



حالت‌های آدرس‌دهی و مجموعه دستورالعمل‌ها

در میکروکنترلرهای AVR

فهرست مطالب

- حالت‌های آدرس‌دهی (Addressing modes)
- مجموعه دستورالعمل‌ها

اسمبلی میکروکنترلرهای ۸ بیتی AVR

- میکروکنترلر RISC تقویت شده AVR، حالت‌های آدرس‌دهی قدرتمند و موثری را برای دسترسی به حافظه برنامه (فلش) و حافظه داده (SRAM)، فایل ثبات، حافظه I/O و حافظه توسعه‌یافته I/O فراهم می‌نماید.

پرچم‌ها (خلاصه)

علائم و اختصارات ثبات وضعیت (SREG) حاوی بیت های وضعیت (پرچم‌ها)

نام پرچم	علامت پرچم
پرچم نقلی	C
پرچم صفر	Z
پرچم منفی	N
پرچم سرریز	V
پرچم تست مقادیر علامت‌دار	$S: N \oplus V$
پرچم نیمه‌توازن	H
بیت انتقال (مورد استفاده در دستورالعمل‌های BST و BLD)	T
پرچم وقفه سراسری	I

علائم و اختصارات ثبات‌ها و عملوندها

علائم و اختصارات ثبات‌ها و عملوندها و منظور از استفاده از آنها در دستورالعمل‌های میکروکنترلرهای ۸ بیتی AVR

نام ثبات	عملیات
Rd	ثبات مقصد (و منبع) در فایل ثبات
Rr	ثبات منبع در فایل ثبات
R	نتیجه بعد از اجرای دستورالعمل
K	داده ثابت
k	آدرس ثابت
b	بیت در فایل ثبات یا ثبات I/O (سه بیت)
s	بیت در ثبات وضعیت (سه بیت)
X,Y,Z	ثبات آدرس‌دهی غیر مستقیم (X=R27:R26, Y=R29:R28 and Z=R31:R30)
A	آدرس محل I/O

علائم و اختصارات ثبات‌ها و عملوندها

علائم و اختصارات ثبات‌های ورودی/خروجی و منظور از استفاده از آنها در دستورالعمل‌های میکروکنترلرهای ۸ بیتی AVR:

- ثبات‌های **RAMPX**، **RAMPY** و **RAMPZ** ثبات‌های متصل به ثبات‌های **X**، **Y** و **Z** هستند که آدرس‌دهی غیرمستقیم کل فضای داده میکروکنترلرهای با بیش از 64K بایت فضای داده و واکنشی داده ثابت در میکروکنترلرهای خانواده ATmega با بیش از 64K بایت حافظه برنامه را امکان‌پذیر می‌سازند.

- ثبات **RAMPD** ثبات متصل به ثبات **Z** است که آدرس‌دهی مستقیم کل فضای داده میکروکنترلرهای با بیش از 64K بایت فضای داده را امکان‌پذیر می‌سازد.

علائم و اختصارات ثبات‌ها و عملوندها

- ثبات **EIND** ثبات متصل به ثبات **Z** است که پرش و فراخوانی غیرمستقیم به کل فضای داده میکروکنترلرهای با بیش از **64K** کلمه (**128K** بایت) و فضای برنامه را امکان‌پذیر می‌سازد.
- پشته (**STACK**) برای ذخیره‌سازی آدرس بازگشت و ثبات‌های پوش شده، و ثبات **SP** ثبات اشاره‌گر پشته می‌باشد.

علائم و اختصارات وضعیت پرچم‌ها

علائم مورد استفاده جهت بیان چگونگی تغییر وضعیت پرچم‌ها بازاء اجراء هر دستورالعمل:

⇔: پرچم توسط دستورالعمل تغییر کرده است.

0: پرچم توسط دستورالعمل صفر شده است.

1: پرچم توسط دستورالعمل یک شده است.

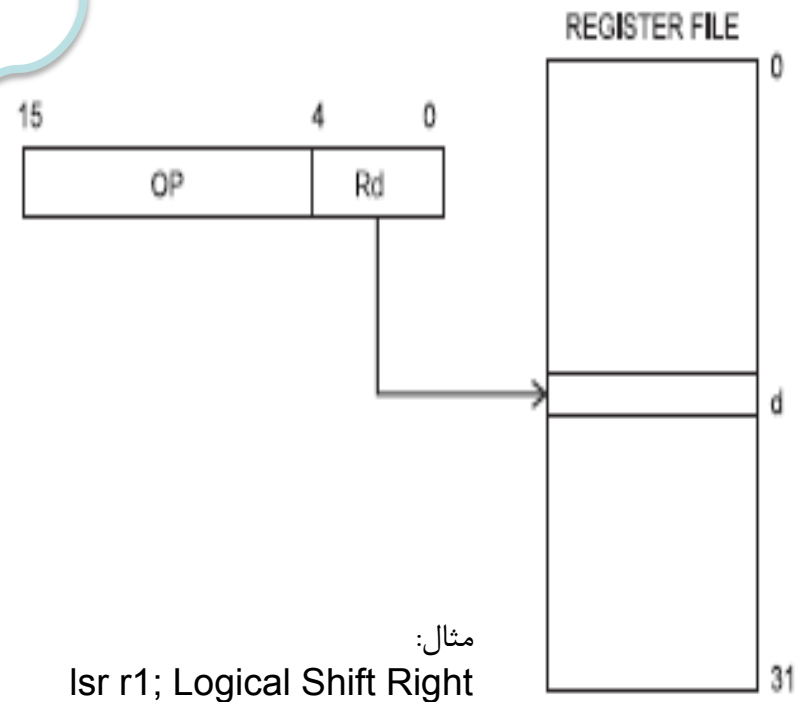
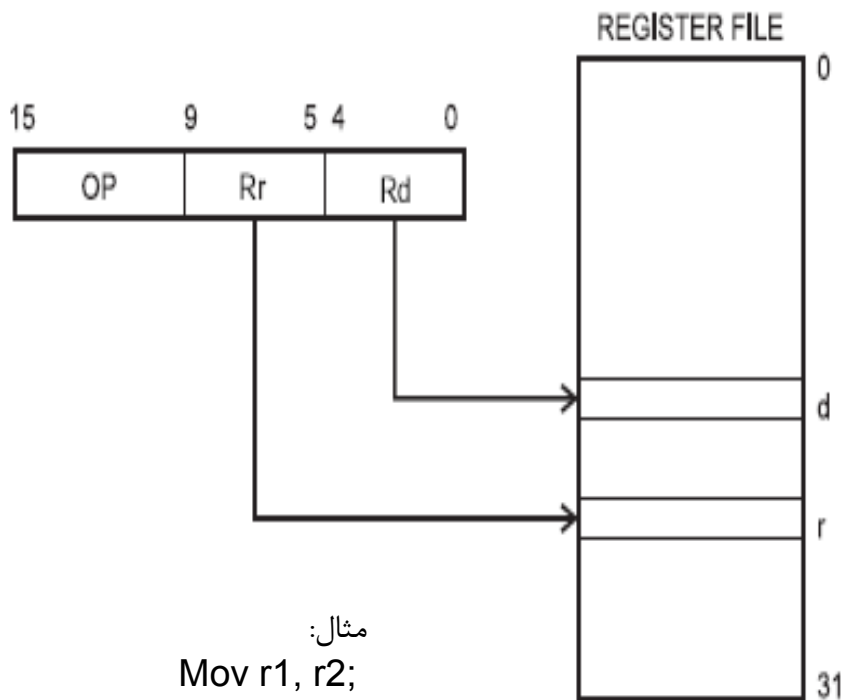
-: پرچم توسط دستورالعمل تغییر نکرده است.

توجه: در مباحث و جداولی که در ادامه می‌آیند، OP به معنای بخش کد عملیاتی کلمه دستورالعمل است.

کلمات **RAMEND** و **FLASHEND** بالاترین محل در فضای داده و برنامه را نشان می‌دهند.

حالت‌های آدرس‌دهی

(ب) حالت آدرس‌دهی مستقیم توسط ثبات (با دو ثبات Rd و Rr)



(الف) حالت آدرس‌دهی مستقیم توسط ثبات (با تنها یک ثبات Rd)

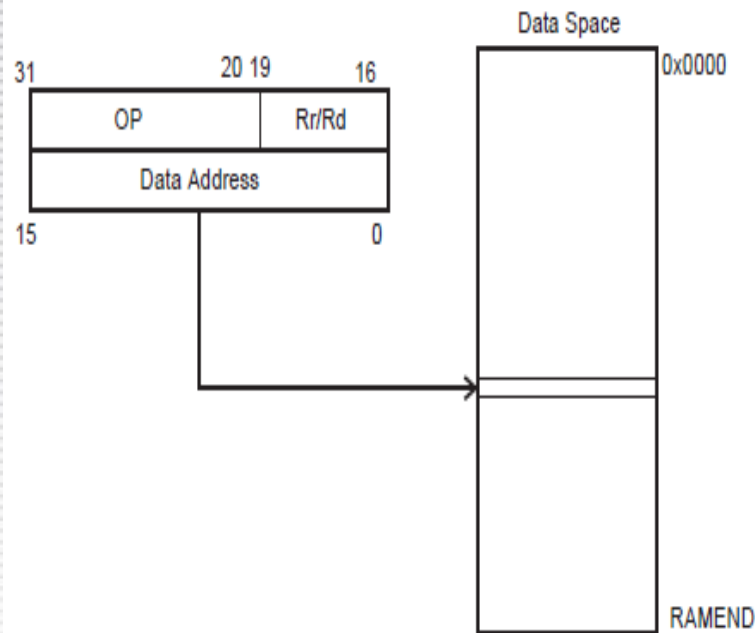
حالت‌های آدرس‌دهی

الف) حالت آدرس‌دهی مستقیم توسط ثبات (با تنها یک ثبات R_d). در این مود، عملوند دستورالعمل در ثبات R_d قرار می‌گیرد.

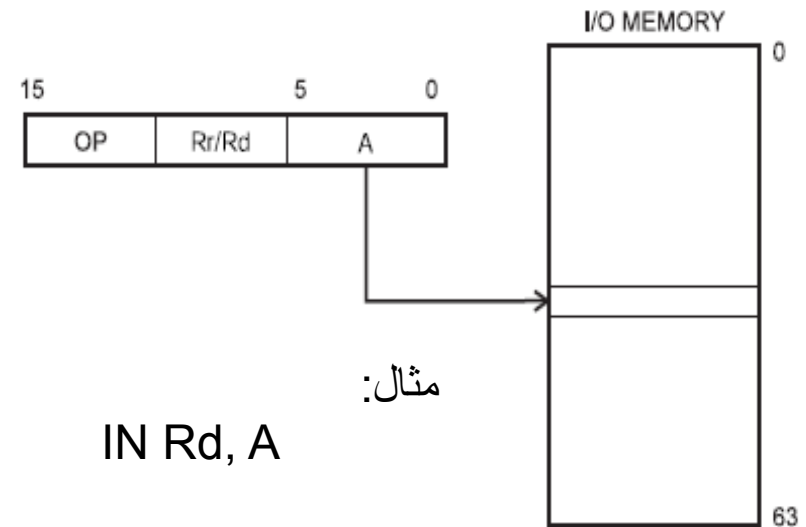
ب) حالت آدرس‌دهی مستقیم توسط ثبات (با دو ثبات R_d و R_r). در این مود، عملوندها در ثبات‌های R_d و R_r قرار می‌گیرند. نتیجه در ثبات R_d گذاشته می‌شود

حالت‌های آدرس‌دهی

(د) حالت آدرس‌دهی مستقیم داده



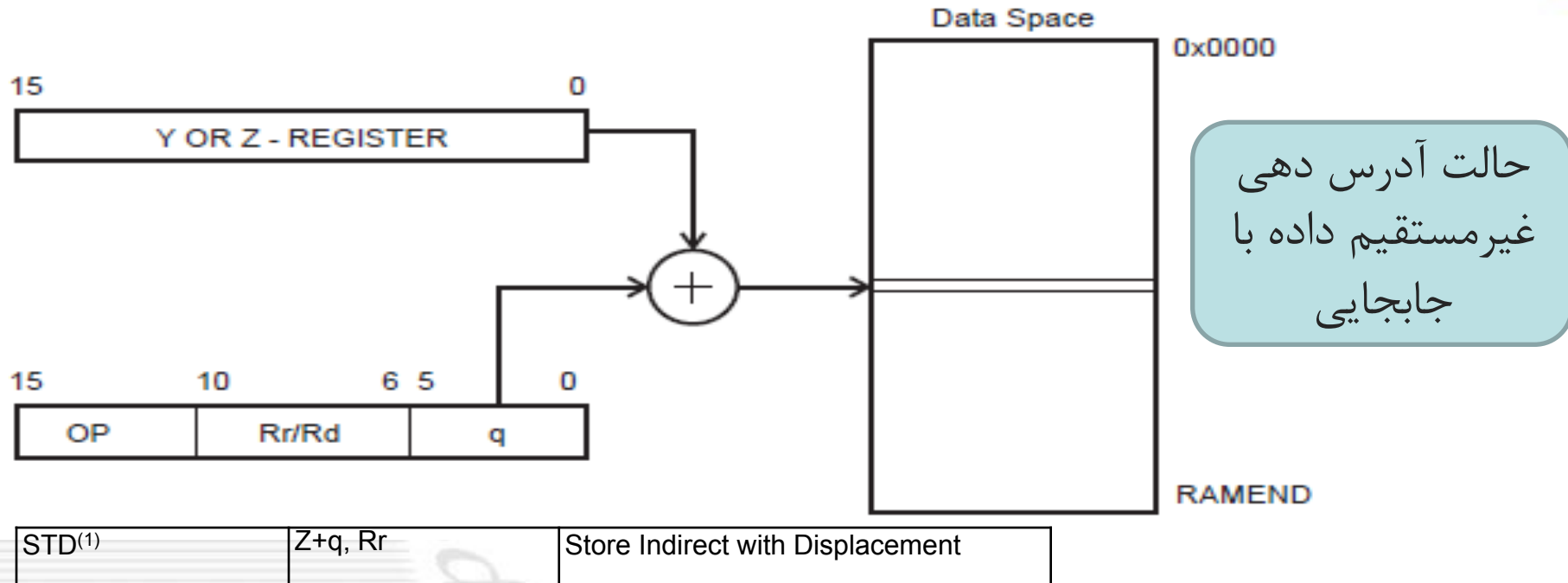
(ج) حالت آدرس‌دهی مستقیم I/O



(ج) حالت آدرس‌دهی مستقیم ورود/خروجی. در این
مود، آدرس عملوند در ۶ بیت از کلمه دستورالعمل
قرار می‌گیرد. A آدرس ثبات مبداء یا مقصد است.

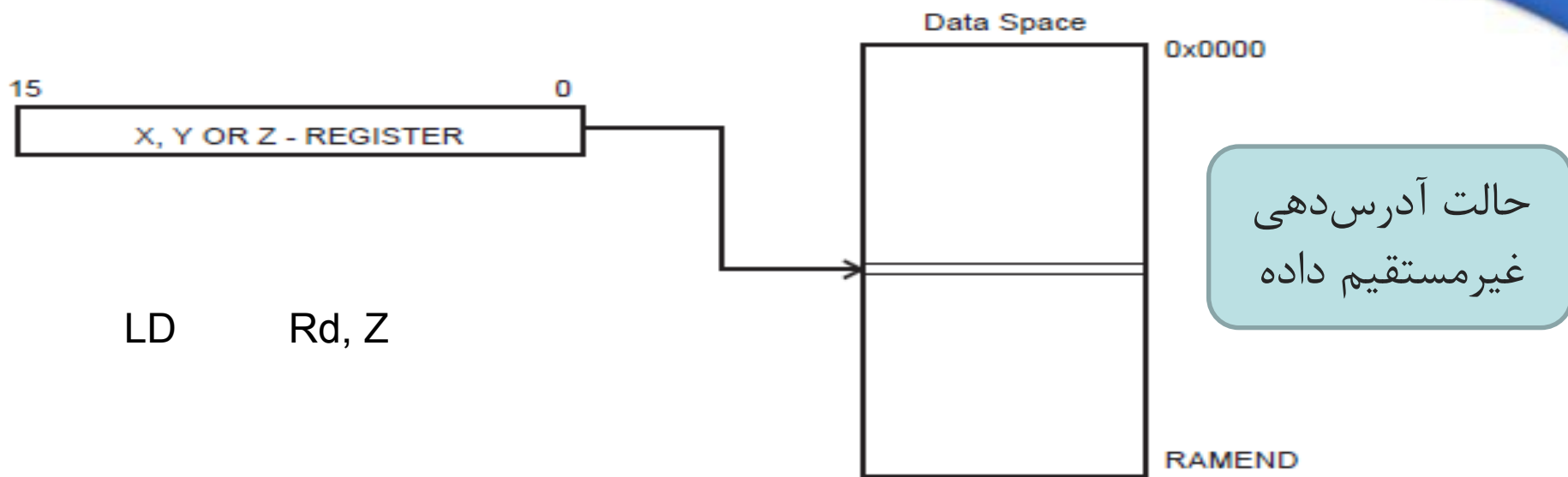
(د) حالت آدرس‌دهی مستقیم داده. در این مود،
یک آدرس داده ۱۶ بیتی در ۱۶ بیت کم ارزش
یک دستورالعمل دو کلمه‌ای قرار دارد. Rd/Rr
ثبات مبداء یا مقصد را تعریف می‌کنند.

حالت‌های آدرس‌دهی



- در این مود، آدرس عملوند نتیجه حاصل جمع محتوای ثبات‌های **Y** یا **Z** با ۶ بیت آدرس موجود در کلمه دستورالعمل می‌باشد. Rr و Rd ثبات‌های مبدا و مقصد را مشخص می‌کنند

حالت‌های آدرس دهی

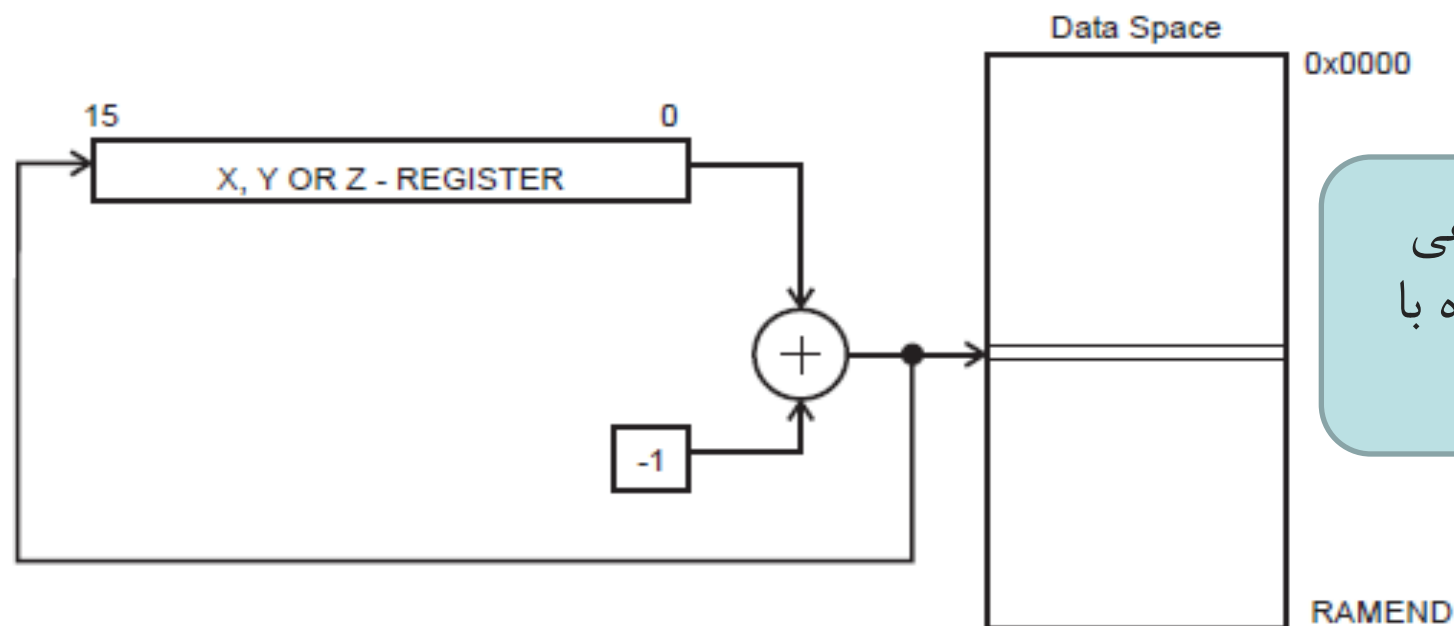


- در این مود، آدرس عملوند، محتوای یکی از ثبات های **X, Y یا Z** است.

- آدرس دهی غیرمستقیم ثبات زیرمجموعه‌ای از حالت آدرس دهی غیرمستقیم داده است، چرا که فضای داده از ۰ تا ۳۱ همان فایل ثبات را تشکیل می‌دهد.

- در میکروکنترلرهای AVR بدون حافظه SRAM، حالت آدرس دهی غیرمستقیم داده را حالت آدرس دهی غیرمستقیم ثبات می‌نامند.

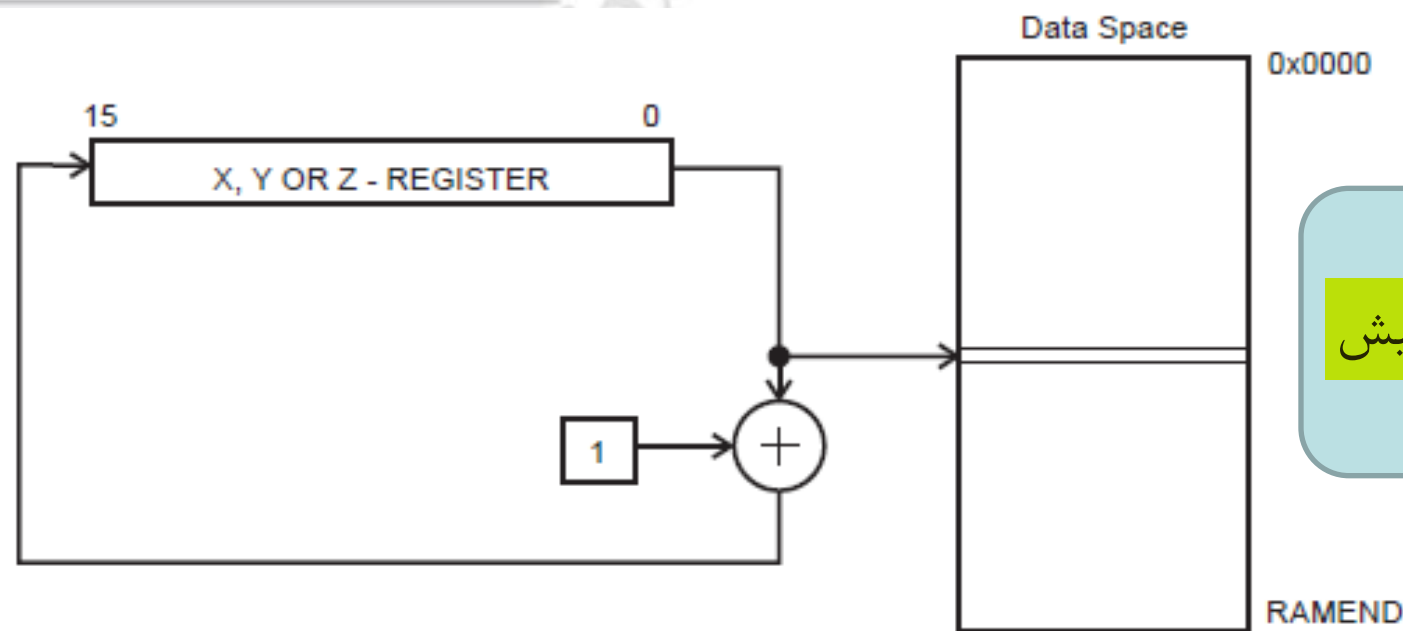
حالت‌های آدرس دهی



- در این حالت ثبات X ، Y یا Z قبل از عملیات، یک واحد کاهش می یابد. آدرس عملوند کاهش یافته محتوای یکی از ثبات‌های X ، Y یا Z است.

ST ⁽²⁾	-Z, Rr	Store Indirect and Pre-Decrement
-------------------	--------	----------------------------------

حالت‌های آدرس دهی

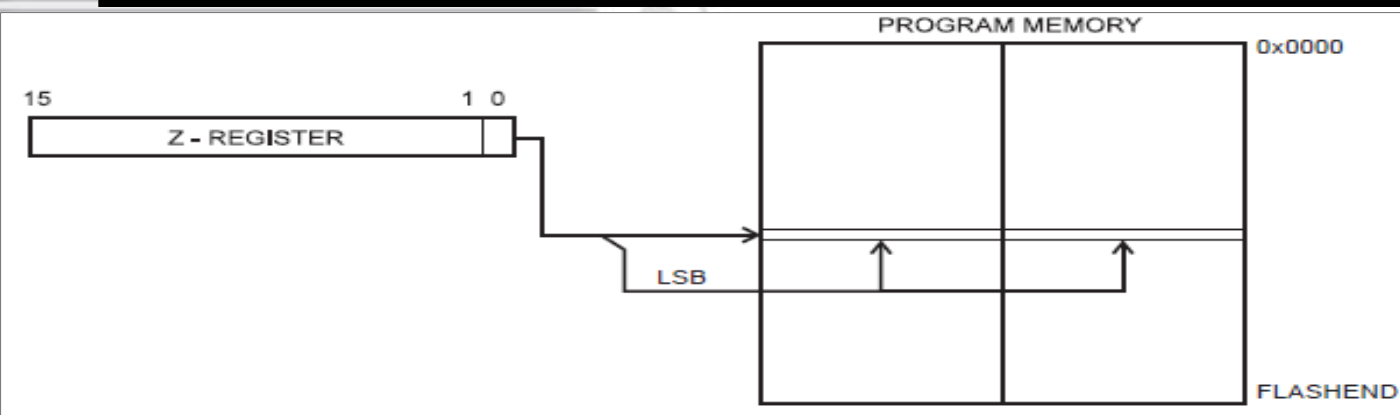


حالت آدرس دهی
غیرمستقیم داده با پیش
افزایش

- در این حالت ثبات X ، Y یا Z قبل از عملیات، یک واحد افزایش می‌یابد. آدرس عملوند افزایش یافته محتوای یکی از ثبات‌های X ، Y یا Z است.

ST(2)	Z+, Rr	Store Indirect and Post-Increment
-------	--------	-----------------------------------

حالت‌های آدرس دهی



حالت آدرس دهی
حافظه برنامه با
آدرس ثابت

LPM Rd, Z; Load Program memory; $Rd \leftarrow (Z)$

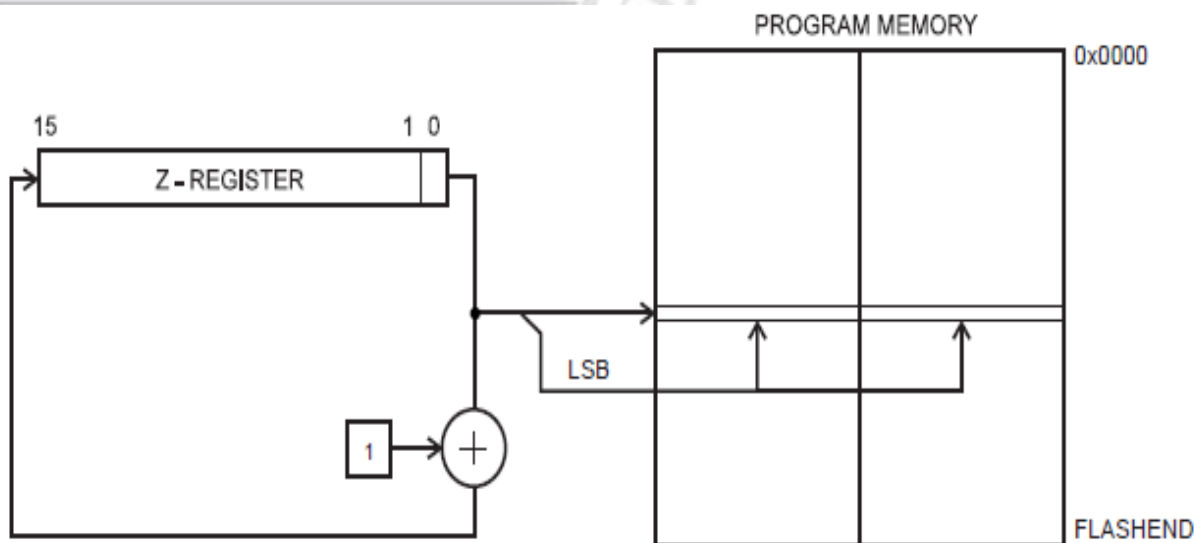
- در این مود، آدرس بایتی معینی در حافظه برنامه توسط محتوای ثابت **Z** تعریف می شود.
- بیت ۱۵ پر ارزش آدرس کلمه را انتخاب می کنند.

- برای **LPM**، اگر بیت کم ارزش ۰ شود، بایت پائین انتخاب و اگر ۱ باشد، بایت پر ارزش انتخاب می شود. اگر **ELPM** بکار رود، ثابت **RAMPZ** برای توسعه ثابت **Z** بکار می رود.

ELPM Rd, Z; Extended Load Program memory; $Rd \leftarrow (RAMPZ: Z)$

SPM z+; Store Program Memory and Post- Increment by 2; $(RAMPZ:Z) \leftarrow R1:R0$ $Z \leftarrow Z+2$

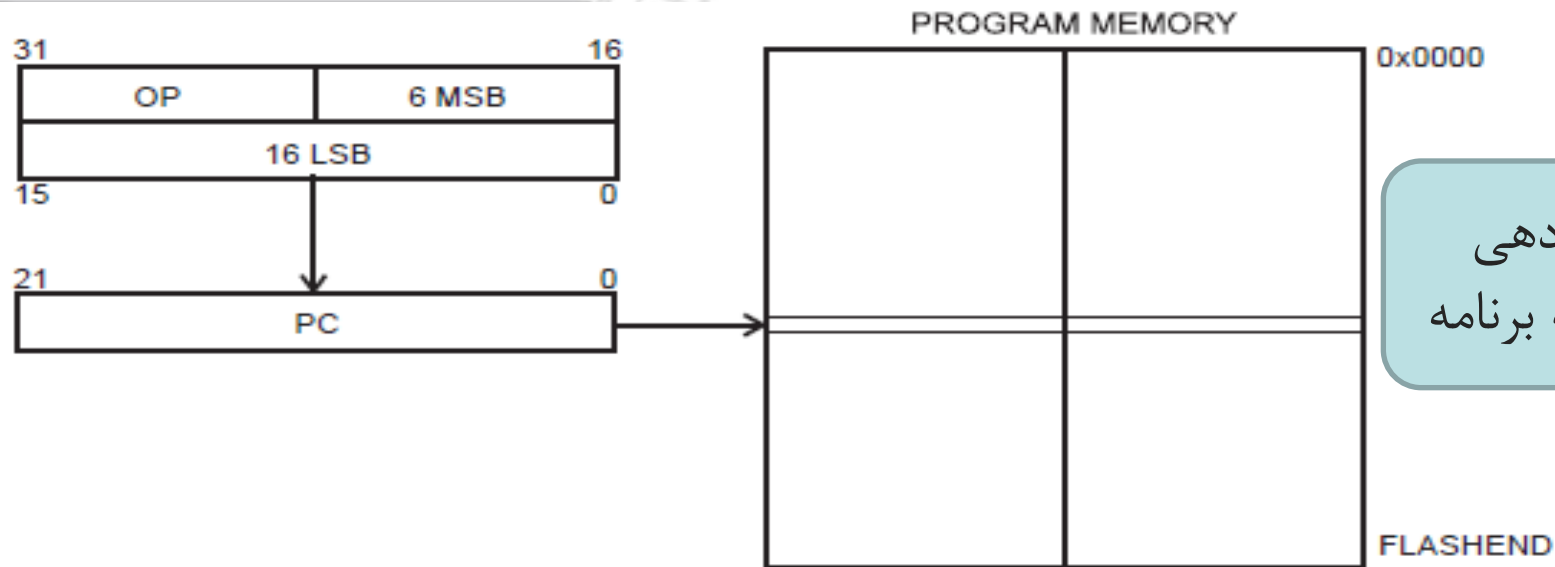
حالت‌های آدرس دهی



حالت آدرس دهی حافظه
برنامه با پس افزایش با استفاده
از LPM Z+ و ELPM Z+

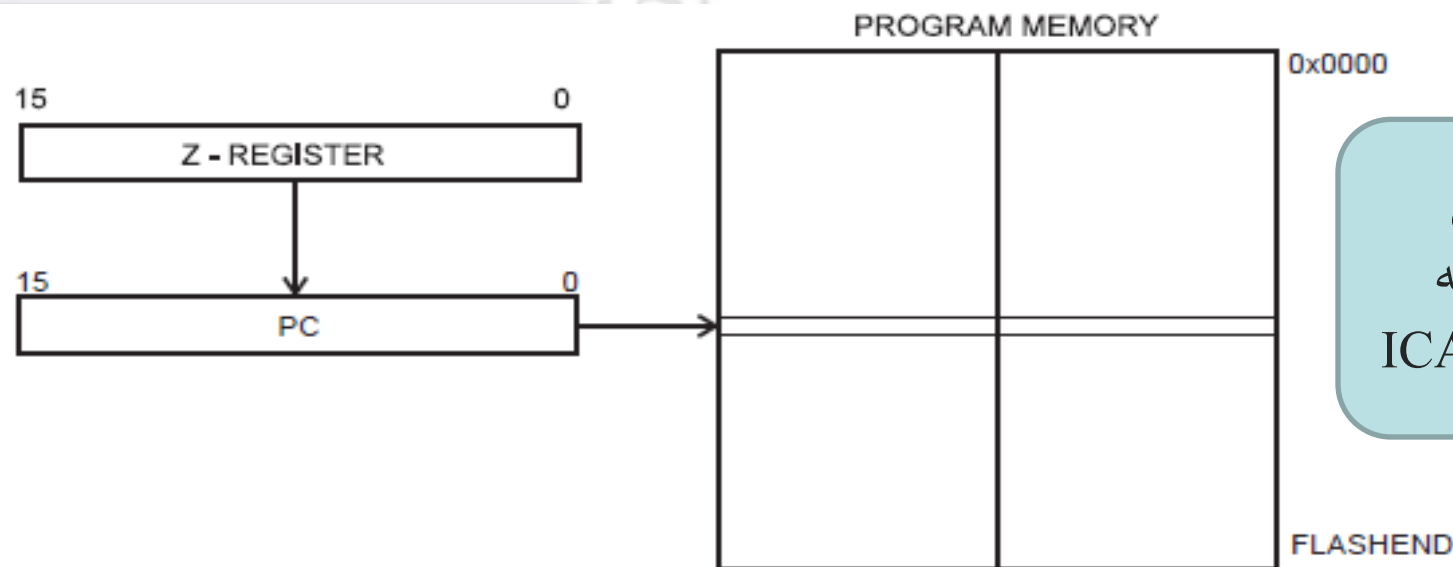
- در این حالت آدرس بایتی ثابت توسط محتوای ثابت **Z** تعریف می شود.
- **۱۵ بیت** پرارزش آدرس کلمه را انتخاب می کنند.
- اگر بیت کم ارزش صفر شود، **بایت پائین** انتخاب و اگر **یک** باشد، **بایت پرارزش** انتخاب می شود.
- اگر **ELPM Z+** بکار رود، ثابت **RAMPZ** برای توسعه ثابت **Z** بکار می رود.

حالت‌های آدرس‌دهی



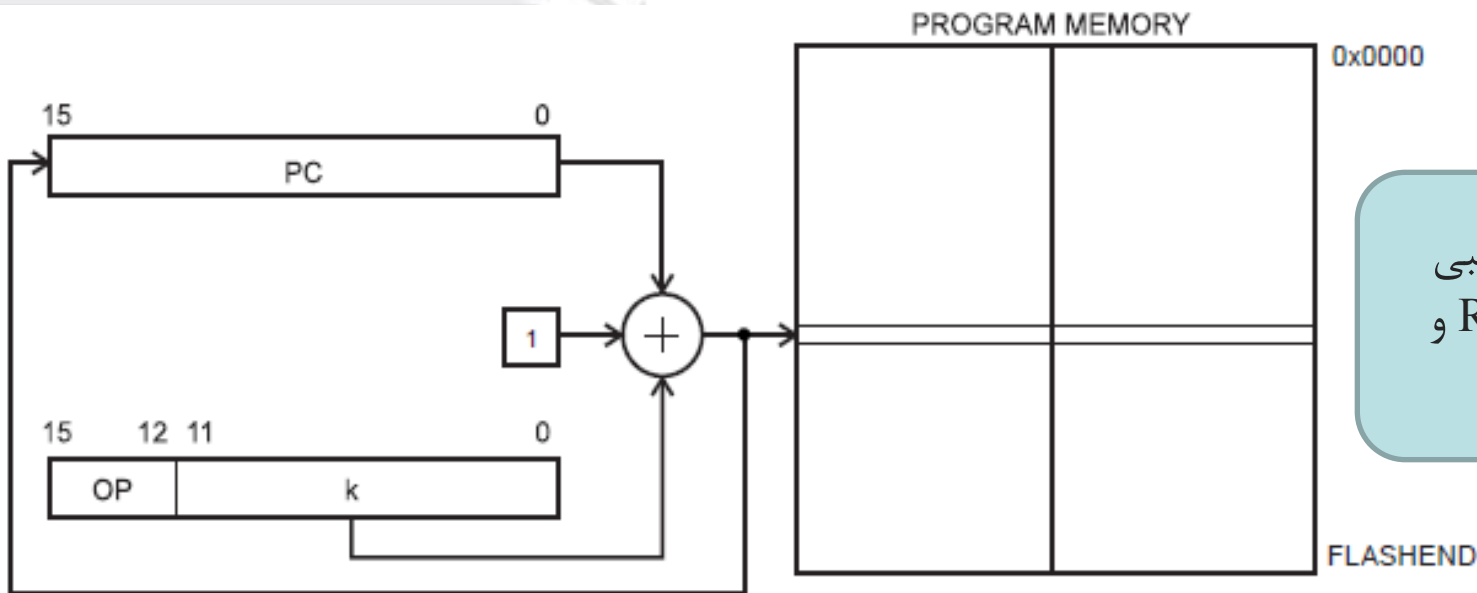
حالت آدرس‌دهی
مستقیم حافظه برنامه

حالت‌های آدرس دهی



- در این حالت اجرای برنامه از آدرس موجود در **ثبات Z** ادامه می‌یابد (یعنی اینکه مقدار موجود در ثبات Z در PC قرار می‌گیرد).

حالت‌های آدرس‌دهی



حالت آدرس‌دهی نسبی
حافظه برنامه، RJMP و
RCALL

- در این حالت اجرای برنامه از آدرس $PC+K+1$ ادامه می‌یابد. آدرس نسبی k از 2048- تا 2048 تغییر می‌نماید.

مجموعه دستورالعمل‌های میکروکنترلرهای ۸ بیتی AVR

- دستورالعمل‌های میکروکنترلرهای ۸ بیتی AVR شامل گروه‌های زیر هستند:
 - الف- دستورالعمل‌های حسابی و منطقی
 - ب- دستورالعمل‌های انشعاب
 - ج- دستورالعمل‌های انتقال داده
 - د- دستورالعمل‌های بیتی و تست بیت
 - ه- دستورالعمل‌های کنترل میکروکنترلر

مجموعه دستورالعمل‌های میکروکنترلرهای ۸ بیتی AVR

توجه: معانی اعداد نوشته شده در داخل پرانتزها در ستون‌های ۶ و ۷ در اسلایدهای بعد به شرح زیر است:

- (۱) این دستورالعمل‌ها در تمام میکروکنترلرهای ۸ بیتی AVR وجود ندارد.
- (۲) تمام گونه‌های این دستورالعمل در تمام میکروکنترلرهای ۸ بیتی AVR فراهم نیستند
- (۳) زمان‌های چرخه برای تمام دسترسی‌های به حافظه داده فرض بر دسترسی به حافظه RAM داخلی دارند و برای دسترسی از طریق واسطه RAM خارجی معتبر نیستند.
- (۴) در صورت دسترسی به SRAM داخلی یک چرخه ساعت دیگر اضافه شود.
- (۵) تعداد چرخه‌های ساعت برای میکروکنترلر ATtiny10.

مجموعه دستورالعمل‌های میکروکنترلرهای ۸ بیتی AVR

Mnemonics	Operands	Description	Operation	Flags	#Clocks	#Clocks-XMEGA
ADD	Rd, Rr	Add without Carry	$Rd \leftarrow Rd + Rr$	Z,C,N,V,S,H	1	
ADC	Rd, Rr	Add with Carry	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,S,H	1	
ADIW ⁽¹⁾	Rd, K	Add Immediate to Word	$Rd + 1:Rd \leftarrow Rd + 1:Rd + K$	Z,C,N,V,S	2	
SUB	Rd, Rr	Subtract without Carry	$Rd \leftarrow Rd - Rr$	Z,C,N,V,S,H	1	
SUBI	Rd, K	Subtract Immediate	$Rd \leftarrow Rd - K$	Z,C,N,V,S,H	1	
SBC	Rd, Rr	Subtract with Carry	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,S,H	1	
SBCI	Rd, K	Subtract Immediate with Carry	$Rd \leftarrow Rd - K - C$	Z,C,N,V,S,H	1	
SBIW ⁽¹⁾	Rd, K	Subtract Immediate from Word	$Rd + 1:Rd \leftarrow Rd + 1:Rd - K$	Z,C,N,V,S	2	
AND	Rd, Rr	Logical AND	$Rd \leftarrow Rd \cdot Rr$	Z,N,V,S	1	
ANDI	Rd, K	Logical AND with Immediate	$Rd \leftarrow Rd \cdot K$	Z,N,V,S	1	
OR	Rd, Rr	Logical OR	$Rd \leftarrow Rd \vee Rr$	Z,N,V,S	1	
ORI	Rd, K	Logical OR with Immediate	$Rd \leftarrow Rd \vee K$	Z,N,V,S	1	
EOR	Rd, Rr	Exclusive OR	$Rd \leftarrow Rd \oplus Rr$	Z,N,V,S	1	

مجموعه دستورالعمل‌های میکروکنترلرهای ۸ بیتی AVR

COM	Rd	One's Complement	$Rd \leftarrow \$FF - Rd$	Z,C,N,V,S	1	
NEG	Rd	Two's Complement	$Rd \leftarrow \$00 - Rd$	Z,C,N,V,S,H	1	
SBR	Rd,K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V,S	1	
CBR	Rd,K	Clear Bit(s) in Register	$Rd \leftarrow Rd \cdot (\$FFh - K)$	Z,N,V,S	1	
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V,S	1	
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V,S	1	
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \cdot Rd$	Z,N,V,S	1	
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V,S	1	
SER	Rd	Set Register	$Rd \leftarrow \$FF$	None	1	
MUL ⁽¹⁾	Rd,Rr	Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr \text{ (UU)}$	Z,C	2	
MULS ⁽¹⁾	Rd,Rr	Multiply Signed	$R1:R0 \leftarrow Rd \times Rr \text{ (SS)}$	Z,C	2	
MULSU ⁽¹⁾	Rd,Rr	Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr \text{ (SU)}$	Z,C	2	
FMUL ⁽¹⁾	Rd,Rr	Fractional Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr \ll 1 \text{ (UU)}$	Z,C	2	
FMULS ⁽¹⁾	Rd,Rr	Fractional Multiply Signed	$R1:R0 \leftarrow Rd \times Rr \ll 1 \text{ (SS)}$	Z,C	2	
FMULSU ⁽¹⁾	Rd,Rr	Fractional Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr \ll 1 \text{ (SU)}$	Z,C	2	
DES	K	Data Encryption	if (H = 0) then R15:R0 \leftarrow Encrypt(R15:R0, K) else if (H = 1) then R15:R0 \leftarrow Decrypt(R15:R0, K)			1/2

مجموعه دستورالعمل‌های میکروکنترلرهای ۸ بیتی AVR

DES Instruction Description:

The module is an instruction set extension to the AVR CPU, performing DES iterations. The 64-bit data block (plaintext or cipher text) is placed in the CPU register file, registers R0-R7, where LSB of data is placed in LSB of R0 and MSB of data is placed in MSB of R7. The full 64-bit key (including parity bits) is placed in registers R8-R15, organized in the register file with LSB of key in LSB of R8 and MSB of key in MSB of R15. Executing one DES instruction performs one round in the DES algorithm. Sixteen rounds must be executed in increasing order to form the correct DES cipher text or plaintext. Intermediate results are stored in the register file (R0-R15) after each DES instruction. The instruction's operand (K) determines which round is executed, and the half carry flag (H) determines whether encryption or decryption is performed.

مجموعه دستورالعمل‌های میکروکنترلرهای ۸ بیتی AVR

Mnemonics	Operands	Description	Operation	Flags	#Clocks	#Clocks-XMEGA
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2	
IJMP ⁽¹⁾		Indirect Jump to (Z)	$PC(15:0) \leftarrow Z$ $PC(21:16) \leftarrow 0$	None	2	
EIJMP ⁽¹⁾		Extended Indirect Jump to (Z)	$PC(15:0) \leftarrow Z$ $PC(21:16) \leftarrow EIND$	None	2	
JMP ⁽¹⁾	k	Jump	$PC \leftarrow k$	None	3	
RCALL	k	Relative Call Subroutine	$PC \leftarrow PC + k + 1$	None	3/4 ⁽³⁾ (5)	2/3 ⁽³⁾
ICALL ⁽¹⁾		Indirect Call to (Z)	$PC(15:0) \leftarrow Z$, $PC(21:16) \leftarrow 0$	None	3/4 ⁽³⁾	2/3 ⁽³⁾
EICALL ⁽¹⁾		Extended Indirect Call to (Z)	$PC(15:0) \leftarrow Z$, $PC(21:16) \leftarrow EIND$	None	4 ⁽³⁾	3 ⁽³⁾
CALL ⁽¹⁾	k	call Subroutine	$PC \leftarrow k$	None	4 / 5 ⁽³⁾	3/4 ⁽³⁾
RET		Subroutine Return	$PC \leftarrow STACK$	None	4 / 5 ⁽³⁾	
RETI		Interrupt Return	$PC \leftarrow STACK$	I	4 / 5 ⁽³⁾	
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) $PC \leftarrow PC + 2$ or 3	None	1/2/3	
CP	Rd,Rr	Compare	Rd - Rr	Z,C,N,V,S,H	1	
CPC	Rd,Rr	Compare with Carry	Rd - Rr - C	Z,C,N,V,S,H	1	
CPI	Rd,K	Compare with Immediate	Rd - K	Z,C,N,V,S,H	1	
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b) = 0) $PC \leftarrow PC + 2$ or 3	None	1/2/3	
SBRSC	Rr, b	Skip if Bit in Register Set	if (Rr(b) = 1) $PC \leftarrow PC + 2$ or 3	None	1/2/3	
SBIC	A, b	Skip if Bit in I/O Register Cleared	if (I/O(A,b) = 0) $PC \leftarrow PC + 2$ or 3	None	1/2/3	2/3/4
SBIS	A, b	Skip if Bit in I/O Register Set	If (I/O(A,b) = 1) $PC \leftarrow PC + 2$ or 3	None	1/2/3	2/3/4
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then $PC \leftarrow PC + k + 1$	None	1/2	

مجموعه دستورالعمل‌های میکروکنترلرهای ۸ بیتی AVR

BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC \leftarrow PC + k + 1	None	½	
BREQ	k	Branch if Equal	if (Z = 1) then PC \leftarrow PC + k + 1	None	½	
BRNE	k	Branch if Not Equal	if (Z = 0) then PC \leftarrow PC + k + 1	None	½	
BRCS	k	Branch if Carry Set	if (C = 1) then PC \leftarrow PC + k + 1	None	½	
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC \leftarrow PC + k + 1	None	½	
BRSH	k	Branch if Same or Higher	if (C = 0) then PC \leftarrow PC + k + 1	None	½	
BRLO	k	Branch if Lower	if (C = 1) then PC \leftarrow PC + k + 1	None	½	
BRMI	k	Branch if Minus	if (N = 1) then PC \leftarrow PC + k + 1	None	½	
BRPL	k	Branch if Plus	if (N = 0) then PC \leftarrow PC + k + 1	None	½	
BRGE	k	Branch if Greater or Equal, Signed	if (N \oplus V = 0) then PC \leftarrow PC + k + 1	None	½	
BRLT	k	Branch if Less Than, Signed	if (N \oplus V = 1) then PC \leftarrow PC + k + 1	None	½	
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC \leftarrow PC + k + 1	None	½	
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC \leftarrow PC + k + 1	None	½	
BRTS	k	Branch if T Flag Set	if (T = 1) then PC \leftarrow PC + k + 1	None	½	
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC \leftarrow PC + k + 1	None	½	
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC \leftarrow PC + k + 1	None	½	
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC \leftarrow PC + k + 1	None	½	
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC \leftarrow PC + k + 1	None	½	
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC \leftarrow PC + k + 1	None	½	

مجموعه دستورالعمل‌های میکروکنترلرهای ۸ بیتی AVR

Mnemonics	Operands	Description	Operation	Flags	#Clocks	#Clocks XMEGA
MOV	Rd, Rr	Copy Register	$Rd \leftarrow Rr$	None	1	
MOVW ⁽¹⁾	Rd, Rr	Copy Register Pair	$Rd+1:Rd \leftarrow Rr+1:Rr$	None	1	
LDI	Rd, K	Load Immediate	$Rd \leftarrow K$	None	1	
LDS ⁽¹⁾	Rd, k	Load Direct from data space	$Rd \leftarrow (k)$	None	1 ⁽⁵⁾ /2 ⁽³⁾	2 ⁽³⁾ /4
LD ⁽²⁾	Rd, X	Load Indirect	$Rd \leftarrow (X)$	None	1 ⁽⁵⁾ /2 ⁽³⁾	1 ⁽³⁾ /4
LD ⁽²⁾	Rd, X+	Load Indirect and Post-Increment	$Rd \leftarrow (x)$ $X \leftarrow X+1$	None	2 ⁽³⁾	1 ⁽³⁾ /4
LD ⁽²⁾	Rd, -X	Load Indirect and Pre-Decrement	$X \leftarrow X - 1$, $Rd \leftarrow (X)$	None	2 ⁽³⁾ /3 ⁽⁵⁾	2 ⁽³⁾ /4
LD ⁽²⁾	Rd, Y	Load Indirect	$Rd \leftarrow (Y)$	None	1 ⁽⁵⁾ /2 ⁽³⁾	1 ⁽³⁾ /4
LD ⁽²⁾	Rd, Y+	Load Indirect and Post-Increment	$Rd \leftarrow (Y)$ $Y \leftarrow Y+1$	None	2 ⁽³⁾	1 ⁽³⁾ /4
LD ⁽²⁾	Rd, -Y	Load Indirect and Pre-Decrement Y	$Y \leftarrow Y-1$ $Rd \leftarrow (Y)$	None	2 ⁽³⁾ /3 ⁽⁵⁾	2 ⁽³⁾ /4
LDD ⁽¹⁾	Rd, Y+q	Load Indirect with Displacement	$Rd \leftarrow (Y + q)$	None	2 ⁽³⁾	2 ⁽³⁾ /4

مجموعه دستورالعمل‌های میکروکنترلرهای ۸ بیتی AVR

LD ⁽²⁾	Rd, Z	Load Indirect	$Rd \leftarrow (Z)$	None	1 ⁽⁵⁾ /2 ⁽³⁾	1 ⁽³⁾ (4)
LD ⁽²⁾	Rd, Z+	Load Indirect and Post-Increment	$Rd \leftarrow (Z)$ $Z \leftarrow Z+1$	None	2 ⁽³⁾	1 ⁽³⁾ (4)
LD ⁽²⁾	Rd, -Z	Load Indirect and Pre-Decrement	$Z \leftarrow Z-1$ $Rd \leftarrow (Z)$	None	2 ⁽³⁾ /3 ⁽⁵⁾	2 ⁽³⁾ (4)
LDD ⁽¹⁾	Rd, Z+q	Load Indirect with Displacement	$Rd \leftarrow (Z + q)$	None	2 ⁽³⁾	2 ⁽³⁾ (4)
STS ⁽¹⁾	k, Rr	Store Direct to Data Space	$(k) \leftarrow Rr$	None	1 ⁽⁵⁾ /2 ⁽³⁾	2 ⁽³⁾
ST ⁽²⁾	X, Rr	Store Indirect	$(X) \leftarrow Rr$	None	1 ⁽⁵⁾ /2 ⁽³⁾	1 ⁽³⁾
ST ⁽²⁾	X+, Rr	Store Indirect and Post-Increment	$(X) \leftarrow Rr$ $X \leftarrow X+1$	None	1 ⁽⁵⁾ /2 ⁽³⁾	1 ⁽³⁾
ST ⁽²⁾	-X, Rr	Store Indirect and Pre-Decrement	$X \leftarrow X-1$ $(X) \leftarrow Rr$	None	2 ⁽³⁾	2 ⁽³⁾
ST ⁽²⁾	Y, Rr	Store Indirect	$(Y) \leftarrow Rr$	None	1 ⁽⁵⁾ /2 ⁽³⁾	1 ⁽³⁾
ST ⁽²⁾	Y+, Rr	Store Indirect and Post-Increment	$(Y) \leftarrow Rr$ $Y \leftarrow Y+1$	None	1 ⁽⁵⁾ /2 ⁽³⁾	1 ⁽³⁾
ST ⁽²⁾	-Y, Rr	Store Indirect and Pre-Decrement	$Y \leftarrow Y-1$ $(Y) \leftarrow Rr$	None	2 ⁽³⁾	2 ⁽³⁾
STD ⁽¹⁾	Y+q, Rr	Store Indirect with Displacement	$(Y + q) \leftarrow Rr$	None	2 ⁽³⁾	2 ⁽³⁾
ST ⁽²⁾	Z, Rr	Store Indirect	$(Z) \leftarrow Rr$	None	1 ⁽⁵⁾ /2 ⁽³⁾	1 ⁽³⁾

مجموعه دستورالعمل‌های میکروکنترلرهای ۸ بیتی AVR

ST ⁽²⁾	Z+, Rr	Store Indirect and Post-Increment	$(Z) \leftarrow Rr$ $Z \leftarrow Z+1$	None	1 ⁽⁵⁾ /2 ⁽³⁾	1 ⁽³⁾
ST ⁽²⁾	-Z, Rr	Store Indirect and Pre-Decrement	$Z \leftarrow Z-1$	None	2 ⁽³⁾	2 ⁽³⁾
STD ⁽¹⁾	Z+q,Rr	Store Indirect with Displacement	$(Z + q) \leftarrow Rr$	None	2 ⁽³⁾	2 ⁽³⁾
LP ^{(1) (2)}		Load Program Memory	$R0 \leftarrow (Z)$	None	3	3
LPM ^{(1) (2)}	Rd, Z	Load Program Memory	$Rd \leftarrow (Z)$	None	3	3
LPM ^{(1) (2)}	Rd, Z+	Load Program Memory and Post-Increment	$Rr \leftarrow (Z)$ $Z \leftarrow Z+1$	None	3	3
ELPM ⁽¹⁾		Extended Load Program Memory	$R0 \leftarrow (RAMPZ:Z)$	None	3	
ELPM ⁽¹⁾	Rd, Z	Extended Load Program Memory	$Rd \leftarrow (RAMPZ:Z)$	None	3	
ELPM ⁽¹⁾	Rd, Z+	Extended Load Program Memory and Post-Increment	$Rd \leftarrow (RAMPZ:Z)$ $Z \leftarrow Z+1$	None	3	

مجموعه دستورالعمل‌های میکروکنترلرهای ۸ بیتی AVR

SPM ⁽¹⁾		Store Program Memory	$(RAMPZ:Z) \leftarrow R1:R0$	None	-	-
SPM ⁽¹⁾	Z+	Store Program Memory and Post-Increment by 2	$(RAMPZ:Z) \leftarrow R1:R0$ $Z \leftarrow Z+2$	None	-	-
IN	Rd, A	In From I/O Location	$Rd \leftarrow I/O(A)$	None	1	
OUT	A, Rr	Out To I/O Location	$I/O(A) \leftarrow Rr$	None	1	
PUSH ⁽¹⁾	Rr	Push Register on Stack	$STACK \leftarrow Rr$	None	2	1 ⁽³⁾
POP ⁽¹⁾	Rd	Pop Register from Stack	$Rd \leftarrow STACK$	None	2	2 ⁽³⁾

مجموعه دستورالعمل‌های میکروکنترلرهای ۸ بیتی AVR

Mnemonics	Operands	Description	Operation	Flags	#Clocks	#Clocks XMEGA
LSL	Rd	Logical Shift Left	$Rd(n+1) \leftarrow Rd(n)$ $Rd(0) \leftarrow 0$ $C \leftarrow Rd(7)$	Z,C,N,V,H	1	
LSR	Rd	Logical Shift Right	$Rd(n) \leftarrow Rd(n+1)$ $Rd(7) \leftarrow 0$ $C \leftarrow Rd(0)$	Z,C,N,V	1	
ROL	Rd	Rotate Left Through Carry	$Rd(0) \leftarrow C$ $Rd(n+1) \leftarrow Rd(n)$ $C \leftarrow Rd(7)$	Z,C,N,V,H	1	
ROR	Rd	Rotate Right Through Carry	$Rd(7) \leftarrow C$ $Rd(n) \leftarrow Rd(n+1)$ $C \leftarrow Rd(0)$	Z,C,N,V	1	
ASR	Rd	Arithmetic Shift Right	$Rd(n) \leftarrow Rd(n+1), n=0..6$	Z,C,N,V	1	
SWAP	s	Swap Nibbles	$Rd(3..0) \leftrightarrow Rd(7..4)$	None	1	
BSET	s	Flag Set	$SREG(s) \leftarrow 1$	SREG(s)	1	
BCLR	A, b	Flag Clear	$SREG(s) \leftarrow 0$	SREG(s)	$1^{(5)}/2$	1
SBI	A, b	Set Bit in I/O Register	$I/O(A, b) \leftarrow 1$	None	$1^{(5)}/2$	1
CBI	Rr, b	Clear Bit in I/O Register	$I/O(A, b) \leftarrow 0$	None	1	
BST	Rd, b	Bit Store from Register to T	$T \leftarrow Rr(b)$	T	1	

مجموعه دستورالعمل‌های میکروکنترلرهای ۸ بیتی AVR

BLD		Bit load from T to Register	$Rd(b) \leftarrow T$	None	1	
SEC		Set Carry	$C \leftarrow 1$	C	1	
CLC		Clear Carry	$C \leftarrow 0$	C	1	
SEN		Set Negative Flag	$N \leftarrow 1$	N	1	
CLN		Clear Negative Flag	$N \leftarrow 0$	N	1	
SEZ		Set Zero Flag	$Z \leftarrow 1$	Z	1	
CLZ		Clear Zero Flag	$Z \leftarrow 0$	Z	1	
SEI		Global Interrupt Disable	$I \leftarrow 1$	I	1	
CLI		Global Interrupt Disable	$I \leftarrow 0$	I	1	
SES		Set Signed Test Flag	$S \leftarrow 1$	S	1	
CLS		Clear Signed Test Flag	$S \leftarrow 0$	S	1	
SEV		Set Two's Complement Overflow	$V \leftarrow 1$	V	1	
CLV		Clear Two's Complement Overflow	$V \leftarrow 0$	V	1	
SET		Set T in SREG	$T \leftarrow 1$	T	1	
CLT		Clear T in SREG	$T \leftarrow 0$	T	1	
SEH		Set Half Carry Flag in SREG	$H \leftarrow 1$	H	1	

مجموعه دستورالعمل‌های میکروکنترلرهای ۸ بیتی AVR

Mnemonics	Operands	Description	Operation	Flags	#Clocks	#Clocks XMEGA
BREAK(1)		Break	(See specific descr. for BREAK)	None	1	
NOP		No Operation		None	1	
SLEEP		Sleep	(see specific descr. for Sleep)	None	1	
WDR		Sleep	(see specific descr. for WDR)	None	1	

مجموعه دستورالعمل‌های کنترل میکروکنترلر در میکروکنترلرهای ۸ بیتی AVR

توجه ۱: دستورالعمل BREAK توسط سیستم debug سوار بر تراشه استفاده شده و توسط برنامه‌های کاربردی قابل استفاده نیست.