

# Wykład 14

---

## Mikrokomputery jedoukładowe rodziny AVR Część 2 Wbudowane układy we-wy

**B**artłomiej Zieliński, PhD, DSc

# AVR (2)

---

Program:

(poprzednio)

- Rodziny i struktura AVR
- Porty we-wy

(dziś)

- Wbudowane układy we-wy
  - Układ czasowo-licznikowy
  - Układy analogowe
  - Port szeregowy
  - Układ przerwań
- Rozkazy

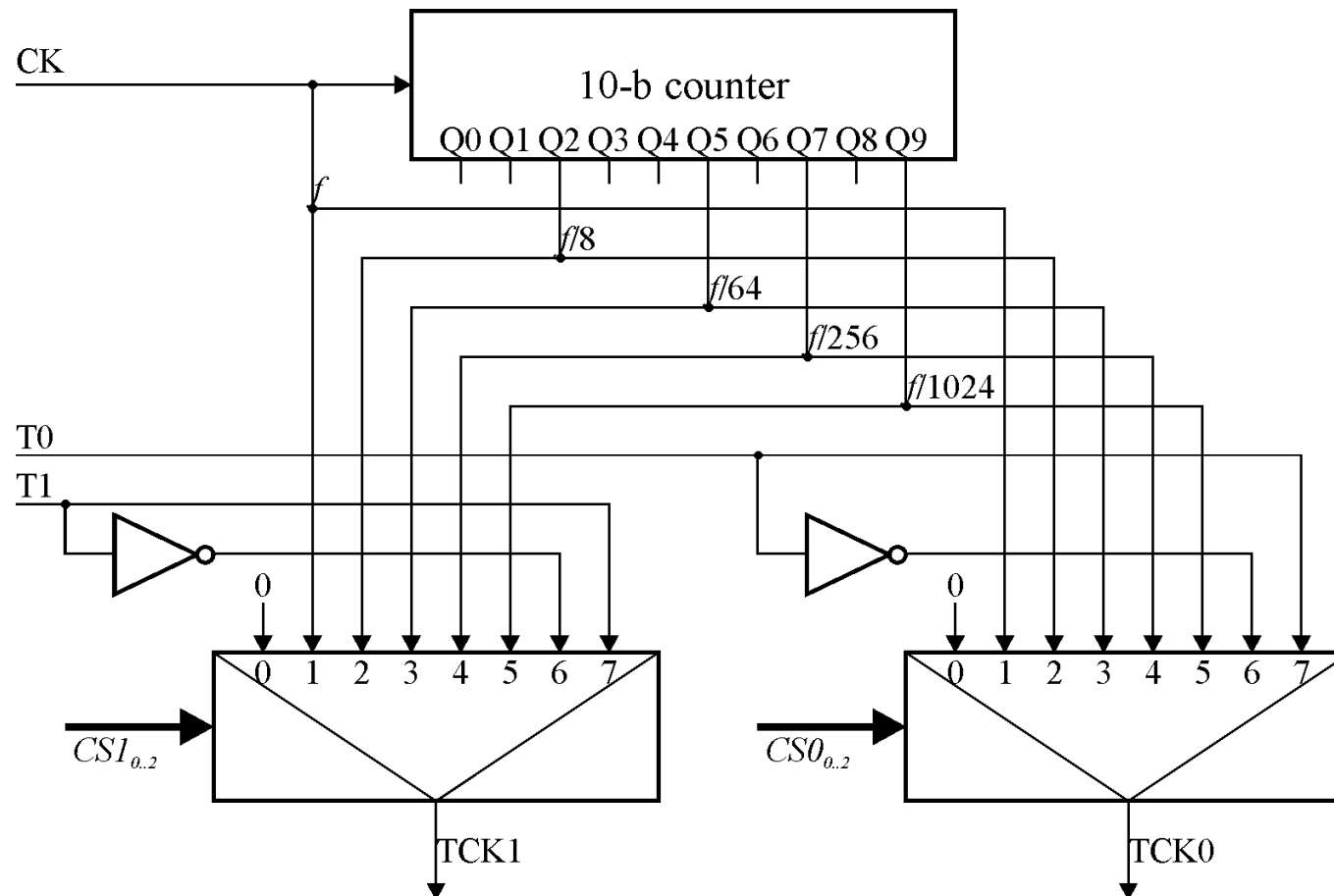
# AVR (2)

---

- Układ czasowo-licznikowy
  - TC0 (8-b), TC1 (16-b)
  - Wspólny preskaler (10-b)
  - Rejestry:
    - TIMSK – maska przerwań (wspólny dla TC1 i TC2)
    - TIFR – znaczniki przerwań (wspólne dla TC1 i TC2)
    - TCNT0 – rejestr roboczy TC0
    - TCNT1 – rejestr roboczy TC1
    - ICR1 – przechwytywanie TC1 (?)
    - TCCR1A, TCCR1B – rejestr sterujący TC1

# AVR (2)

- Układ czasowo-licznikowy
  - Wspólny preskaler dla TC0 i TC1



# AVR (2)

---

- Układ czasowo-licznikowy

- TCO

- Wejście T0 synchronizowane z zegarem

- $f_{TC0} \leq \frac{f_{XTAL}}{2}$

- T0 – wejście niezależnie od kierunku wyprowadzenia

- Programowe sterowanie procesem zliczania

- Zliczanie w górę

- 0 = przepiętnienie

- Odczyt i zapis TCNT0

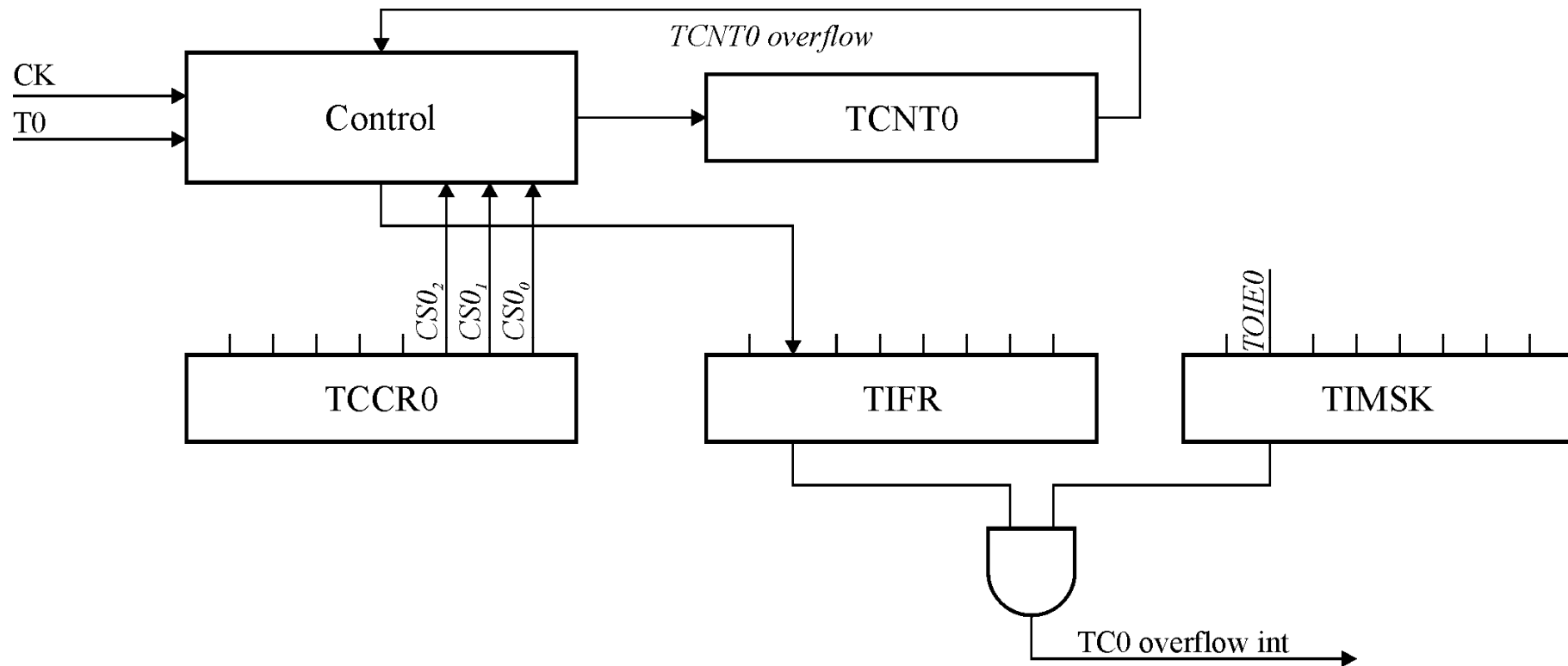
- Zapis podczas impulsu TCO:

- » 1. Inkrementacja TCNT0

- » 2. Zapis nowej wartości

# AVR (2)

- Układ czasowo-licznikowy  
– TC0



# AVR (2)

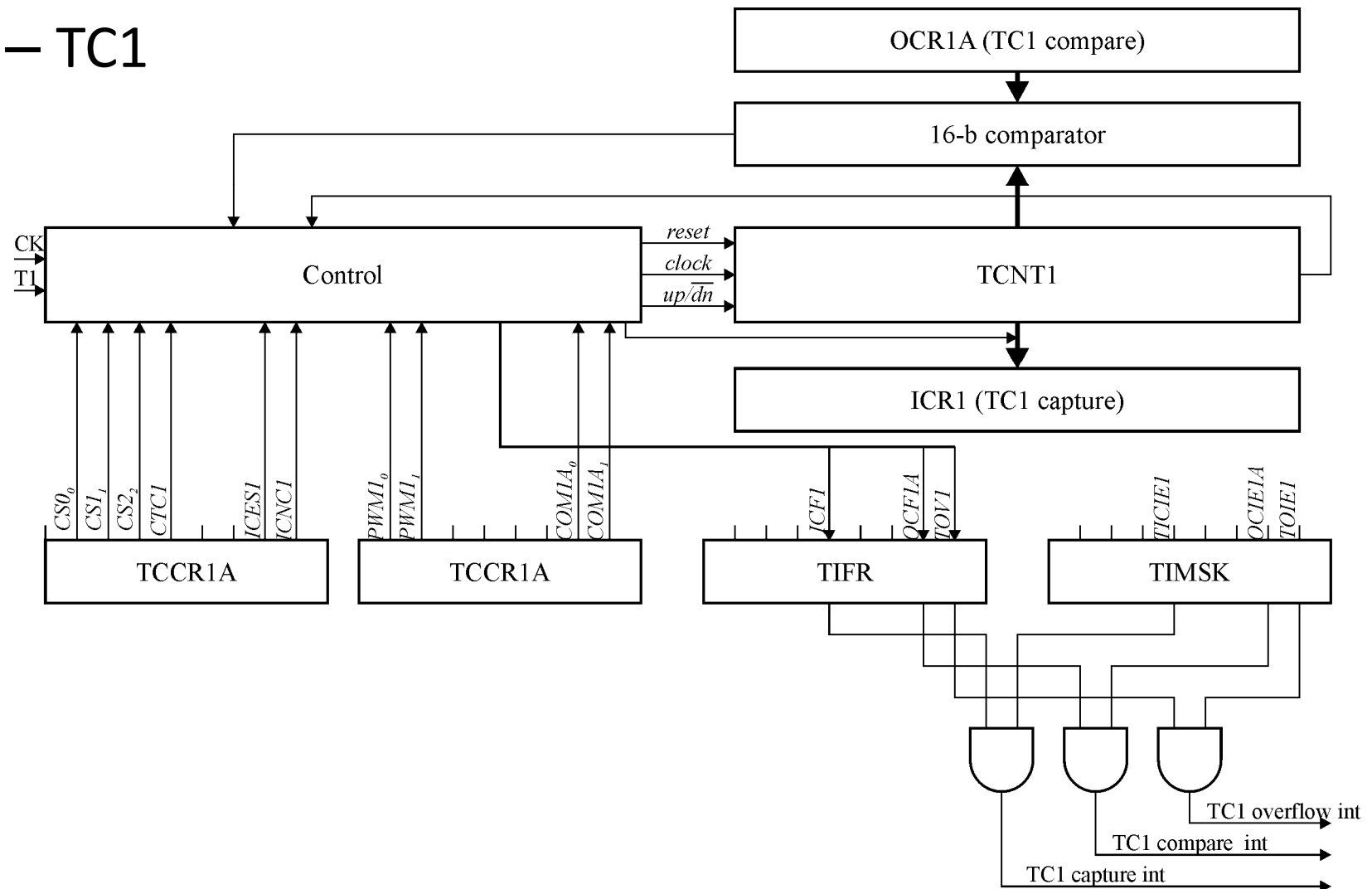
---

- Układ czasowo-licznikowy
  - TC1 (w AT90S2313)
    - Wejście T0 synchronizowane z zegarem
      - $f_{TC0} \leq \frac{f_{XTAL}}{2}$
    - Tryby pracy:
      - Czasomierz
      - Licznik
      - Wyjście porównania
      - Wejście przechwytywania
      - PWM

# AVR (2)

- Układ czasowo-licznikowy

– TC1





# AVR (2)

---

- Układ czasowo-licznikowy
  - Rejestry TC1
    - TCCR1A
      - COM1A1, COM1A0
        - » Wyjście OC1 odłączone od układu TC1
        - » Zgodność porównania → zmiana OC1
        - » Zgodność porównania → OC1=0
        - » Zgodność porównania → OC1=1
      - PWM11, PWM10
        - » PWM wyłączony dla TC1
        - » TC1 jako 8-b PWM
        - » TC1 jako 9-b PWM
        - » TC1 jako 10-b PWM

# AVR (2)

---

- Układ czasowo-licznikowy
  - Rejestry TC1
    - TCCR1B
      - ICNC1 – usuwanie zakłóceń dla TC1
        - » Off → wyzwolenie przez 1. aktywne zboczu wejścia ICP
        - » On → wyzwolenie, gdy 4 kolejne próbki ICP są równe
        - » Użyteczne przy pracy z komparatorem analogowym
      - ICES1 – TCNT1 → ICR1 przy narast.(1)/opad.(0) zboczu ICP
      - CTC1 – 1 → TCNT1=0 po spełnieniu warunków porównania
        - » Zależy od preskalera
        - » Bez znaczenia w trybie PCM
      - CS1<sub>2..0</sub> – wybór źródła taktowania

# AVR (2)

---

- Układ czasowo-licznikowy

- Rejestry TC1

- TCNT1H, TCNT1L (+ TEMP)

- Zapis:

- » Zapis TCNT1H → zapis TEMP

- » Zapis TCNT1L → TCNT1H=TEMP

- » Kolejność zapisu: TCNT1H, TCNT1L

- Odczyt:

- » Odczyt TCNT1L → TEMP=TCNT1H

- » Odczyt TCNT1H → odczyt TEMP

- » Kolejność odczytu: TCNT1L, TCNT1H

- Jeśli TEMP jest używany podczas przerw

- » Zamaskować przerwanie podczas odczytu/zapisu

# AVR (2)

---

- Układ czasowo-licznikowy
  - TC1 – tryb porównania
    - Ciągłe sprawdzanie, czy  $OCR1A=TCNT1$
    - Jeśli równe, akcja na wyjściu OC1 wg ustaleń w TCCR1A
      - $TIFR.OCF1A=1$  (znacznik przewania)
      - $CTC1=1 \rightarrow TCNT1=0$
      - OC1 musi być ustawione jako wyjście!
    - Odczyt/zapis OCR1A jako TCCR1
  - TC1 – tryb przechwytywania
    - Aktywne zbocze ICP ( $TCCR1B.ICES1$ )  $\rightarrow ICR1A=TCNT1$ 
      - $ICF1=1$
    - Odczyt/zapis ICR1A jako TCCR1

# AVR (2)

---

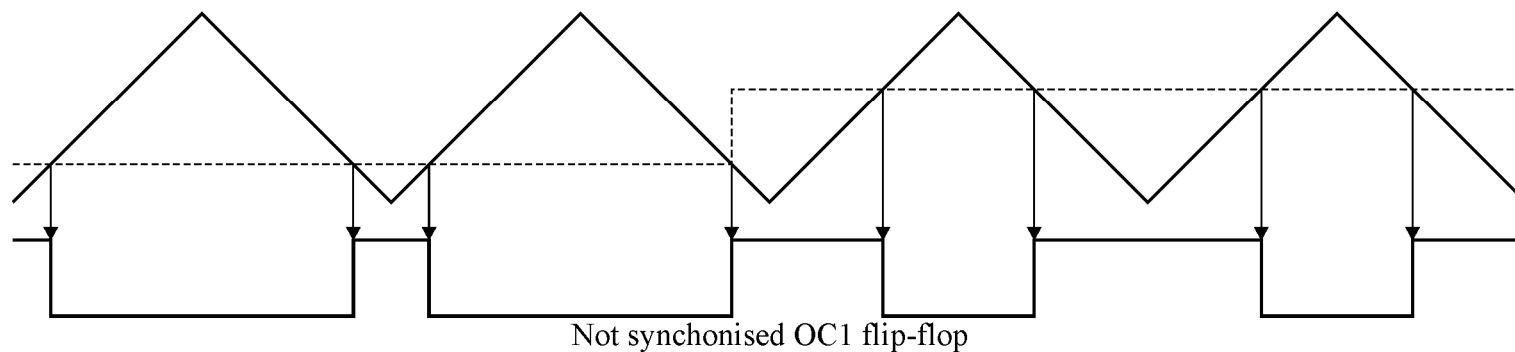
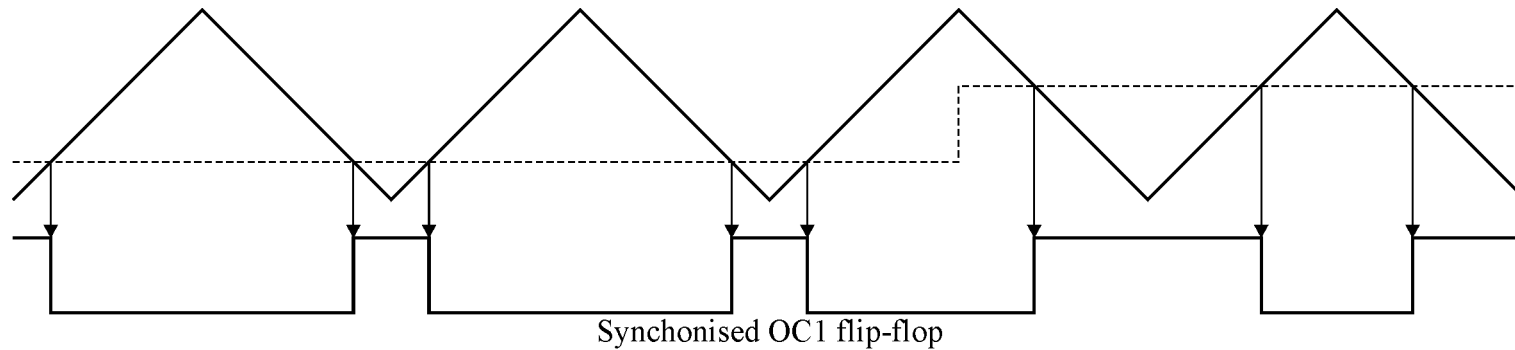
- Układ czasowo-licznikowy
  - TC1 – tryb PWM
    - Wykorzystuje rejestr OCR1
    - OC1=wyjście modulatora
    - Zmiana stopnia wypełniania bez zakłóceń
    - Cykl TC1:  $0 \rightarrow \text{max}$ ,  $\text{max} \rightarrow 0$ 
      - Gdy  $\text{TCNT1} = \text{OCR1A}$ , OC1 ustawione/zerowane
        - » Wg ustawień  $\text{TCCR1A.COM1A}_{1..0}$  – PWM/odwrotny PWM

Rozdzielczość PWM	Wartość max.	Częstotliwość PWM
8 bits	\$00FF (255)	$f_{\text{TC1}}/510$
9 bits	\$01FF (511)	$f_{\text{TC1}}/1022$
10 bits	\$03FF (1023)	$f_{\text{TC1}}/2046$

# AVR (2)

---

- Układ czasowo-licznikowy
  - TC1 – tryb PWM
    - OCR1A efektywny, gdy TCNT1=max
      - Brak zakłóceń



# AVR (2)

---

- Układ czasowo-licznikowy
  - TC1 – tryb PWM
    - Jeśli OCR1A=max
      - Preskaler nie używany → brak wyjścia PWM
        - » Porównane wartości dla zliczania „w górę” i „w dół” występują jednocześnie
      - Preskaler używany → pojedynczy impuls
        - » PWM=1, gdy osiągnięto wartość max.
        - » Warunek nie badany podczas zliczania „w dół”
    - Przerwania w trybie PWM
      - Mogą działać normalnie (TIMSK.TOIE1=1, SREG.I=1)
      - TOV=1, gdy TCNT1=\$0000 podczas zliczania „w górę”
        - » Za wyjątkiem 1. cyklu po włączeniu

# AVR (2)

---

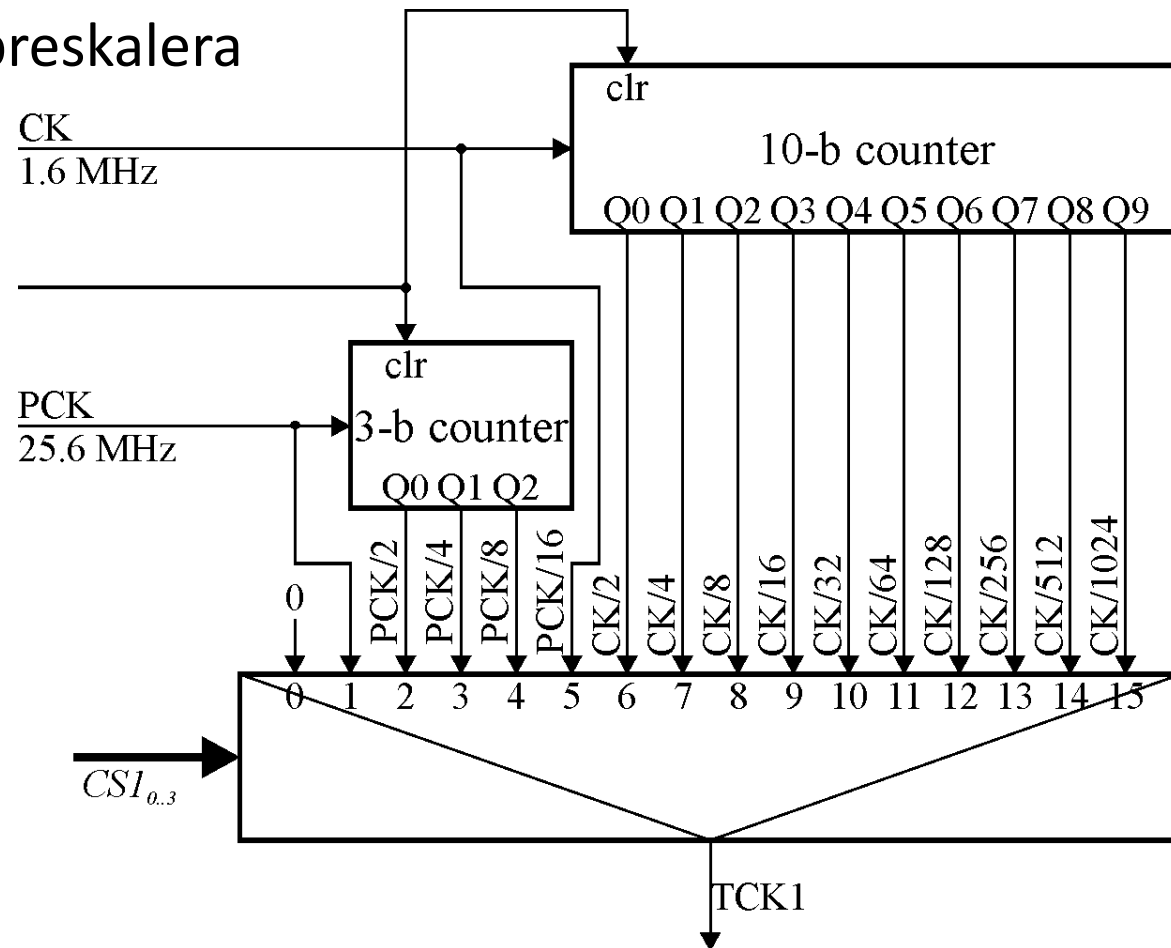
- Układ czasowo-licznikowy
  - TC1 w ATtiny15
    - Inny preskaler
    - Inna struktura
    - CK=wewnętrzny oscylator,  $F \approx 1.6$  MHz
    - PCK=mnożnik częstotliwości PLL,  $F \approx 25.6$  MHz
      - Stabilność częstotliwości zależy od kalibracji CK
      - TC1 nie działa z PCK, gdy  $F_{CK} > 1.75$  MHz
    - Oba preskaler można wyzerować (SFOR.PSR1)
    - Dwa rejestry OCR1
      - Sterowanie stopnia wypełnienia
      - Sterowanie częstotliwością



# AVR (2)

- Układ czasowo-licznikowy
  - TC1 w ATtiny15

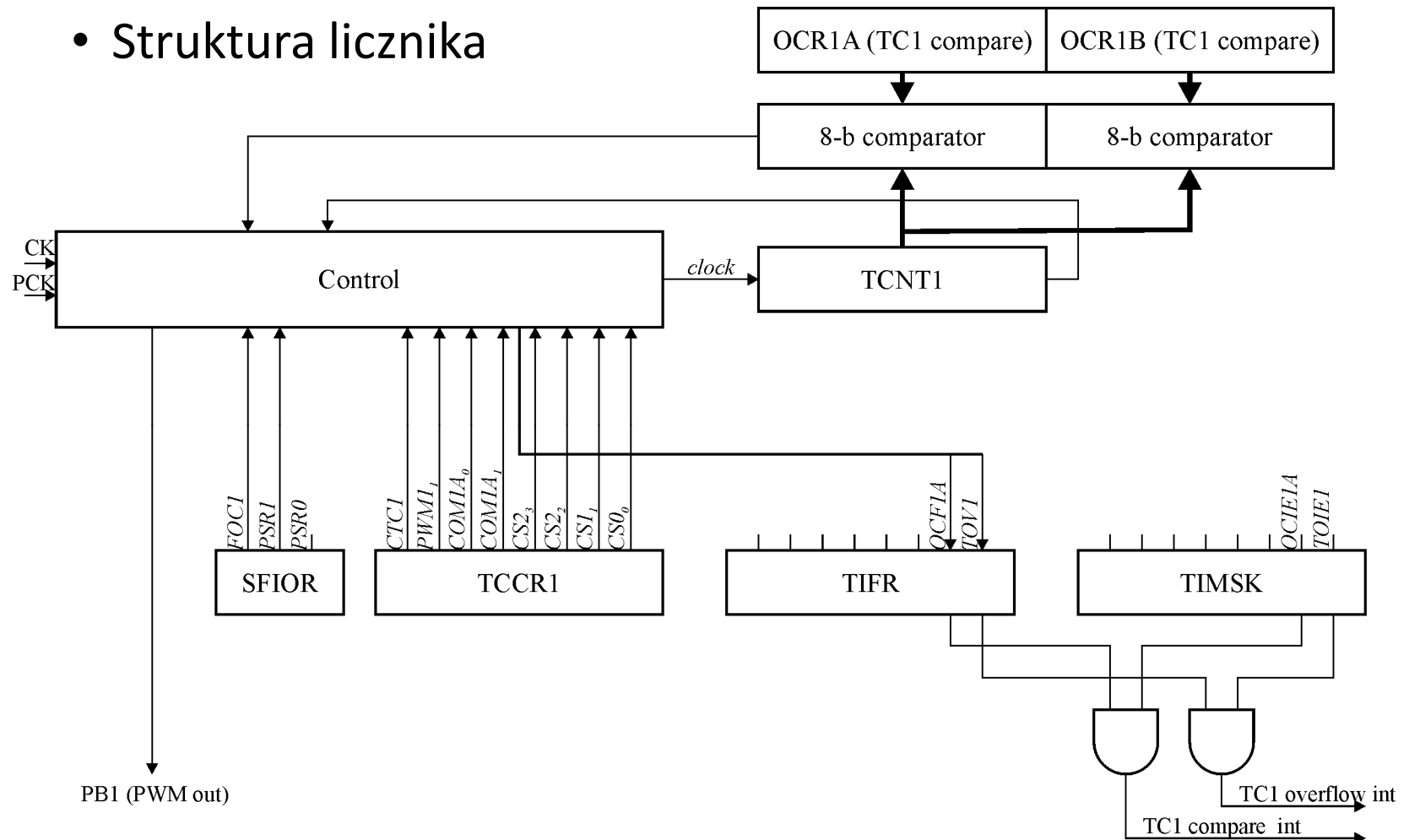
- Struktura preskalera



# AVR (2)

- Układ czasowo-licznikowy
  - TC1 w ATtiny15

- Struktura licznika



# AVR (2)

---

- UART
  - Wbudowany generator prędkości transmisji
  - Wysokie prędkości dla niskich częstotliwości zegara
  - 8/9-b słowo danych
  - Usuwanie zakłóceń
  - Wykrywanie przepełnienia bufora odbiorczego
  - Wykrywanie błędu ramki
  - Wykrywanie błędu bitu startu
  - 3 przerwania:
    - Koniec transmisji
    - Bufor nadajnika pusty
    - Bufor odbiornika pełny

# AVR (2)

---

- UART

- Rejestry

- UDR (*UART data register*)

- Odrębne rejestry nadajnika i odbiornika

- USR (*UART status register*)

- RxC – koniec odbioru (gotowość odbiornika)

- TxC – koniec nadawania (gotowość nadajnika)

- UDRE – UDR pusty

- FE – błąd ramki (np. bit stopu=0)

- OR – przepełnienie (nadpisanie bufora odbiornika)

# AVR (2)

---

- UART

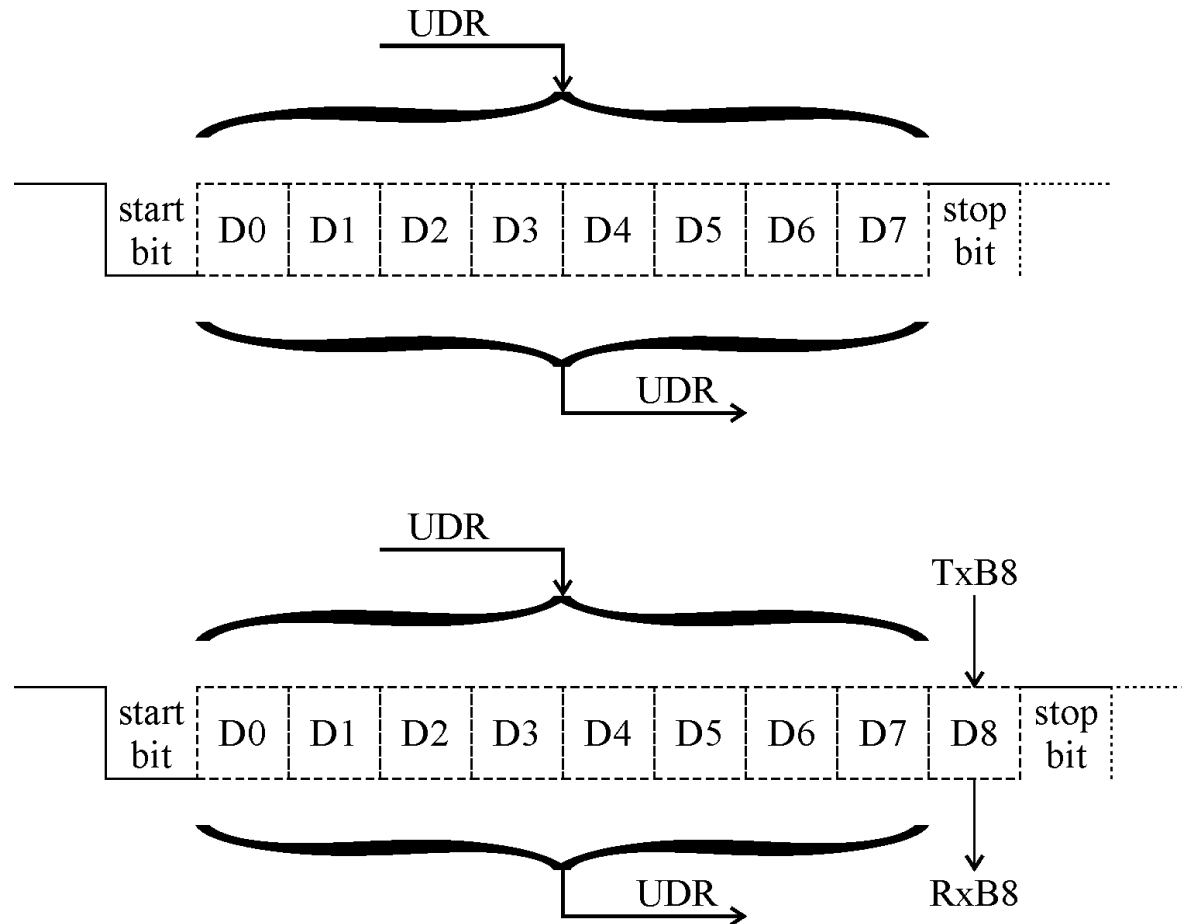
- Rejestry

- UCR (*UART control register*)

- RxCIE – wł/wył. przerwania RxC
      - TxCIE – wł/wył. przerwania TxC
      - UDRIE – wł/wył. przerwania UDRE
      - RxEN – wł/wył. odbiornika
      - TxEN – wł/wył. nadajnika
      - CHR9 – 8/9-b długość słowa danych
      - RxB8 – odebrany bit 8
      - TxB8 – nadawany bit 8

# AVR (2)

- UART
  - Format danych



# AVR (2)

---

- UART

- Rejestry

- UBRR (*UART Baud Rate Register*)

- $R = \frac{f_{CK}}{16(UBRR+1)}$

- Zalecane częstotliwości zegara

- 11.059, 7.3728, (4.608), 3.6864, 1.8432 MHz

- » Najpopularniejsze prędkości transmisji wyznaczone dokładnie

# AVR (2)

---

- UART
  - Usuwanie zakłóceń w odbiorniku
    - 16 próbek na bit
    - Bit startu
      - RxD=1→0 → bit startu
      - Porównanie próbki 8, 9 i 10
      - Jeśli  $\geq 2$  próbki równe „1”, bit startu unieważniony
    - Pozostałe bity
      - Porównanie próbki 8, 9 i 10
      - Wartość bitu = większość próbek
        - » Niekoniecznie kolejne próbki równe
        - » Np. 101 → 1



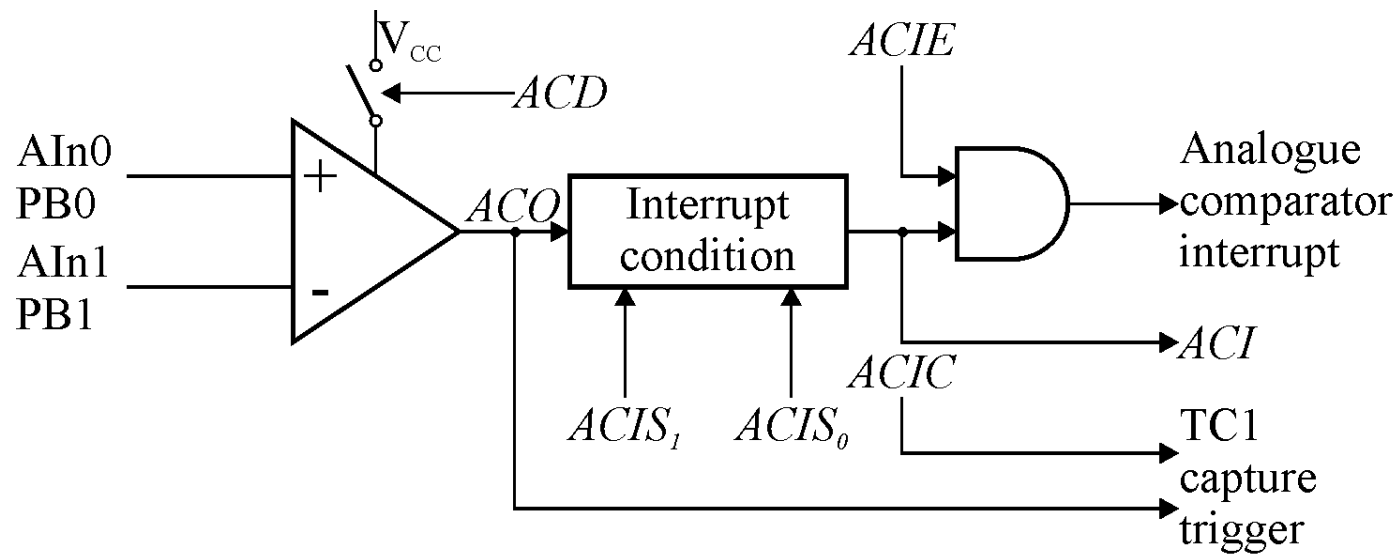
# AVR (2)

---

- Komparator analogowy
  - Znacznie tańszy niż pełny przetwornik A/C
    - Części analogowe i cyfrowe zakłócają się wzajemnie
    - Złożony projekt układu cyfrowego
  - W AT90S2313
    - $ACO=1$ , gdy  $U_{AIN0} > U_{AIN1}$
    - ACO może wyzwolić przechwytywanie TC1
    - Może zgłosić przerwanie
      - $ACO=1$ ,  $ACO=0$ ,  $ACO=NOT ACO$
    - Może być wyłączony programowo

# AVR (2)

- Komparator analogowy



# AVR (2)

---

- Komparator analogowy
  - ACSR – rejestr sterujący/stanu
    - ACD – 1 → wył. komparatora
      - ACSR.ACIE=0, by nie było fałszywego zgłoszenia przerwania
    - ACO – wyjście komparatora (kopia wyjścia ACO)
    - ACI – znacznik przerwania komparatora analogowego
    - ACIE – wł/wył. przerwania komparatora analogowego
    - ACIC – przechwycenie TC1 wyzwolone przez komparator
    - ACIS<sub>1..0</sub> – warunki przerwania
      - ACO=NOT ACO
      - Opadające zbocze ACO
      - Narastające zbocze ACO

# AVR (2)

---

- Przerwania
  - 10 źródeł
    - Zewnętrzne
      - INT1, INT0
    - Wewnętrzne – układ czasowo-licznikowy
      - Przepiętnienie TC0
      - Przechwycenie TC1
      - Przepiętnienie TC1
      - Porównanie TC1
    - Wewnętrzne – UART
      - RxC
      - TxC
      - UDRE
    - Wewnętrzne – komparator analogowy

# AVR (2)

---

- Przerwania
  - Przerwania zewnętrzne
    - Znaczniki niedostępne programowo
      - Ustawiane automatycznie
      - Zerowane, gdy zaczyna się procedura obsługi przerwania
    - Programowe wywołanie przerwania
      - PD2 (INT0), PD3 (INT1) ustawione jako wyjścia

# AVR (2)

---

- Przerwania
  - Rejestry – przerwania zewnętrzne
    - SREG.I
      - Global interrupt enable
    - GIMSK (*General Interrupt Mask Register*)
      - INT1 – wł/wył przerwania Int 1
      - INT0 – wł/wył przerwania Int 0
    - GIFR (*General Interrupt Flag Register*)
      - INTF1 – znacznik przerwania Int 1
      - INTF0 – znacznik przerwania Int 0
    - MCUCR
      - Jak zgłasza się przerwania zewnętrzne INT0, INT1?
        - » Poziom niski/zbocze narastające/zbocze opadające

# AVR (2)

---

- Przerwania

- rejestry – układ czasowo-licznikowy

- TIMSK (*T/C Interrupt Mask Register*)

- TOIE1 – *TC1 Overflow Int Enable*

- OCIE1A – *TC1 Output Compare Match Int Enable*

- TICIE1 – *TC1 Input Capture Int Enable*

- TOIE0 – *TC0 Overflow Int Enable*

- TIFR (*T/C Interrupt Flag Register*)

- TOV1 – *TC1 Overflow Flag*

- OCF1A – *TC1 Output Compare Flag*

- ICF1 – *TC1 Input Capture Flag*

- TOV0 – *TC0 Overflow Flag*

# AVR (2)

---

- Przerwania

- Rejestry – UART

- *USR (UART status register)*

- RxC – zakończono odbiór (gotowość odbiornika)
      - TxC – zakończono nadawanie (gotowość nadajnika)
      - UDRE – UDR pusty

- *UCR (UART control register)*

- RxCIE – wł/wył przerwania RxC
      - TxCIE – Wł/wył przerwania TxC
      - UDRIE – wł/wył przerwania UDRE

- Rejestry – komparator analogowy

- ACSR

- ACI – znacznik przerwania komparatora analogowego
      - ACIE – wł/wył przerwania komparatora analogowego



# AVR (2)

---

- Przerwania
  - Przyjęcie przerwania (zwykle 8 cykli zegara)
    - Ustawienie znaczników (i potem 4 cykle zegara)
    - SREG.I=0
    - Adres powrotu → stos (2 cykle zegara)
    - Wywołanie procedury obsługi (2 cykle zegara)
    - Zerowanie znaczników przerw
  - Wielopoziomowy system przerw
    - SREG.I=1 w procedurze obsługi przerwania
  - RETI (zwykle 4 cykle zegara)
    - Pobranie adresu powrotu ze stosu (2 takty zegara)
    - SREG.I=1

# AVR (2)

---

- Przerwania
  - Wektory przerwań

Numer	Adres	Źródło
1	\$0000	Zerowanie
2	\$0001	INT0
3	\$0002	INT1
4	\$0003	Przechwycenie TC1
5	\$0004	Porównanie TC1
6	\$0005	Przepetnienie TC1
7	\$0006	Przepetnienie TC0
8	\$0007	UART Rx
9	\$0008	UART UDRE
10	\$0009	UART Tx
11	\$000A	Komparator analogowy

# AVR (2)

---

- Przerwania
  - Wektory przerwań – przykładowy fragment programu

```
$0000    rjmp  Reset
$0001    rjmp  ExtInt0
$0002    rjmp  ExtInt1
$0003    rjmp  TC1_Capt
$0004    rjmp  TC1_Comp
$0005    rjmp  TC1_Ovl
```

# AVR (2)

---

- Tryby energooszczędne
  - Tryb jałowy (*Idle*)
    - Zatrzymanie CPU
    - Układy we-wy pracują
    - Przerwanie → wybudzenie, CPU rozpoczyna pracę
  - Tryb obniżonego poboru mocy (*Power Down*)
    - Zatrzymanie CPU
    - Zatrzymanie układów we-wy
    - Układ nadzorujący i przerwania zewnętrzne działają (jeśli włączone)
    - Zerowanie, układ nadzorujący, przerwanie zewnętrzne (wyzwalane poziomem) → wybudzenie

# AVR (2)

---

- Tryby energooszczędne
  - Rejestr MCUCR
    - SE – 1=*Sleep Enable* (uśpienie po rozkazie `sleep`)
      - Włączenie bezpośrednio przed rozkazem `sleep`
    - SM – *Sleep Mode*:
      - 0: tryb jałowy (*Idle*)
      - 1: tryb obniżonego poboru mocy (*Power Down*)

# AVR (2)

---

- Rozkazy
  - Rozkazy arytmetyczno-logiczne
    - Dodawanie i odejmowanie
      - `add; Rd+=Rs`
      - `adc; Rd+=Rs+C`
      - `adiw; RRh:RRI+=c63 (16-b + stała)`
      - `sub; Rd-=Rs`
      - `subi; Rh-=c255`
      - `sbiw; RRh:RRI-=c63 (16-b + stała)`
      - `sbc; Rd-=Rs-C`
      - `sbc i; Rd-=Rh-c255-C`
      - `inc; Rd+=1`
      - `dec; Rd-=1`

# AVR (2)

---

- Rozkazy
  - Rozkazy arytmetyczno-logiczne
    - Rozkazy logiczne
      - `and; Rd&&=Rs`
      - `andi; Rh&&=c255`
      - `or; Rd||=Rs`
      - `ori; Rh||=c255`
      - `eor; Rd^=Rs`
      - `com; Rd=$ff-Rd`
      - `neg; Rd=$00-Rd`
      - `sbr; Rh||=c255`
      - `cbr; Rh&&=c255`
      - `ser; Rh=$ff`
      - `tst; Rd&&=Rd`
      - `clr; Rd^=Rd`

# AVR (2)

---

- Rozkazy

- Rozkazy arytmetyczno-logiczne

- Mnożenie całkowite

- `mul`; (bez znaku)  $R1:R0 = Rd \times Rs$

- `muls`; (ze znakiem)  $R1:R0 = Rh_d \times Rh_s$

- `mulsu`; (ze znakiem  $\times$  bez znaku)  $R1:R0 = Rh_d \times Rh_s$

- Mnożenie ułamkowe

- `fmul`;  $R1:R0 = (Rd \times Rs) \ll 1$

- `fmuls`;  $R1:R0 = (Rd \times Rs) \ll 1$

- `fmulsu`;  $R1:R0 = (Rd \times Rs) \ll 1$



# AVR (2)

---

- Rozkazy

- Operacje bitowe

- ustawienie i zerowanie bitu w porcie we-wy

- `sbi`;  $P(b)=1$

- `cbi`;  $P(b)=0$

- Przesunięcia i rotacje

- `lsl`, `lsr`; przesunięcie logiczne

- `rol`, `ror`; rotacja z przeniesieniem

- `asr`; przesunięcie arytmetyczne

- Pozostałe

- `swap`;  $Rd_{4..7} \leftrightarrow Rd_{3..0}$

- `bset`, `bclr`;  $SREG(b)=1, 0$

- `bst`;  $T=Rs(b)$

- `bld`;  $Rd(b)=T$

# AVR (2)

---

- Rozkazy
  - Operacje bitowe
    - Ustawianie i zerowanie znaczników
      - sec, clc; C=1, 0
      - sen, cln; N=1, 0
      - sez, clz; Z=1, 0
      - sei, cli; I=1, 0
      - ses, cls; S=1, 0
      - sev, clv; V=1, 0
      - set, clt; T=1, 0
      - seh, clh; H=1, 0

# AVR (2)

---

- Rozkazy

- Przesyły danych

- Międzyrejestrowe

- `mov; Rd=Rs`

- `movw; Rd:Rd+1=Rs:Rs+1` (kopia 16-b)

- Operacje we-wy

- `in; Rd=P`

- `out; P=Rs`

- Do i ze stosu

- `push; (SPL)=Rs; SPL-=1`

- `pop; SPL+=1; Rd=(SPL)`

# AVR (2)

---

- Rozkazy

- Przesyły danych

- Ładowanie

- `ldi`; `Rh=c255`

- `ld`; adres pośredni (+postinkr, predekr)

- `ldd`; adres pośredni + przemieszczenie (c63)

- `lds`; adres bezpośredni

- Składowanie

- `sti`; adres pośredni (+postinkr, predekr)

- `std`; adres pośredni + przemieszczenie (c63)

- `sts`; adres bezpośredni

# AVR (2)

---

- Rozkazy
  - Przesyły danych
    - Odczyt z pamięci programu
      - $lpm$ ; adres pośredni (+ postinkr.)
        - »  $Rd=(Z)$ ,  $Rd=(Z++)$
      - $elpm$ ; adres pośredni rozszerzony
        - »  $Rd=(rampz:z)$ ,  $Rd=(rampz:z++)$
      - $spm$ ; adres pośredni
        - »  $(Z)=R1:R0$

# AVR (2)

---

- Rozkazy
  - Skoki i wywołania
    - Rozkazy skoków
      - `rjmp`; względny,  $PC += c1024 + 1$
      - `ijmp`; pośredni,  $PC = Z$
      - `eijmp`;  $PC = EIND.Z$
      - `jmp`;  $PC = \text{adr}4M$
    - Rozkazy wywołań (jak rozkazy skoków)
      - `rcall`
      - `icall`
      - `ecall`
      - `call`
    - Powroty
      - `ret`
      - `reti`

# AVR (2)

---

- Rozkazy
  - Skoki i wywołania
    - Porównanie rejestrów
      - `cpse`; porównanie i przeskok, gdy równe
      - `cp`; porównanie rejestrów
      - `cpc`; porównanie rejestrów z przeniesieniem
      - `cpî`; porównanie rejestru i stałej
    - Przeskoki (warunkowe pominięcie następnego rozkazu)
      - `sbrç`; przeskok, gdy bit rejestru=0
      - `sbrs`; przeskok, gdy bit rejestru=1
      - `sbiç`; przeskok, gdy bit portu=0
      - `sbrç`; przeskok, gdy bit portu=1

# AVR (2)

---

- Rozkazy

- Skoki i wywołania

- Skoki warunkowe

- brbs, brbc; jeśli znacznik w SREG =1/=0
      - breq, brne; jeśli równe/nierówne
      - brcs, brcc; jeśli Carry =1/=0
      - brsh, brlo; jeśli  $\geq$ / $<$  (bez znaku)
      - brmi, brpl; jeśli dodatnie/ujemne
      - brge, brlt; jeśli  $\geq$ / $<$  (ze znakiem)
      - brhs, brhc; jeśli Half carry =1/=0
      - brts, btrs; jeśli T =1/=0
      - brvs, brvc; jeśli V =1/=0
      - bric, brid; jeśli przerwania włączone/wyłączone



# AVR (2)

---

- Rozkazy
  - Sterowanie procesorem
    - `Sleep`; tryb energooszczędny
    - `wdr`; zerowanie układu nadzorującego
    - `break`; używane tylko przez debugery
    - `nop`;