

Mikrokontrolery AVR ATmega

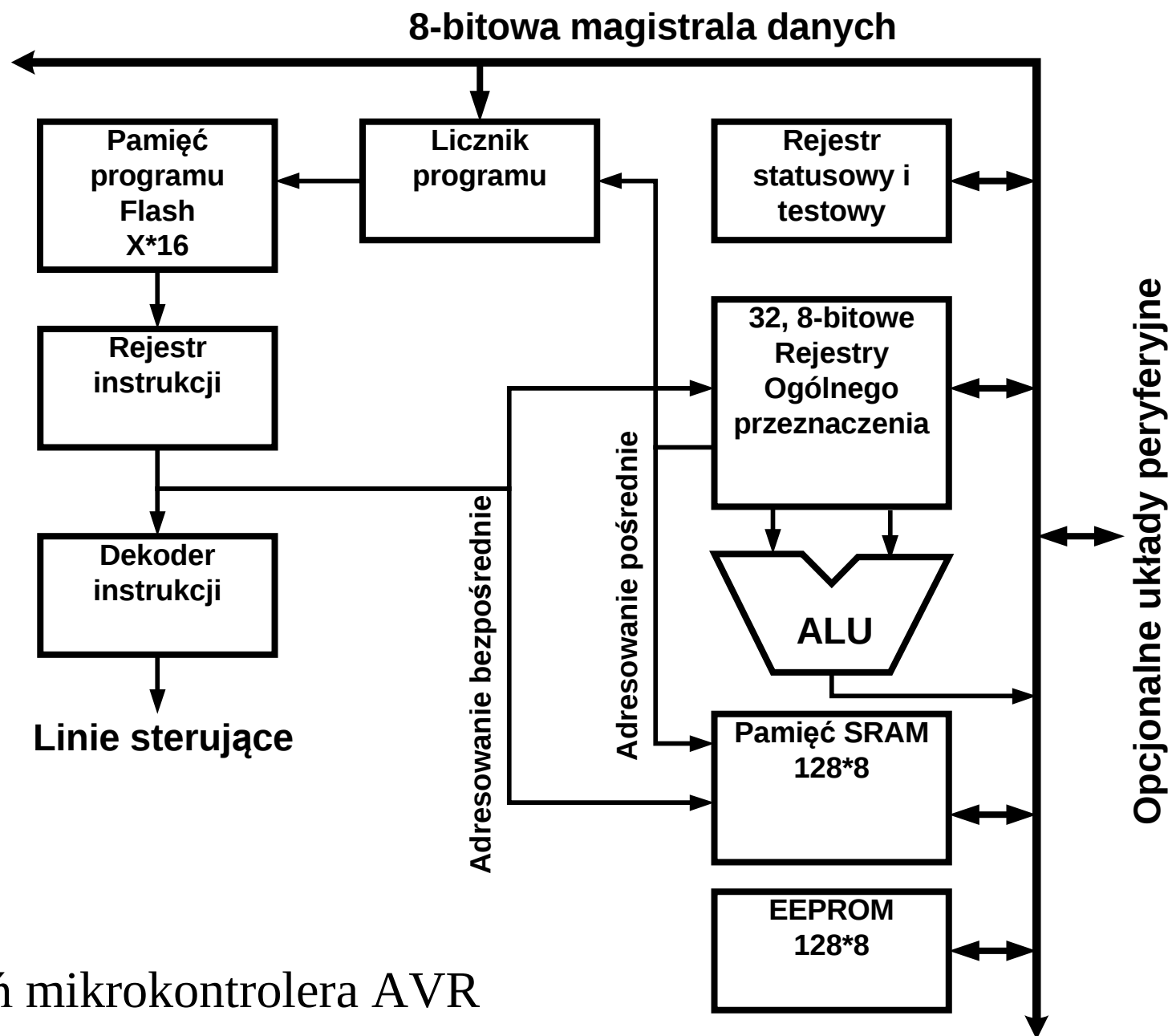
Literatura:

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

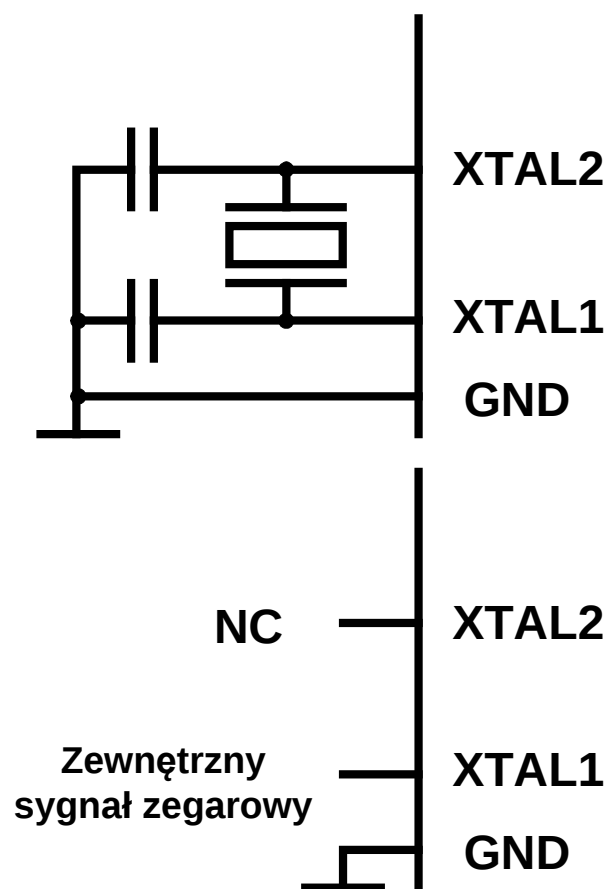
8-bit AVR Instruction Set [www.atmel.com]

Baranowski Rafał, Mikrokontrolery AVR Atmega, BTC, Warszawa 2005

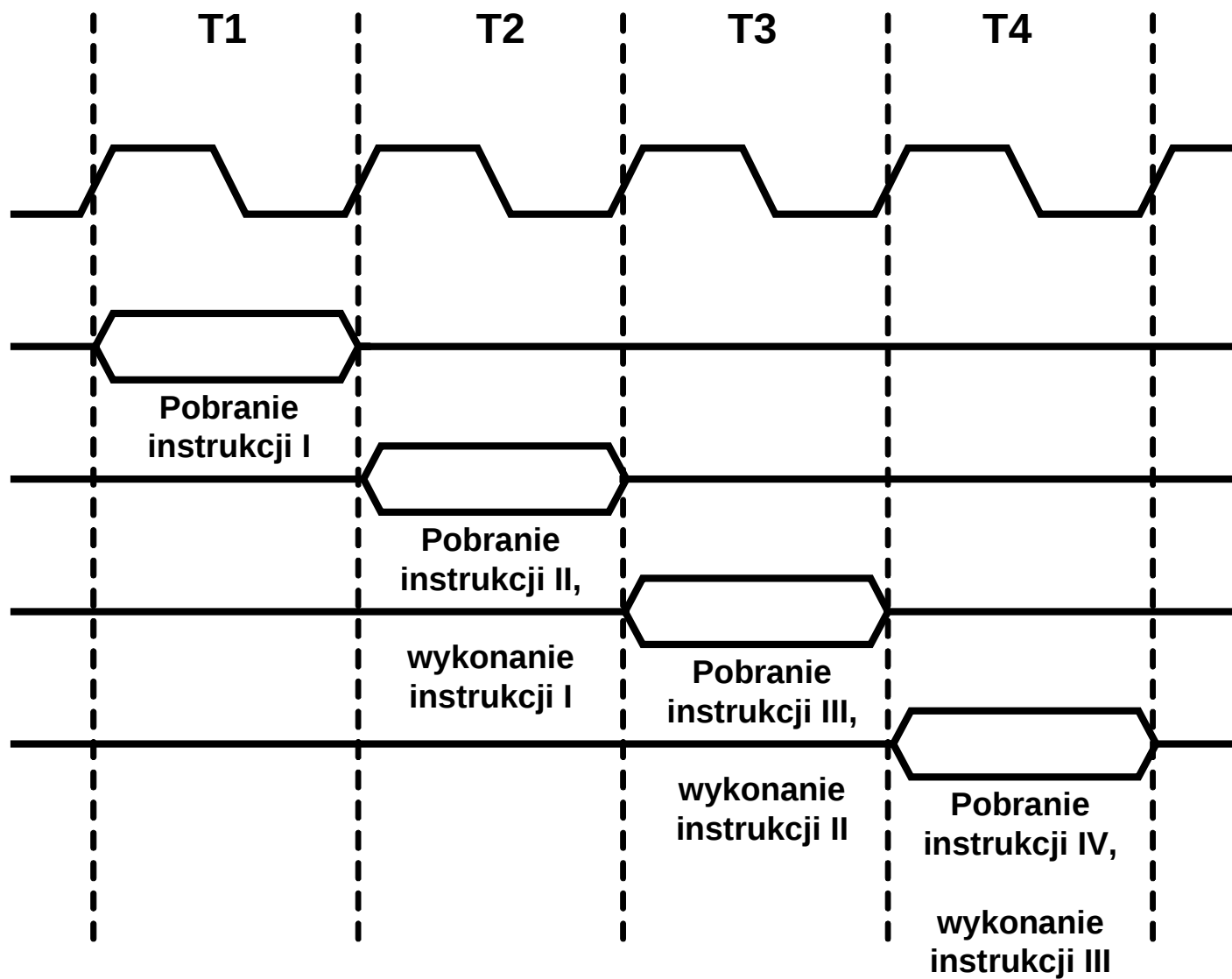




Rdzeń mikrokontrolera AVR

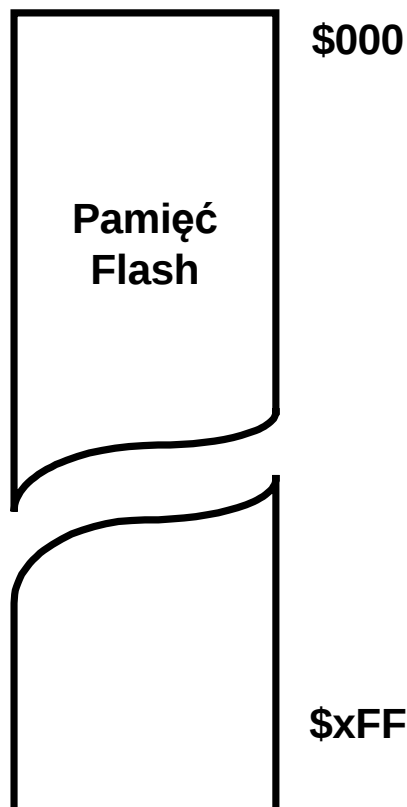


Generator zegarowy

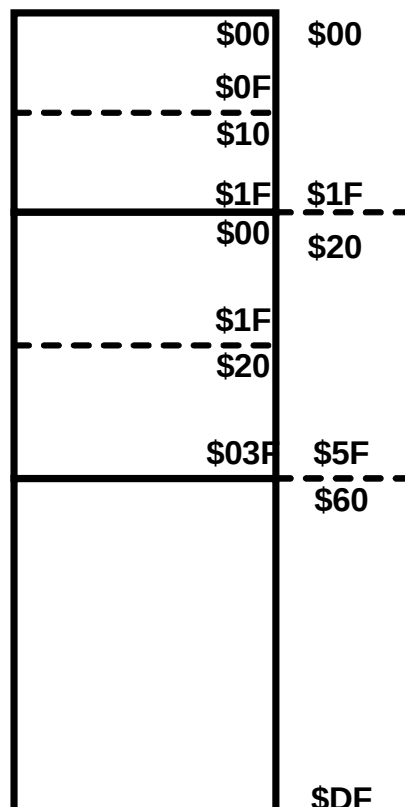


Przetwarzanie potokowe

Pamięć programu



Pamięć SRAM



32 rejestry
ogólnego
przeznaczenia

64 rejestry we/
wy (dostęp przy
pomocy
rozkazów IN i
OUT)

Jeżeli rejestry we/wy są adresowane jako komórki
pamięci SRAM należy do ich adresu dodać \$20

Pamięć EEPROM



Pamięć
dostępna
pośredni przy
pomocy
rejestrów
EEADR, EEDR

Organizacja pamięci mikrokontrolerów AVR

	Adres	
R0	0x0000	Rejestry ogólnego przeznaczenia
R1	0x0001	
R2	0x0002	
R3	0x0003	
R15	0x000F	
R16	0x0010	
R25	0x0019	
R26	0x001A	XL (LSB rejestru X)
R27	0x001B	XH (MSB rejestru X)
R28	0x001C	YL (LSB rejestru Y)
R29	0x001D	YH (MSB rejestru Y)
R30	0x001E	ZL (LSB rejestru Z)
R31	0x001F	ZH (MSB rejestru Z)

7	6	5	4	3	2	1	0
I	T	H	S	V	N	Z	C

SREG

Rejestr statusu- SREG

I- globalne zezwolenie na przerwanie (odmaskowanie *sei* I=„1”, zamaskowanie *cli* I=„0”, automatycznie odtwarzana po *reti*)

T-rejestr bitowy do kopiowania bitów

H- przeniesienie połówkowe

S- bit znaku (jeżeli nie wystąpiło V)

V- przepełnienie przy operacjach w kodzie U2

N- znacznik wartości ujemnej

Z- znacznik zera

C- przeniesienie lub pożyczka

Tryby adresacji

**Adresowanie pamięci danych SRAM (pamięć SRAM,
rejstry robocze, rejstry funkcyjne)**

Adresowanie bezpośrednie pamięci danych

W rozkazie umieszczono dwubajtowy adres komórki pamięci np..

lds R0,0X0060; załaduj do R0 zawartość komórki pamięci o adresie
0060 hex

Adresowanie pośrednie pamięci danych

Adres umieszczono w jednym z trzech 16-to bitowych rejestrów X, Y, Z np.

ldi XH,0

ldi XL, 0x62

st X,R0; zapisz zawartość R0 do komórki pamięci o adresie 0062 hex

Adresowanie pośrednie z postinkrementacją

Po wykonaniu przesłania adres zwarty w 16-to bitowym rejestrze adresowym jest zwiększany o 1 np.

ldi XH,0

ldi XL,0x60; X=0x0060

ld R0,X+; zapis do rejestru R0 komórki o adresie 0060 hex

ld r1,X; zapis do rejestru R1 komórki o adresie 0061 hex

Adresowanie pośrednie z predekrementacją

Przed wykonaniem przesłania adres zwarty w 16-to bitowym rejestrze adresowym jest zmniejszany o 1 np.

ldi XH,0

ldi XL,0x63; X=0x0063

ld R0,-X; zapis do rejestru R0 komórki o adresie 0062 hex

ld r1,-X; zapis do rejestru R1 komórki o adresie 0061 hex

Adresowanie pośrednie z przemieszczeniem

Adres jest sumą adresu bazowego zwanego w jednym z rejestrów 16-to bitowych i przesunięcia z zakresu 0-63 np.

ldi ZH,0

ldi ZL,0x60; Z=0x060

std Z+5,R0; zapisz zawartość rejestru R0 do komórki o adresie 0065 hex

Adresowanie rejestrów roboczych

Bezpośrednie adresowanie rejestrów roboczych

Kod wybranej instrukcji zawiera adres rejestru np.

clr R1; zerowanie zawartości rejestru R1

Bezpośrednie adresowanie dwóch rejestrów roboczych

W kodzie rozkazu znajdują się adresy dwóch rejestrów np.

mov R1,R0; przesłanie zawartości rejestru R1 do R0

Adresowanie pośrednie rejestru roboczego

Adres rejestru znajduje się w 16-to bitowym rejestrze adresowym np.

ldi XH,0x00

ldi XL,0x00; adres rejestru R0 (0x0000) w X

ld R1,X

Adresowanie przestrzeni wejścia-wyjścia

Bezpośrednie adresowanie przestrzeni wejścia wyjścia jako osobnej przestrzeni adresowej

Za kodem instrukcji znajduje się adres urządzenia we-wy np.

in R1,SREG; załaduj R1 zawartością rejestru statusowego o adresie 0x3F

Bezpośrednie adresowanie przestrzeni wejścia wyjścia jako części przestrzeni adresowej pamięci danych

Za kodem rozkazu podany jest adres rejestru we-wy w przestrzenie adresowej SRAM np.

Lds R1,SREG+0x20; załaduj R1 zawartością komórki pamięci o adresie 0x3F+0x20 czyli rejestru SREG

Adresowanie pamięci programu- odczyt stałych

Tylko adresowanie pośrednie z wykorzystaniem rejestru Z np.

Stala: ,dw 0x1234 ; deklaracja stałej 2 bajtowej

ldi ZH, high(Stala*2+1)

ldi ZL, high(Stala*2+1)

lpm R0,Z; pobranie starszego bajtu stałej spod adresu Stala do R0

Możliwość adresacji z postinkrementacją

Adresowanie skoków w pamięci programu

Względne adresowanie pamięci programu

rjpm **dalej;** skok bez śladu

.

dalej:

Do zawartości PC dodawane jest przesunięcie od –2048 do +2047

Bezpośrednie adresowanie pamięci programu

call wykonanie; wywołanie podprogramu

.

wykonanie:

Pełny, 16-to bitowy adres skoku

Pośrednie adresowanie pamięci programu

Wykorzystuje rejestr Z w roli wskaźnika np.

ldi ZH,high(wykonaj); załadowanie starszej części adresu procedury

ldi ZL,low(wykonaj); załadowanie młodszej części adresu procedury

Icall; wywołanie procedury

•

wykonaj:

•

ret

Rejestry funkcyjne

Wskaźnik stosu:

SPH (MSB)

SPL (LSB)

Typowa inicjalizacja stosu:

ldi R15,high(RAM_koniec)

out SPH,R15

ldi R15,low(RAM_koniec)

out SPL,R15

Rejestry GPIOR- rejestry ogólnego przeznaczenia zwykle trzy GPIOR0-GPIOR3. Dostępne również bitowo.

Rejestr RAMPZ, zaimplementowany jest tylko jeden rejestr RAMPZ, niezbędny do pośredniego adresowania pamięci danych w mikrokontrolerach o pamięci programu większej niż 64kB. Przy pomocy najmłodszego bitu rejestru RAMPZ można wybrać stronę pamięci odczytywaną rozkazem **elpm**

Lista rozkazów

Instrukcje arytmetyczne i logiczne

Dodawanie bez CARRY

ADD Rd,Rr

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	rrrr
------	------	------	------

Dodawanie z CARRY

ADC Rd,Rr

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	rrrr
------	------	------	------

Dodawanie danej natychmiastowej do słowa

ADIW Rd,K

Operation:

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

Syntax:

Operands:

Program Counter:

- (i) ADIW Rd+1:Rd,K $d \in \{24,26,28,30\}, 0 \leq K \leq 63$

$PC \leftarrow PC + 1$

16-bit Opcode:

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

Odejmowanie bez CARRY

SUB Rd,Rr

Operation:

(i) $Rd \leftarrow Rd - Rr$

Syntax:

(i) SUB Rd,Rr

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0001	10rd	dddd	rrrr
------	------	------	------

Odejmowanie danej natychmiastowej

SUBI Rd,K

Operation:

(i) $Rd \leftarrow Rd - K$

Syntax:

(i) SUBI Rd,K

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0101	KKKK	dddd	KKKK
------	------	------	------

Odejmowanie z CARRY

SBC Rd,Rr

Operation:

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

Syntax:

(i) SBC Rd,Rr

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0000	10rd	dddd	rrrr
------	------	------	------

Odejmowanie danej natychmiastowej z CARRY

SBCI Rd,K

Operation:

(i) $Rd \leftarrow Rd - K - C$

Syntax:

(i) SBCI Rd,K

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0100	KKKK	dddd	KKKK
------	------	------	------

Odejmowanie danej natychmiastowej od słowa

SBIW Rd,K

Operation:

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

Syntax:

Operands:

Program Counter:

- (i) SBIW Rd+1:Rd,K $d \in \{24,26,28,30\}, 0 \leq K \leq 63$

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	0111	KKdd	KKKK
------	------	------	------

Mnożenie logiczne

AND Rd,Rr

Operation:

(i) $Rd \leftarrow Rd \bullet Rr$

Syntax:

(i) AND Rd,Rr

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0010	00rd	dddd	rrrr
------	------	------	------

Mnożenie logiczne z daną natychmiastową

ANDI Rd,K

Operation:

(i) $Rd \leftarrow Rd \bullet K$

Syntax:

(i) ANDI Rd,K

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0111	KKKK	dddd	KKKK
------	------	------	------

Suma logiczna

OR Rd,Rr

Operation:

(i) $Rd \leftarrow Rd \vee Rr$

Syntax:

(i) OR Rd,Rr

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0010	10rd	dddd	rrrr
------	------	------	------

Suma logiczna z daną natychmiastową

ORI Rd,K

Operation:

(i) $Rd \leftarrow Rd \vee K$

Syntax:

(i) ORI Rd,K

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0110	KKKK	dddd	KKKK
------	------	------	------

Exclusiv OR

EOR Rd,Rr

Operation:

(i) $Rd \leftarrow Rd \oplus Rr$

Syntax:

(i) EOR Rd,Rr

Operands:

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

Program Counter:

$$PC \leftarrow PC + 1$$

16-bit Opcode:

0010	01rd	dddd	rrrr
------	------	------	------

Complement

COM Rd

Operation:

(i) $Rd \leftarrow \$FF - Rd$

Syntax:

(i) COM Rd

Operands:

$0 \leq d \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	010d	dddd	0000
------	------	------	------

Negacja

NEG Rd

Operation:

(i) $Rd \leftarrow \$00 - Rd$

Syntax:

(i) NEG Rd

Operands:

$0 \leq d \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	010d	dddd	0001
------	------	------	------

Ustawienie bitów w rejestrze

SBR Rd,K

Operation:

(i) $Rd \leftarrow Rd \vee K$

Syntax:

(i) SBR Rd,K

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0110	KKKK	dddd	KKKK
------	------	------	------

Zerowanie bitów w rejestrze

CBR Rd,K

Operation:

(i) $Rd \leftarrow Rd \bullet (\$FF - K)$

Syntax:

(i) CBR Rd,K

Operands:

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

Program Counter:

$$PC \leftarrow PC + 1$$

16-bit Opcode: (see ANDI with K complemented)

Zwiększ zawartość rejestru o 1

INC Rd

Operation:

(i) $Rd \leftarrow Rd + 1$

Syntax:

(i) INC Rd

Operands:

$0 \leq d \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	010d	dddd	0011
------	------	------	------

Zmniejsz zawartość rejestru o 1

DEC Rd

Operation:

(i) $Rd \leftarrow Rd - 1$

Syntax:

(i) DEC Rd

Operands:

$0 \leq d \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	010d	dddd	1010
------	------	------	------

Testuj na zero lub minus

TST Rd

Operation:

(i) $Rd \leftarrow Rd \bullet Rd$

Syntax:

(i) TST Rd

Operands:

$0 \leq d \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

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

0010	00dd	dddd	dddd
------	------	------	------

Zeruj rejestr

CLR Rd

Operation:

(i) $Rd \leftarrow Rd \oplus Rd$

Syntax:

(i) CLR Rd

Operands:

$$0 \leq d \leq 31$$

Program Counter:

$$PC \leftarrow PC + 1$$

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

0010	01dd	dddd	dddd
------	------	------	------

Wpisz do rejestru 0xFF

SER Rd

Operation:

(i) $R_d \leftarrow \$FF$

Syntax:

(i) SER Rd

Operands:

$16 \leq d \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1110	1111	dddd	1111
------	------	------	------

Množenie bez znaku

MUL Rd,Rr

Operation:

- (i) $R1:R0 \leftarrow R_d \times R_r$ (unsigned $\leftarrow$ unsigned $\times$ unsigned)

Syntax:

- (i) MUL Rd,Rr

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

1001	11rd	dddd	rrrr
------	------	------	------

Mnożenie liczb ze znakiem

MULS Rd,Rr

Operation:

- (i) $R1:R0 \leftarrow R_d \times R_r$ (signed $\leftarrow$ signed $\times$ signed)

Syntax:

- (i) MULS Rd,Rr

Operands:

$16 \leq d \leq 31, 16 \leq r \leq 31$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0000	0010	dddd	rrrr
------	------	------	------

Mnożenie liczby ze znakiem i bez znaku

MULSU Rd,Rr

Operation:

- (i) $R1:R0 \leftarrow R_d \times R_r$ (signed $\leftarrow$ signed $\times$ unsigned)

Syntax:

- (i) MULSU Rd,Rr

Operands:

$16 \leq d \leq 23, 16 \leq r \leq 23$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0000	0011	0ddd	0rrr
------	------	------	------

Mnożenie ułamkowe bez znaku

FMUL Rd,Rr

Operation:

- (i) $R1:R0 \leftarrow R_d \times R_r$ (unsigned (1.15) $\leftarrow$ unsigned (1.7) $\times$ unsigned (1.7))

Syntax:

- (i) FMUL Rd,Rr

Operands:

- $16 \leq d \leq 23, 16 \leq r \leq 23$

Program Counter:

- $PC \leftarrow PC + 1$

16-bit Opcode:

0000	0011	0ddd	1rrr
------	------	------	------

Mnożenie ułamkowe ze znakiem

FMULS Rd,Rr

Operation:

- (i) $R1:R0 \leftarrow R_d \times R_r$ (signed (1.15) $\leftarrow$ signed (1.7) $\times$ signed (1.7))

Syntax:

- (i) FMULS Rd,Rr

Operands:

$16 \leq d \leq 23, 16 \leq r \leq 23$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0000	0011	1ddd	0rrr
------	------	------	------

Mnożenie ułamkowe ze znakiem i bez znaku

FMULSU Rd,Rr

Operation:

- (i) $R1:R0 \leftarrow R_d \times R_r$ (signed (1.15) $\leftarrow$ signed (1.7) $\times$ unsigned (1.7))

Syntax:

- (i) FMULSU Rd,Rr

Operands:

$16 \leq d \leq 23, 16 \leq r \leq 23$

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0000	0011	1ddd	1rrr
------	------	------	------

Instrukcje skoków

Skok względny bezwarunkowy

RJMP k

Operation:

(i) $PC \leftarrow PC + k + 1$

Syntax:

(i) RJMP k

Operands:

$-2K \leq k < 2K$

Program Counter:

$PC \leftarrow PC + k + 1$

Stack

Unchanged

16-bit Opcode:

1100	kkkk	kkkk	kkkk
------	------	------	------

Skok bezpośredni pod adres zawarty w Z

IJMP

Operation:

- (i) $PC \leftarrow Z(15:0)$ Devices with 16 bits PC, 128K bytes Program memory maximum.
- (ii) $PC(15:0) \leftarrow Z(15:0)$ Devices with 22 bits PC, 8M bytes Program memory maximum.
 $PC(21:16) \leftarrow 0$

Syntax:

(i),(ii)

IJMP

Operands:

None

Program Counter:

See Operation

Stack:

Not Affected

16-bit Opcode:

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

Rozszerzony skok bezpośredni pod adres zawarty w Z

EI JMP

Operation:

- (i) $PC(15:0) \leftarrow Z(15:0)$
 $PC(21:16) \leftarrow EIND$

Syntax:

Operands:

Program Counter:

Stack:

- (i) EI JMP None See Operation Not Affected

16-bit Opcode:

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

Skok bezpośredni

JMP k

Operation:

(i) $PC \leftarrow k$

Syntax:

(i) JMP k

Operands:

$0 \leq k < 4M$

Program Counter:

$PC \leftarrow k$

Stack:

Unchanged

32-bit Opcode:

1001	010k	kkkk	110k
kkkk	kkkk	kkkk	kkkk

Wywołanie procedury z adresacją względną

RCALL k

Operation:

- (i) $PC \leftarrow PC + k + 1$ Devices with 16 bits PC, 128K bytes Program memory maximum.
- (ii) $PC \leftarrow PC + k + 1$ Devices with 22 bits PC, 8M bytes Program memory maximum.

Syntax:

- (i) RCALL k

Operands:

$$-2K \leq k < 2K$$

Program Counter:

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

Stack:

$$STACK \leftarrow PC + 1$$

$$SP \leftarrow SP - 2 \text{ (2 bytes, 16 bits)}$$

- (ii) RCALL k

$$-2K \leq k < 2K$$

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

$$STACK \leftarrow PC + 1$$

$$SP \leftarrow SP - 3 \text{ (3 bytes, 22 bits)}$$

16-bit Opcode:

1101	k k k k	k k k k	k k k k
------	---------	---------	---------

Wywołanie procedury z adresacją bezpośrednią zawartością Z

ICALL

Operation:

- (i) $PC(15:0) \leftarrow Z(15:0)$ Devices with 16 bits PC, 128K bytes Program memory maximum.
- (ii) $PC(15:0) \leftarrow Z(15:0)$ Devices with 22 bits PC, 8M bytes Program memory maximum.
 $PC(21:16) \leftarrow 0$

	Syntax:	Operands:	Program Counter:	Stack:
(i)	ICALL	None	See Operation	$STACK \leftarrow PC + 1$ $SP \leftarrow SP - 2$ (2 bytes, 16 bits)
(ii)	ICALL	None	See Operation	$STACK \leftarrow PC + 1$ $SP \leftarrow SP - 3$ (3 bytes, 22 bits)

Wywołanie procedury z adresacją rozszerzoną bezpośrednią zawartością Z

EICALL

Operation:

- (i) $PC(15:0) \leftarrow Z(15:0)$
 $PC(21:16) \leftarrow EIND$

Syntax:

- (i) EICALL

Operands:

None

Program Counter:

See Operation

Stack:

$STACK \leftarrow PC + 1$
 $SP \leftarrow SP - 3$ (3 bytes, 22 bits)

16-bit Opcode:

1001	0101	0001	1001
------	------	------	------

Wywołanie procedury z adresem bezpośrednim

CALL k

Operation:

- (i) $PC \leftarrow k$ Devices with 16 bits PC, 128K bytes Program memory maximum.
- (ii) $PC \leftarrow k$ Devices with 22 bits PC, 8M bytes Program memory maximum.

Syntax:

Operands:

Program Counter

Stack:

- (i) CALL k $0 \leq k < 64K$ $PC \leftarrow k$ $STACK \leftarrow PC+2$
SP $\leftarrow$ SP-2, (2 bytes, 16 bits)
- (ii) CALL k $0 \leq k < 4M$ $PC \leftarrow k$ $STACK \leftarrow PC+2$
SP $\leftarrow$ SP-3 (3 bytes, 22 bits)

32-bit Opcode:

1001	010k	kkkk	111k
kkkk	kkkk	kkkk	kkkk

Powrót z procedury

RET

Operation:

- (i) $PC(15:0) \leftarrow \text{STACKDevices with 16 bits PC, 128K bytes Program memory maximum.}$
- (ii) $PC(21:0) \leftarrow \text{STACKDevices with 22 bits PC, 8M bytes Program memory maximum.}$

	Syntax:	Operands:	Program Counter:	Stack:
(i)	RET	None	See Operation	$SP \leftarrow SP + 2$, (2bytes, 16 bits)
(ii)	RET	None	See Operation	$SP \leftarrow SP + 3$, (3bytes, 22 bits)

16-bit Opcode:

1001	0101	0000	1000
------	------	------	------

Powrót z przerwania

RETI

Operation:

- (i) $PC(15:0) \leftarrow STACKDevices$ with 16 bits PC, 128K bytes Program memory maximum.
- (ii) $PC(21:0) \leftarrow STACKDevices$ with 22 bits PC, 8M bytes Program memory maximum.

	Syntax:	Operands:	Program Counter:	Stack
(i)	RETI	None	See Operation	$SP \leftarrow SP + 2$ (2 bytes, 16 bits)
(ii)	RETI	None	See Operation	$SP \leftarrow SP + 3$ (3 bytes, 22 bits)

16-bit Opcode:

1001	0101	0001	1000
------	------	------	------

Porównaj i pomiń jeśli równy

CPSE Rd,Rr

Operation:

- (i) If $Rd = Rr$ then $PC \leftarrow PC + 2$ (or 3) else $PC \leftarrow PC + 1$

Syntax:

- (i) CPSE Rd,Rr

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$, Condition false - no skip
 $PC \leftarrow PC + 2$, Skip a one word instruction
 $PC \leftarrow PC + 3$, Skip a two word instruction

16-bit Opcode:

0001	00rd	dddd	rrrr
------	------	------	------

Porównaj

CP Rd,Rr

(i) Operation:
Rd - Rr

(i) Syntax: CP Rd,Rr
Operands: $0 \leq d \leq 31, 0 \leq r \leq 31$

Program Counter:
 $PC \leftarrow PC + 1$

16-bit Opcode:

0001	01rd	dddd	rrrr
------	------	------	------

Porównaj z CARRY

CPC Rd,Rr

Operation:

(i) $Rd - Rr - C$

Syntax:

(i) CPC Rd,Rr

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0000	01rd	dddd	rrrr
------	------	------	------

Porównaj z daną natychmiastową

CPI Rd,K

Operation:

(i) Rd - K

Syntax:

(i) CPI Rd,K

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$

16-bit Opcode:

0011	KKKK	dddd	KKKK
------	------	------	------

Pomiń jeśli bit w rejestrze wyzerowany

SBRC Rd,b

Operation:

- (i) If $Rr(b) = 0$ then $PC \leftarrow PC + 2$ (or 3) else $PC \leftarrow PC + 1$

Syntax:

- (i) SBRC Rr,b

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$, Condition false - no skip

$PC \leftarrow PC + 2$, Skip a one word instruction

$PC \leftarrow PC + 3$, Skip a two word instruction

16-bit Opcode:

1111	110r	rrrr	0bbb
------	------	------	------

Pomiń jeśli bit w rejestrze ustawiony

SBRS Rd,b

Operation:

- (i) If $Rr(b) = 1$ then $PC \leftarrow PC + 2$ (or 3) else $PC \leftarrow PC + 1$

Syntax:

- (i) SBRS Rr,b

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$, Condition false - no skip

$PC \leftarrow PC + 2$, Skip a one word instruction

$PC \leftarrow PC + 3$, Skip a two word instruction

16-bit Opcode:

1111	111r	rrrr	0bbb
------	------	------	------

Pomiń jeśli bit w rejestrze we-wy wyzerowany

SBIC A,b

Operation:

- (i) If $I/O(A,b) = 0$ then $PC \leftarrow PC + 2$ (or 3) else $PC \leftarrow PC + 1$

Syntax:

- (i) SBIC A,b

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$, Condition false - no skip

$PC \leftarrow PC + 2$, Skip a one word instruction

$PC \leftarrow PC + 3$, Skip a two word instruction

16-bit Opcode:

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

Pomiń jeśli bit w rejestrze we-wy ustawiony

SBIS A,b

Operation:

- (i) If $I/O(A,b) = 1$ then $PC \leftarrow PC + 2$ (or 3) else $PC \leftarrow PC + 1$

Syntax:

- (i) SBIS A,b

Operands:

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

Program Counter:

$PC \leftarrow PC + 1$, Condition false - no skip

$PC \leftarrow PC + 2$, Skip a one word instruction

$PC \leftarrow PC + 3$, Skip a two word instruction

16-bit Opcode:

1001	1011	AAAA	Abbb
------	------	------	------

Skocz jeśli bit w rejestrze statusowym ustawiony

BRBS s,k

Operation:

- (i) If $SREG(s) = 1$ then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRBS s,k

Operands:

$$0 \leq s \leq 7, -64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	00kk	kkkk	ksss
------	------	------	------

Skocz jeśli bit w rejestrze statusowym wyzerowany

BRBC s,k

Operation:

- (i) If $SREG(s) = 0$ then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRBC s,k

Operands:

$$0 \leq s \leq 7, -64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	01kk	kkkk	ksss
------	------	------	------

Skocz jeśli równy

BREQ k

Operation:

- (i) If $R_d = R_r$ ($Z = 1$) then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BREQ k

Operands:

$$-64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	00kk	kkkk	k001
------	------	------	------

Skocz jeśli różny

BRNE k

Operation:

- (i) If $Rd \neq Rr$ ($Z = 0$) then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRNE k

Operands:

$-64 \leq k \leq +63$

Program Counter:

$PC \leftarrow PC + k + 1$

$PC \leftarrow PC + 1$, if condition is false

16-bit Opcode:

1111	01kk	kkkk	k001
------	------	------	------

Skocz jeśli bit CARRY ustawiony

BRCS k

Operation:

- (i) If $C = 1$ then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRCS k

Operands:

$$-64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	00kk	kkkk	k000
------	------	------	------

Skocz jeśli bit CARRY skasowany

BRCC k

Operation:

- (i) If $C = 0$ then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRCC k

Operands:

$$-64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	01kk	kkkk	k000
------	------	------	------

Skocz jeśli większy lub równy

BRSK k

Operation:

- (i) If $R_d \geq R_r$ ($C = 0$) then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRSH k

Operands:

$$-64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	01kk	kkkk	k000
------	------	------	------

Skocz jeśli mniejszy

BRLO k

Operation:

- (i) If $R_d < R_r$ ($C = 1$) then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRLO k

Operands:

$$-64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	00kk	kkkk	k000
------	------	------	------

Skocz jeśli minus

BRMI k

Operation:

- (i) If $N = 1$ then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRMI k

Operands:

$$-64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	00kk	kkkk	k010
------	------	------	------

Skocz jeśli plus

BRPL k

Operation:

- (i) If $N = 0$ then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRPL k

Operands:

$$-64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	01kk	kkkk	k010
------	------	------	------

Skocz jeśli większy lub równy (ze znakiem)

BRGE k

Operation:

- (i) If $R_d \geq R_r$ ($N \oplus V = 0$) then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRGE k

Operands:

$$-64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	01kk	kkkk	k100
------	------	------	------

Skocz jeśli mniejszy niż (ze znakiem)

BRLT k

Operation:

- (i) If $R_d < R_r$ ($N \oplus V = 1$) then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRLT k

Operands:

$$-64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	00kk	kkkk	k100
------	------	------	------

Skocz jeśli Half Carry jest ustawiony

BRHS k

Operation:

- (i) If $H = 1$ then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRHS k

Operands:

$$-64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	00kk	kkkk	k101
------	------	------	------

Skocz jeśli Half Carry jest wyzerowany

BRHC k

Operation:

- (i) If $H = 0$ then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRHC k

Operands:

$$-64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	01kk	kkkk	k101
------	------	------	------

Skocz jeśli bit T wyzerowany

BRTC k

Operation:

- (i) If $T = 0$ then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRTC k

Operands:

$$-64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	01kk	kkkk	k110
------	------	------	------

Skocz jeśli bit T Ustawiony

BRTS k

Operation:

- (i) If $T = 1$ then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRTS k

Operands:

$$-64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	00kk	kkkk	k110
------	------	------	------

Skocz jeśli bit przepełnienia ustawiony

BRVS k

Operation:

- (i) If $V = 1$ then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRVS k

Operands:

$$-64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	00kk	kkkk	k011
------	------	------	------

Skocz jeśli bit przepełnienia wyzerowany

BRVC k

Operation:

- (i) If $V = 0$ then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRVC k

Operands:

$$-64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	01kk	kkkk	k011
------	------	------	------

Skocz jeśli bit maski przerwań ustawiony

BRIE k

Operation:

- (i) If I = 1 then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRIE k

Operands:

$$-64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	00kk	kkkk	k111
------	------	------	------

Skocz jeśli bit maski przerwań wyzerowany

BRID k

Operation:

- (i) If $I = 0$ then $PC \leftarrow PC + k + 1$, else $PC \leftarrow PC + 1$

Syntax:

- (i) BRID k

Operands:

$$-64 \leq k \leq +63$$

Program Counter:

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

$$PC \leftarrow PC + 1, \text{ if condition is false}$$

16-bit Opcode:

1111	01kk	kkkk	k111
------	------	------	------