-
Перехід, якщо прапорець T встановлений	BRTS k	Якщо $T = 1$	1/2	-
Перехід, якщо прапорець T скинутий	BRTC k	Якщо $T = 0$	1/2	-
Перехід по переповненню додаткового коду	BRVS k	Якщо $V = 1$	1/2	-
Перехід, якщо немає переповнення додаткового коду	BRVC k	Якщо $V = 0$	1/2	-
Перехід, якщо переривання заборонені	BRID k	Якщо $I = 0$	1/2	-
Перехід, якщо переривання дозволені	BRIE k	Якщо $I = 1$	1/2	-