Proiectare cu Microprocesoare

Curs 2

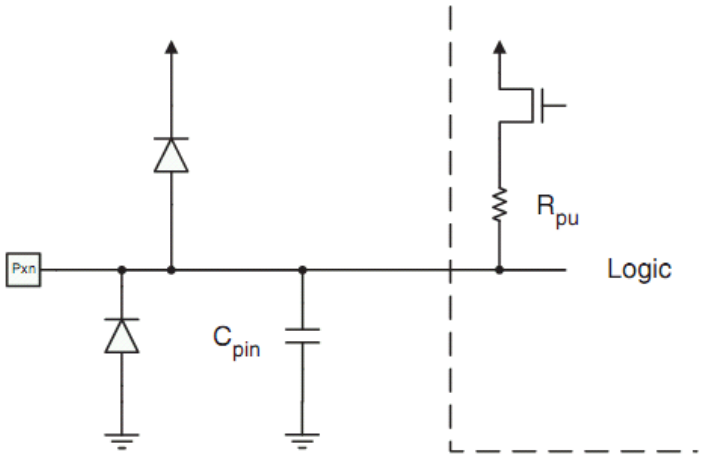
I/O si intreruperi la microcontrollerele AVR

An 3 CTI
Semestrul I

Lector: Răzvan Itu

Intrare / Iesire

- Porturile de intrare/iesire:
  - ATmega 328P (UNO): **PORT B, C, D**
  - ATmega 2560 (MEGA): **PORT A, B, C, D, E, F, G, H, J, K, L**
- PORTA... PORTE pot fi accesate prin instructiuni speciale **in**, **out**
- PORTF, PORTG accesibile doar prin **ld**, **st** – spatiul de adrese I/O extins
- Fiecare bit din fiecare port poate fi configurat ca intrare sau ca iesire, prin scrierea registrului de directie DDRx
- Scrierea portului se face prin registrul PORTx
- Citirea starii pinilor se face prin PINx
- Diode de protectie, impotriva electricitatii statice
- Rezistenta “pull up”, care poate fi activata/ dezactivata prin logica



Address (HEX)	
0 - 1F	32 Registers
20 - 5F	64 I/O Registers
60 - 1FF	416 External I/O Registers
200	Internal SRAM (8192 × 8)
21FF	
2200	External SRAM (0 - 64K × 8)
FFFF	

Intrare / Ieşire

Pentru fiecare port sunt alocate trei locaţii în spaţiul de adrese I/O

- Registrul de date – PORTx,
- Registrul de direcţie – DDRx
- Starea pinilor de intrare – PINx

Exemplu (PORTA):

PORTA – Port A Data Register

Bit	7	6	5	4	3	2	1	0	
0x02 (0x22)	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
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	

DDRA – Port A Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x01 (0x21)	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
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	

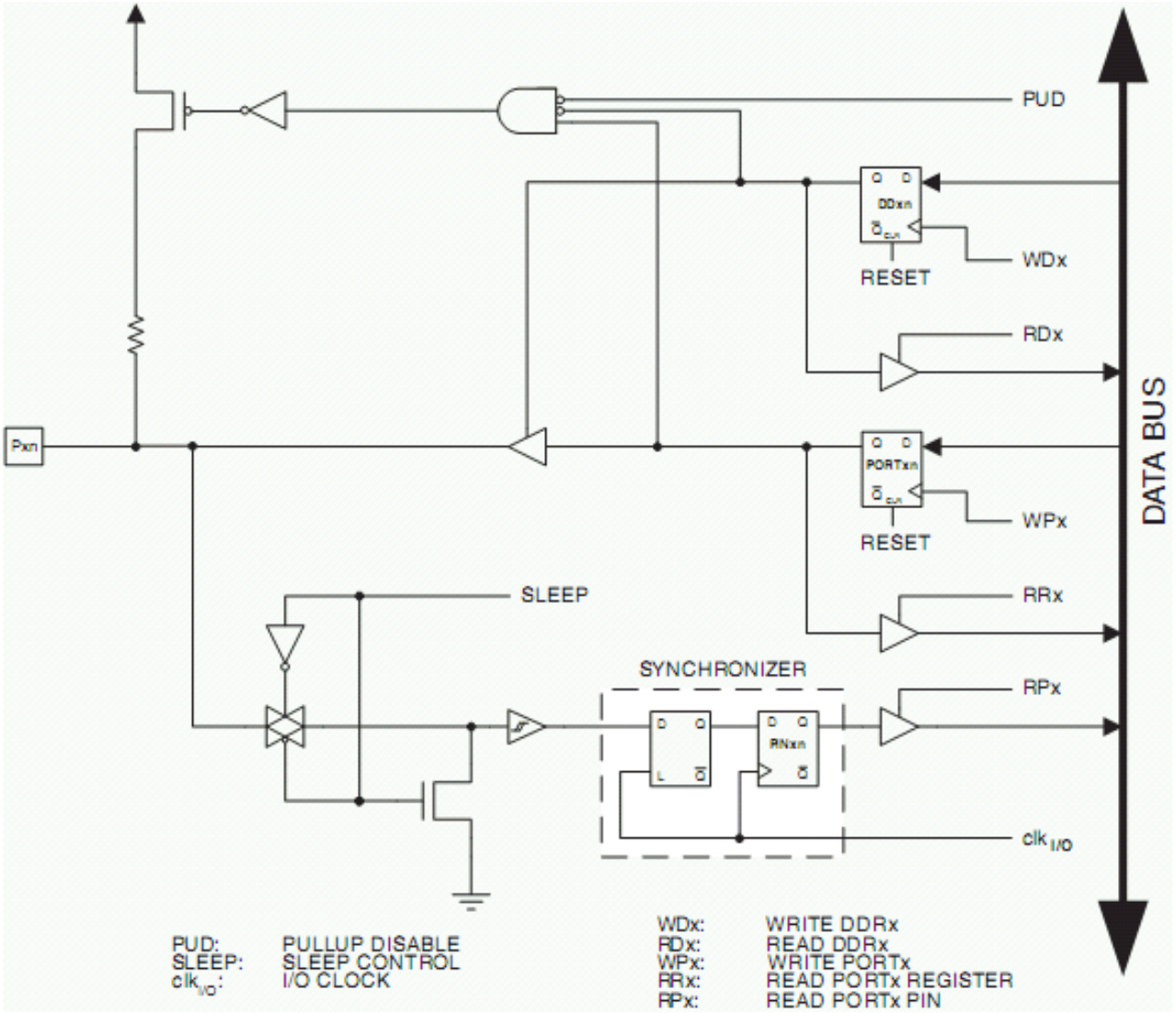
PINA – Port A Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x00 (0x20)	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Notaţie: PORTxn = pin n în PORTx (ex: PORTB3 – pentru bitul 3 din Port B).

Intrare / Iesire

- Schema generala pentru 1 bit dintr-un port I/O



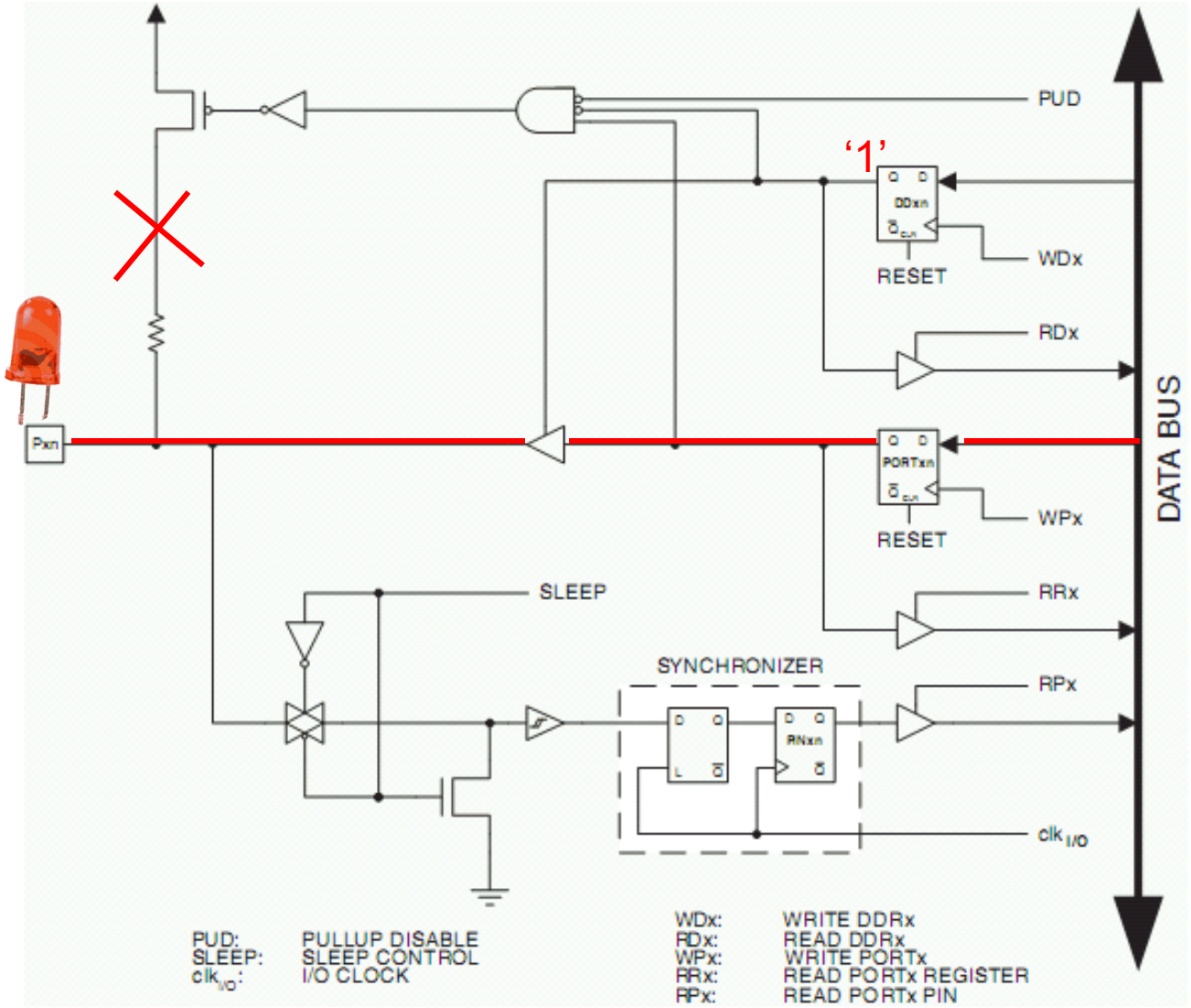
Control Directie

Datele ce vor fi trimise la iesire

Datele citite de pe intrare

Intrare / Iesire

- Configuratia pentru iesire

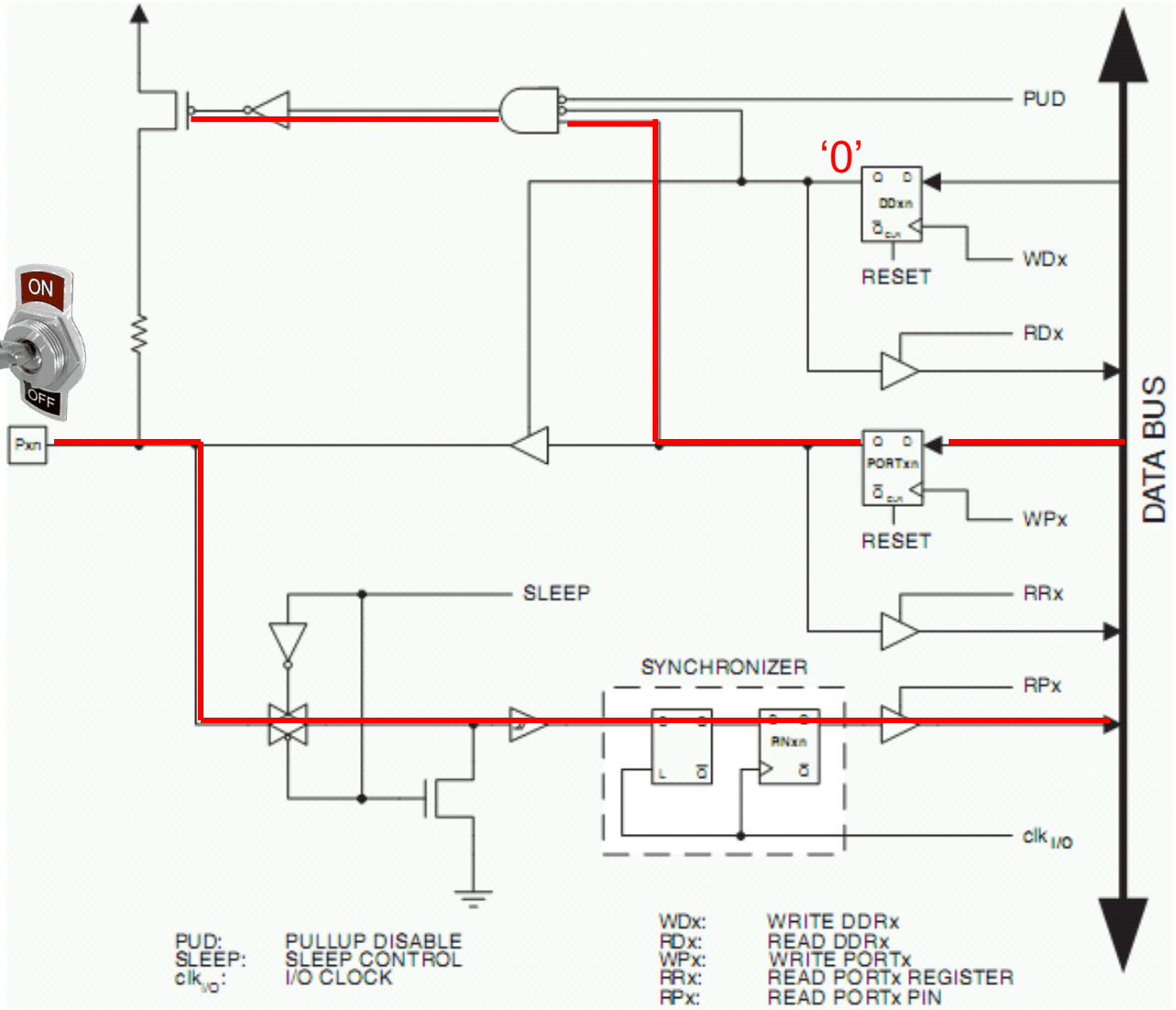


Directie = 1

Datele scrise in PORTx sunt trimise la iesire

Intrare / Iesire

- Configuratia pentru intrare



Directie = 0

'1' scris in PORTx activeaza rezistenta pull up

Datele devin disponibile ca PINx

Intrare / Iesire

- Stari posibile ale pinilor I/O

DDxn	PORTxn	PUD (MCUCR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

- PUD – Pull Up Disable – GLOBAL
  - Valoarea ‘1’ a bitului 4 din MCUCR dezactiveaza toate rezistentele pull up

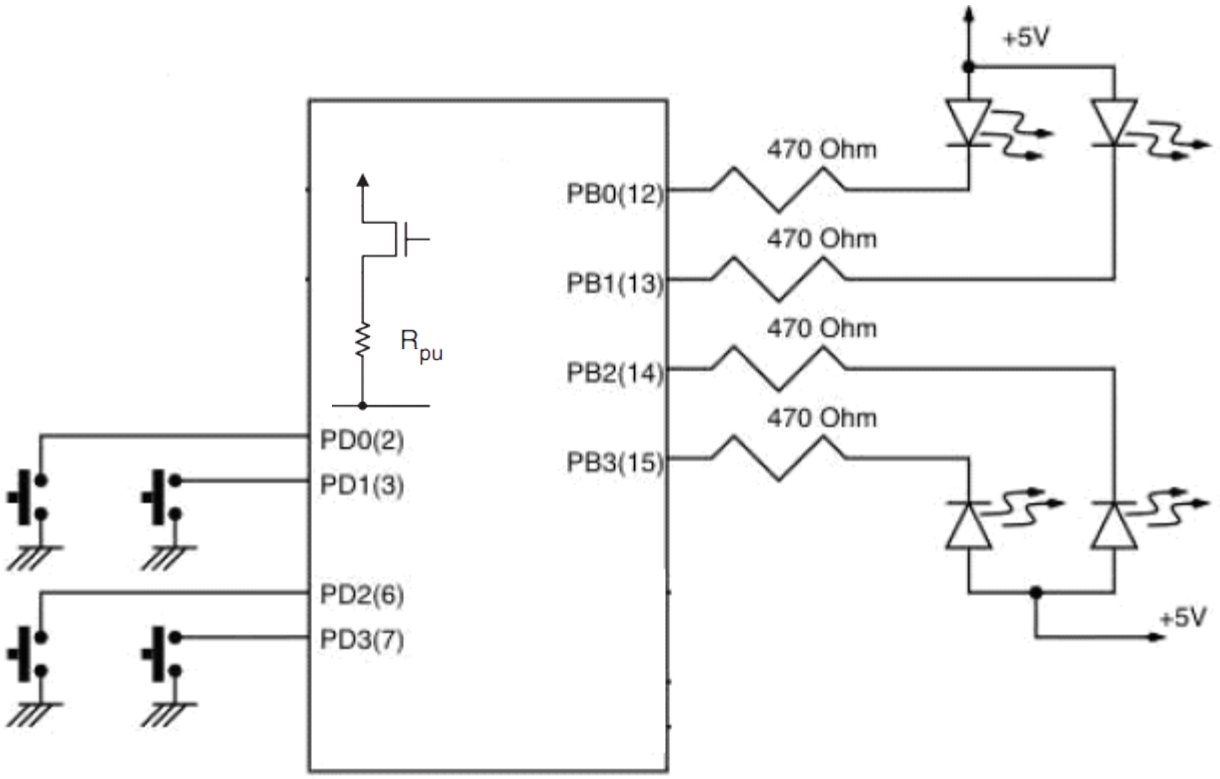
Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	JTD	–	–	PUD	–	–	IVSEL	IVCE	MCUCR
Read/Write	R/W	R	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

```
in r17, MCUCR
ori r17, 0b00010000
out MCUCR, r17
```

```
sbi MCUCR, 4
```

Intrare / Iesire

- Exemplu – Butoane si LED-uri
- Rezistentele pull-up asigura nivelul ‘1’ pe pin cand butonul este in repaus
- Cand butonul este apasat, nivelul pinului este ‘0’ prin legare la GND
- Un nivel ‘0’ pe pinii de iesire (B) cauzeaza diferenta de potential pe LED-uri, provocand aprinderea lor
- Nivelul ‘1’ pe pinii B stinge LED-urile



Intrare / Iesire

- Exemplu – Butoane si LED-uri – Scrierea programului

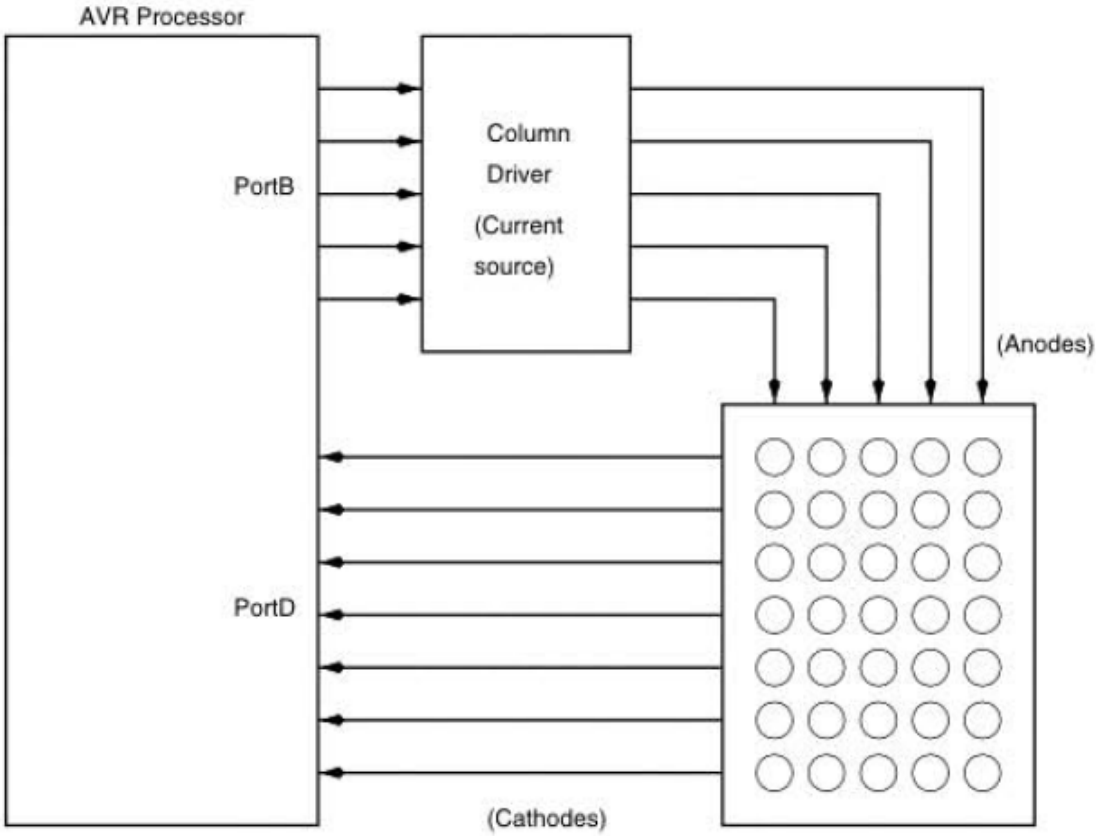
```
ldi r16, 0x00
out DDRD, r16        Directia portului D - intrare
ldi r16, 0xFF
out PORTD, r16       '1' in PORTD – rezistente pull up activate
ldi r16, 0xFF
out DDRB, r16        Directia portului B - iesire
loop:
    in r16, PIND     Citire port D
    out PORTB, r16   Scriere port B
rjmp loop
```

- Atentie!!

```
in r16, PIND         Citeste starea pinilor exteriori, modificata de
                     activitate exterioara
in r16, PORTD        Citeste starea registrului PORTD, setat din interiorul
                     microcontrollerului prin program
```

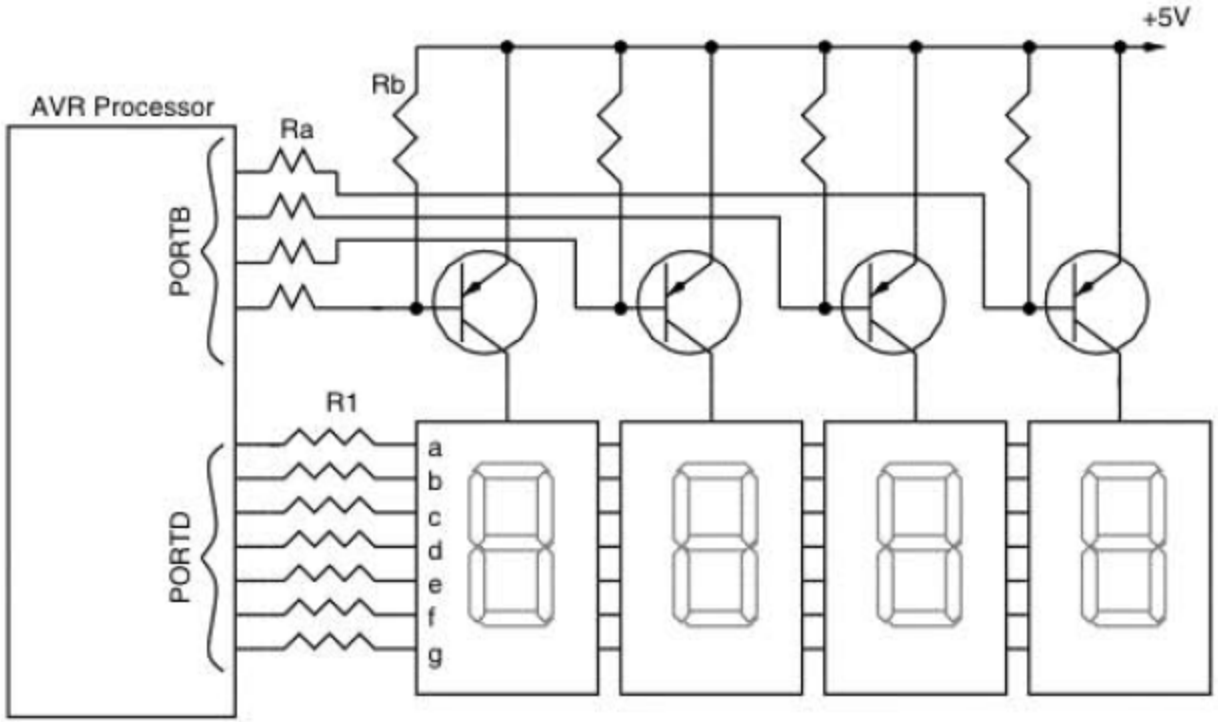
Intrare / Iesire

- Exemplu – Matrice de LED-uri
  - Ambele porturi (D si B) sunt iesire
  - Pentru ca un LED sa se aprinda, anodul trebuie sa fie in ‘1’ si catodul in ‘0’
  - Se poate controla o linie sau o coloana simultan
  - Pentru utilizarea intregii matrici - **baleiere**



Intrare / Iesire

- Exemplu – Bloc 4x7 segmente
  - Fiecare cifra este alcatuita din 7 led-uri, cu anod comun
  - Nivelul ‘1’ pe anod activeaza cifra – una singura activa la un moment dat
  - Valori selective de ‘0’ pe fiecare catod realizeaza modelul cifrei
  - Baleiere pentru utilizarea intregului dispozitiv



Calculul rezistenţei LED-urilor

- Fiecare tip de LED are o cădere tipică de tensiune, V_f (forward voltage)
- Diferenţa de tensiune dintre ieşirea unui pin digital V_{CC} şi V_f este căderea de tensiune pe rezistenţă
- Intensitatea curentului prin pinul digital este:

$$I = \frac{V_R}{R} = \frac{V_{CC} - V_F}{R}$$

- Cu cât intensitatea e mai mare, cu atât LED-ul va fi mai strălucitor.
- *I* trebuie limitat la sub 20 mA, pentru protecţia microcontrollerului (şi a LED-ului).

Typical LED Characteristics			
Semiconductor Material	Wavelength	Colour	V_F @ 20mA
GaAs	850-940nm	Infra-Red	1.2v
GaAsP	630-660nm	Red	1.8v
GaAsP	605-620nm	Amber	2.0v
GaAsP:N	585-595nm	Yellow	2.2v
AlGaP	550-570nm	Green	3.5v
SiC	430-505nm	Blue	3.6v
GaInN	450nm	White	4.0v

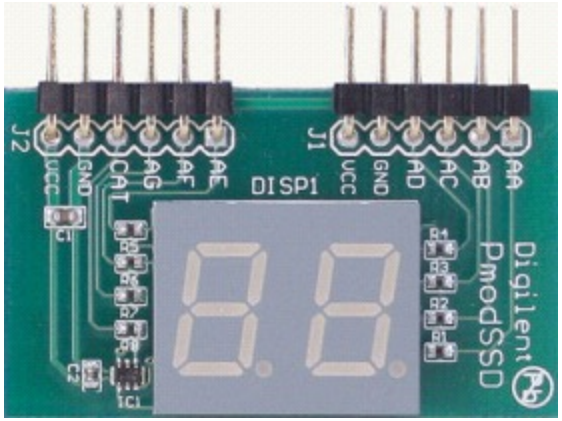
- Exemplu: 1 Led roşu
- V_F = 1.8 V , V_{CC} = 5 V
- **Pentru I = 10 mA**
- R = (5 V – 1.8 V) / 0.01A = 320 Ohm

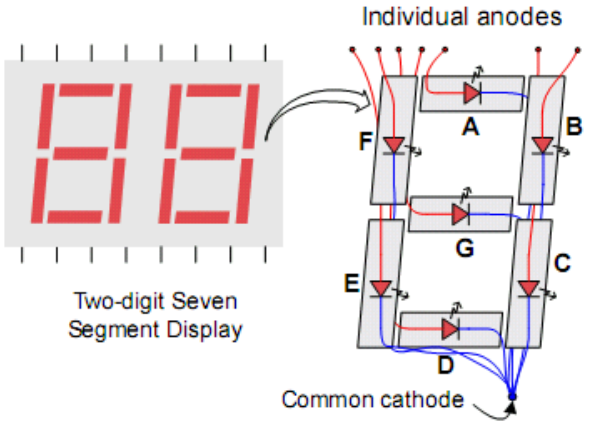
- **Pentru I = 20mA**
- R = (5 V – 1.8 V) / 0.02A = 160 Ohm

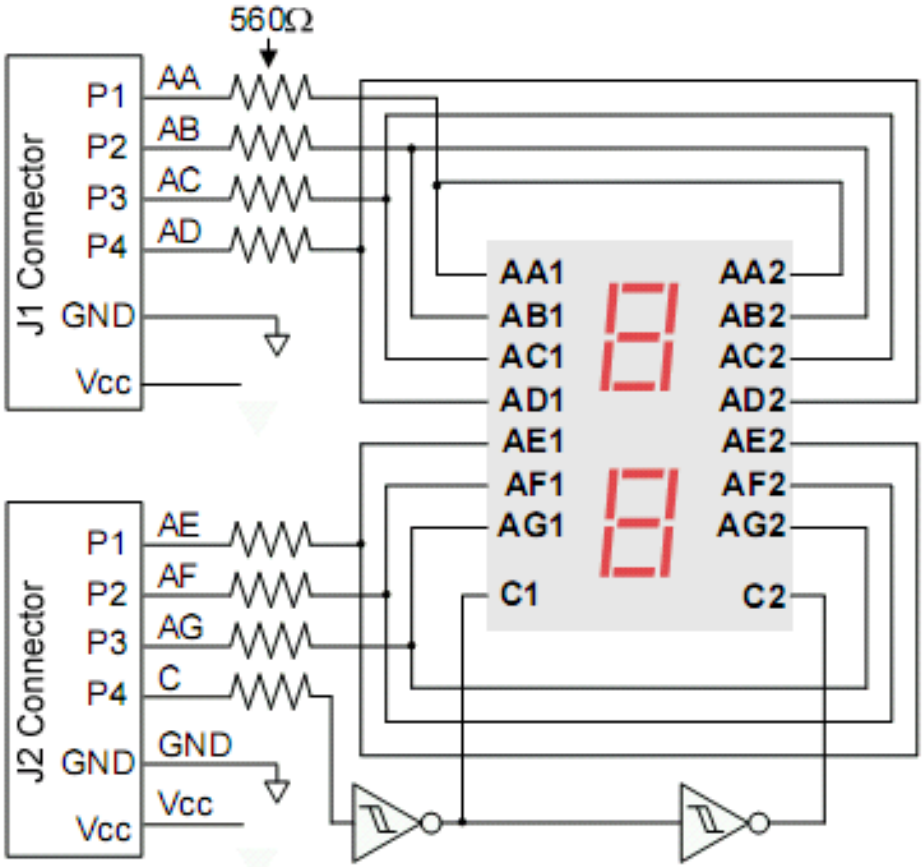
http://www.electronics-tutorials.ws/diode/diode_8.html

Intrare / Iesire

- Exemplu – Digilent PMOD SSD (2x7 segmente)



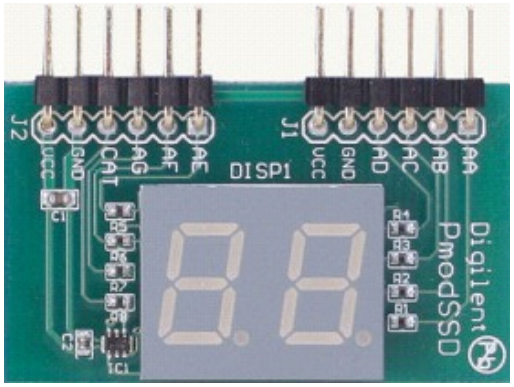




Exemplu

- Exemplu – Afisare pe modul 7 segmente – functia de conversie

```
; conversia pentru afisare pe 7-segmente
; input r16, packed BCD number
; output r17, r18 - sevseg number
convssg:
        push r20
        push r30
        push r31
        in r17, sreg
        push r17
        push r16

        ldi zl, low (sevsegtable*2)
        ldi zh, high (sevsegtable*2)
        ldi r20, 0

        andi r16, 0x0F
        add zl, r16
        adc zh, r20
        lpm r17, Z  ; r17, cifra unitatilor

        ldi zl, low (sevsegtable*2)
        ldi zh, high (sevsegtable*2)
        pop r16     ; reface r16, salveaza din nou
        push r16

        lsr r16
        lsr r16
        lsr r16
        lsr r16

        add zl, r16
        adc zh, r20
        lpm r18, Z  ; r18, cifra zecilor

        pop r16
        pop r20
        out sreg, r20
        pop r31
        pop r30
        pop r20

ret
```




```
sevsegtable: ; tabela pentru activarea led-urilor pentru fiecare cifra
.db     0b00111111, 0b00000110, 0b01011011, 0b01001111, 0b01100110, 0b01101101, 0b01111101,
0b00000111, 0b01111111, 0b01101111
```

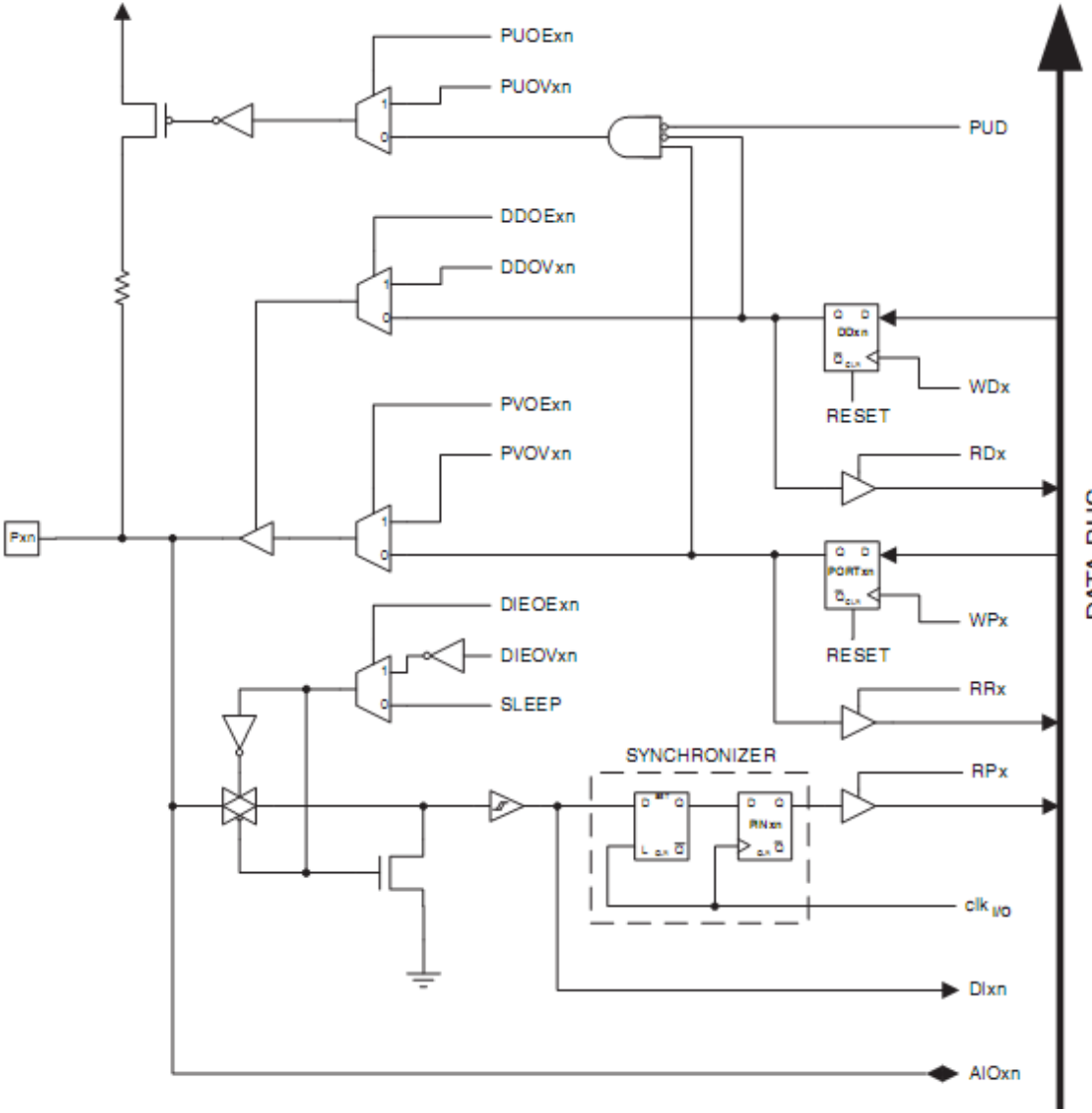
Exemplu

- Exemplu – Afisare pe modul 7 segmente – apelul functiei

```
main:
        ldi r16, 0x35       ; numarul de afisat
        rcall convssg       ; conversie, rezultatul in r17 si r18
        out PORTA, r17      ; SSG este atasat la portul A / conectorul JA
        rcall delay         ; cod de asteptare
        ori r18, 0x80       ; aceasta operatie pune r18(7) pe '1', pentru a activa
                            ; cifra zecilor
        out PORTA, r18
        rcall delay         ; cod de asteptare
rjmp main
```

Intrare / Iesire

- Functii alternative ale pinilor I/O la AVR



PUOExn:	Pxn PULL-UP OVERRIDE ENABLE
PUOVxn:	Pxn PULL-UP OVERRIDE VALUE
DDOExn:	Pxn DATA DIRECTION OVERRIDE ENABLE
DDOVxn:	Pxn DATA DIRECTION OVERRIDE VALUE
PVOExn:	Pxn PORT VALUE OVERRIDE ENABLE
PVOVxn:	Pxn PORT VALUE OVERRIDE VALUE
DIEOExn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE ENABLE
DIEOVxn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE VALUE
SLEEP:	SLEEP CONTROL

PUD:	PULLUP DISABLE
WDx:	WRITE DDRx
RDx:	READ DDRx
RRx:	READ PORTx REGISTER
WPx:	WRITE PORTx
RPx:	READ PORTx PIN
$clk_{I/O}$:	I/O CLOCK
DIxn:	DIGITAL INPUT PIN n ON PORTx
AIOxn:	ANALOG INPUT/OUTPUT PIN n ON PORTx

Intrare / Iesire

- Functii alternative ale pinilor I/O la AVR
  - Exemplu – functiile alternative ale portului **B**

Port Pin	Alternate Functions
PB7	OC2/OC1C[1] (Output Compare and PWM Output for Timer/Counter2 or Output Compare and PWM Output C for Timer/Counter1)
PB6	OC1B (Output Compare and PWM Output B for Timer/Counter1)
PB5	OC1A (Output Compare and PWM Output A for Timer/Counter1)
PB4	OC0 (Output Compare and PWM Output for Timer/Counter0)
PB3	MISO (SPI Bus Master Input/Slave Output)
PB2	MOSI (SPI Bus Master Output/Slave Input)
PB1	SCK (SPI Bus Serial Clock)
PB0	$\overline{SS}$ (SPI Slave Select input)

Generare de semnale prin intermediul timer-elor interne

Semnalele pentru comunicatia SPI – controller intern

Intrare / Iesire

- Functii alternative ale pinilor I/O la AVR
  - Exemplu – functiile alternative ale portului **D**

Port Pin	Alternate Function
PD7	T2 (Timer/Counter2 Clock Input)
PD6	T1 (Timer/Counter1 Clock Input)
PD5	XCK1[1] (USART1 External Clock Input/Output)
PD4	ICP1 (Timer/Counter1 Input Capture Pin)
PD3	INT3/TXD1[1] (External Interrupt3 Input or UART1 Transmit Pin)
PD2	INT2/RXD1[1] (External Interrupt2 Input or UART1 Receive Pin)
PD1	INT1/SDA[1] (External Interrupt1 Input or TWI Serial DAta)
PD0	INT0/SCL[1] (External Interrupt0 Input or TWI Serial CLock)

Intreruperi externe, semnale UART si TWI

Operatii cu stiva

- Stack pointer (16 biti) – indica adresa varfului stivei
- Accesabil prin cele doua jumatati de 8 biti, *SPL* si *SPH*, prin instructiuni de I/O
- Trebuie initializat la inceputul oricarui program care foloseste operatii explicite cu stiva, apeluri de procedura, sau intreruperi
- Trebuie sa fie o adresa din memoria SRAM, mai mare decat 0x60
- Deoarece prin introducerea de date pe stiva SP este decrementat, este bine ca el sa fie initializat cu cea mai mare adresa disponibila din SRAM – *RAMEND*

Bit	15	14	13	12	11	10	9	8	
0x3E (0x5E)	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
0x3D (0x5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	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	0	
	0	0	0	0	0	0	0	0	

- Exemplu initializare SP

```
ldi R16, high(RAMEND)
out SPH, R16
ldi R16, low(RAMEND)
out SPL, R16
```

Valoarea initiala este total nepotrivita pentru utilizare!

Operatii cu stiva

- Instructiuni care opereaza cu stiva

push Rx

Mem(SP) = Rx
SP = SP-1

pop Rx

SP = SP+1
Rx = Mem(SP)

rcall adresa

Mem(SP:SP-1) = PC+1 adresa instructiunii urmatoare, 16 biti
SP = SP-2
PC = adresa* *de fapt se modifica PC cu un offset relativ la pozitia curenta

ret

SP = SP+2
PC = Mem (SP:SP-1)

Intreruperi

- Mecanismul de intreruperi permite microcontrolerului sa raspunda la evenimente externe, sau la evenimente produse de perifericele integrate pe chip.
- In lipsa evenimentelor, procesorul poate executa programul principal, sau poate intra in stare de inactivitate (sleep) pentru a conserva energie.
- Tratarea intreruperilor este activata sau dezactivata prin bit-ul 7 din registrul SREG

Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	**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	

- Instructiuni

  SEI – activeaza sistemul de intreruperi (SREG(7) = 1)

  CLI – dezactiveaza sistemul de intreruperi (SREG(7)=0)

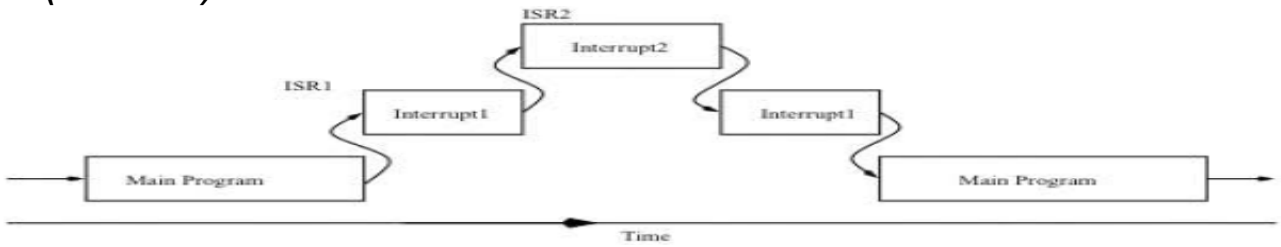
Intreruperi

- Tratarea unei intreruperi

  1. Dispozitivul periferic genereaza cererea de intrerupere
  2. Se finalizeaza executia instructiunii curente
  3. PC se salveaza pe stiva
     TOS = PC
     SP = SP – 2
  4. Accesarea vectorului specific tipului de intrerupere
  5. Executia saltului la Procedura de Tratare a Intreruperii (*Interrupt Service Routine, ISR*)
  6. Blocare flag intreruperi (**CLI**)
  7. Executie ISR
  8. Revenire din ISR (**reti**)
     SP = SP + 2
     PC = TOS
     Activare flag intreruperi (**SEI**)

reti este echivalent cu **sei + ret**

Daca in ISR se activeaza intreruperile prin apel SEI, se pot executa intreruperi incuibate (nested).



Intreruperi

- **Sursele de întreruperi sunt prezentate în tabela Vectorilor de Întrerupere**
- Întreruperile pot fi tratate doar după finalizarea execuţiei instrucţiunii curente
- Timp de răspuns: > = 4 .. 5 cycles

  PC (2/3 bytes) salvat pe stivă (push)

  Stack Pointer ← Stack Pointer – 2/3;

  Salt la adresa specificată de tabela de vectori

  Sistemul de întreruperi este blocat, bitul I, SREG(7) ← 0
- După 4..5 cicli începe execuţia procedurii ISR (cazul optim)
- Dacă sistemul de întreruperi este folosit pentru revenirea din modul “sleep”, timpul de răspuns este crescut cu 4..5 cicli
- Revenirea din ISR (RETI): 4..5 cicli

  PC ← PC salvat (scos din stivă)

  Stack Pointer ← Stack Pointer + 2/3;

  Sistemul de întreruperi este re-activat, : bitul I, SREG(7) ← 1
- Bitul I poate fi activat sau de-activat direct prin instrucţiunile SEI & CLI
- **Prioritate:** scade pe măsură ce numărul întreruperii creşte
- **Prioritate maximă : Reset**

Intreruperi

- Surse si vectori de intrerupere (1)

Adrese absolute in memoria program (flash)

Vector No.	Program Address(2)	Source	Interrupt Definition
1	0x0000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 1
4	0x0006	INT2	External Interrupt Request 2
5	0x0008	INT3	External Interrupt Request 3
6	0x000A	INT4	External Interrupt Request 4
7	0x000C	INT5	External Interrupt Request 5
8	0x000E	INT6	External Interrupt Request 6
9	0x0010	INT7	External Interrupt Request 7
10	0x0012	TIMER2 COMP	Timer/Counter2 Compare Match
11	0x0014	TIMER2 OVF	Timer/Counter2 Overflow
12	0x0016	TIMER1 CAPT	Timer/Counter1 Capture Event
13	0x0018	TIMER1 COMPA	Timer/Counter1 Compare Match A
14	0x001A	TIMER1 COMPB	Timer/Counter1 Compare Match B
15	0x001C	TIMER1 OVF	Timer/Counter1 Overflow
16	0x001E	TIMER0 COMP	Timer/Counter0 Compare Match
17	0x0020	TIMER0 OVF	Timer/Counter0 Overflow
18	0x0022	SPI, STC	SPI Serial Transfer Complete

Intreruperi

- Surse si vectori de intrerupere (2)

Vector No.	Program Address(2)	Source	Interrupt Definition
19	0x0024	USART0, RX	USART0, Rx Complete
20	0x0026	USART0, UDRE	USART0 Data Register Empty
21	0x0028	USART0, TX	USART0, Tx Complete
22	0x002A	ADC	ADC Conversion Complete
23	0x002C	EE READY	EEPROM Ready
24	0x002E	ANALOG COMP	Analog Comparator
25	0x0030(3)	TIMER1 COMPC	Timer/Countre1 Compare Match C
26	0x0032(3)	TIMER3 CAPT	Timer/Counter3 Capture Event
27	0x0034(3)	TIMER3 COMPA	Timer/Counter3 Compare Match A
28	0x0036(3)	TIMER3 COMPB	Timer/Counter3 Compare Match B
29	0x0038(3)	TIMER3 COMPC	Timer/Counter3 Compare Match C
30	0x003A(3)	TIMER3 OVF	Timer/Counter3 Overflow
31	0x003C(3)	USART1, RX	USART1, Rx Complete
32	0x003E(3)	USART1, UDRE	USART1 Data Register Empty
33	0x0040(3)	USART1, TX	USART1, Tx Complete
34	0x0042(3)	TWI	Two-wire Serial Interface
35	0x0044(3)	SPM READY	Store Program Memory Ready

Intreruperi

- Intreruperi externe – cauzate de activitate pe pinii externi INT7…INT0
- Pinii INT7:INT0 sunt comuni cu pinii porturilor **D** si **E** – daca porturile sunt configurate ca iesire, se pot declansa intreruperi software prin scrierea acestor porturi.
- Configurarea modului de sesizare a intreruperilor externe – registrii **EICRA** si **EICRB** – in total 16 biti, 2 biti / intrerupere

Bit	7	6	5	4	3	2	1	0	
(0x6A)	ISC31	ISC30	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00	EICRA
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	

Bit	7	6	5	4	3	2	1	0	
0x3A (0x5A)	ISC71	ISC70	ISC61	ISC60	ISC51	ISC50	ISC41	ISC40	EICRB
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	

ISCn1	ISCn0	
0	0	Nivel ‘0’ pe INTn genereaza cerere de intrerupere
0	1	Schimbarea nivelului pinului generează cerere de întrerupere
1	0	Front descrescator pe INTn genereaza cerere de intrerupere
1	1	Front crescator pe INTn genereaza cerere de intrerupere

Intreruperi

- Scrierea/citirea registrilor de control a intreruperilor externe
  EICRB se poate scrie/citi cu **in**, **out**
  EICRA se poate scrie/citi cu **lds**, **sts**

- Activarea punctuala a intreruperilor externe – folosirea registrului **EIMSK**
- Fiecare intrerupere este controlata de un bit al acestui registru
- Setarea la ‘1’ a bitului corespunzator activeaza intreruperea
- Registrul **EIMSK** se poate citi/scrie cu **in**, **out**

Bit	7	6	5	4	3	2	1	0	
0x39 (0x59)	**INT7**	**INT6**	**INT5**	**INT4**	**INT3**	**INT2**	**INT1**	**INT0**	**EIMSK**
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	

Intreruperi

- Exemplu – incrementarea unui numarator prin apasarea unui buton, folosind mecanismul intreruperilor externe
  - Butonul este conectat la intreruperea externa INT0

```
.org 0x0000                     ; adresa vectorului pentru intreruperea reset
        rjmp main
.org 0x0002                     ; adresa vectorului pentru intreruperea externa INT0
        rjmp isr_INT0

main:
        ldi r16, high(RAMEND)   ; initializare stiva – necesara cand folosim intreruperi!
        out SPH, R16
        ldi r16, low(RAMEND)
        out SPL, R16

        ldi r16, 0b00000011     ; configurare mod de tratare INT0 – front crescator
        sts EICRA, r16
        ldi r16, 0b00000001     ; activare intrerupere INT0
        out EIMSK, r16

        ldi r16, 0xFF           ; setam directia portului E – iesire pentru numarator
        out DDRE, r16

        ldi r17,0               ; valoarea initiala a numaratorului
        sei                     ; activare globala a sistemului de intreruperi
```

Intreruperi

- Exemplu – incrementarea unui numarator prin apasarea unui buton, folosind mecanismul intreruperilor externe - continuare

```
loop:
        out PORTE, r17          ; scriem numaratorul pe portul E (LED-uri)
rjmp loop

isr_INT0:                       ; inceputul rutinei de tratare a intreruperii INT0
        inc r17                 ; doar incrementam numaratorul
reti                            ; revenire din intrerupere
```

- Exercitiu: cum se rezolva problema incrementarii unui numarator prin apasarea unui buton, fara folosirea sistemului de intreruperi?
- Exercitiu 2: Modificaţi exemplul anterior pentru a utiliza două butoane, conectate la INT1 şi INT2. INT1 ar trebui să incrementeze contorul cu 1, iar INT2 să decrementeze cu 1.
- Exercitiu 3: Puteţi să rezolvaţi problema numărării apăsării butoanelor, fără a cunoaşte dinainte ce fel de front este generat de buton? (Crescător / descrescător)