

Mikrokontrolery AVR ATmega

Literatura:

8-bit Microcontroller AVR with 32KBytes In-System Programmable Flash ATmega32 [www.atmel.com]

8-bit AVR Instruction Set [www.atmel.com]

Baranowski Rafał, Mikrokontrolery AVR Atmega, BTC, Warszawa 2005



Instrukcje przesyłu danych

Kopiowanie rejestrów

MOV Rd,Rr

Operation:

(i) $Rd \leftarrow Rr$

Syntax:

(i) MOV Rd,Rr

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0010	11rd	dddd	rrrr
------	------	------	------

Kopiowanie par rejestrów

MOVW Rd,Rr

Operation:

- (i) $Rd+1:Rd \leftarrow Rr+1:Rr$

Syntax:

Operands:

- (i) **MOVW** $Rd+1:Rd, Rr+1$ $Rd \in \{0,2,\dots,30\}, r \in \{0,2,\dots,30\}$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0000	0001	dddd	rrrr
------	------	------	------

Ładowanie danej natychmiastowej do rejestru

LDI Rd,K

Operation:

(i) $Rd \leftarrow K$

Syntax:

(i) LDI Rd,K

Operands:

$16 \leq d \leq 31, 0 \leq K \leq 255$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1110	KKKK	dddd	KKKK
------	------	------	------

Ładowanie pośrednie danej z pamięci do rejestru

LDS Rd,k

Operation:

(i) $Rd \leftarrow (k)$

Syntax:

(i) LDS Rd,k

Operands:

$0 \leq d \leq 31, 0 \leq k \leq 65535$

Program Counter:

$PC \leftarrow PC + 2$

32-bit Opcode:

1001	000d	dddd	0000
kkkk	kkkk	kkkk	kkkk

Ładowanie pośrednie danej z pamięci do rejestru (16 bitów)

LDS (16bit) Rd,k

Operation:

(i) $Rd \leftarrow (k)$

Syntax:

(i) LDS Rd,k

Operands:

$16 \leq d \leq 31, 0 \leq k \leq 127$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1010	0kkk	dddd	kkkk
------	------	------	------

Ładowanie pośrednie danej do rejestru

LD Rd,X

Using the X-pointer:

	Operation:		Comment:
(i)	$Rd \leftarrow (X)$		X: Unchanged
(ii)	$Rd \leftarrow (X)$	$X \leftarrow X + 1$	X: Post incremented
(iii)	$X \leftarrow X - 1$	$Rd \leftarrow (X)$	X: Pre decremented
	Syntax:	Operands:	Program Counter:
(i)	LD Rd, X	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(ii)	LD Rd, X+	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iii)	LD Rd, -X	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$

Ładowanie pośrednie danej do rejestru z postinkrementacją

LD Rd,X+

Using the X-pointer:

Operation:		Comment:
(i)	$Rd \leftarrow (X)$	X: Unchanged
(ii)	$Rd \leftarrow (X)$ $X \leftarrow X + 1$	X: Post incremented
(iii)	$X \leftarrow X - 1$ $Rd \leftarrow (X)$	X: Pre decremented
Syntax:		Program Counter:
(i)	LD Rd, X	$PC \leftarrow PC + 1$
(ii)	LD Rd, X+	$PC \leftarrow PC + 1$
(iii)	LD Rd, -X	$PC \leftarrow PC + 1$

Operands:

$0 \leq d \leq 31$

$0 \leq d \leq 31$

$0 \leq d \leq 31$

Ładowanie pośrednie danej do rejestru z predekrementacją

LD Rd,-X

Using the X-pointer:

Operation:			Comment:
(i)	$Rd \leftarrow (X)$		X: Unchanged
(ii)	$Rd \leftarrow (X)$	$X \leftarrow X + 1$	X: Post incremented
(iii)	$X \leftarrow X - 1$	$Rd \leftarrow (X)$	X: Pre decremented
Syntax:		Operands:	Program Counter:
(i)	LD Rd, X	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(ii)	LD Rd, X+	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iii)	LD Rd, -X	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$

Ładowanie pośrednie danej do rejestru

LD Rd,Y

Using the Y-pointer:

	Operation:		Comment:
(i)	$Rd \leftarrow (Y)$		Y: Unchanged
(ii)	$Rd \leftarrow (Y)$	$Y \leftarrow Y + 1$	Y: Post incremented
(iii)	$Y \leftarrow Y - 1$	$Rd \leftarrow (Y)$	Y: Pre decremented
(iv)	$Rd \leftarrow (Y+q)$		Y: Unchanged, q: Displacement

	Syntax:	Operands:	Program Counter:
(i)	LD Rd, Y	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(ii)	LD Rd, Y+	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iii)	LD Rd, -Y	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iv)	LDD Rd, Y+q	$0 \leq d \leq 31, 0 \leq q \leq 63$	$PC \leftarrow PC + 1$

Ładowanie pośrednie danej do rejestru z postinkrementacją

LD Rd,Y+

Using the Y-pointer:

	Operation:		Comment:
(i)	$Rd \leftarrow (Y)$		Y: Unchanged
(ii)	$Rd \leftarrow (Y)$	$Y \leftarrow Y + 1$	Y: Post incremented
(iii)	$Y \leftarrow Y - 1$	$Rd \leftarrow (Y)$	Y: Pre decremented
(iv)	$Rd \leftarrow (Y+q)$		Y: Unchanged, q: Displacement
	Syntax:	Operands:	Program Counter:
(i)	LD Rd, Y	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(ii)	LD Rd, Y+	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iii)	LD Rd, -Y	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iv)	LDD Rd, Y+q	$0 \leq d \leq 31, 0 \leq q \leq 63$	$PC \leftarrow PC + 1$

Ładowanie pośrednie danej do rejestru z predekrementacją

LD Rd,-Y

Using the Y-pointer:

	Operation:		Comment:
(i)	$Rd \leftarrow (Y)$		Y: Unchanged
(ii)	$Rd \leftarrow (Y)$	$Y \leftarrow Y + 1$	Y: Post incremented
(iii)	$Y \leftarrow Y - 1$	$Rd \leftarrow (Y)$	Y: Pre decremented
(iv)	$Rd \leftarrow (Y+q)$		Y: Unchanged, q: Displacement
	Syntax:	Operands:	Program Counter:
(i)	LD Rd, Y	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(ii)	LD Rd, Y+	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iii)	LD Rd, -Y	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iv)	LDD Rd, Y+q	$0 \leq d \leq 31, 0 \leq q \leq 63$	$PC \leftarrow PC + 1$

Ładowanie pośrednie danej do rejestru

LD Rd,Z

Using the Z-pointer:

	Operation:	Comment:	
(i)	$Rd \leftarrow (Z)$		Z: Unchanged
(ii)	$Rd \leftarrow (Z)$	$Z \leftarrow Z + 1$	Z: Post increment
(iii)	$Z \leftarrow Z - 1$	$Rd \leftarrow (Z)$	Z: Pre decrement
(iv)	$Rd \leftarrow (Z+q)$		Z: Unchanged, q: Displacement
	Syntax:	Operands:	Program Counter:
(i)	LD Rd, Z	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(ii)	LD Rd, Z+	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iii)	LD Rd, -Z	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iv)	LDD Rd, Z+q	$0 \leq d \leq 31, 0 \leq q \leq 63$	$PC \leftarrow PC + 1$

Ładowanie pośrednie danej do rejestru z postinkrementacją

LD Rd,Z+

Using the Z-pointer:

	Operation:	Comment:	
(i)	$Rd \leftarrow (Z)$		Z: Unchanged
(ii)	$Rd \leftarrow (Z)$	$Z \leftarrow Z + 1$	Z: Post increment
(iii)	$Z \leftarrow Z - 1$	$Rd \leftarrow (Z)$	Z: Pre decrement
(iv)	$Rd \leftarrow (Z+q)$		Z: Unchanged, q: Displacement
	Syntax:	Operands:	Program Counter:
(i)	LD Rd, Z	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(ii)	LD Rd, Z+	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iii)	LD Rd, -Z	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iv)	LDD Rd, Z+q	$0 \leq d \leq 31, 0 \leq q \leq 63$	$PC \leftarrow PC + 1$

Ładowanie pośrednie danej do rejestru z predekrementacją

LD Rd,-Z

Using the Z-pointer:

	Operation:	Comment:	
(i)	$Rd \leftarrow (Z)$		Z: Unchanged
(ii)	$Rd \leftarrow (Z)$	$Z \leftarrow Z + 1$	Z: Post increment
(iii)	$Z \leftarrow Z - 1$	$Rd \leftarrow (Z)$	Z: Pre decrement
(iv)	$Rd \leftarrow (Z+q)$		Z: Unchanged, q: Displacement
	Syntax:	Operands:	Program Counter:
(i)	LD Rd, Z	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(ii)	LD Rd, Z+	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iii)	LD Rd, -Z	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iv)	LDD Rd, Z+q	$0 \leq d \leq 31, 0 \leq q \leq 63$	$PC \leftarrow PC + 1$

Ładowanie pośrednie danej do rejestru z przesunięciem

LD Rd,Y+q

Using the Z-pointer:

	Operation:	Comment:	
(i)	$Rd \leftarrow (Z)$		Z: Unchanged
(ii)	$Rd \leftarrow (Z)$	$Z \leftarrow Z + 1$	Z: Post increment
(iii)	$Z \leftarrow Z - 1$	$Rd \leftarrow (Z)$	Z: Pre decrement
(iv)	$Rd \leftarrow (Z+q)$		Z: Unchanged, q: Displacement
	Syntax:	Operands:	Program Counter:
(i)	LD Rd, Z	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(ii)	LD Rd, Z+	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iii)	LD Rd, -Z	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iv)	LDD Rd, Z+q	$0 \leq d \leq 31, 0 \leq q \leq 63$	$PC \leftarrow PC + 1$

Ładowanie pośrednie danej do rejestru z przesunięciem

LD Rd,Z+q

Using the Z-pointer:

	Operation:	Comment:	
(i)	$Rd \leftarrow (Z)$		Z: Unchanged
(ii)	$Rd \leftarrow (Z)$	$Z \leftarrow Z + 1$	Z: Post increment
(iii)	$Z \leftarrow Z - 1$	$Rd \leftarrow (Z)$	Z: Pre decrement
(iv)	$Rd \leftarrow (Z+q)$		Z: Unchanged, q: Displacement
	Syntax:	Operands:	Program Counter:
(i)	LD Rd, Z	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(ii)	LD Rd, Z+	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iii)	LD Rd, -Z	$0 \leq d \leq 31$	$PC \leftarrow PC + 1$
(iv)	LDD Rd, Z+q	$0 \leq d \leq 31, 0 \leq q \leq 63$	$PC \leftarrow PC + 1$

Wysłanie danej z rejestru do komórki pamięci o adresie bezpośrednim

STS k,Rr

Operation:

(i) $(k) \leftarrow Rr$

Syntax:

(i) STS k,Rr

Operands:

$0 \leq r \leq 31, 0 \leq k \leq 65535$

Program Counter:

$PC \leftarrow PC + 2$

32-bit Opcode:

1001	001d	dddd	0000
kkkk	kkkk	kkkk	kkkk

Wysłanie danej z rejestru do komórki pamięci o adresie bezpośrednim
(16 bitów)

STS k,Rr

Operation:

(i) $(k) \leftarrow Rr$

Syntax:

(i) STS k,Rr

Operands:

$16 \leq r \leq 31, 0 \leq k \leq 127$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1010	1kkk	dddd	kkkk
------	------	------	------

Przesłanie danej z rejestru do komórki pamięci o adresie zawartym w X

ST X,Rr

	Operation:		Comment:
(i)	$(X) \leftarrow Rr$		X: Unchanged
(ii)	$(X) \leftarrow Rr$	$X \leftarrow X+1$	X: Post incremented
(iii)	$X \leftarrow X - 1$	$(X) \leftarrow Rr$	X: Pre decremented
	Syntax:	Operands:	Program Counter:
(i)	ST X, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(ii)	ST X+, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(iii)	ST -X, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$

Przesłanie danej z rejestru do komórki pamięci o adresie zawartym w X
z postinkrementacją

ST X+,Rr

Operation:

- | | | |
|-------|----------------------|---------------------|
| (i) | $(X) \leftarrow Rr$ | |
| (ii) | $(X) \leftarrow Rr$ | $X \leftarrow X+1$ |
| (iii) | $X \leftarrow X - 1$ | $(X) \leftarrow Rr$ |

Comment:

X: Unchanged
X: Post incremented
X: Pre decremented

Syntax:

- | | |
|-------|-----------|
| (i) | ST X, Rr |
| (ii) | ST X+, Rr |
| (iii) | ST -X, Rr |

Operands:

$0 \leq r \leq 31$
 $0 \leq r \leq 31$
 $0 \leq r \leq 31$

Program Counter:

$PC \leftarrow PC + 1$
 $PC \leftarrow PC + 1$
 $PC \leftarrow PC + 1$

Przesłanie danej z rejestru do komórki pamięci o adresie zawartym w X
z predekrementacją

ST -X,Rr

	Operation:		Comment:
(i)	$(X) \leftarrow Rr$		X: Unchanged
(ii)	$(X) \leftarrow Rr$	$X \leftarrow X+1$	X: Post incremented
(iii)	$X \leftarrow X - 1$	$(X) \leftarrow Rr$	X: Pre decremented
	Syntax:	Operands:	Program Counter:
(i)	ST X, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(ii)	ST X+, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(iii)	ST -X, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$

Przesłanie danej z rejestru do komórki pamięci o adresie zawartym w Y

ST Y,Rr

	Operation:		Comment:
(i)	$(Y) \leftarrow Rr$		Y: Unchanged
(ii)	$(Y) \leftarrow Rr$	$Y \leftarrow Y+1$	Y: Post incremented
(iii)	$Y \leftarrow Y - 1$	$(Y) \leftarrow Rr$	Y: Pre decremented
(iv)	$(Y+q) \leftarrow Rr$		Y: Unchanged, q: Displacement
	Syntax:	Operands:	Program Counter:
(i)	ST Y, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(ii)	ST Y+, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(iii)	ST -Y, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(iv)	STD Y+q, Rr	$0 \leq r \leq 31, 0 \leq q \leq 63$	$PC \leftarrow PC + 1$

Przesłanie danej z rejestru do komórki pamięci o adresie zawartym w Y
z postinkrementacją

ST Y+,Rr

Operation:		Comment:
(i)	$(Y) \leftarrow Rr$	Y: Unchanged
(ii)	$(Y) \leftarrow Rr$ $Y \leftarrow Y+1$	Y: Post incremented
(iii)	$Y \leftarrow Y - 1$ $(Y) \leftarrow Rr$	Y: Pre decremented
(iv)	$(Y+q) \leftarrow Rr$	Y: Unchanged, q: Displacement

Syntax:		Operands:	Program Counter:
(i)	ST Y, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(ii)	ST Y+, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(iii)	ST -Y, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(iv)	STD Y+q, Rr	$0 \leq r \leq 31, 0 \leq q \leq 63$	$PC \leftarrow PC + 1$

Przesłanie danej z rejestru do komórki pamięci o adresie zawartym w Y
z predekrementacją

ST -Y,Rr

	Operation:		Comment:
(i)	$(Y) \leftarrow Rr$		Y: Unchanged
(ii)	$(Y) \leftarrow Rr$	$Y \leftarrow Y+1$	Y: Post incremented
(iii)	$Y \leftarrow Y - 1$	$(Y) \leftarrow Rr$	Y: Pre decremented
(iv)	$(Y+q) \leftarrow Rr$		Y: Unchanged, q: Displacement
	Syntax:	Operands:	Program Counter:
(i)	ST Y, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(ii)	ST Y+, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(iii)	ST -Y, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(iv)	STD Y+q, Rr	$0 \leq r \leq 31, 0 \leq q \leq 63$	$PC \leftarrow PC + 1$

Przesłanie danej z rejestru do komórki pamięci o adresie zawartym w Y
z przrsumięciem

STD Y+q,Rr

Operation:		Comment:
(i)	$(Y) \leftarrow Rr$	Y: Unchanged
(ii)	$(Y) \leftarrow Rr$ $Y \leftarrow Y+1$	Y: Post incremented
(iii)	$Y \leftarrow Y - 1$ $(Y) \leftarrow Rr$	Y: Pre decremented
(iv)	$(Y+q) \leftarrow Rr$	Y: Unchanged, q: Displacement

Syntax:		Operands:	Program Counter:
(i)	ST Y, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(ii)	ST Y+, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(iii)	ST -Y, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(iv)	STD Y+q, Rr	$0 \leq r \leq 31, 0 \leq q \leq 63$	$PC \leftarrow PC + 1$

Przesłanie danej z rejestru do komórki pamięci o adresie zawartym w Z

ST Z,Rr

	Operation:		Comment:
(i)	$(Z) \leftarrow Rr$		Z: Unchanged
(ii)	$(Z) \leftarrow Rr$	$Z \leftarrow Z+1$	Z: Post incremented
(iii)	$Z \leftarrow Z - 1$	$(Z) \leftarrow Rr$	Z: Pre decremented
(iv)	$(Z+q) \leftarrow Rr$		Z: Unchanged, q: Displacement
	Syntax:	Operands:	Program Counter:
(i)	ST Z, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(ii)	ST Z+, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(iii)	ST -Z, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(iv)	STD Z+q, Rr	$0 \leq r \leq 31, 0 \leq q \leq 63$	$PC \leftarrow PC + 1$

Przesłanie danej z rejestru do komórki pamięci o adresie zawartym w Z z postinkrementacją

ST Z+,Rr

	Operation:		Comment:
(i)	$(Z) \leftarrow Rr$		Z: Unchanged
(ii)	$(Z) \leftarrow Rr$	$Z \leftarrow Z+1$	Z: Post incremented
(iii)	$Z \leftarrow Z - 1$	$(Z) \leftarrow Rr$	Z: Pre decremented
(iv)	$(Z+q) \leftarrow Rr$		Z: Unchanged, q: Displacement
	Syntax:	Operands:	Program Counter:
(i)	ST Z, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(ii)	ST Z+, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(iii)	ST -Z, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(iv)	STD Z+q, Rr	$0 \leq r \leq 31, 0 \leq q \leq 63$	$PC \leftarrow PC + 1$

Przesłanie danej z rejestru do komórki pamięci o adresie zawartym w Z z predekrementacją

ST -Z,Rr

	Operation:		Comment:
(i)	$(Z) \leftarrow Rr$		Z: Unchanged
(ii)	$(Z) \leftarrow Rr$	$Z \leftarrow Z+1$	Z: Post incremented
(iii)	$Z \leftarrow Z - 1$	$(Z) \leftarrow Rr$	Z: Pre decremented
(iv)	$(Z+q) \leftarrow Rr$		Z: Unchanged, q: Displacement
	Syntax:	Operands:	Program Counter:
(i)	ST Z, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(ii)	ST Z+, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(iii)	ST -Z, Rr	$0 \leq r \leq 31$	$PC \leftarrow PC + 1$
(iv)	STD Z+q, Rr	$0 \leq r \leq 31, 0 \leq q \leq 63$	$PC \leftarrow PC + 1$

Przesłanie danej z rejestru do komórki pamięci o adresie zawartym w Z z przsunieniem

STD Z+q,Rr

Operation:

- | | | |
|-------|-----------------------|---------------------|
| (i) | $(Z) \leftarrow Rr$ | |
| (ii) | $(Z) \leftarrow Rr$ | $Z \leftarrow Z+1$ |
| (iii) | $Z \leftarrow Z - 1$ | $(Z) \leftarrow Rr$ |
| (iv) | $(Z+q) \leftarrow Rr$ | |

Comment:

Z: Unchanged
Z: Post incremented
Z: Pre decremented
Z: Unchanged, q: Displacement

Syntax:

- | | |
|-------|-------------|
| (i) | ST Z, Rr |
| (ii) | ST Z+, Rr |
| (iii) | ST -Z, Rr |
| (iv) | STD Z+q, Rr |

Operands:

- | |
|--------------------------------------|
| $0 \leq r \leq 31$ |
| $0 \leq r \leq 31$ |
| $0 \leq r \leq 31$ |
| $0 \leq r \leq 31, 0 \leq q \leq 63$ |

Program Counter:

- | |
|------------------------|
| $PC \leftarrow PC + 1$ |
| $PC \leftarrow PC + 1$ |
| $PC \leftarrow PC + 1$ |
| $PC \leftarrow PC + 1$ |

Załaduj daną z pamięci programu (spod adresu zawartego w Z) do rejestru R0

LPM

Operation:		Comment:
(i)	$R0 \leftarrow (Z)$	Z: Unchanged, R0 implied destination register
(ii)	$Rd \leftarrow (Z)$	Z: Unchanged
(iii)	$Rd \leftarrow (Z) \quad Z \leftarrow Z + 1$	Z: Post incremented
Syntax:		Program Counter:
(i)	LPM	$PC \leftarrow PC + 1$
(ii)	LPM Rd, Z	$PC \leftarrow PC + 1$
(iii)	LPM Rd, Z+	$PC \leftarrow PC + 1$
Operands:		
None, R0 implied		
$0 \leq d \leq 31$		
$0 \leq d \leq 31$		

Załaduj daną z pamięci programu (spod adresu zawartego w Z) do rejestru Rd

LPM Rd,Z

Operation:

- (i) $R0 \leftarrow (Z)$
- (ii) $Rd \leftarrow (Z)$
- (iii) $Rd \leftarrow (Z) \quad Z \leftarrow Z + 1$

Comment:

- Z: Unchanged, R0 implied destination register
- Z: Unchanged
- Z: Post incremented

Syntax:

- (i) LPM
- (ii) LPM Rd, Z
- (iii) LPM Rd, Z+

Operands:

- None, R0 implied
- $0 \leq d \leq 31$
- $0 \leq d \leq 31$

Program Counter:

- $PC \leftarrow PC + 1$
- $PC \leftarrow PC + 1$
- $PC \leftarrow PC + 1$

Załaduj daną z pamięci programu (spod adresu zawartego w Z) do rejestru Rd z postinkrementacją

LPM Rd,Z+

Operation:		Comment:
(i)	$R0 \leftarrow (Z)$	Z: Unchanged, R0 implied destination register
(ii)	$Rd \leftarrow (Z)$	Z: Unchanged
(iii)	$Rd \leftarrow (Z) \quad Z \leftarrow Z + 1$	Z: Post incremented
Syntax:		Program Counter:
(i)	LPM	$PC \leftarrow PC + 1$
(ii)	LPM Rd, Z	$PC \leftarrow PC + 1$
(iii)	LPM Rd, Z+	$PC \leftarrow PC + 1$
Operands:		
None, R0 implied		
$0 \leq d \leq 31$		
$0 \leq d \leq 31$		

Załaduj daną z pamięci programu (spod adresu zawartego w Z i RAMPZ) do rejestru R0

ELPM

Operation:		Comment:
(i)	$R0 \leftarrow (RAMPZ:Z)$	RAMPZ:Z: Unchanged, R0 implied destination register
(ii)	$Rd \leftarrow (RAMPZ:Z)$	RAMPZ:Z: Unchanged
(iii)	$Rd \leftarrow (RAMPZ:Z) \quad (RAMPZ:Z) \leftarrow (RAMPZ:Z) + 1$	RAMPZ:Z: Post incremented
Syntax:		Program Counter:
(i)	ELPM	$PC \leftarrow PC + 1$
(ii)	ELPM Rd, Z	$PC \leftarrow PC + 1$
(iii)	ELPM Rd, Z+	$PC \leftarrow PC + 1$
Operands:		
None, R0 implied		
$0 \leq d \leq 31$		
$0 \leq d \leq 31$		

Załaduj daną z pamięci programu (spod adresu zawartego w Z i RAMPZ) do rejestru Rd

ELPM Rd,Z

Operation:

- (i) $R0 \leftarrow (RAMPZ:Z)$
- (ii) $Rd \leftarrow (RAMPZ:Z)$
- (iii) $Rd \leftarrow (RAMPZ:Z) \quad (RAMPZ:Z) \leftarrow (RAMPZ:Z) + 1$

Comment:

RAMPZ:Z: Unchanged, R0 implied destination register
RAMPZ:Z: Unchanged
RAMPZ:Z: Post incremented

Syntax:

- (i) ELPM
- (ii) ELPM Rd, Z
- (iii) ELPM Rd, Z+

Operands:

None, R0 implied
 $0 \leq d \leq 31$
 $0 \leq d \leq 31$

Program Counter:

$PC \leftarrow PC + 1$
 $PC \leftarrow PC + 1$
 $PC \leftarrow PC + 1$

Załaduj daną z pamięci programu (spod adresu zawartego w Z i RAMPZ) do rejestru Rd z postinkrementacją

ELPM Rd,Z+

Operation:		Comment:
(i)	$R0 \leftarrow (RAMPZ:Z)$	RAMPZ:Z: Unchanged, R0 implied destination register
(ii)	$Rd \leftarrow (RAMPZ:Z)$	RAMPZ:Z: Unchanged
(iii)	$Rd \leftarrow (RAMPZ:Z) \quad (RAMPZ:Z) \leftarrow (RAMPZ:Z) + 1$	RAMPZ:Z: Post incremented
Syntax:		Program Counter:
(i)	ELPM	$PC \leftarrow PC + 1$
(ii)	ELPM Rd, Z	$PC \leftarrow PC + 1$
(iii)	ELPM Rd, Z+	$PC \leftarrow PC + 1$

Operands:
None, R0 implied
 $0 \leq d \leq 31$
 $0 \leq d \leq 31$

Wyślij daną 16-to bitową z rejestrów R0,R1 do komórki pamięci programu (o adresie zawartym w Z + RAMPZ)

SPM

Operation:

- (i) $(RAMPZ:Z) \leftarrow \$ffff$
- (ii) $(RAMPZ:Z) \leftarrow R1:R0$
- (iii) $(RAMPZ:Z) \leftarrow R1:R0$
- (iv) $(RAMPZ:Z) \leftarrow TEMP$
- (v) $BLBITS \leftarrow R1:R0$

Comment:

Erase Program memory page
Write Program memory word
Write temporary page buffer
Write temporary page buffer to Program memory
Set Boot Loader Lock bits

Syntax:

(i)-(v) SPM

Operands:

Z+

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0101	1110	1000
------	------	------	------

Wyślij daną 16-to bitową z rejestrów R0,R1 do komórki pamięci programu (o adresie zawartym w Z + RAMPZ) z postinkrementacją

SPM Z+

Operation:

- (i) (RAMPZ:Z) $\leftarrow$ \$ffff
- (ii) (RAMPZ:Z) $\leftarrow$ R1:R0
- (iii) (RAMPZ:Z) $\leftarrow$ R1:R0
- (iv) (RAMPZ:Z) $\leftarrow$ TEMP
- (v) BLBITS $\leftarrow$ R1:R0

Comment:

Erase Program memory page
Write Program memory word
Write temporary page buffer
Write temporary page buffer to Program memory
Set Boot Loader Lock bits

Syntax:

(i)-(v) SPM

Operands:

Z+

Program Counter:

PC $\leftarrow$ PC + 1

16-bit Opcode:

1001	0101	1110	1000
------	------	------	------

Ładowanie danej z urządzenia we-wy o adresie A do rejestru Rd

IN Rd,A

Operation:

(i) $Rd \leftarrow I/O(A)$

Syntax:

(i) IN Rd,A

Operands:

$0 \leq d \leq 31, 0 \leq A \leq 63$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1011	0AA d	d ddd	AAAA
------	---------	---------	------

Załaduj daną z rejestru Rr do urządzenia we-wy o adresie A

OUT A,Rr

Operation:

(i) $I/O(A) \leftarrow Rr$

Syntax:

(i) OUT A,Rr

Operands:

$0 \leq r \leq 31, 0 \leq A \leq 63$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1011	1AAr	rrrr	AAAA
------	------	------	------

Składanie na stosie zawartości rejestru Rr

PUSH Rr

Operation:

(i) $STACK \leftarrow Rr$

Syntax:

(i) **PUSH Rr**

Operands:

$0 \leq r \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

Stack:

$SP \leftarrow SP - 1$

16-bit Opcode:

1001	001d	dddd	1111
------	------	------	------

Ściągnięcie ze stosu danej do rejestru Rd

POP Rd

Operation:

(i) $Rd \leftarrow \text{STACK}$

Syntax:

(i) POP Rd

Operands:

$0 \leq d \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

Stack:

$SP \leftarrow SP + 1$

16-bit Opcode:

1001	000d	d	1111
------	------	---	------

Wymiana zawartości rejestru Rd z zawartością komórki pamięci o adresie zawartym w Z

XCH Z,Rd

Operation:

(i) $(Z) \leftarrow Rd, Rd \leftarrow (Z)$

Syntax:

(i) XCH Z,Rd

Operands:

$0 \leq d \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	001r	rrrr	0100
------	------	------	------

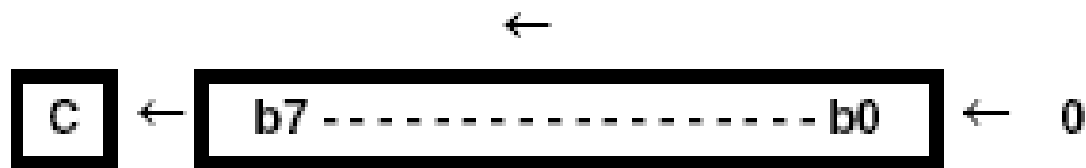
Operacje na bitach i testowanie bitów

Przesunięcie logiczne w lewo

LSL Rd

Operation:

(i)



Syntax:

Operands:

Program Counter:

(i)

LSL Rd

$0 \leq d \leq 31$

$PC \leftarrow PC + 1$

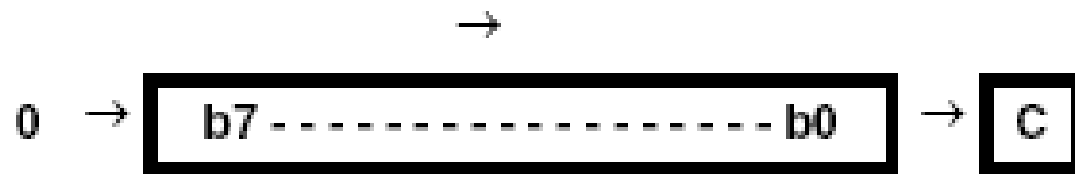
16-bit Opcode: (see ADD Rd,Rd)

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

Przesunięcie logiczne w prawo

LSR Rd

Operation:



Syntax:

Operands:

Program Counter:

(i) LSR Rd

$0 \leq d \leq 31$

$PC \leftarrow PC + 1$

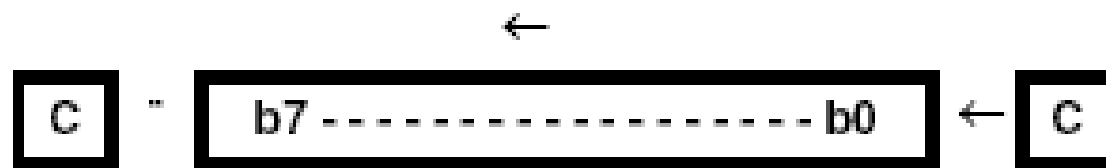
16-bit Opcode:

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

Rotacja w lewo z Carry

ROL Rd

Operation:



Syntax:

Operands:

(i)

ROL Rd

$0 \leq d \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

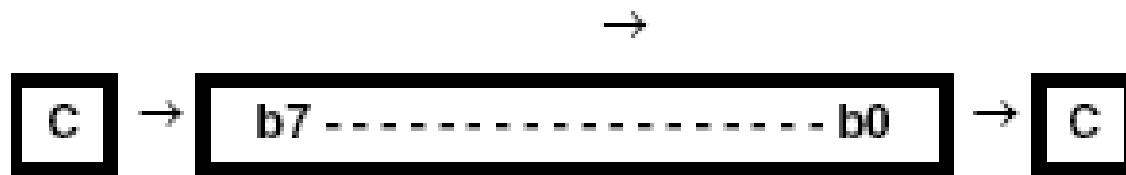
16-bit Opcode: (see ADC Rd,Rd)

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

Rotacja w prawo z Carry

ROR Rd

Operation:



Syntax:

(i) ROR Rd

Operands:

$0 \leq d \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

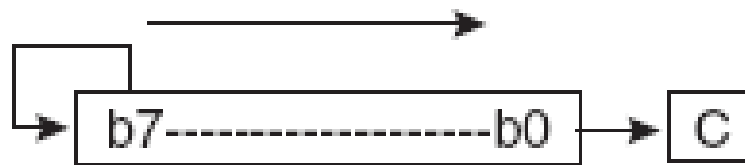
1001	010d	dddd	0111
------	------	------	------

Przesunięcie arytmetyczne w prawo

ASR Rd

Operation:

(i)



Syntax:

(i)

ASR Rd

Operands:

$0 \leq d \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	010d	dddd	0101
------	------	------	------

Zamiana czterech starszych bitów rejestru z czterema starszymi bitami

SWAP

Operation:

(i) $R(7:4) \leftarrow R_d(3:0), R(3:0) \leftarrow R_d(7:4)$

Syntax:

(i) SWAP R_d

Operands:

$0 \leq d \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	010d	dddd	0010
------	------	------	------

Ustaw bit w rejestrze statusowym SREG

BSET s

Operation:

(i) $\text{SREG}(s) \leftarrow 1$

Syntax:

(i) **BSET s**

Operands:

$0 \leq s \leq 7$

Program Counter:

$\text{PC} \leftarrow \text{PC} + 1$

16-bit Opcode:

1001	0100	0sss	1000
------	------	------	------

Zeruj bit w rejestrze statusowym SREG

BCLR s

Operation:

(i) $\text{SREG}(s) \leftarrow 0$

Syntax:

(i) **BCLR s**

Operands:

$0 \leq s \leq 7$

Program Counter:

$\text{PC} \leftarrow \text{PC} + 1$

16-bit Opcode:

1001	0100	1sss	1000
------	------	------	------

Ustawienie bitu b w rejestrze we-wy o adresie A

SBI A,b

Operation:

(i) $I/O(A,b) \leftarrow 1$

Syntax:

(i) SBI A,b

Operands:

$0 \leq A \leq 31, 0 \leq b \leq 7$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	1010	AAAA	Abbb
------	------	------	------

Zeruj bit b w rejestrze we-wy o adresie A

CBI A,b

Operation:

(i) $I/O(A,b) \leftarrow 0$

Syntax:

(i) CBI A,b

Operands:

$0 \leq A \leq 31, 0 \leq b \leq 7$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	1000	AAAA	Abbb
------	------	------	------

Skopiuuj bit b z rejestru Rr do bitu T

BST Rr,b

Operation:

(i) $T \leftarrow R_d(b)$

Syntax:

(i) BST R_d,b

Operands:

$0 \leq d \leq 31, 0 \leq b \leq 7$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1111	101d	dddd	0bbb
------	------	------	------

Skopiuj bit T do bitu b w rejestrze Rd

BLD Rd,b

Operation:

(i) $Rd(b) \leftarrow T$

Syntax:

(i) BLD Rd,b

Operands:

$0 \leq d \leq 31, 0 \leq b \leq 7$

Program Counter:

$PC \leftarrow PC + 1$

16 bit Opcode:

1111	100d	dddd	0bbb
------	------	------	------

Ustawianie bitu Carry

SEC

Operation:

(i) $C \leftarrow 1$

Syntax:

(i) SEC

Operands:

None

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0100	0000	1000
------	------	------	------

Kasowanie bitu Carry

CLC

Operation:

(i) $C \leftarrow 0$

Syntax:

(i) CLC

Operands:

None

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0100	1000	1000
------	------	------	------

Ustawianie bitu N

SEN

Operation:

(i) $N \leftarrow 1$

Syntax:

(i) SEN

Operands:

None

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0100	0010	1000
------	------	------	------

Kasowanie bitu N

CLN

Operation:

(i) $N \leftarrow 0$

Syntax:

(i) CLN

Operands:

None

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0100	1010	1000
------	------	------	------

Ustawianie bitu zera

SEZ

Operation:

(i) $Z \leftarrow 1$

Syntax:

(i) SEZ

Operands:

None

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0100	0001	1000
------	------	------	------

Zerowanie bitu Z

CLZ

Operation:

(i) $Z \leftarrow 0$

Syntax:

(i) CLZ

Operands:

None

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0100	1001	1000
------	------	------	------

Ustawianie bitu globalnej maski przerwań

SEI

Operation:

(i) $I \leftarrow 1$

Syntax:

(i) SEI

Operands:

None

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0100	0111	1000
------	------	------	------

Kasowanie bitu globalnej maski przerwań

CLI

Operation:

(i) $I \leftarrow 0$

Syntax:

(i) CLI

Operands:

None

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0100	1111	1000
------	------	------	------

Ustawianie bitu S

SES

Operation:

(i) $S \leftarrow 1$

Syntax:

(i) SES

Operands:

None

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0100	0100	1000
------	------	------	------

Kasowanie bitu S

CLS

Operation:

(i) $S \leftarrow 0$

Syntax:

(i) CLS

Operands:

None

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0100	1100	1000
------	------	------	------

Ustawianie bitu V

SEV

(i) Operation:
 $V \leftarrow 1$

	Syntax:	Operands:	Program Counter:
(i)	SEV	None	$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0100	0011	1000
------	------	------	------

Zerowanie bitu V

CLV

Operation:

(i) $V \leftarrow 0$

Syntax:

(i) CLV

Operands:

None

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0100	1011	1000
------	------	------	------

Ustawianie bitu T

SET

Operation:

(i) $T \leftarrow 1$

Syntax:

(i) SET

Operands:

None

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0100	0110	1000
------	------	------	------

Kasowanie bitu T

CLT

Operation:

(i) $T \leftarrow 0$

Syntax:

(i) CLT

Operands:

None

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0100	1110	1000
------	------	------	------

Ustawianie bitu H

SEH

Operation:

(i) $H \leftarrow 1$

Syntax:

(i) SEH

Operands:

None

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0100	0101	1000
------	------	------	------

Kasowanie bitu H

CLH

(i) Operation:
 $H \leftarrow 0$

(i) Syntax: CLH Operands: None Program Counter:
 $PC \leftarrow PC + 1$

16-bit Opcode:

1001	0100	1101	1000
------	------	------	------

Operacje sterujące

Normalnie nie używana w programie użytkownika służy do debugowania programu on-chip

BREAK

	Syntax:	Operands:	Program Counter:
(i)	BREAK	None	$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0101	1001	1000
------	------	------	------

Operacja pusta

NOP

Operation:

(i)

No

Syntax:

(i)

NOP

Operands:

None

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0000	0000	0000	0000
------	------	------	------

Wprowadzenie do trybu uśpienia

SLEEP

Syntax:

SLEEP

Operands:

None

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0101	1000	1000
------	------	------	------

Zerowanie Watch Doga

WDR

Operation:

- (i) WD timer restart.

Syntax:

- (i) WDR

Operands:

None

Program Counter:

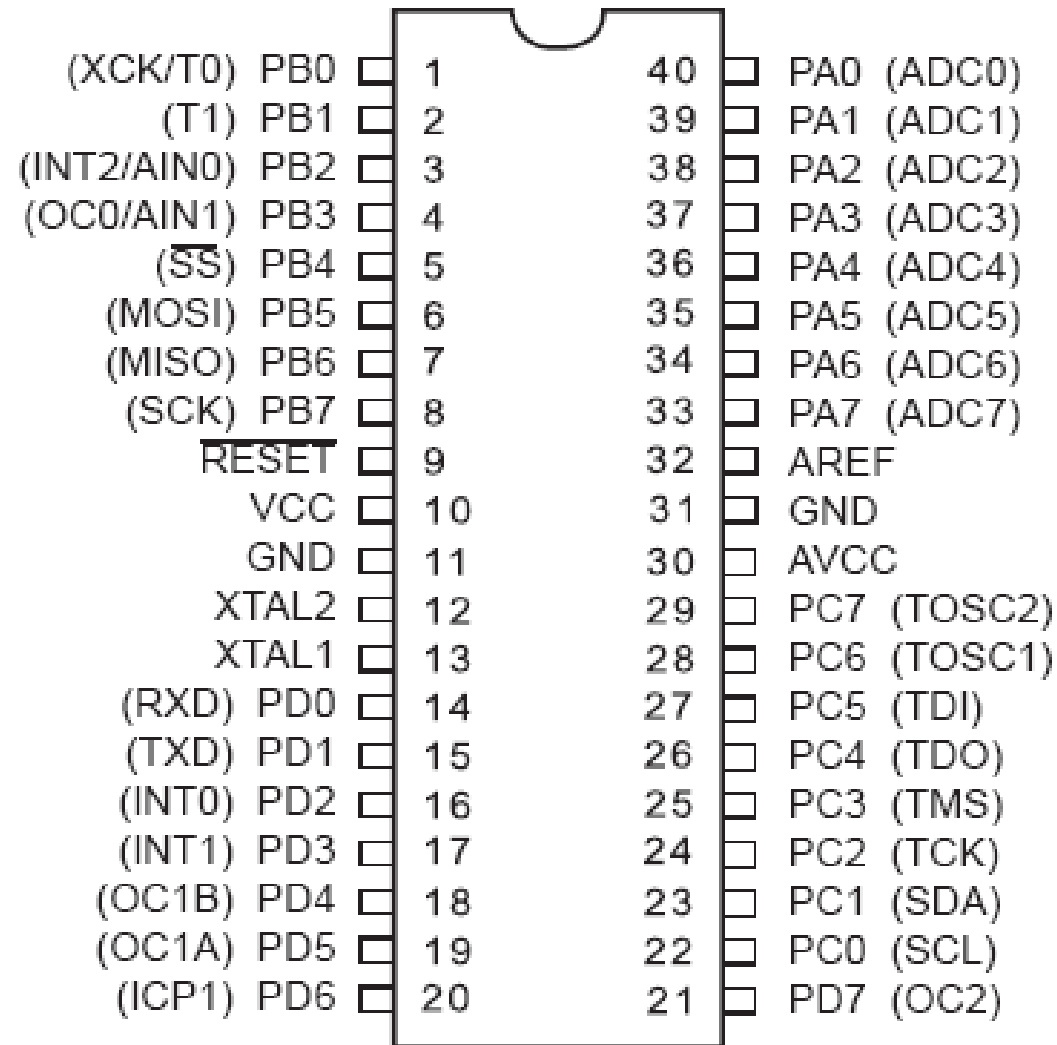
$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0101	1010	1000
------	------	------	------

Wyprowadzenia układu ATmega32

PDIP



VCC: zasilanie

GND: masa

PA0-PA7: port wejścia wyjścia z możliwością dołączania rezystorów podciągających lub wejścia przetwornika analogowo0cyfrowego

PB0-PB7: port wejścia wyjścia z możliwością dołączania rezystorów podciągających lub:

PB0: wejście licznika czasomierza T0 lub wejście taktowania portu szeregowego

PB1: wejście licznika czasomierza T1

PB2: Wejście dodatnie komparatora analogowego lub wejście przerywające INT2

PB3: Wejście ujemne komparatora analogowego lub wyjście Output Compare licznika T0

PB4: SPI wejście Slave Select

PB5: SPI Master Output/Slave Input- MOSI

PB6: SPI Master Input/Slave Output- MISO

PB7: SPI sygnał zegarowy- SCK

PC0-PC7: port wejścia wyjścia z możliwością dołączania rezystorów podciągających lub:

PC0: Sygnał zegarowy magistrali I2C

PC1: Linia danych magistrali I2C

PC2: Zegar złącza JTAG

PC3: Złącze JTAG wybór modu

PC4: Wyjście danych złącza JTAG

PC5: Wejście danych złącza JTAG

PC6: Zewnętrzny oscylator licznika- TOSC1

PC7: Zewnętrzny oscylator licznika- TOSC2

PD0-PD7: port wejścia wyjścia z możliwością dołączania rezystorów podciągających lub:

PD0: RxD- wejście danych asynchronicznego portu szeregowego USART

PD1: TxD- wyjście danych asynchronicznego portu szeregowego USART

PD2: INT0- wejście przerywające 0

PD3: INT1- wejście przerywające 1

PD4: Wyjście A Output Compare licznika-czasomierza T1

PD5: Wyjście B Output Compare licznika-czasomierza T1

PD6: Wejście Input-Capture licznika czasomierza T1

PD7: Wyjście Output Compare licznika-czasomierza T2

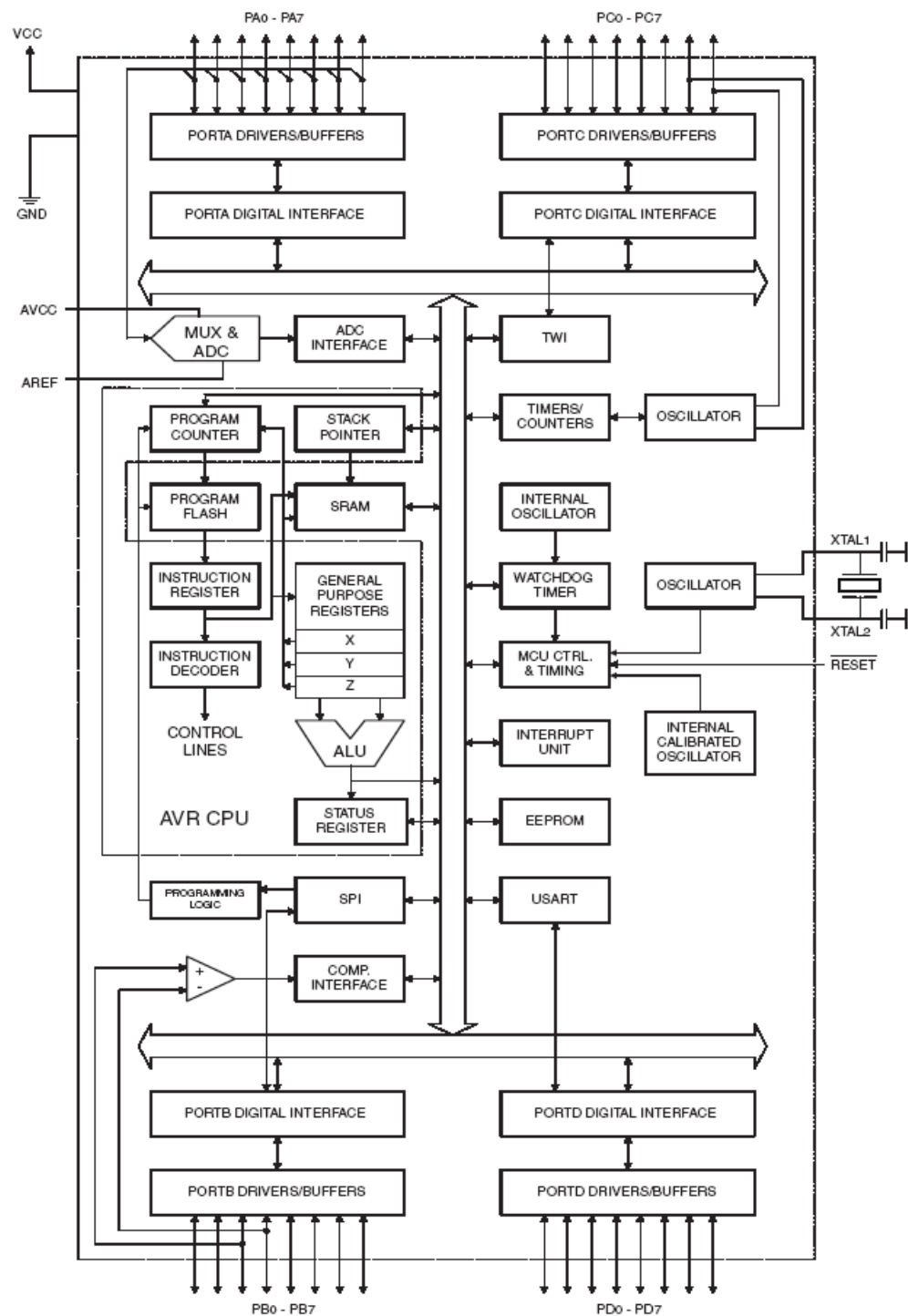
RESET: wejście zerujące

XTAL1: Zewnętrzny oscylator lub wejście zegara zewnętrznego

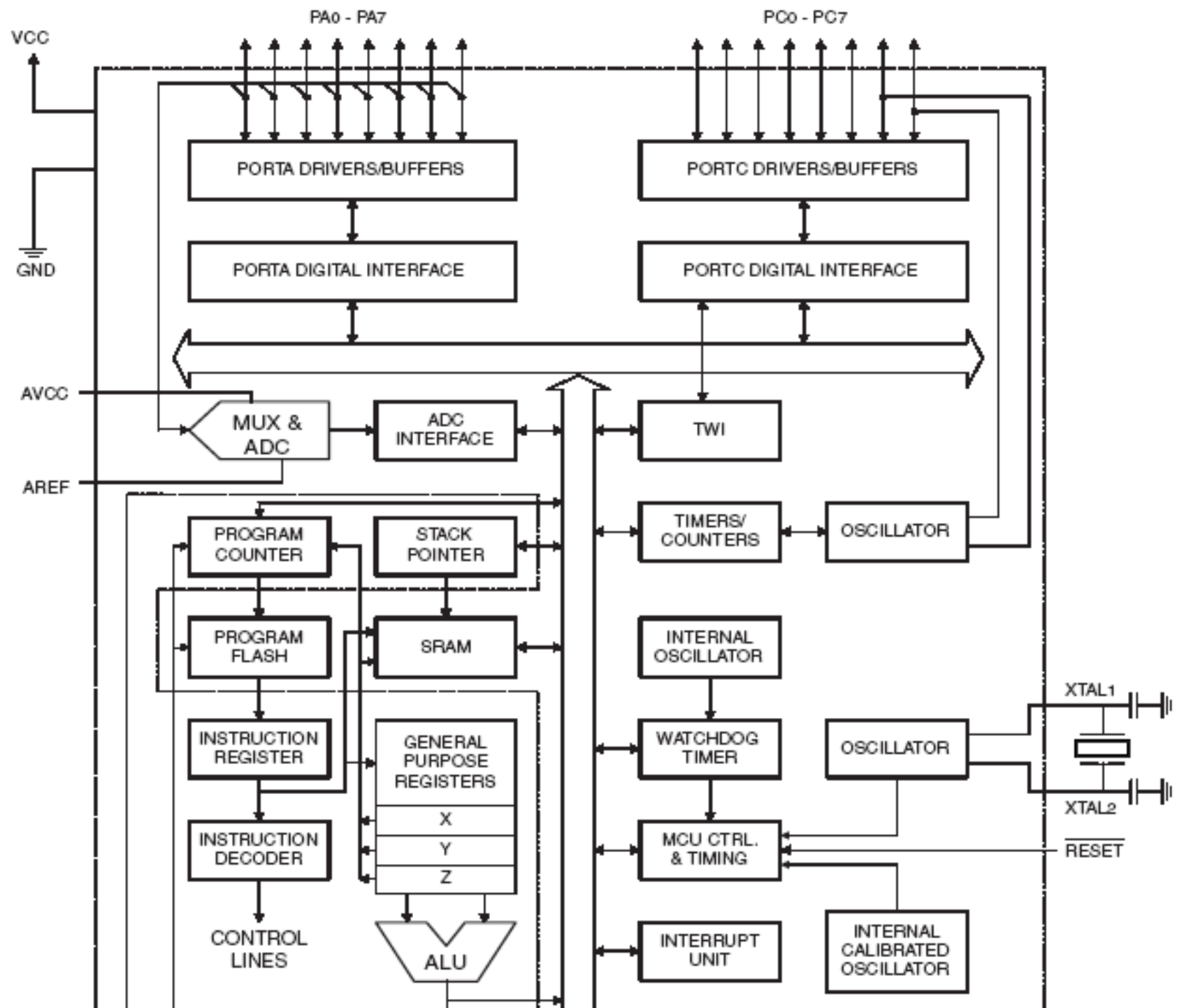
XTAL2: Zewnętrzny oscylator

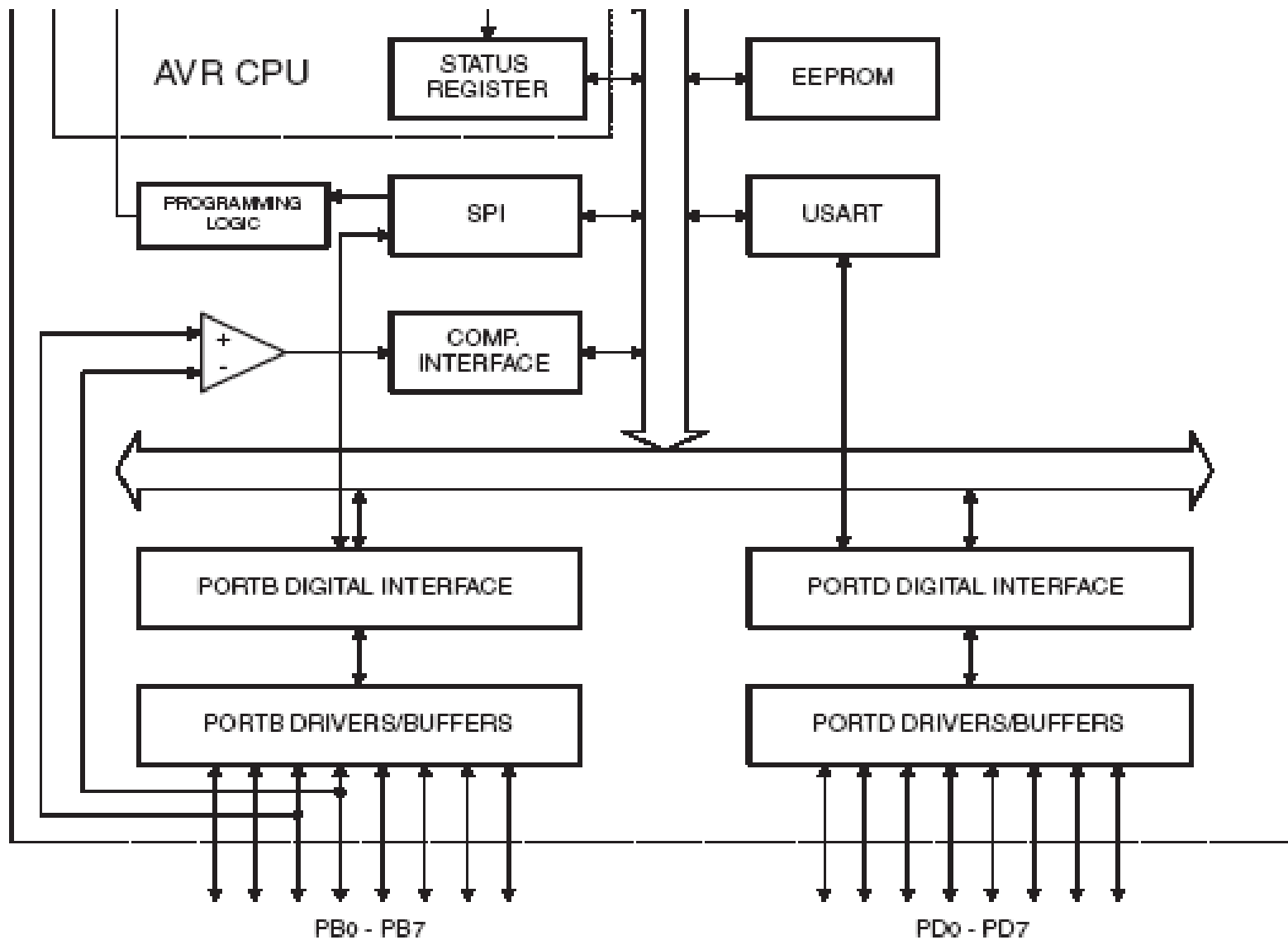
AVCC: wejście zasilania przetwornika AC (gdy nieużywany podłączony do VCC, gdy używany podłączony do VCC przez filtr dolonprzepustowy)

AREF: wejście napięcia referencyjnego dla przetwornika AC



Schemat blokowy





Dostęp do pamięci EEPROM

Dostęp do pamięci EEPROM jest możliwy dzięki odpowiednim rejestrom lokowanym w przestrzeni wejścia-wyjścia.

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	–	–	EEAR9	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/W	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	

Rejestr adresowy EEPROM- EEARH i EEARL

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	

Rejestr danych EEPROM-EEDR

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	EERIE	EEMWE	EEWE	EERE	EECR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	X	0	

Rejestr kontrolny pamięci EEPROM-EECR

BIT3: EERIE- generacja przerwania przy gotowości do zapisu, Gdy ustawiony gotowość pamięci jest równoznaczna ze zgłoszeniem dedykowanego przerwania. Jest ono generowane ciągle dopóki EEW E=„0”

BIT2: EEMWE- zezwolenie na zapis pamięci- musi być ustawiony na cztery cykle zegarowe przed zapisem pamięci EEPROM

BIT1: EEWE- zapis pamięci, jeśli EEMWE=„1” i pamięć nie jest zajęta zapisem poprzednim to ustawienie bitu EEWE powoduje zapis, a jego wyzerowanie informuje o końcu zapisu

BIT0: EERE- odczyt pamięci, jeśli pamięć nie jest zajęta ustawienie bitu powoduje odczyt zaadresowanej komórki do rejestru danych.

Źródła przerwań w Atmega32

nr	Adres obsł.	Źródło	Przyczyna
1	\$000	Reset	Pin, Power on reset, Watch Dog, Brown out reset, JTAG reset
2	\$002	INT0	Sygnał na końcówce INT0
3	\$004	INT1	Sygnał na końcówce INT1
4	\$006	INT2	Sygnał na końcówce INT2
5	\$008	TIMER2 COMP	Licznik T2- funkcja porównania
6	\$00A	TIMER2 OVF	Przepełnienie licznika T2
7	\$00C	TIMER1 CAPT	Licznik T1- funkcja Input Capture
8	\$00E	TIMER1 COMPA	Licznika T1- funkcja porównania A
9	\$010	TIMER1 COMPB	Licznika T1- funkcja porównania B
10	\$012	TIMER1 OVF	Przepełnienie licznika T1

Źródła przerw w ATmega32

nr	Adres obsł.	Źródło	Przyczyna
11	\$014	TIMER0 COMP	Licznik T0- funkcja porównania
12	\$016	TIMER0 OVF	Przepełnienie licznika T0
13	\$018	SPI STC	Transmisja szeregową zakończona
14	\$01A	USART RXC	Dana odebrana przez USART
15	\$01C	USART UDRE	Rejestr danych USART pusty
16	\$01E	USART TXC	Dana wysłana przez USART
17	\$020	ADC	Koniec konwersji AC
18	\$022	EE_RDY	Pamięć EEPROM gotowa
19	\$024	ANA_COMP	Komparator analogowy
20	\$026	TWI	I2C
21	\$028	SPM_RDY	Zapis pamięci programu gotowy

Priorytet przerwania jest zgodny z jego numerem:

Im wyższy numer wektora przerwania tym niższy priorytet.