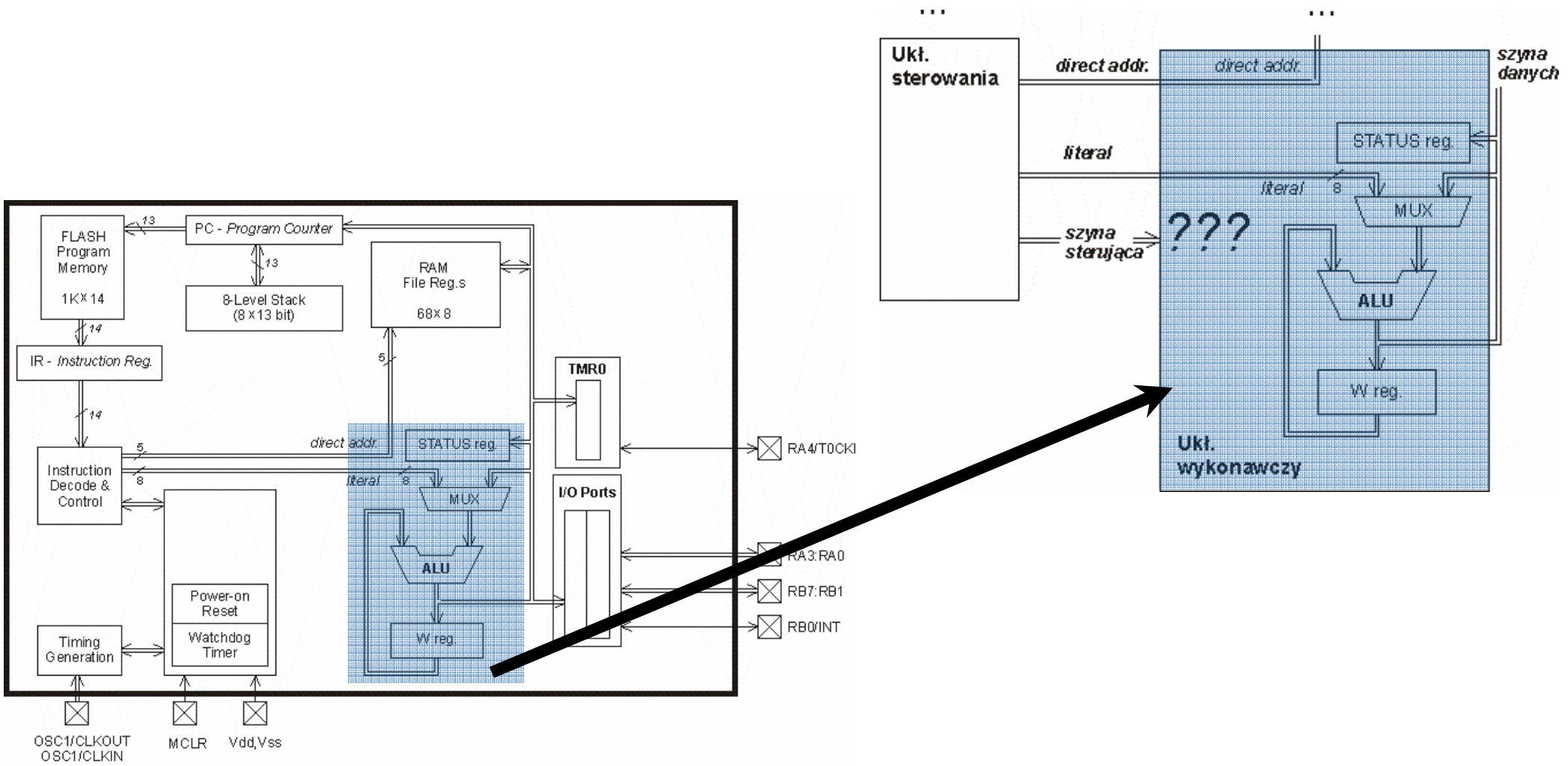
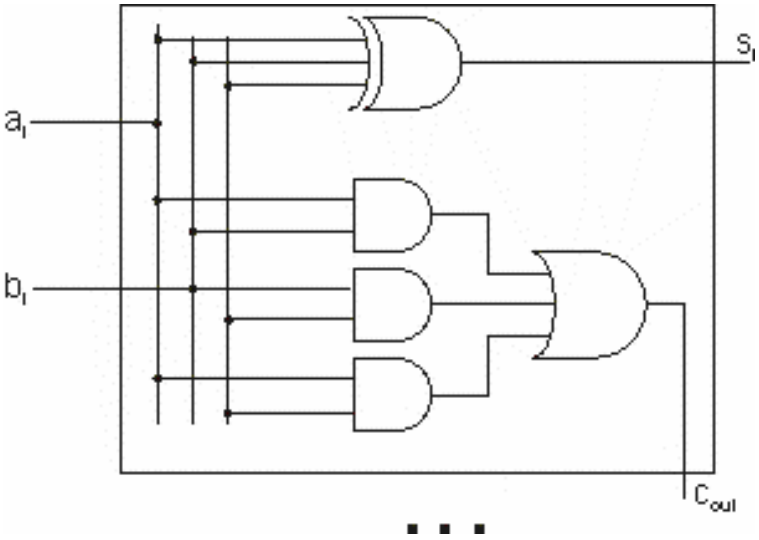
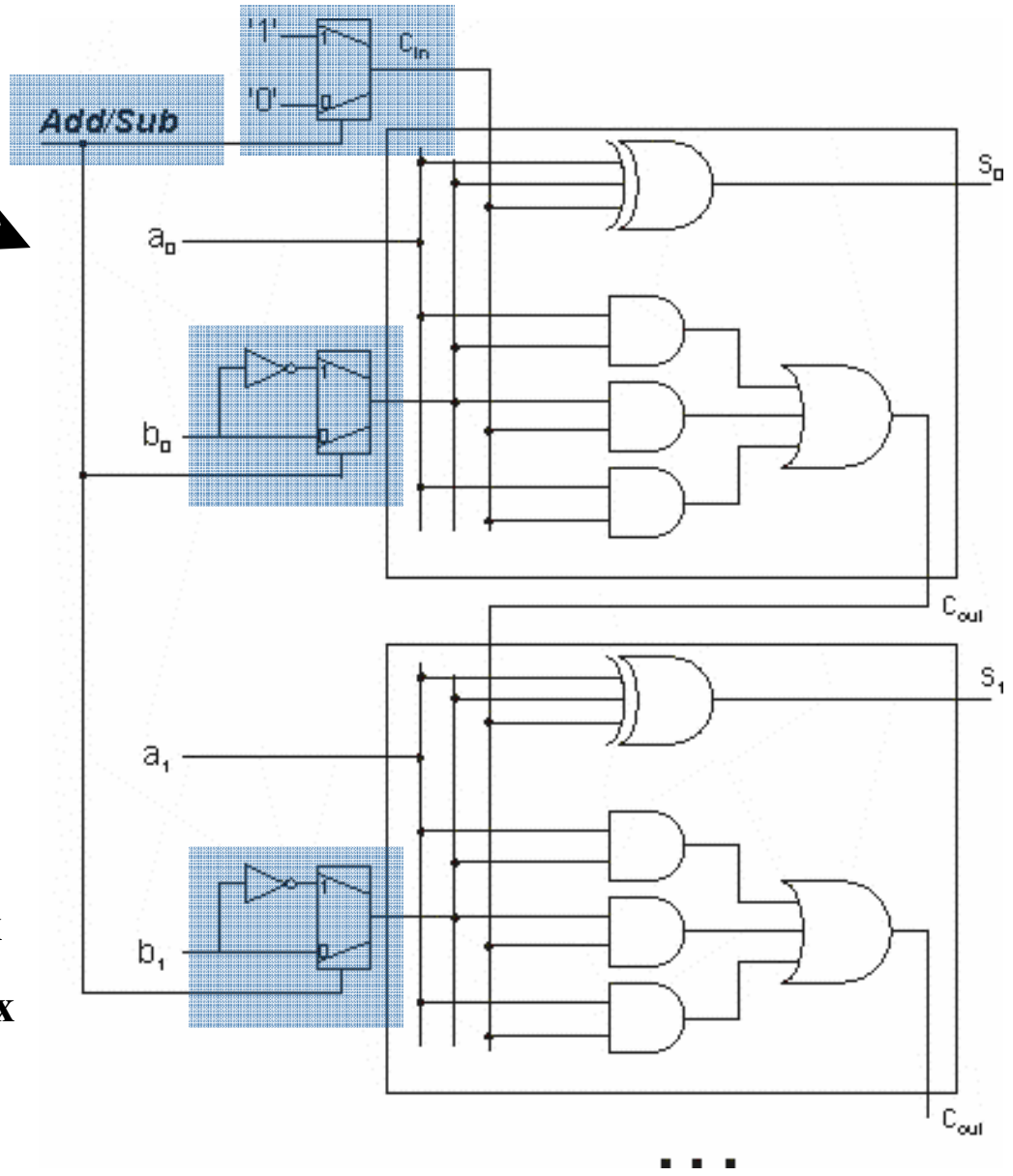
Układ wykonawczy PIC16F84A





ADDxx

INCxxx



ADDxx

SUBxx

INCxxx

DECxxx

Obserwacja 1:

a - b = a + (~b) + 1

Sygnał: *Add/Sub*

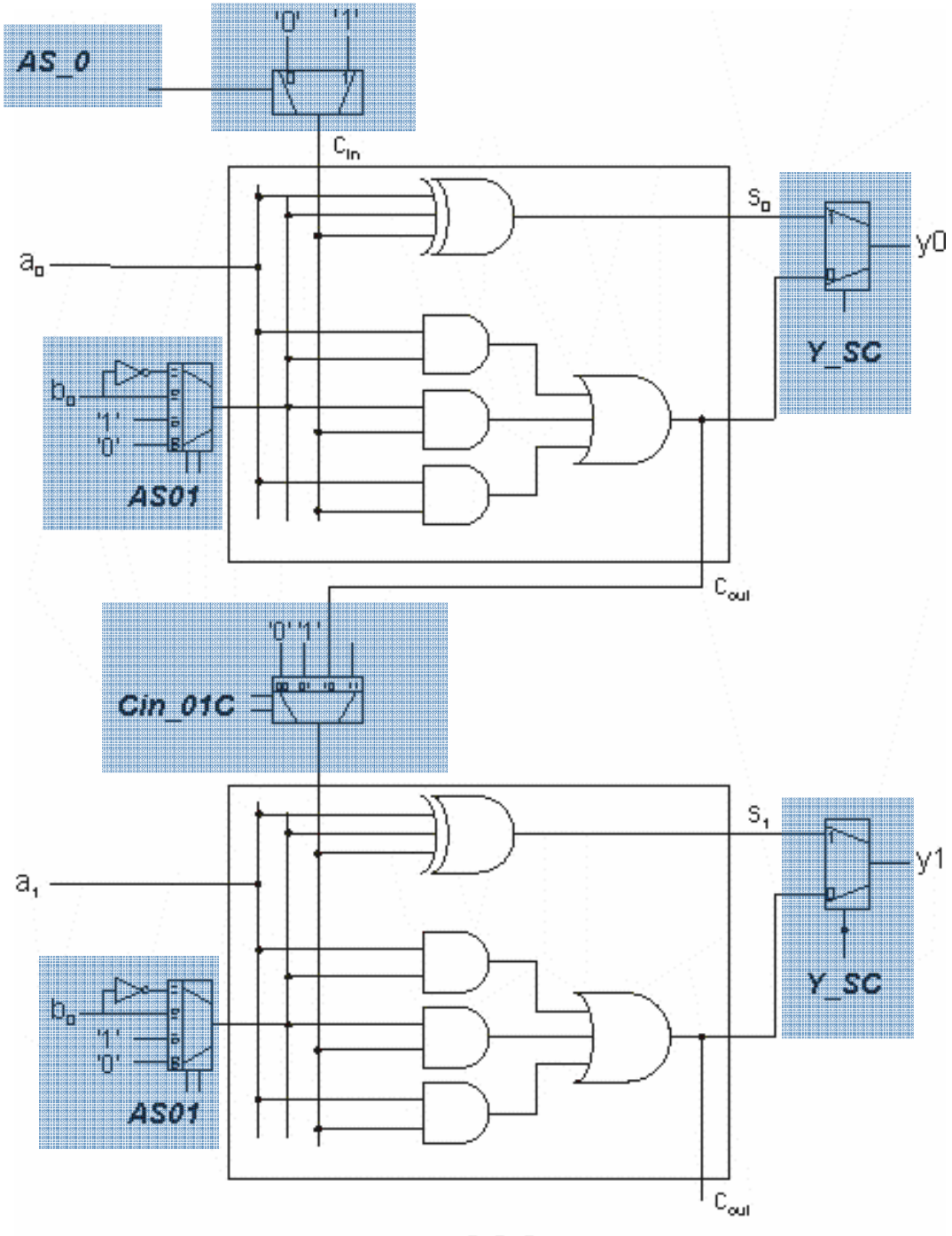
XOR	Cin_01C = 00	Y_SC = 1	AS01 = 10	AS_0 = 0	A = x
AND	Cin_01C = 00	Y_SC = 0	AS01 = 10	AS_0 = 0	A = x
OR	Cin_01C = 01	Y_SC = 0	AS01 = 10	AS_0 = 1	A = x
COMF	Cin_01C = 00	Y_SC = 1	AS01 = 01	AS_0 = 0	A = x
MOVE	Cin_01C = 00	Y_SC = 0	AS01 = 01	AS_0 = 0	A = x
CLR	Cin_01C = 00	Y_SC = 0	AS01 = 00	AS_0 = 0	A = x
NOP	Cin_01C = 00	Y_SC = 0	AS01 = 10	AS_0 = 0	A = 1

Obserwacja 2:

bramki do realizacji wszystkich funkcji logicznych są już w strukturze sumatora

Sygnały:

AS01	***2 bit***
AS_0	***1 bit***
Cin_01C	***2 bit***
Y_SC	***1 bit***
	6 bit

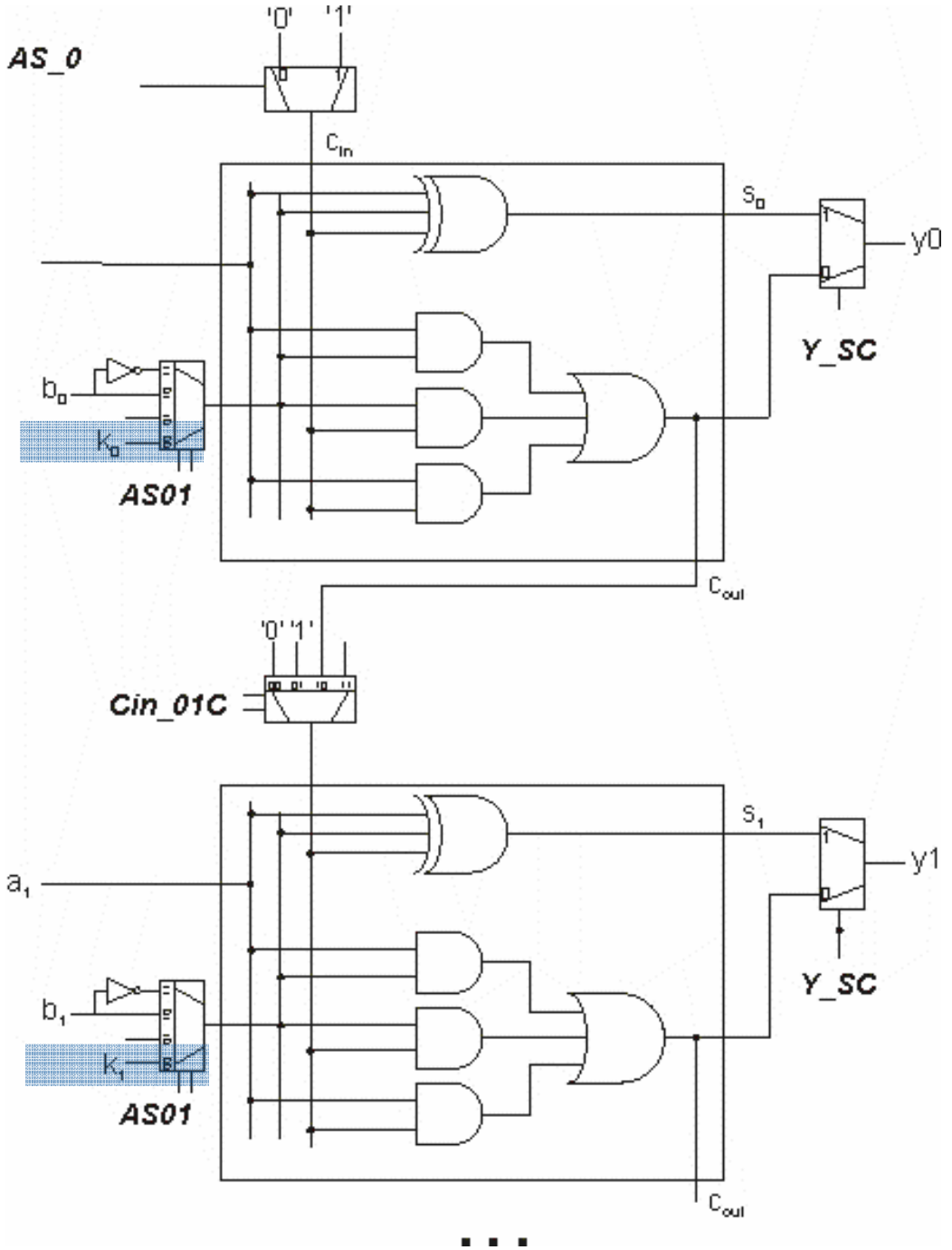


Obserwacja 3:

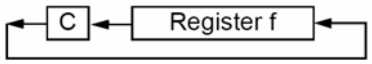
Instrukcje bsf i bcf wymagają indywidualnego sterowania wejściem B

Sygnały dodatk.:

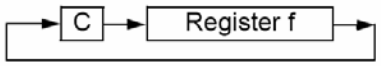
k ***8 bit***



RLF



RRF



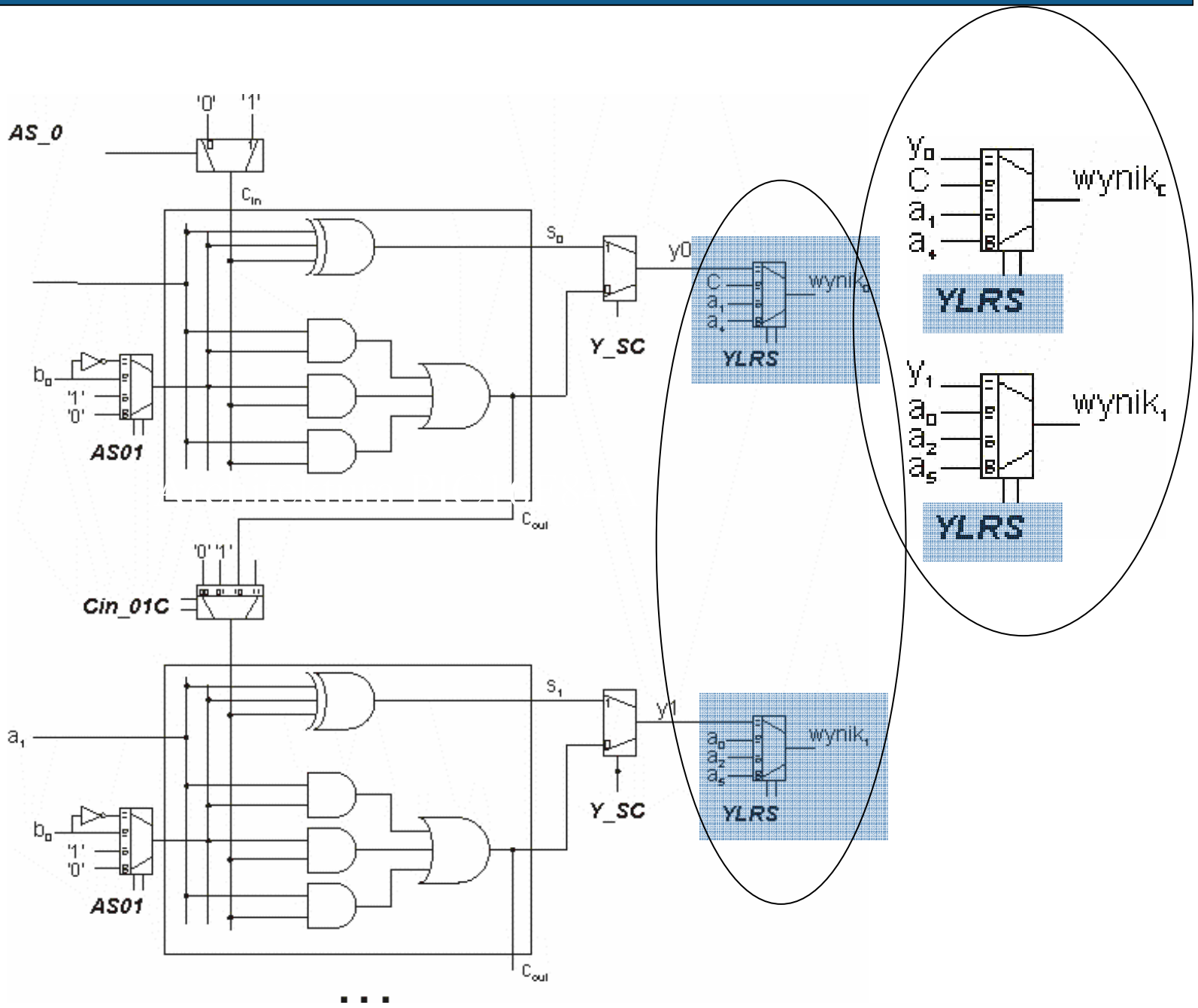
SWAPF

Obserwacja 4:

Osobno wykonać operacje przesunięć

Sygnały dodatk.:

YLRS 2 bit

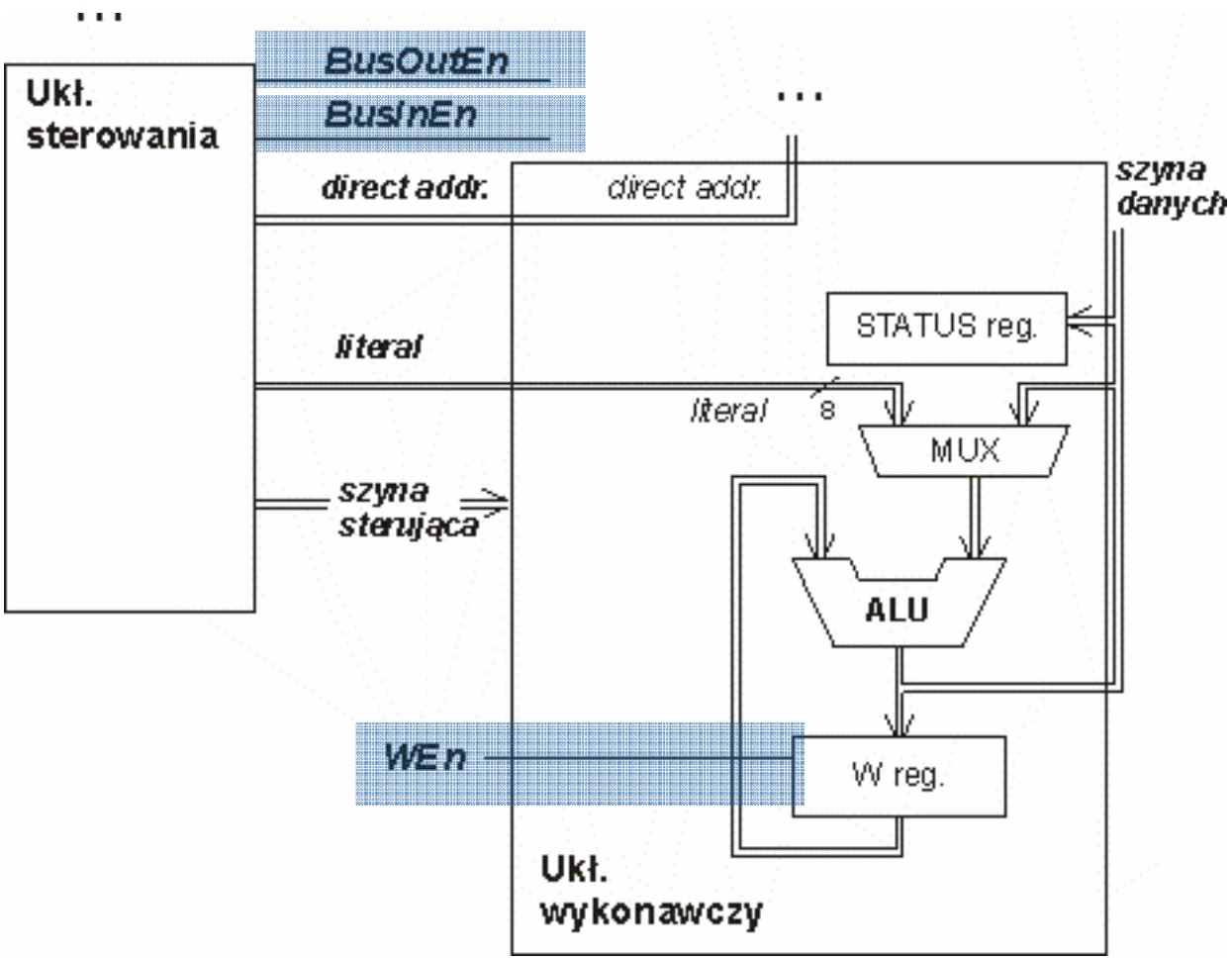


Obserwacja 5:

Nie wszystkie operacje zapisują wynik do W oraz korzystają z F

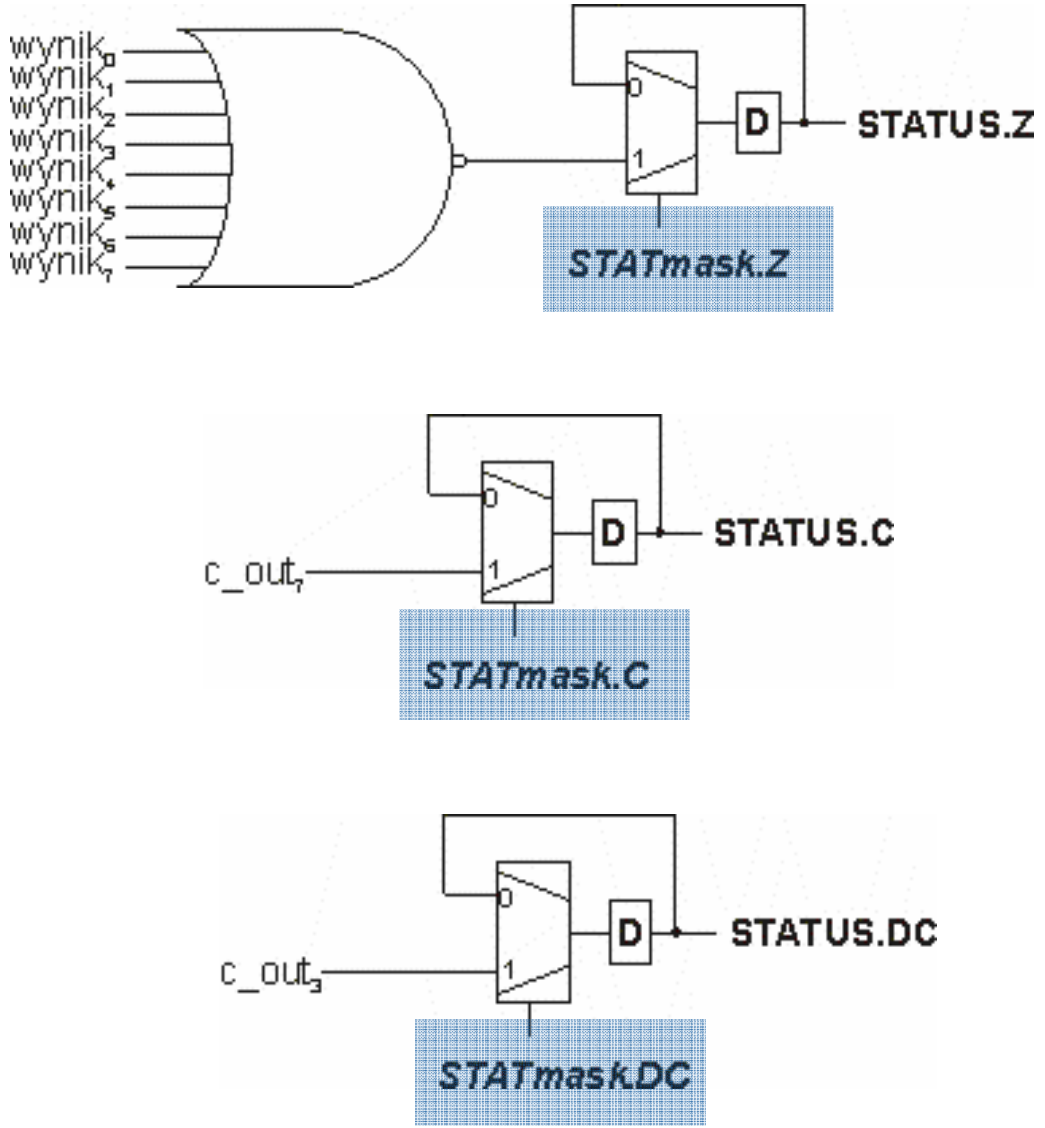
Sygnały dodatk.:

WEn	*1 bit*
BusInEn	*1 bit*
BusOutEn	*1 bit*



Obserwacja 6:
Rej. STATUS jest zmieniany tylko przez niektóre instr. Musi on być maskowany

Sygnały dodatk.:
STATmask 3 bit



Sygnały:		
	AS01	*2 bit*
	AS_0	*1 bit*
	Cin_01C	*2 bit*
	Y_SC	*1 bit*
	k	*8 bit*
	YLRS	*2 bit*
	WEn	*1 bit*
	BusInEn	*1 bit*
	BusOutEn	*1 bit*
	STATmask	*3 bit*
		22 bity

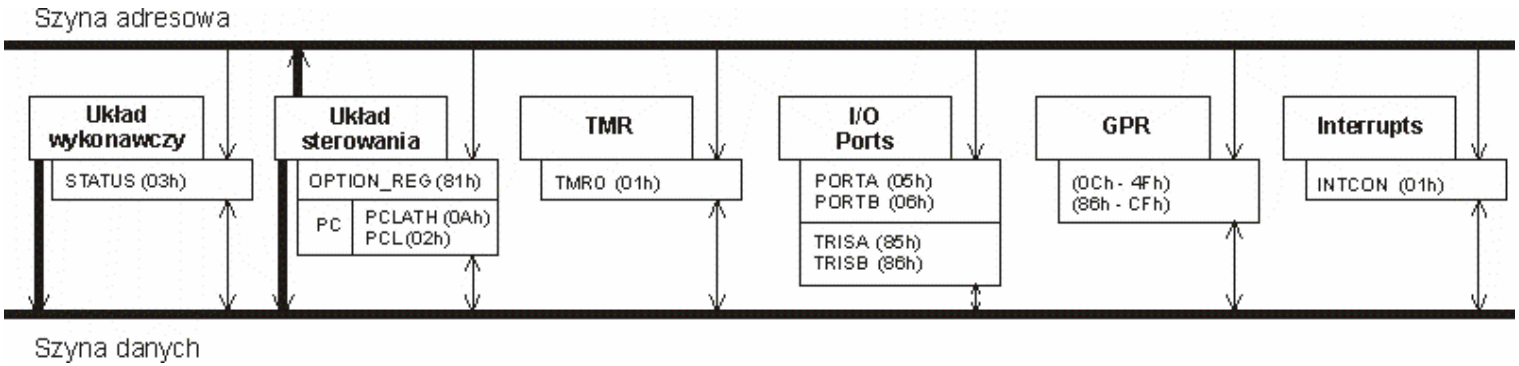
Układ wykonawczy
Podsumowanie

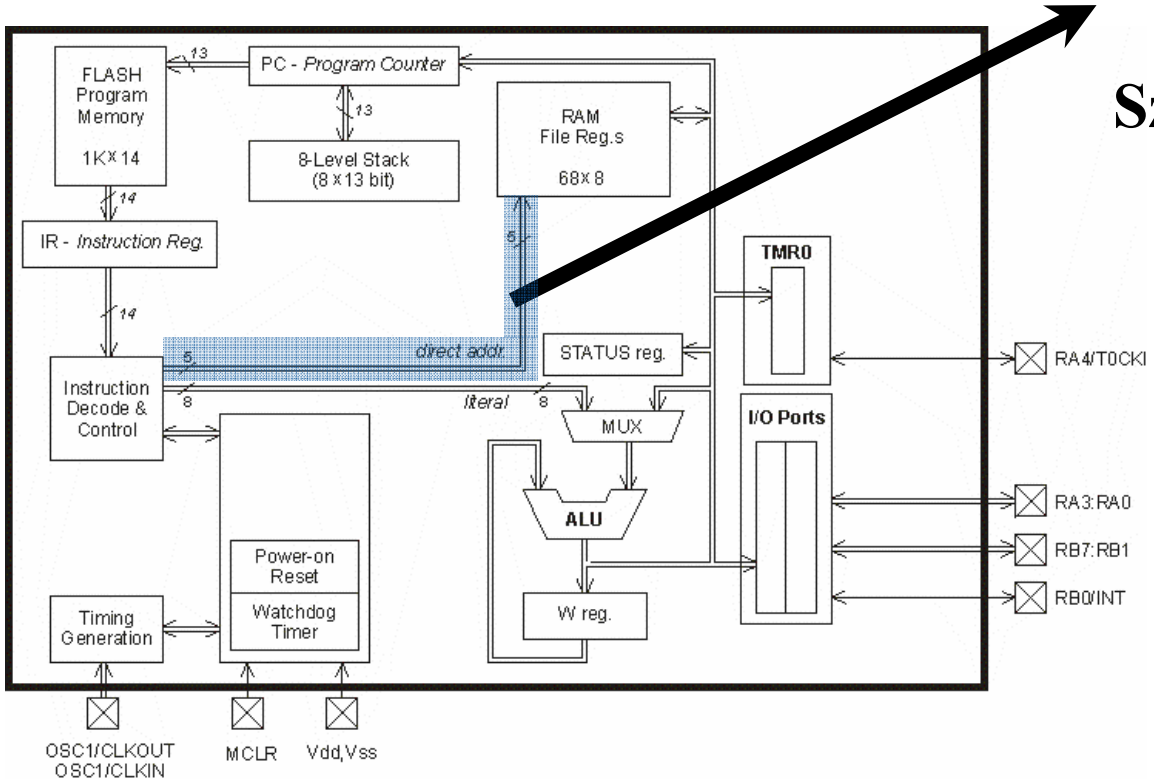
Zadanie

Wykonać tabelę

Instrukcje \ Sygnały sterujące

Szyna adresowa





Szyna adresowa 6-bit

2. Szyna adresowa

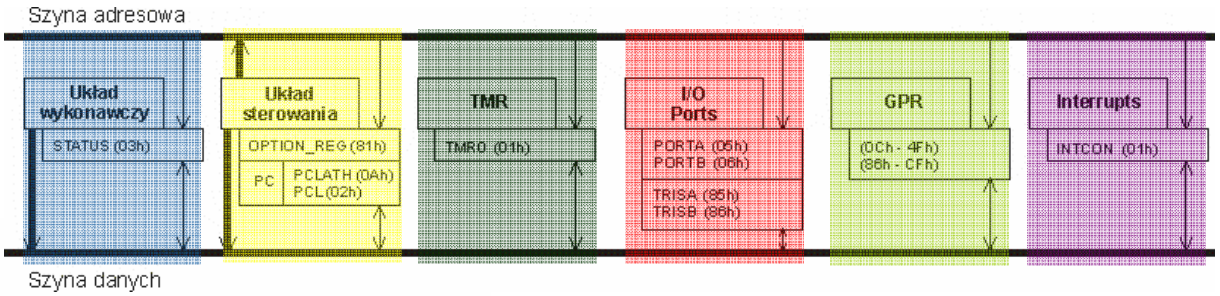
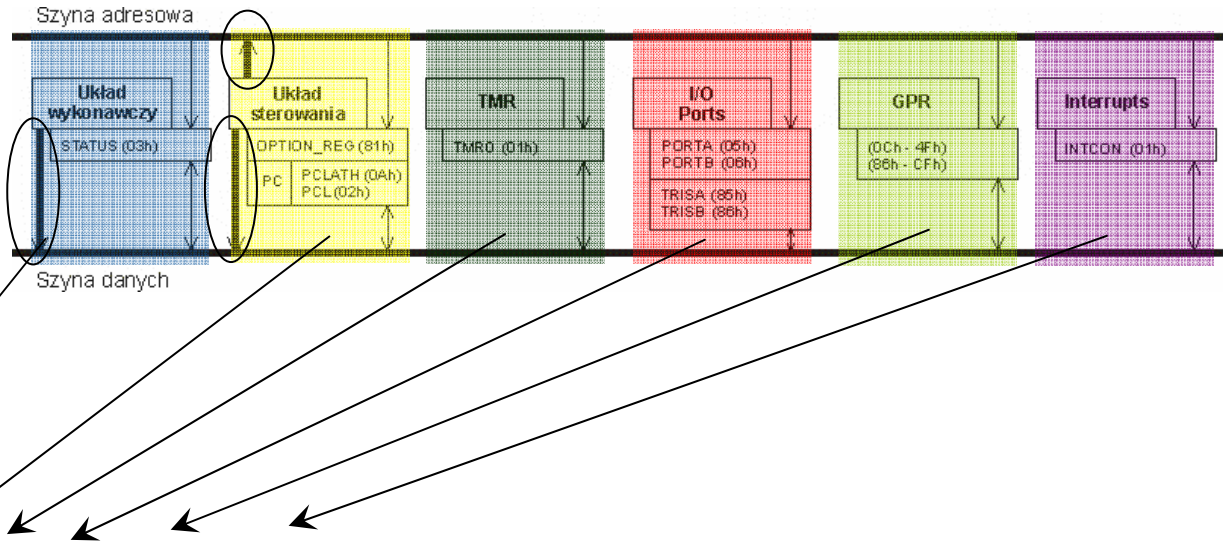
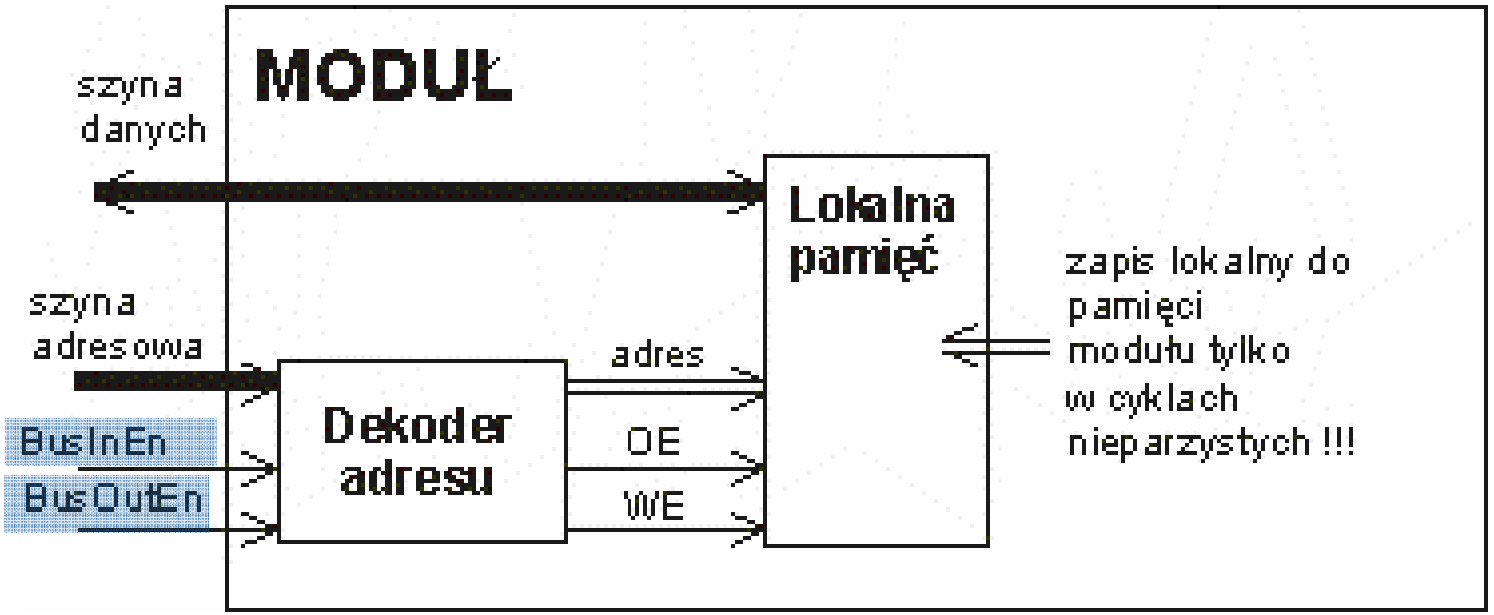


TABLE 2-1: SPECIAL FUNCTION REGISTER FILE SUMMARY

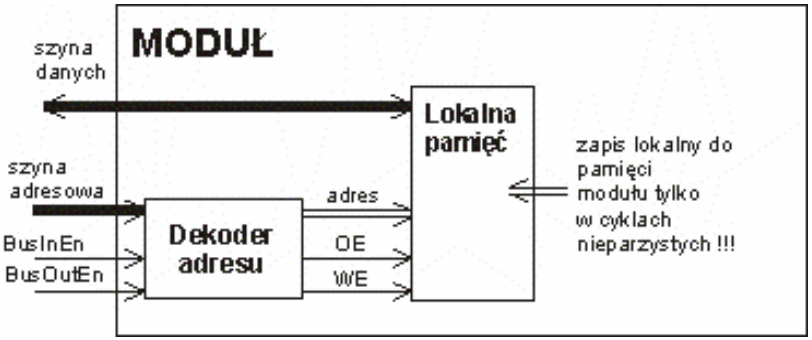
Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on RESET	Details on page	
Bank 0												
00h	INDF	Uses contents of FSR to address Data Memory (not a physical register)								---- ----	11	
01h	TMR0	8-bit Real-Time Clock/Counter								xxxx xxxx	20	**TMR**
02h	PCL	Low Order 8 bits of the Program Counter (PC)								0000 0000	11	**Układ sterowania**
03h	STATUS(2)	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	8	**Układ wykonawczy**
04h	FSR	Indirect Data Memory Address Pointer 0								xxxx xxxx	11	
05h	PORTA(4)	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	16	**I/O Ports**
06h	PORTB(5)	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	18	
07h	—	Unimplemented location, read as '0'								—	—	
08h	EEDATA	EEPROM Data Register								xxxx xxxx	13,14	
09h	EEADR	EEPROM Address Register								xxxx xxxx	13,14	
0Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of the PC(1)					---0 0000	11	**Układ sterowania**
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	10	**Interrupts**
Bank 1												
80h	INDF	Uses Contents of FSR to address Data Memory (not a physical register)								---- ----	11	
81h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	9	**Układ sterowania**
82h	PCL	Low order 8 bits of Program Counter (PC)								0000 0000	11	
83h	STATUS (2)	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	8	**Układ wykonawczy**
84h	FSR	Indirect data memory address pointer 0								xxxx xxxx	11	
85h	TRISA	—	—	—	PORTA Data Direction Register					---1 1111	16	**I/O Ports**
86h	TRISB	PORTB Data Direction Register								1111 1111	18	
87h	—	Unimplemented location, read as '0'								—	—	
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	13	
89h	EECON2	EEPROM Control Register 2 (not a physical register)								---- ----	14	
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC(1)					---0 0000	11	**Układ sterowania**
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	10	**Interrupts**

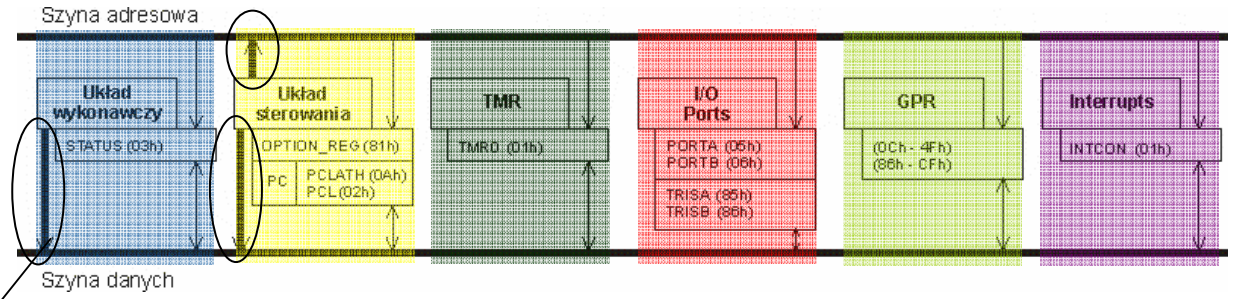
2. Szyna adresowa

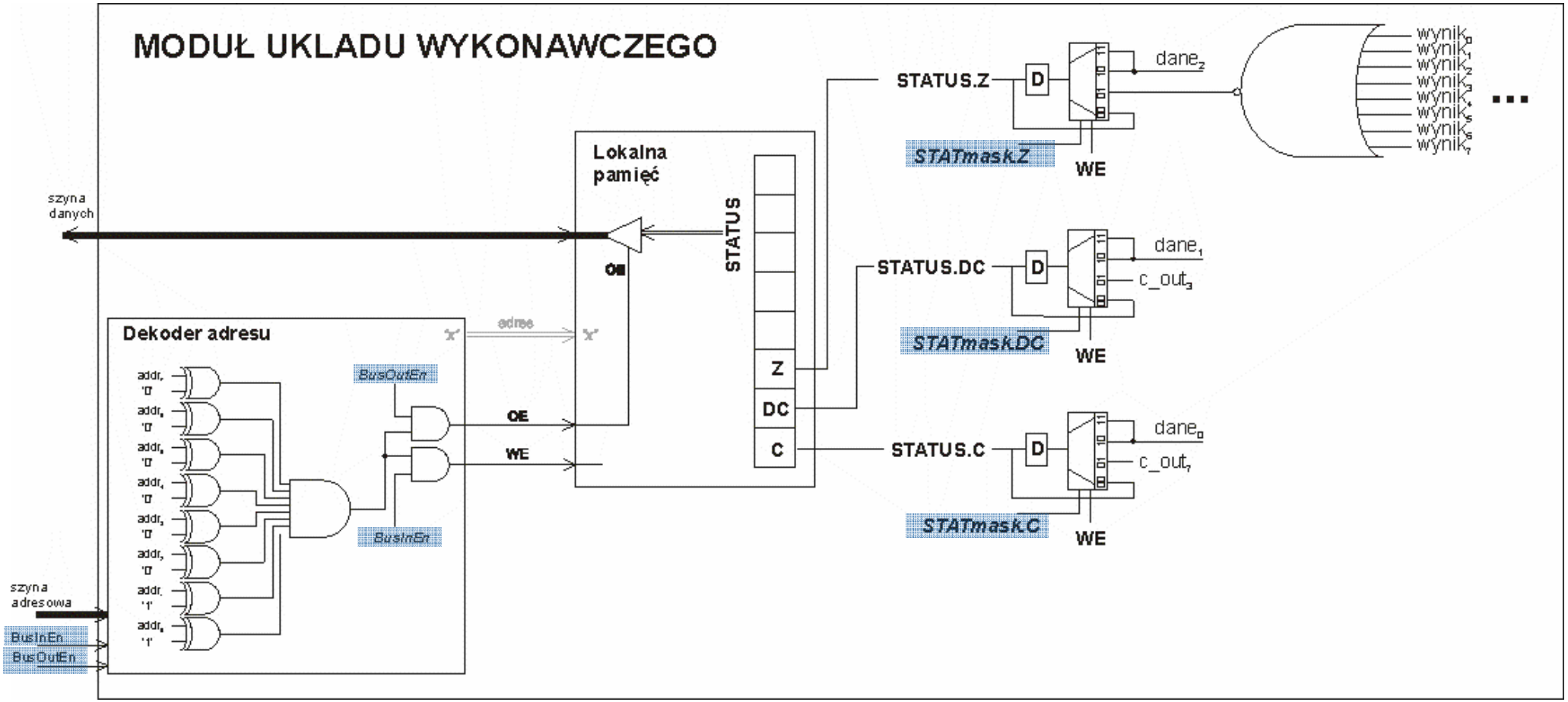




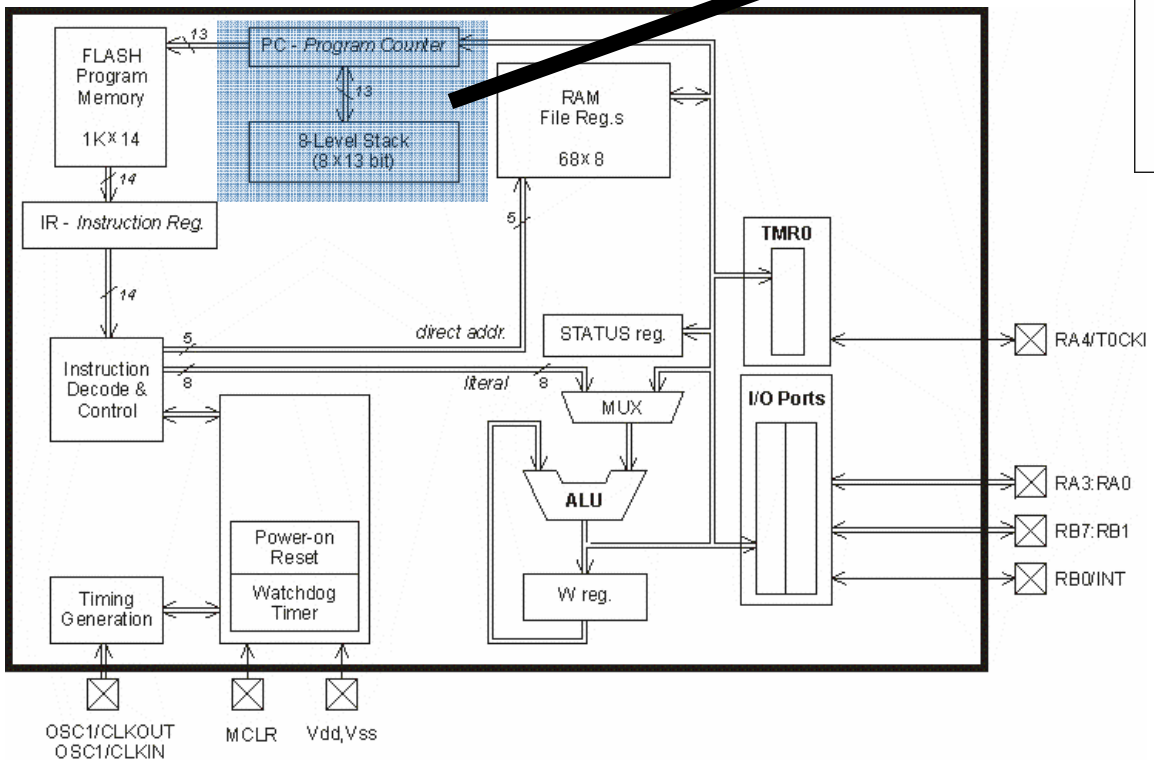
2. Szyna adresowa

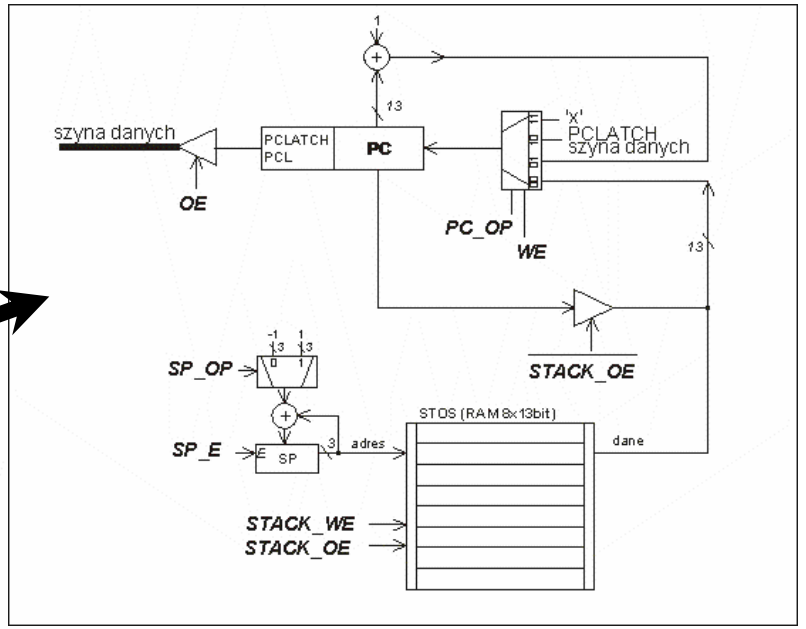


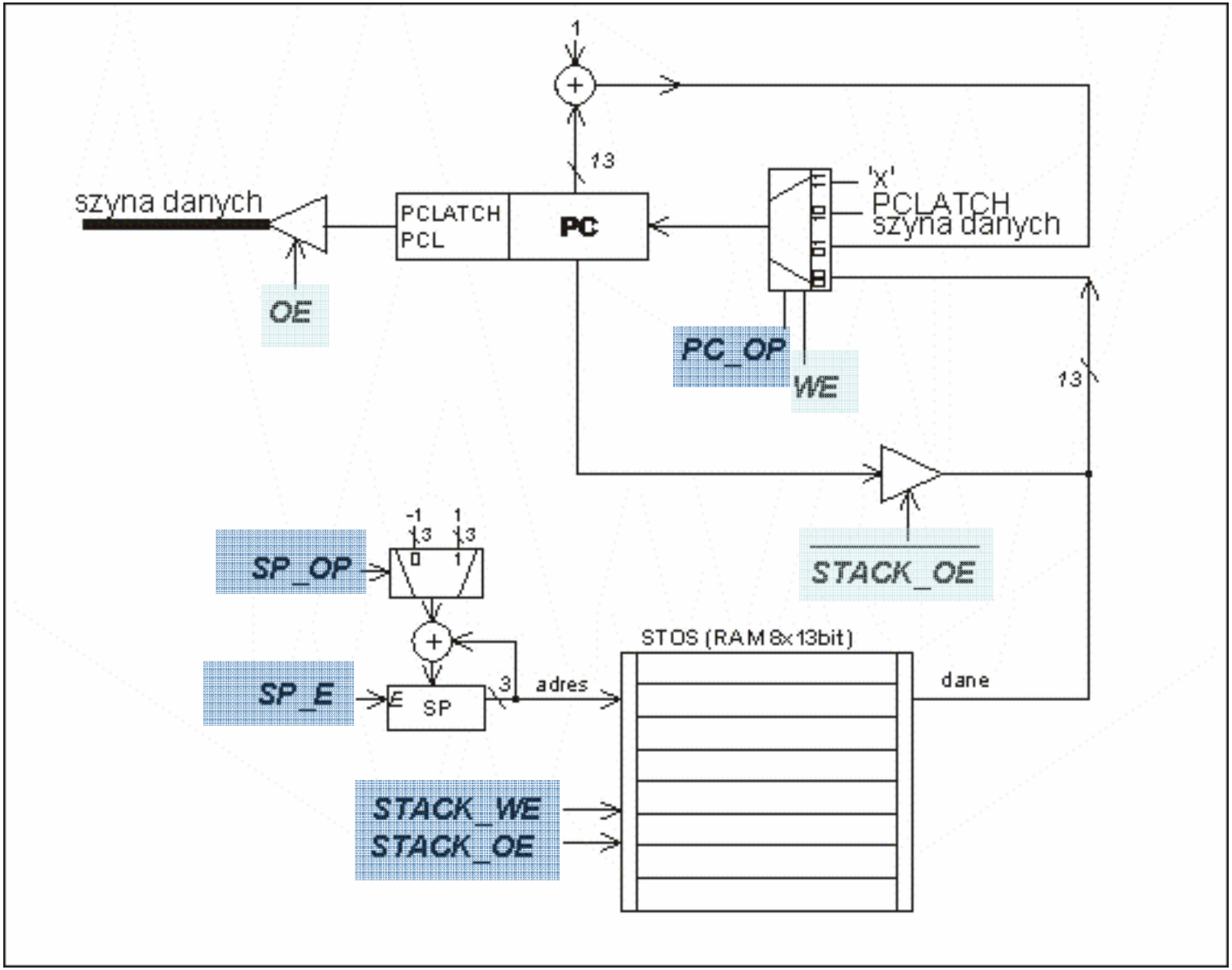




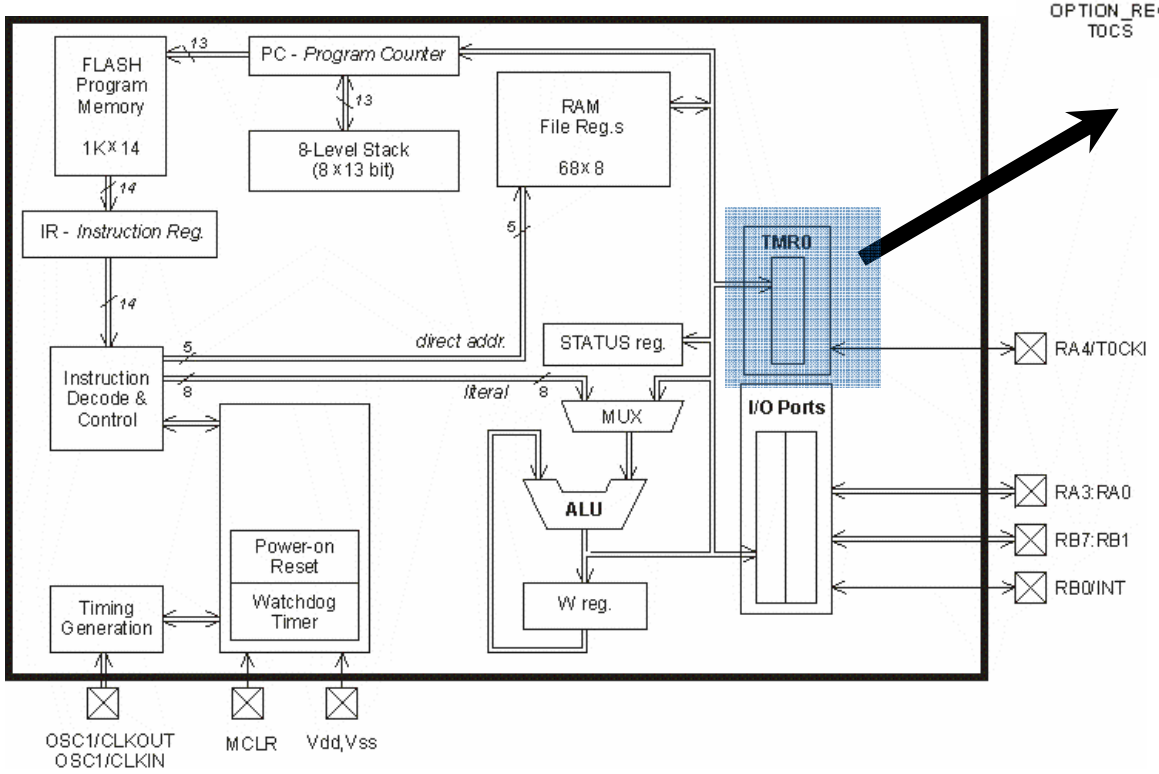
Układ sterujący – sterowanie PC

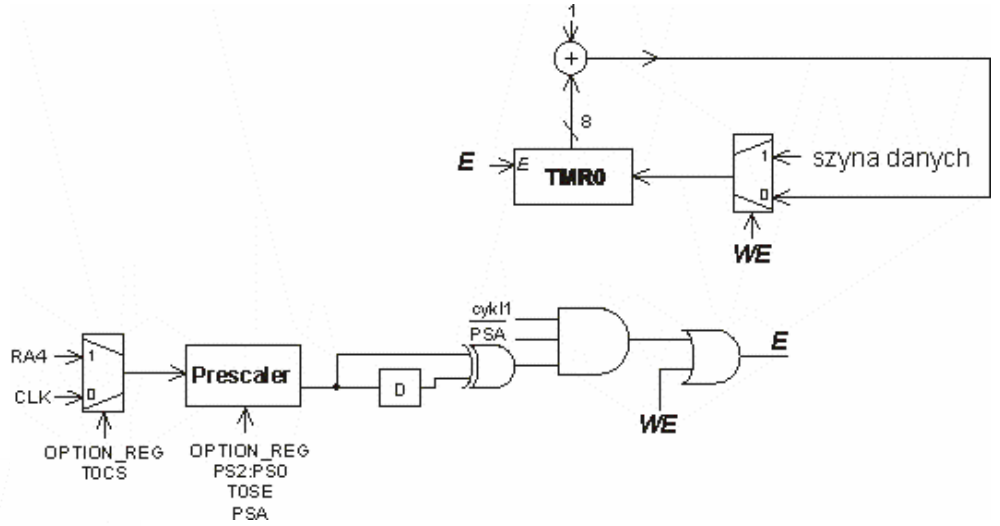


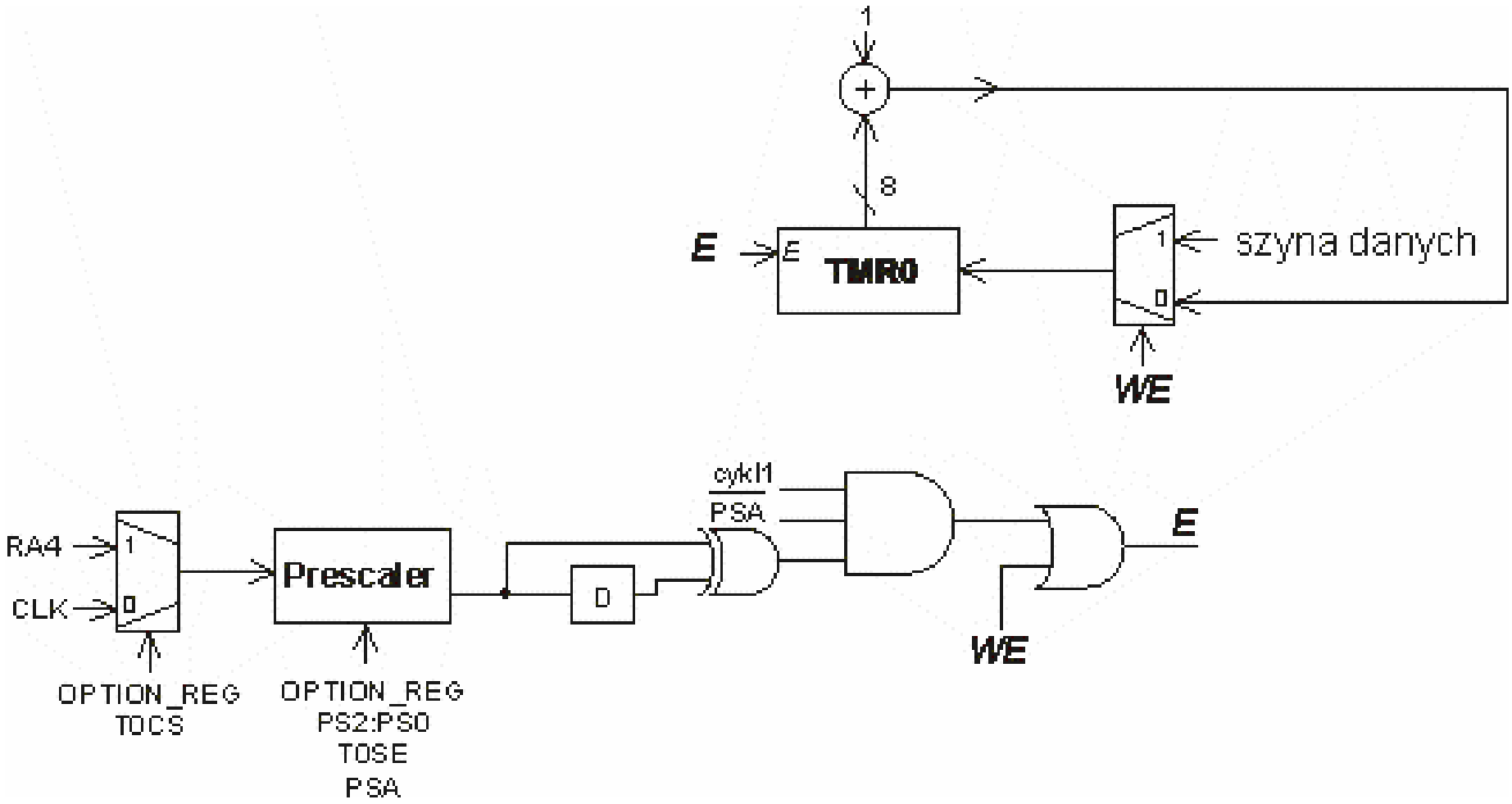



1
13
szyna danych
PCLATCH
PCL
PC
'x'
PCLATCH
szyna danych
OE
PC_OP
WE
13
STACK_OE
-1
1
3
3
SP_OP
0
1
STOS (RAM 8x13bit)
SP_E
E
SP
3
adres
dane
STACK_WE
STACK_OE

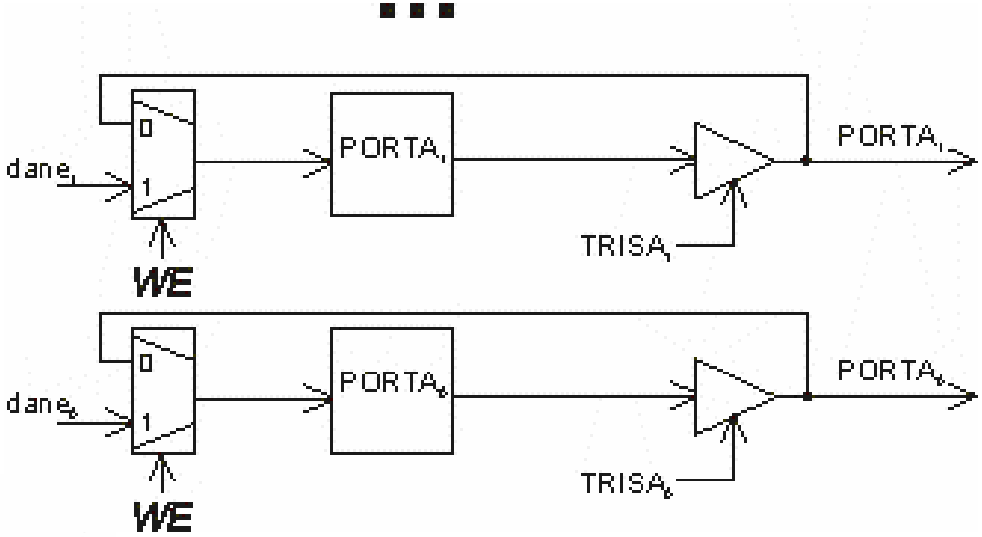
Moduł Timera

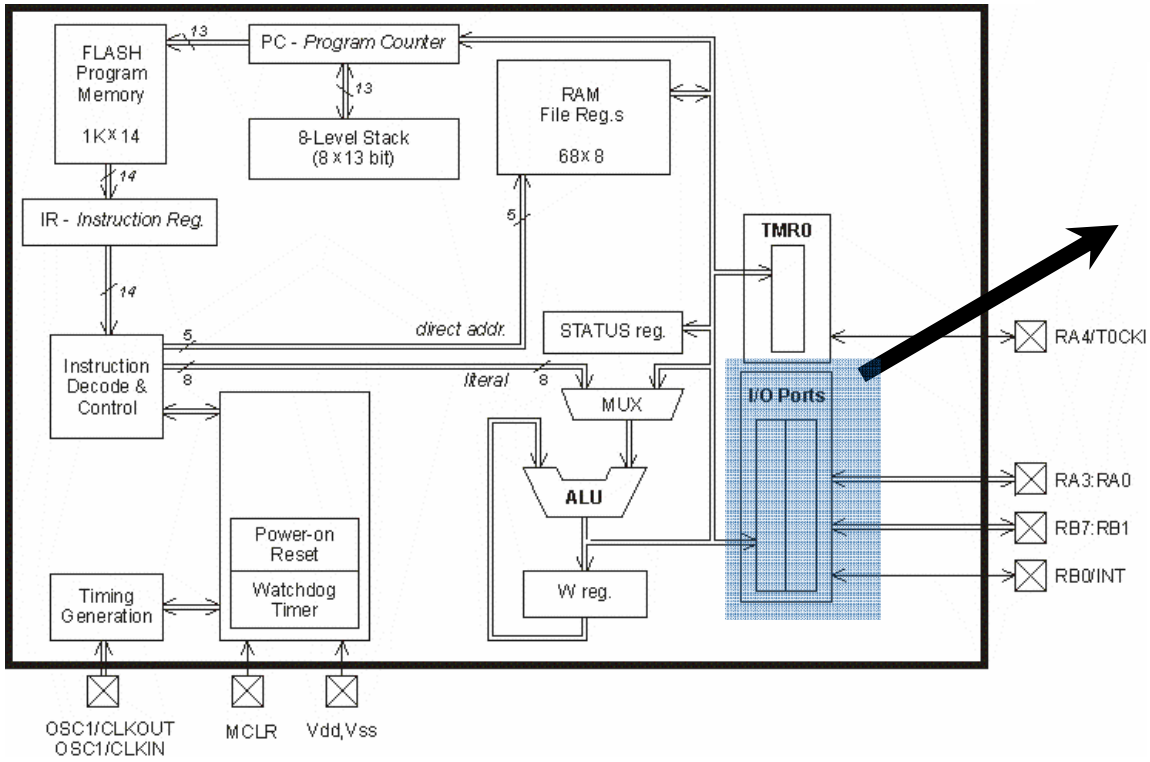


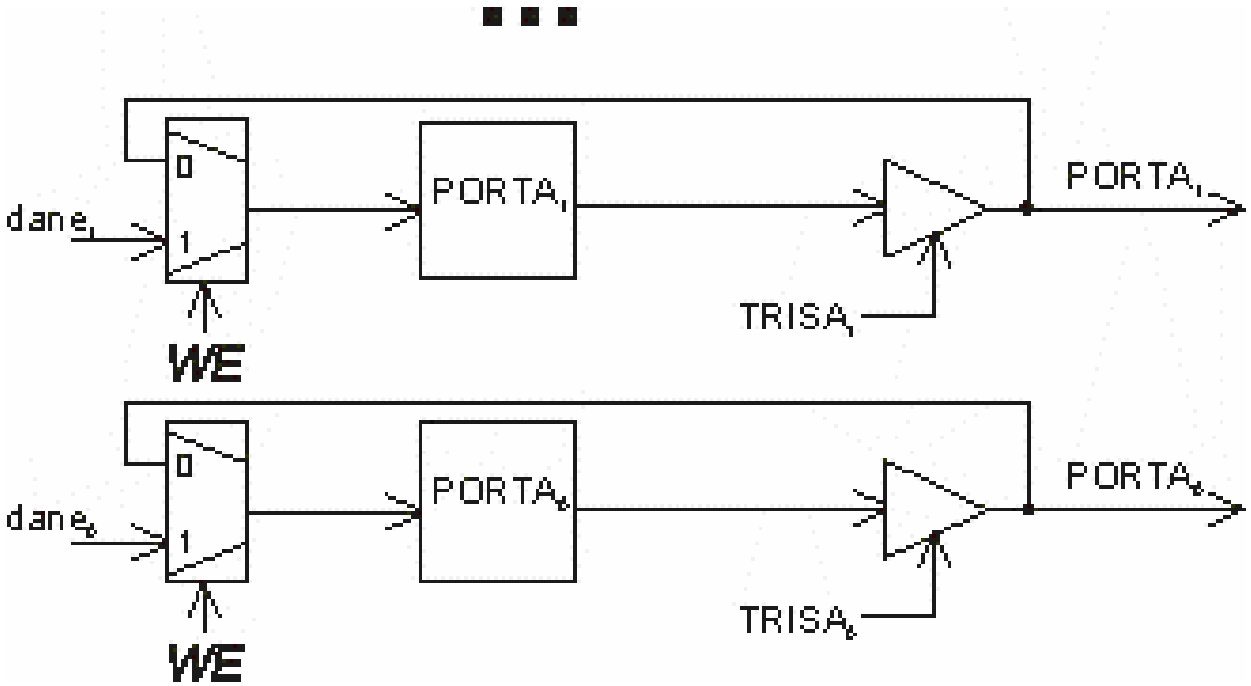



1
8
E
E
TMR0
szyna danych
1
0
WE
cykl1
PSA
RA4
CLK
1
0
Prescaler
D
E
WE
OPTION_REG
TOCS
OPTION_REG
PS2:PS0
TOSE
PSA

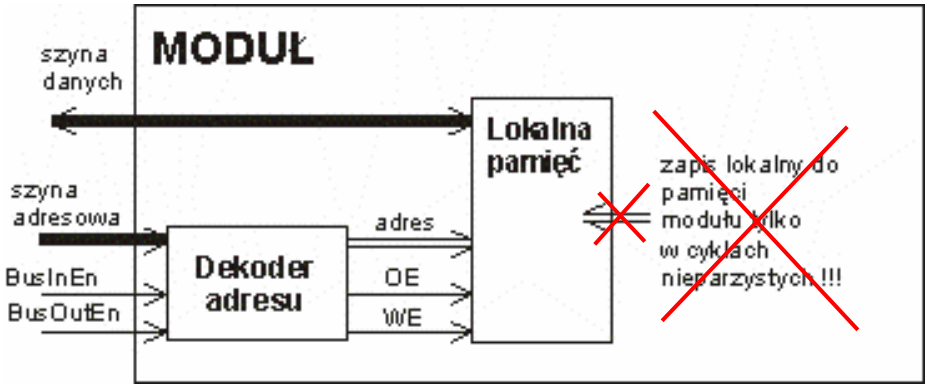
Moduł I/O Ports...

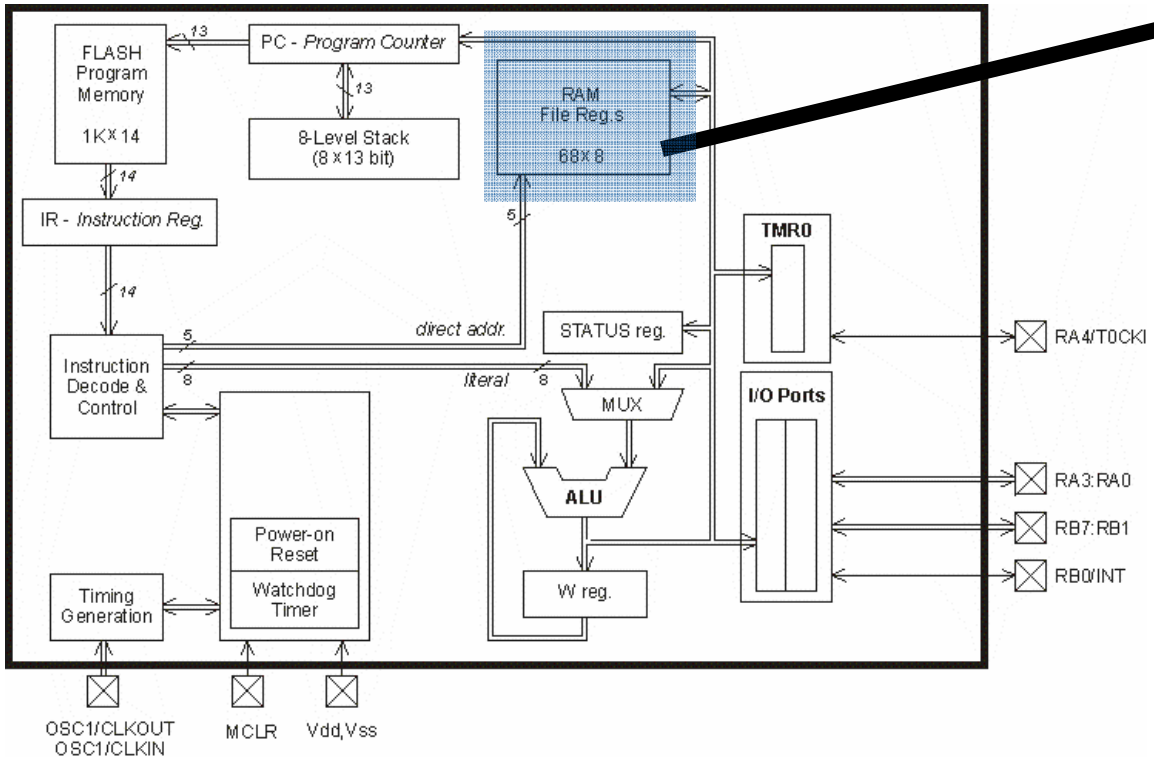





dane$_1$
0
1
WE
PORTA$_1$
TRISA$_1$
PORTA$_1$
dane$_8$
0
1
WE
PORTA$_8$
TRISA$_8$
PORTA$_8$

Moduł GPR


MODUŁ
szyna danych
szyna adresowa
BusInEn
BusOutEn
Dekoder adresu
adres
OE
WE
Lokalna pamięć
zapis lokalny do pamięci modułu tylko w cyklach nieparzystych !!!


FLASH Program Memory 1K x 14
PC - Program Counter
8-Level Stack (8 x 13 bit)
RAM File Reg.s 68x 8
IR - Instruction Reg.
Instruction Decode & Control
direct addr.
literal
STATUS reg.
MUX
ALU
W reg.
TMR0
I/O Ports
RA4/T0CKI
RA3:RA0
RB7:RB1
RB0/INT
Power-on Reset
Watchdog Timer
Timing Generation
OSC1/CLKOUT
OSC1/CLKIN
MCLR
Vdd,Vss