

MC404

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

2009

Prof. Paulo Cesar Centoducatte

ducatte@ic.unicamp.br

www.ic.unicamp.br/~ducatte

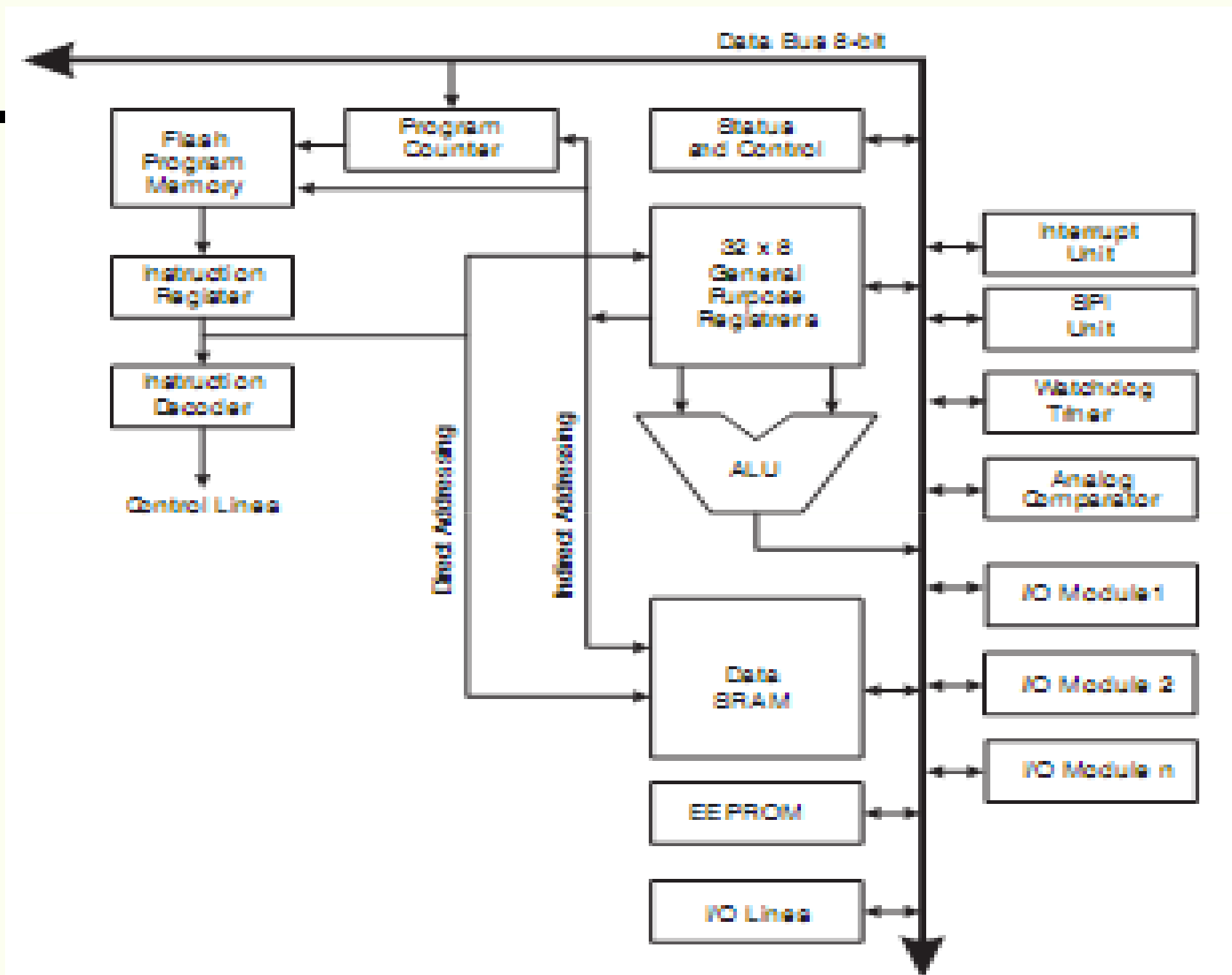
ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

“Organização de Memória e Modos de Endereçamento”

Organização de Memória e Modos de Endereçamento

Sumário

- **Organização de Memória do Atmega88**
 - **Memória de Programa - Flash**
 - **Memória de Dados**
 - **SRAM**
 - **EEPROM**
- **Modos de Endereçamento**



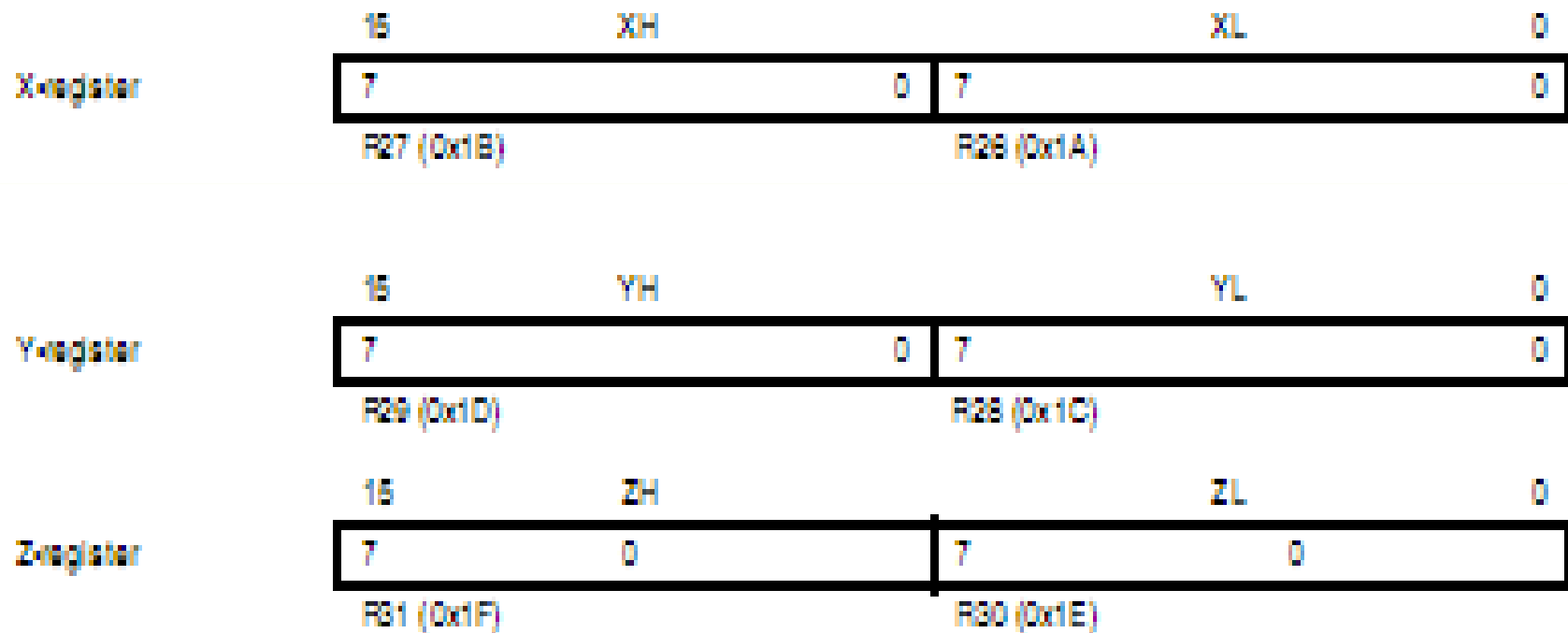
The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figure 4-2. AVR CPU General Purpose Working Registers

			Addr.	
	7	0		
General Purpose Working Registers	R0		0x00	
	R1		0x01	
	R2		0x02	
	...			
	R13		0x0D	
	R14		0x0E	
	R15		0x0F	
	R16		0x10	
	R17		0x11	
	...			
	R26		0x1A	X-register Low Byte
	R27		0x1B	X-register High Byte
	R28		0x1C	Y-register Low Byte
	R29		0x1D	Y-register High Byte
	R30		0x1E	Z-register Low Byte
	R31		0x1F	Z-register High Byte

Figure 4-3. The X-, Y-, and Z-registers

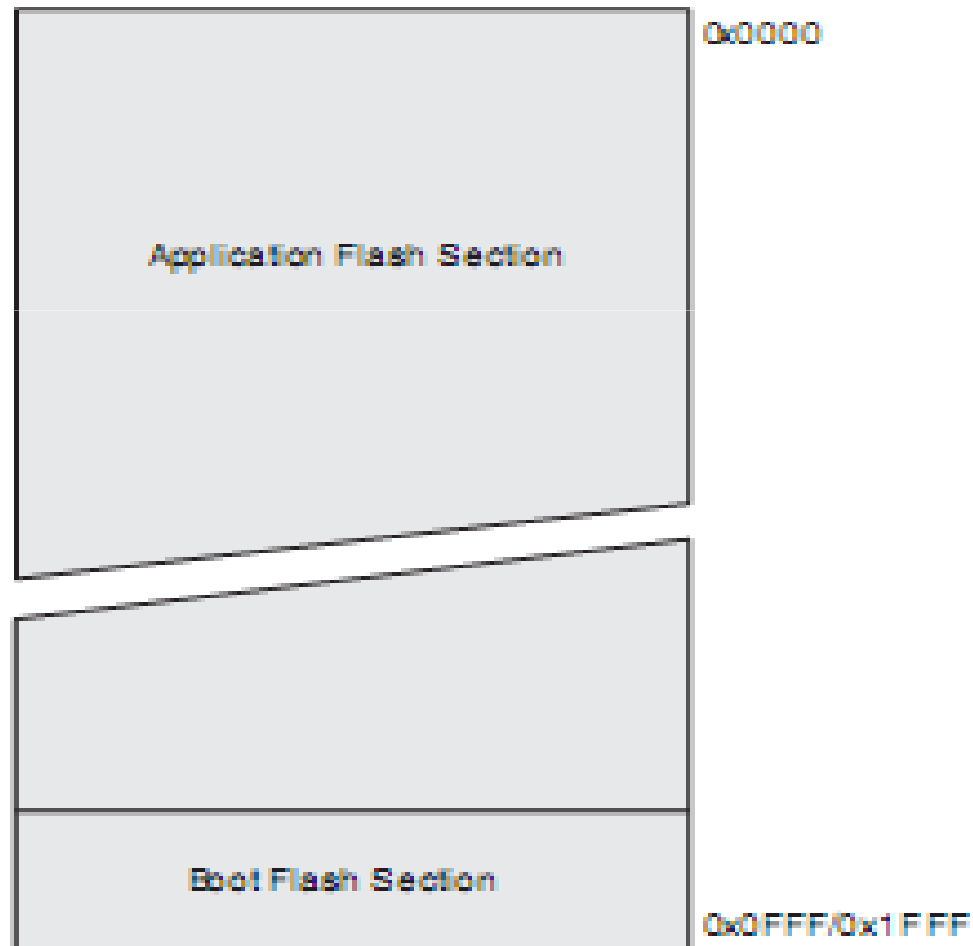


AVR ATmega48/88/168 Memories

In-System Reprogrammable Flash Program Memory

Program Memory Map, ATmega88 and ATmega168

Program Memory



SRAM Data Memory

Data Memory

32 Registers	0x0000 - 0x001F
64 I/O Registers	0x0020 - 0x005F
160 Ext I/O Reg.	0x0060 - 0x00FF
Internal SRAM (512/1024/1024 x 8)	0x0100
	0x02FF/0x04FF/0x04FF

EEPROM Data Memory

The EEPROM Address Register – EEARH and EEARL

Bit	15	14	13	12	11	10	9	8	
	-	-	-	-	-	-	-	EEAR8	EEARH
	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	X	
	X	X	X	X	X	X	X	X	

The EEPROM Data Register – EEDR

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

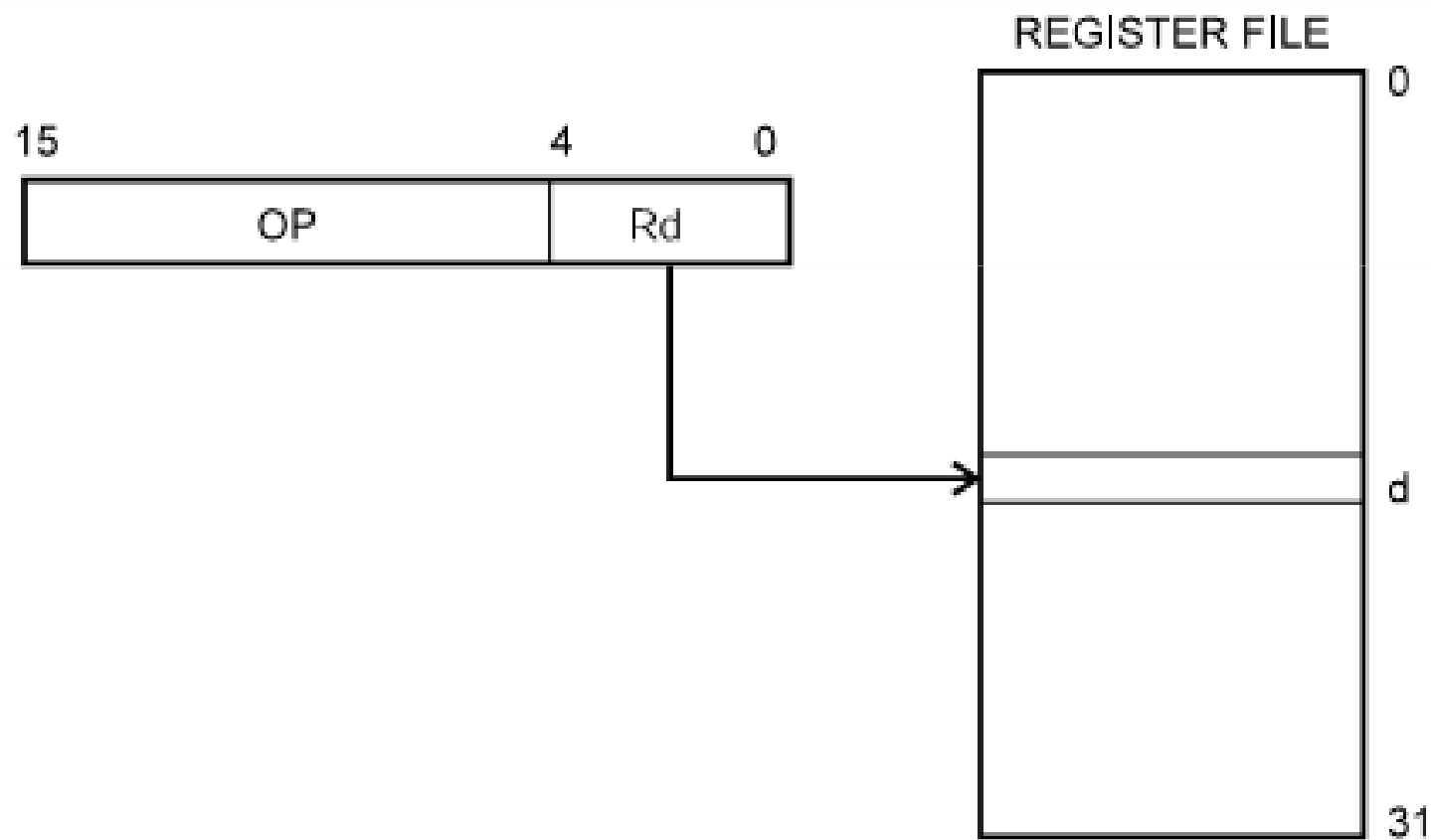
The EEPROM Control Register – EECR

Bit	7	6	5	4	3	2	1	0	
	—	—	EEPM1	EEPM0	EERE	EEMPE	EEPE	EERE	EECR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	X	X	0	0	X	0	

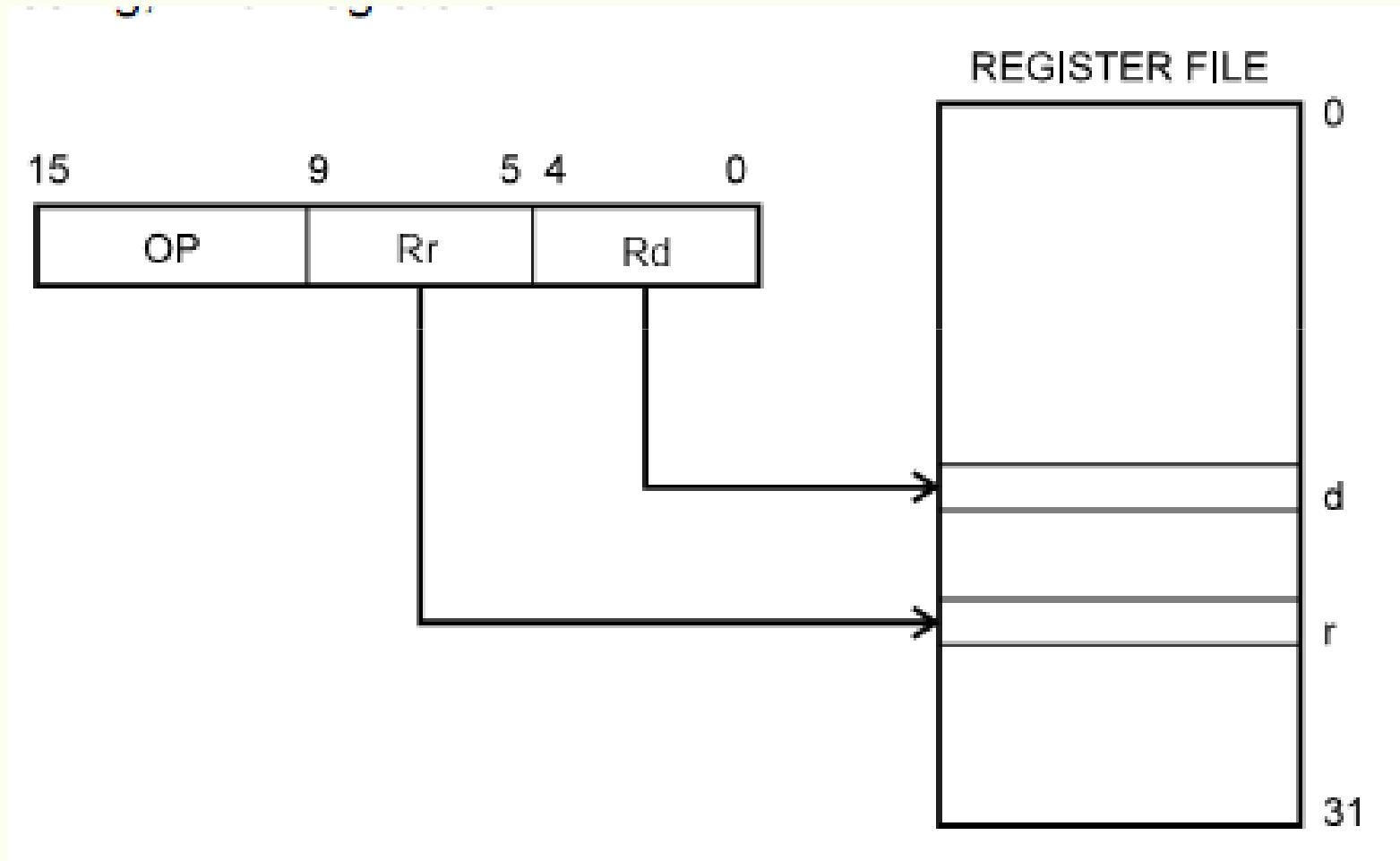
Table 5-1. EEPROM Mode Bits

EEPM1	EEPM0	Programming Time	Operation
0	0	3.4 ms	Erase and Write in one operation (Atomic Operation)
0	1	1.8 ms	Erase Only
1	0	1.8 ms	Write Only
1	1	—	Reserved for future use

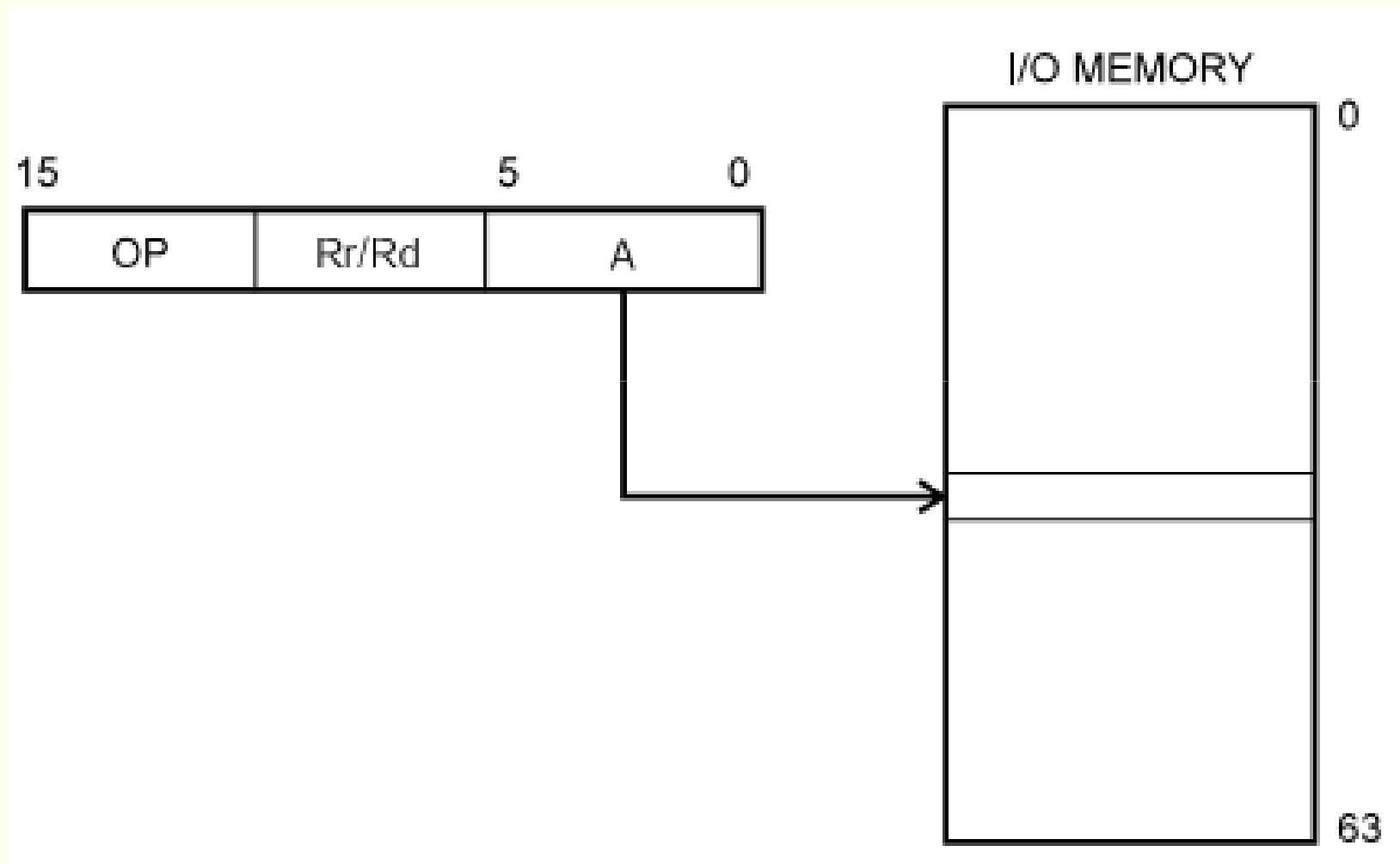
Direct Single Register Addressing



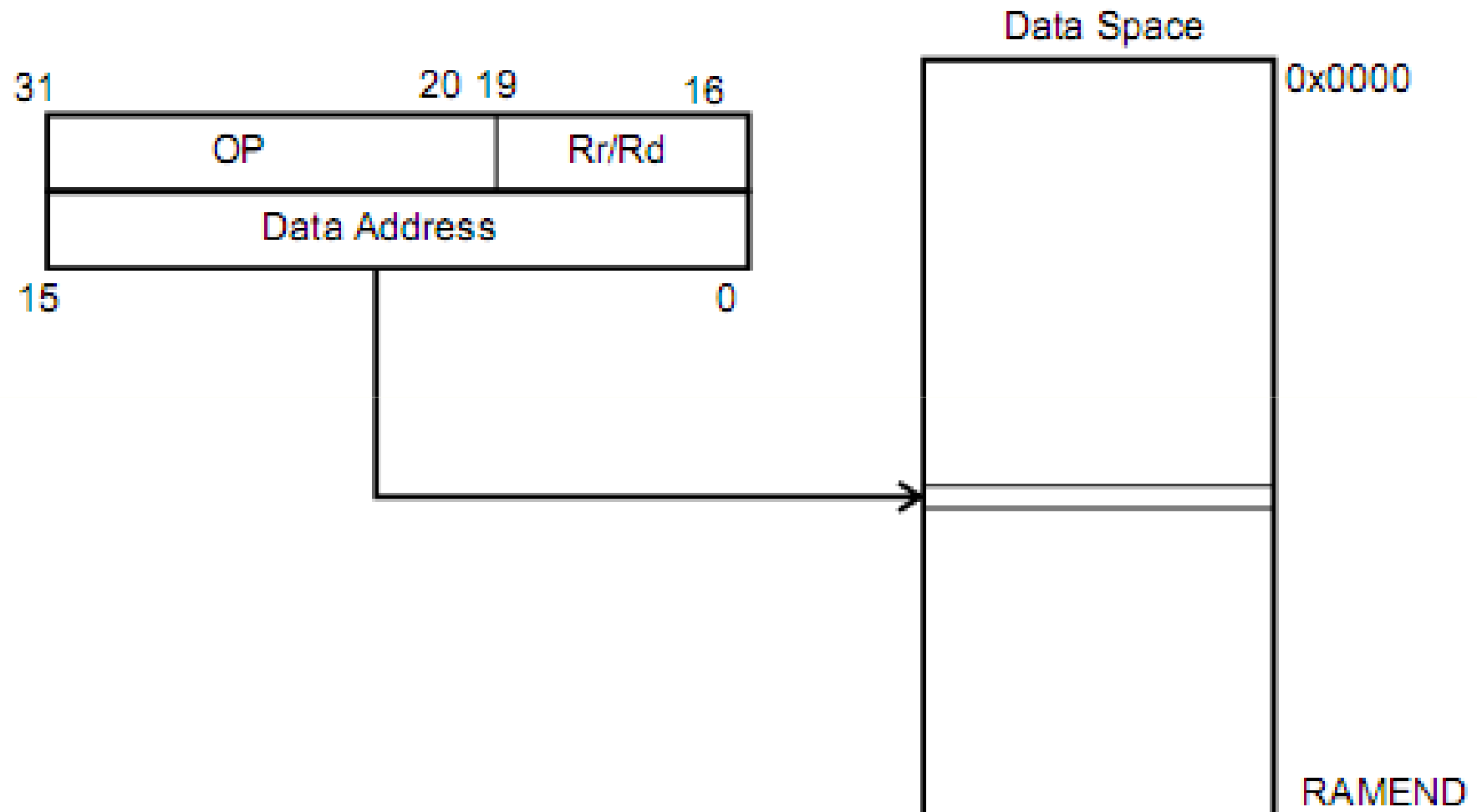
Direct Register Addressing, Two Registers



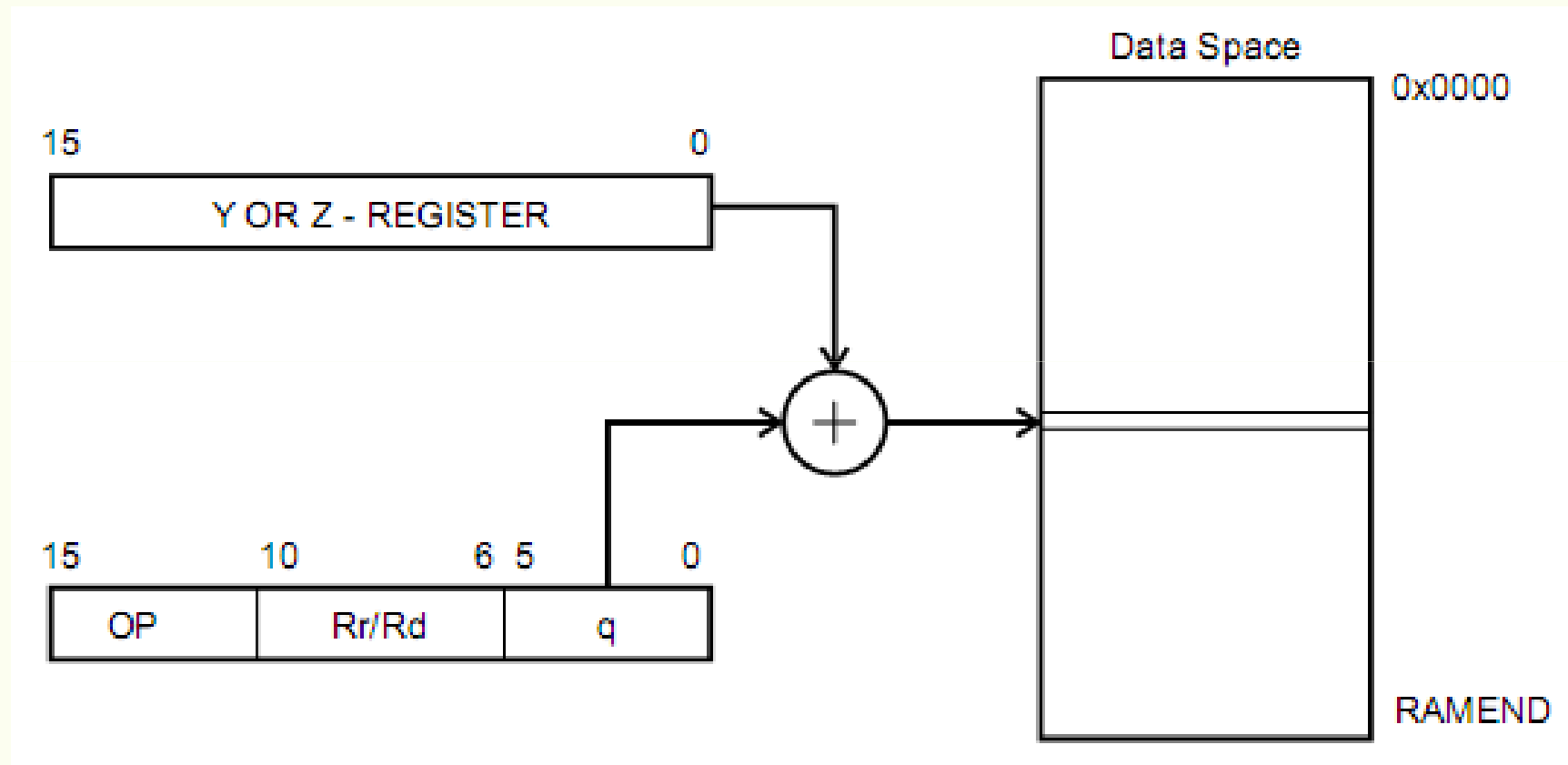
I/O Direct Addressing



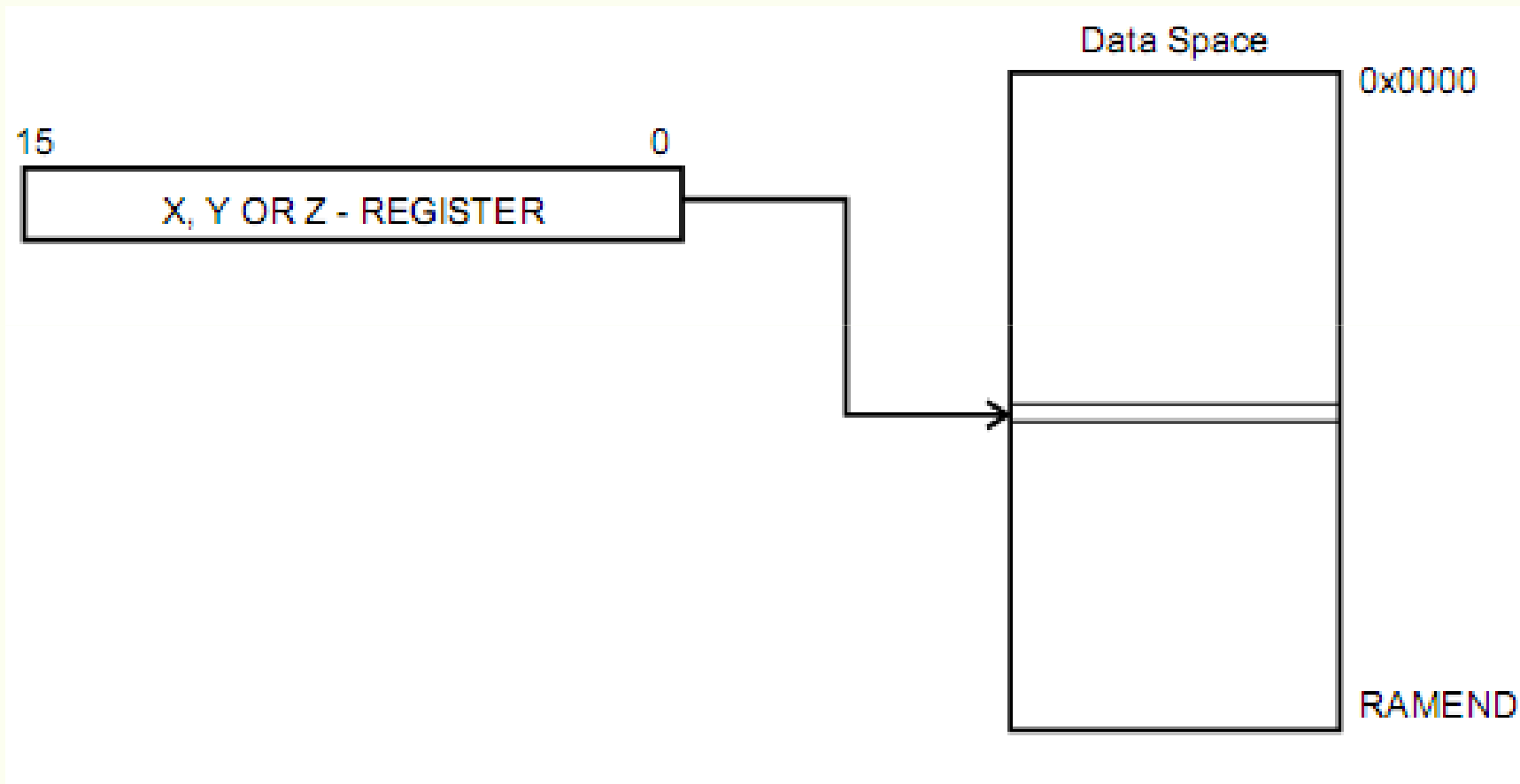
Direct Data Addressing



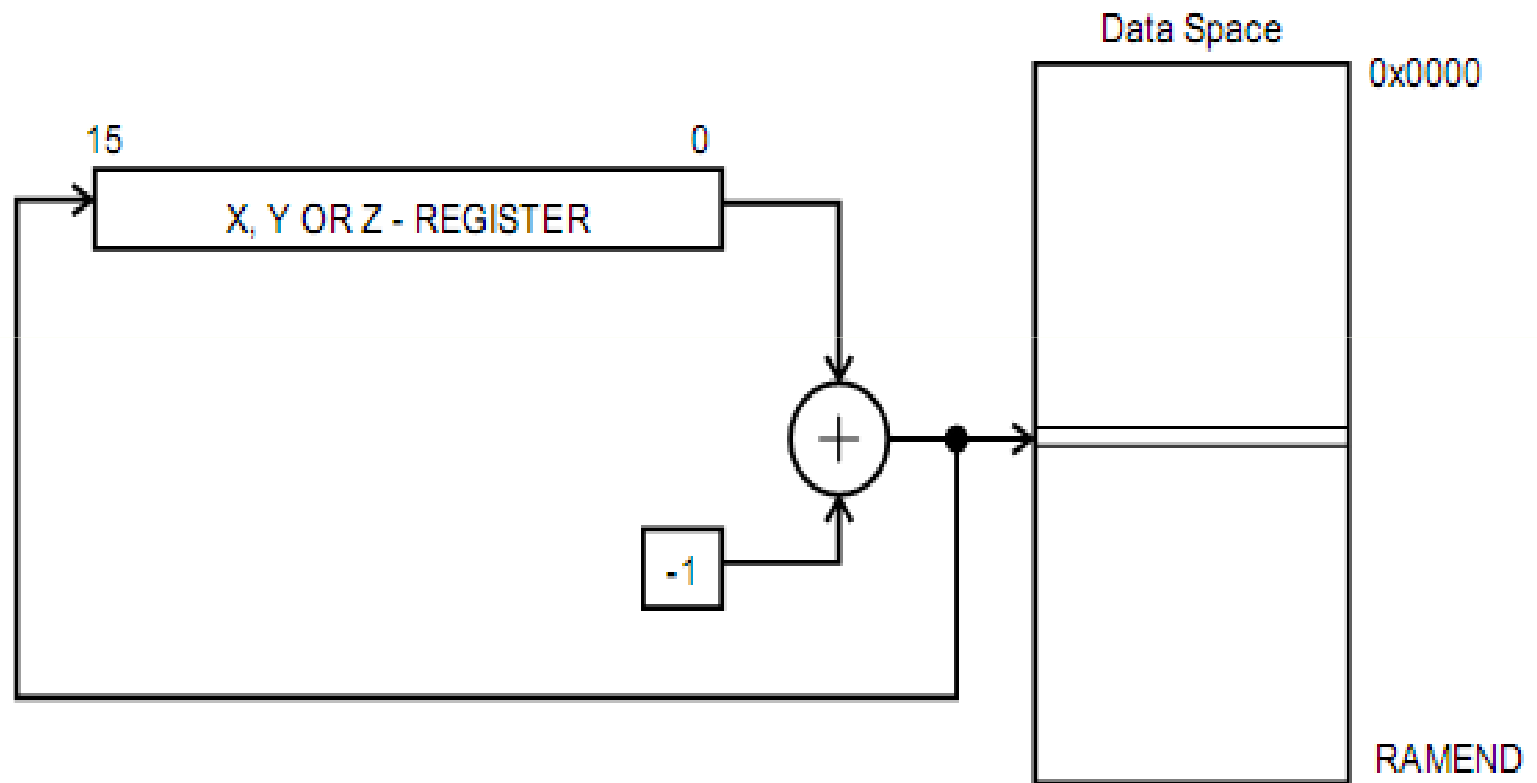
Data Indirect with Displacement



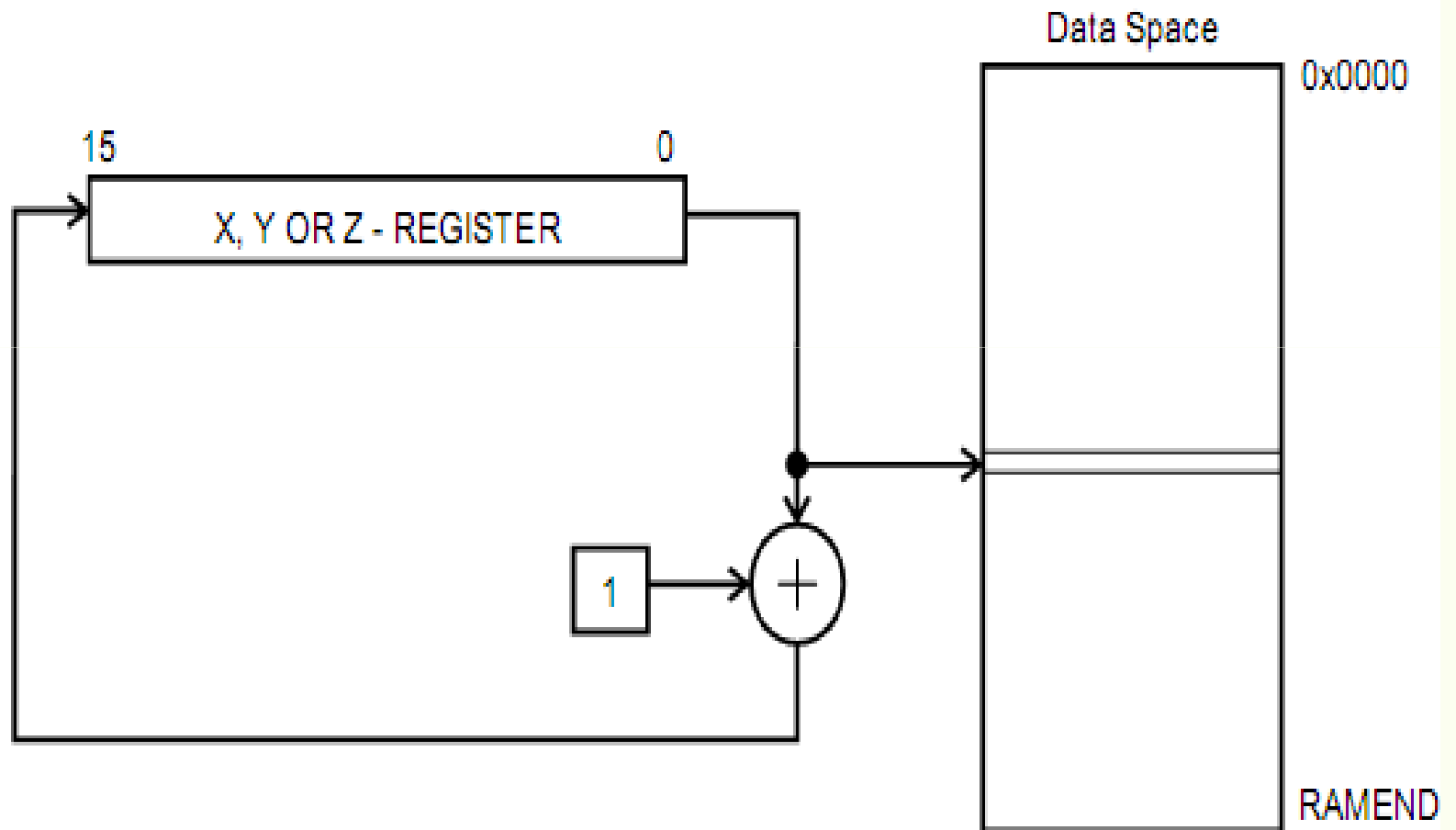
Data Indirect Addressing



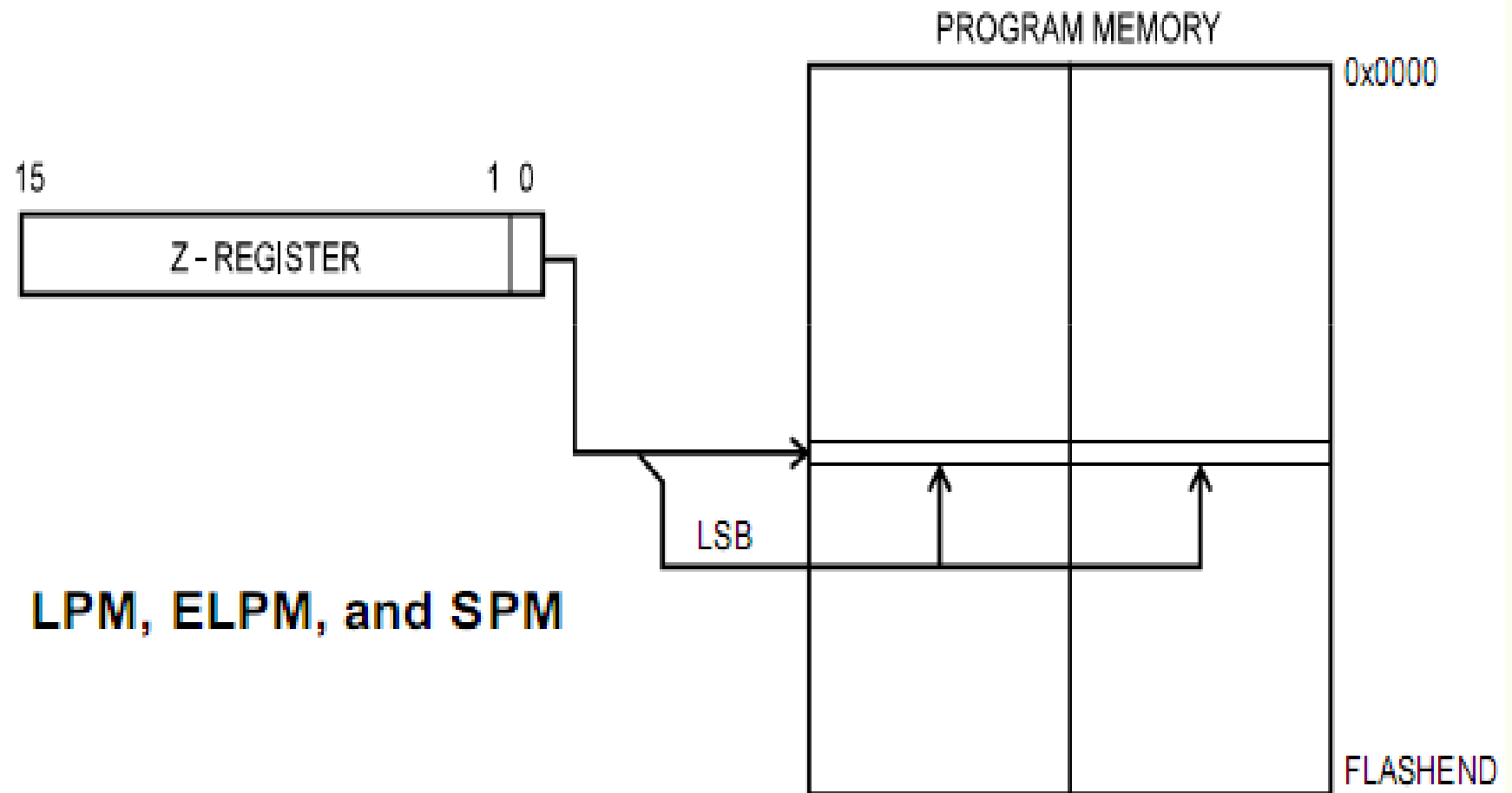
Data Indirect Addressing with Pre-decrement



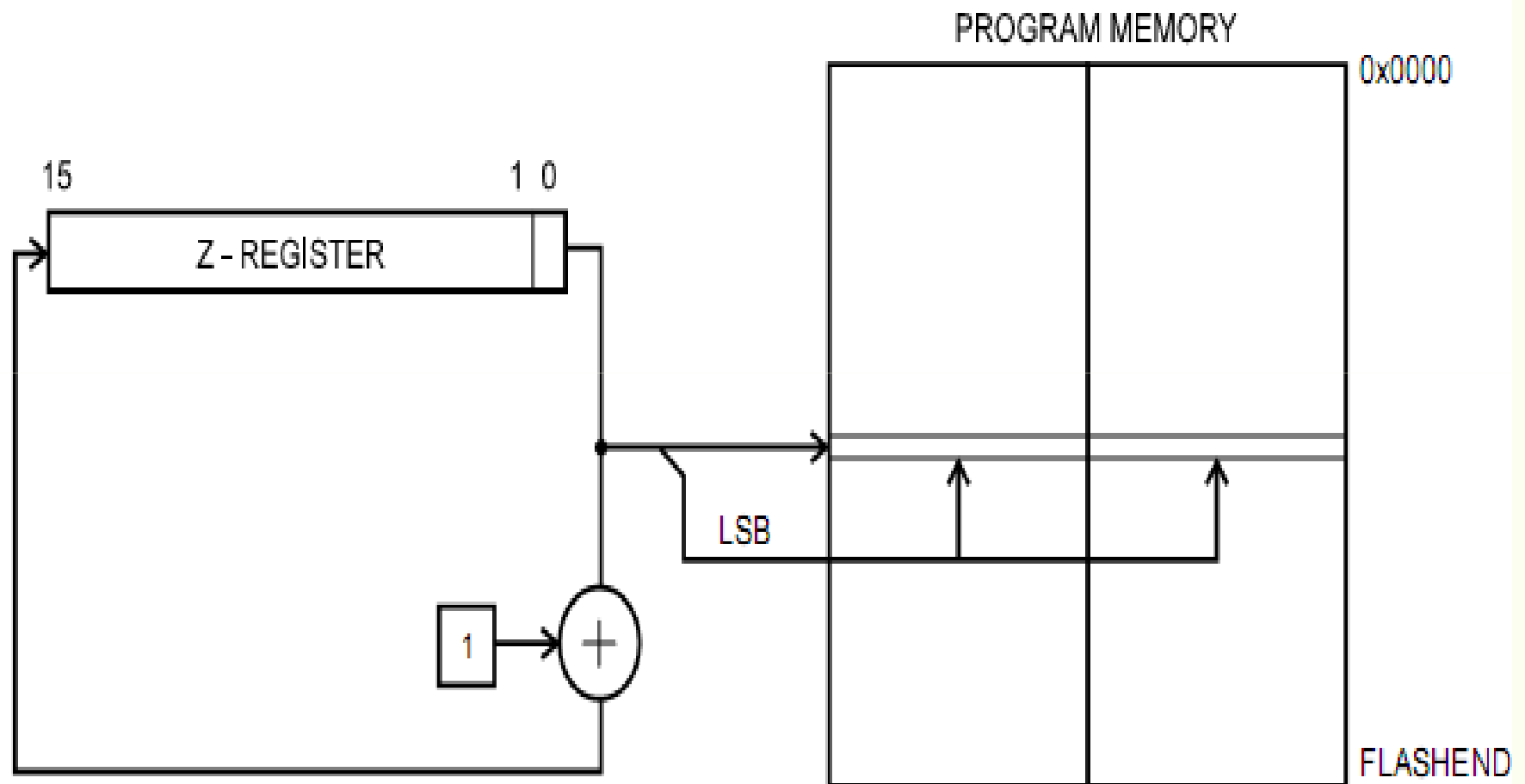
Data Indirect Addressing with Post-increment



Program Memory Constant Addressing

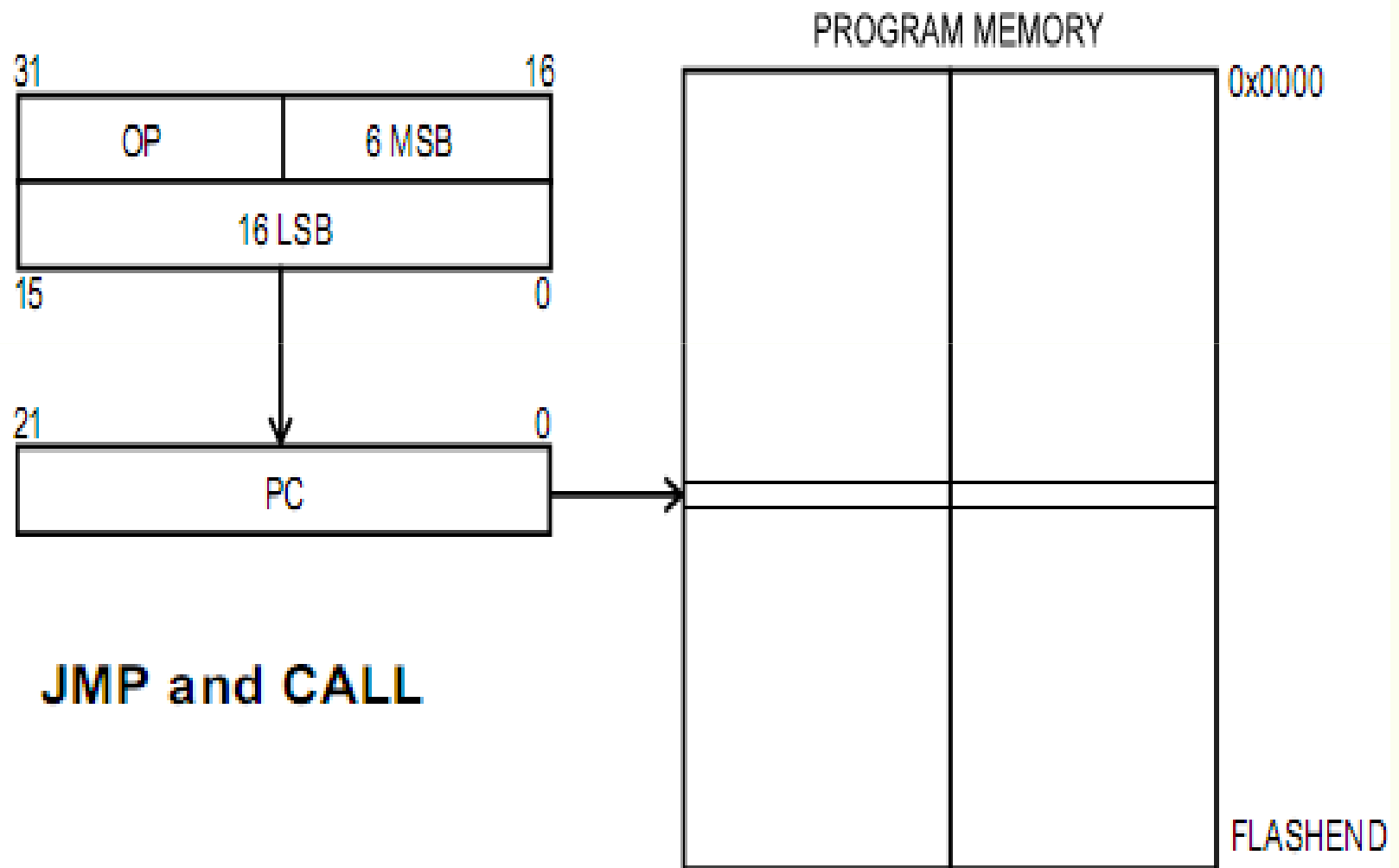


Program Memory Addressing with Post-increment



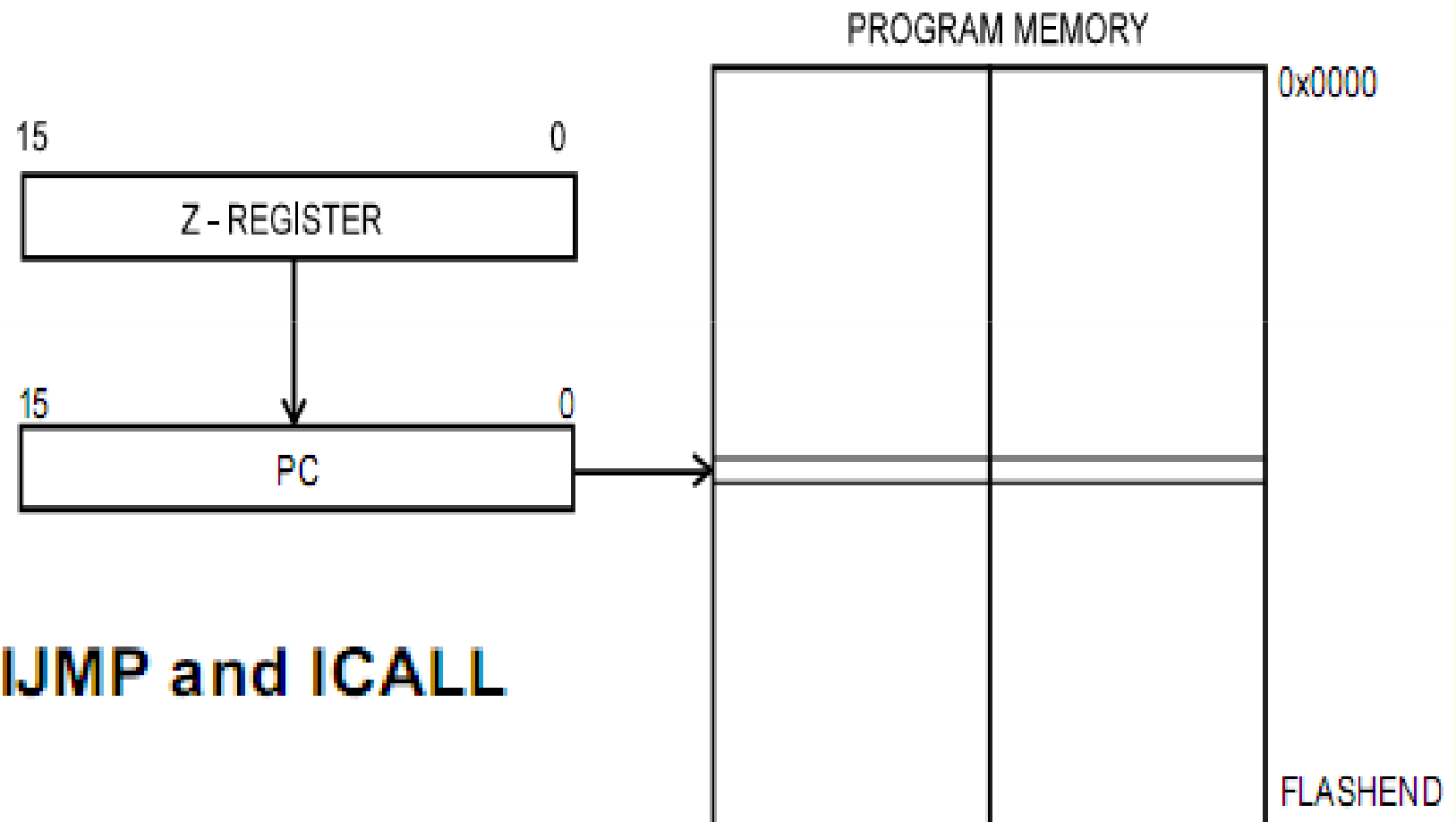
LPM Z+ and ELPM Z+

Direct Program Memory Addressing

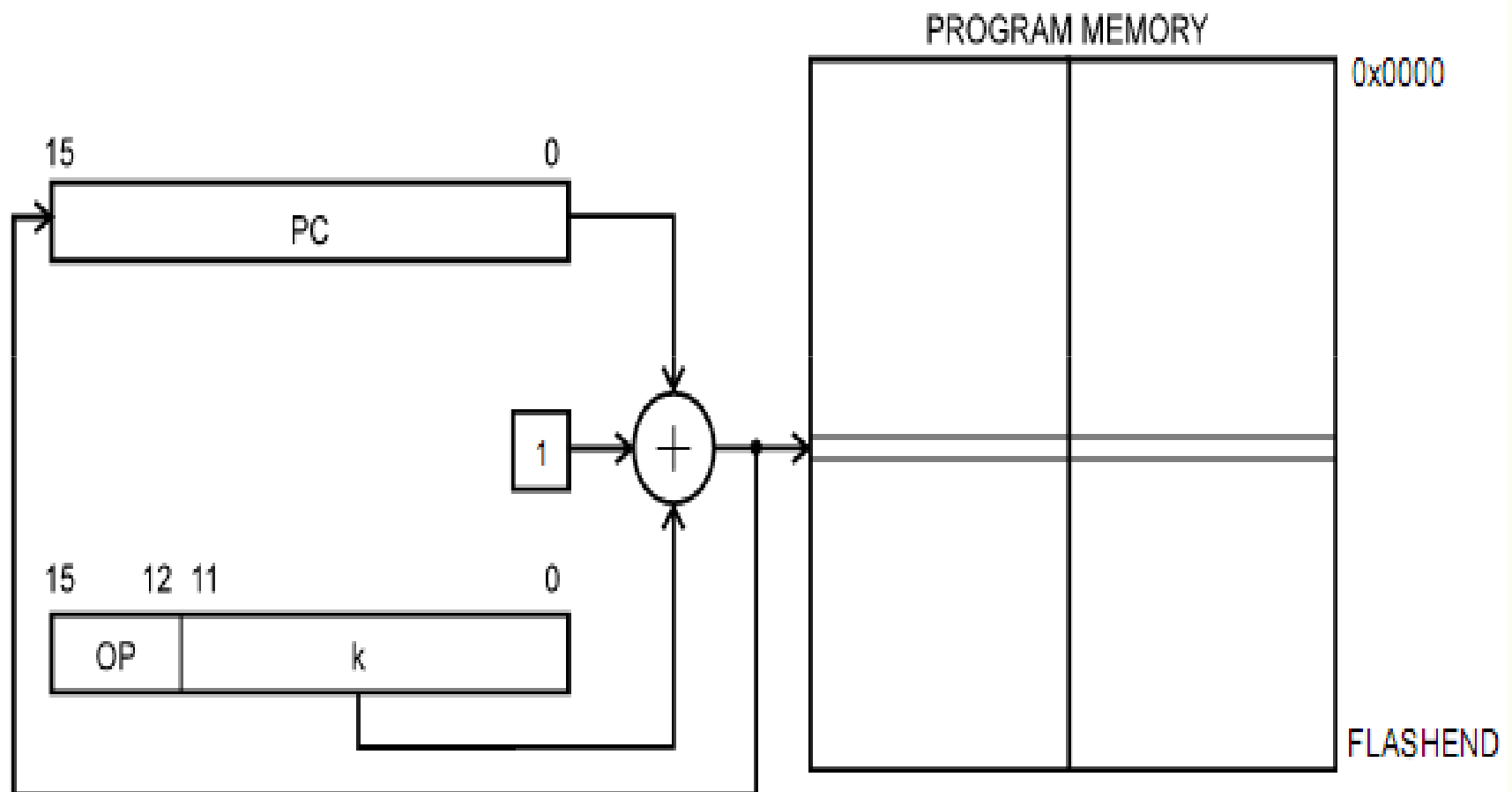


JMP and CALL

Indirect Program Memory Addressing



Relative Program Memory Addressing



RJMP and RCALL

Conditional Branch Summary

Test	Boolean	Mnemonic	Complementary	Boolean	Mnemonic	Comment
$R_d > R_r$	$Z \bullet (N \oplus V) = 0$	BRLT ⁽¹⁾	$R_d \leq R_r$	$Z + (N \oplus V) = 1$	BRGE*	Signed
$R_d \geq R_r$	$(N \oplus V) = 0$	BRGE	$R_d < R_r$	$(N \oplus V) = 1$	BRLT	Signed
$R_d = R_r$	$Z = 1$	BREQ	$R_d \neq R_r$	$Z = 0$	BRNE	Signed
$R_d \leq R_r$	$Z + (N \oplus V) = 1$	BRGE ⁽¹⁾	$R_d > R_r$	$Z \bullet (N \oplus V) = 0$	BRLT*	Signed
$R_d < R_r$	$(N \oplus V) = 1$	BRLT	$R_d \geq R_r$	$(N \oplus V) = 0$	BRGE	Signed
$R_d > R_r$	$C + Z = 0$	BRLO ⁽¹⁾	$R_d \leq R_r$	$C + Z = 1$	BRSH*	Unsigned
$R_d \geq R_r$	$C = 0$	BRSH/BRCC	$R_d < R_r$	$C = 1$	BRLO/BRCS	Unsigned
$R_d = R_r$	$Z = 1$	BREQ	$R_d \neq R_r$	$Z = 0$	BRNE	Unsigned
$R_d \leq R_r$	$C + Z = 1$	BRSH ⁽¹⁾	$R_d > R_r$	$C + Z = 0$	BRLO*	Unsigned
$R_d < R_r$	$C = 1$	BRLO/BRCS	$R_d \geq R_r$	$C = 0$	BRSH/BRCC	Unsigned
Carry	$C = 1$	BRCS	No carry	$C = 0$	BRCC	Simple
Negative	$N = 1$	BRMI	Positive	$N = 0$	BRPL	Simple
Overflow	$V = 1$	BRVS	No overflow	$V = 0$	BRVC	Simple
Zero	$Z = 1$	BREQ	Not zero	$Z = 0$	BRNE	Simple

Note: 1. Interchange R_d and R_r in the operation before the test, i.e., CP $R_d, R_r \rightarrow$ CP R_r, R_d

Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clock Note
Arithmetic and Logic Instructions					
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	Rd, K	Add Immediate to Word	$Rd+1:Rd \leftarrow Rd+1:Rd + K$	Z,C,N,V,S	2 ⁽¹⁾
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	Rd, K	Subtract Immediate from Word	$Rd+1:Rd \leftarrow Rd+1:Rd - K$	Z,C,N,V,S	2 ⁽¹⁾
AND	Rd, Rr	Logical AND	$Rd \leftarrow Rd \bullet Rr$	Z,N,V,S	1
ANDI	Rd, K	Logical AND with Immediate	$Rd \leftarrow Rd \bullet K$	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
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	$Rd \leftarrow Rd \bullet (\$FFh - K)$	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	$Rd \leftarrow Rd \bullet Rd$	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	Rd,Rr	Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr$ (UU)	Z,C	2 ⁽¹⁾
MULS	Rd,Rr	Multiply Signed	$R1:R0 \leftarrow Rd \times Rr$ (SS)	Z,C	2 ⁽¹⁾
MULSU	Rd,Rr	Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr$ (SU)	Z,C	2 ⁽¹⁾
FMUL	Rd,Rr	Fractional Multiply Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$ (UU)	Z,C	2 ⁽¹⁾
FMULS	Rd,Rr	Fractional Multiply Signed	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$ (SS)	Z,C	2 ⁽¹⁾
FMULSU	Rd,Rr	Fractional Multiply Signed with Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$ (SU)	Z,C	2 ⁽¹⁾

Branch Instructions					
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
LMP		Indirect Jump to (Z)	$PC(15:0) \leftarrow Z, PC(21:16) \leftarrow 0$	None	2 ⁽¹⁾
EJMP		Extended Indirect Jump to (Z)	$PC(15:0) \leftarrow Z, PC(21:16) \leftarrow EIND$	None	2 ⁽¹⁾
JMP	k	Jump	$PC \leftarrow k$	None	3 ⁽¹⁾
RCALL	k	Relative Call Subroutine	$PC \leftarrow PC + k + 1$	None	3 / 4 ⁽⁴⁾
ICALL		Indirect Call to (Z)	$PC(15:0) \leftarrow Z, PC(21:16) \leftarrow 0$	None	3 / 4 ⁽¹⁾⁽⁴⁾
EICALL		Extended Indirect Call to (Z)	$PC(15:0) \leftarrow Z, PC(21:16) \leftarrow EIND$	None	4 ⁽¹⁾⁽⁴⁾
CALL	k	Call Subroutine	$PC \leftarrow k$	None	4 / 5 ⁽¹⁾⁽⁴⁾
RET		Subroutine Return	$PC \leftarrow STACK$	None	4 / 5 ⁽⁴⁾
RETI		Interrupt Return	$PC \leftarrow STACK$	I	4 / 5 ⁽⁴⁾
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
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
BRBS	s, k	Branch if Status Flag Set	If (SREG(s) = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	If (SREG(s) = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BREQ	k	Branch if Equal	If (Z = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRNE	k	Branch if Not Equal	If (Z = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRCS	k	Branch if Carry Set	If (C = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRCC	k	Branch if Carry Cleared	If (C = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRSH	k	Branch if Same or Higher	If (C = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRLO	k	Branch if Lower	If (C = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRMI	k	Branch if Minus	If (N = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRPL	k	Branch if Plus	If (N = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	If (N $\oplus$ V = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRLT	k	Branch if Less Than, Signed	If (N $\oplus$ V = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	If (H = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	If (H = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRTS	k	Branch if T Flag Set	If (T = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRTC	k	Branch if T Flag Cleared	If (T = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	If (V = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	If (V = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRIE	k	Branch if Interrupt Enabled	If (I = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2

ADC – Add with Carry

Description:

Adds two registers and the contents of the C Flag and places the result in the destination register Rd.

Operation:

(i) $Rd \leftarrow Rd + Rr + C$

Syntax:

(i) ADC Rd,Rr

Operands:

$0 \leq d \leq 31, 0 \leq r \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0001	11rd	dddd	xxxx
------	------	------	------

Status Register (SREG) Boolean Formula:

I	T	H	S	V	N	Z	C
-	-	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$

ADD – Add without Carry

Description:

Adds two registers without the C Flag and places the result in the destination register Rd.

Operation:

(i) $Rd \leftarrow Rd + Rr$

Syntax:

(i) ADD Rd,Rr

Operands:

$0 \leq d \leq 31, 0 \leq r \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0000	11rd	dddd	xxxx
------	------	------	------

Status Register (SREG) and Boolean Formula:

I	T	H	S	V	N	Z	C
–	–	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$

ADIW – Add Immediate to Word

Description:

Adds an immediate value (0 - 63) to a register pair and places the result in the register pair. This instruction operates on the upper four register pairs, and is well suited for operations on the pointer registers.

This instruction is not available in all devices. Refer to the device specific instruction set summary.

Operation:

(i) $Rd+1:Rd \leftarrow Rd+1:Rd + K$

Syntax:

(i) ADIW Rd+1:Rd,K

Operands:

$d \in \{24,26,28,30\}$, $0 \leq K \leq 63$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0110	KKdd	KKKK
------	------	------	------

Status Register (SREG) and Boolean Formula:

I	T	H	S	V	N	Z	C
-	-	-	=	=	=	=	=

ASR – Arithmetic Shift Right

Description:

Shifts all bits in Rd one place to the right. Bit 7 is held constant. Bit 0 is loaded into the C Flag of the SREG. This operation effectively divides a signed value by two without changing its sign. The Carry Flag can be used to round the result.

Operation:

(i)



	Syntax:	Operands:	Program Counter:
(i)	ASR Rd	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0100	dddd	0101
------	------	------	------

Status Register (SREG) and Boolean Formula:

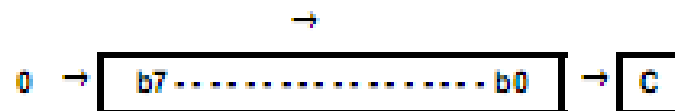
I	T	H	S	V	N	Z	C
-	-	-	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$

LSR – Logical Shift Right

Description:

Shifts all bits in Rd one place to the right. Bit 7 is cleared. Bit 0 is loaded into the C Flag of the SREG. This operation effectively divides an unsigned value by two. The C Flag can be used to round the result.

Operation:



	Syntax:	Operands:	Program Counter:
(i)	LSR Rd	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$

16-bit Opcode:

1001	010d	dddd	0110
------	------	------	------

Status Register (SREG) and Boolean Formula:

I	T	H	S	V	N	Z	C
-	-	-	$\oplus$	$\oplus$	0	$\oplus$	$\oplus$

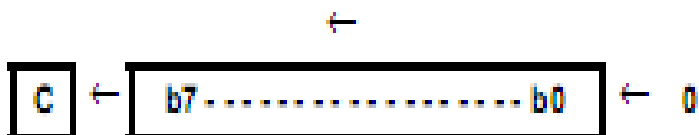
LSL – Logical Shift Left

Description:

Shifts all bits in Rd one place to the left. Bit 0 is cleared. Bit 7 is loaded into the C Flag of the SREG. This operation effectively multiplies signed and unsigned values by two.

Operation:

(i)



	Syntax:	Operands:	Program Counter:
(i)	LSL Rd	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$

16-bit Opcode: (see ADD Rd,Rd)

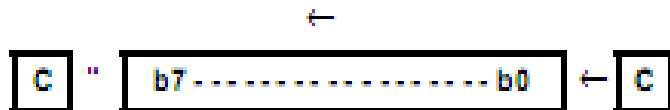
0000	11dd	dddd	dddd
------	------	------	------

ROL – Rotate Left trough Carry

Description:

Shifts all bits in Rd one place to the left. The C Flag is shifted into bit 0 of Rd. Bit 7 is shifted into the C Flag. This operation, combined with LSL, effectively multiplies multi-byte signed and unsigned values by two.

Operation:



	Syntax:	Operands:	Program Counter:
(i)	ROL Rd	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$

16-bit Opcode: (see ADC Rd,Rd)

0001	11dd	dddd	dddd
------	------	------	------

Status Register (SREG) and Boolean Formula:

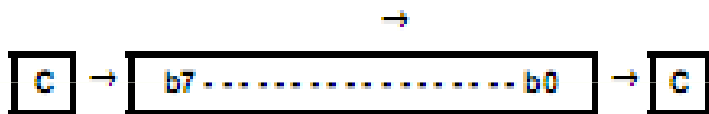
I	T	H	S	V	N	Z	C
-	-	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$

ROR – Rotate Right through Carry

Description:

Shifts all bits in Rd one place to the right. The C Flag is shifted into bit 7 of Rd. Bit 0 is shifted into the C Flag. This operation, combined with ASR, effectively divides multi-byte signed values by two. Combined with LSR it effectively divides multi-byte unsigned values by two. The Carry Flag can be used to round the result.

Operation:



	Syntax:	Operands:	Program Counter:
(i)	ROR Rd	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$

16-bit Opcode:

1001	010d	dddd	0111
------	------	------	------

Status Register (SREG) and Boolean Formula:

I	T	H	S	V	N	Z	C
-	-	-	=	=	=	=	=