

**Katedra Metrologii i Systemów  
Elektronicznych**  
**Wydział Elektroniki, Telekomunikacji  
i Informatyki**  
**Politechniki Gdańskiej**

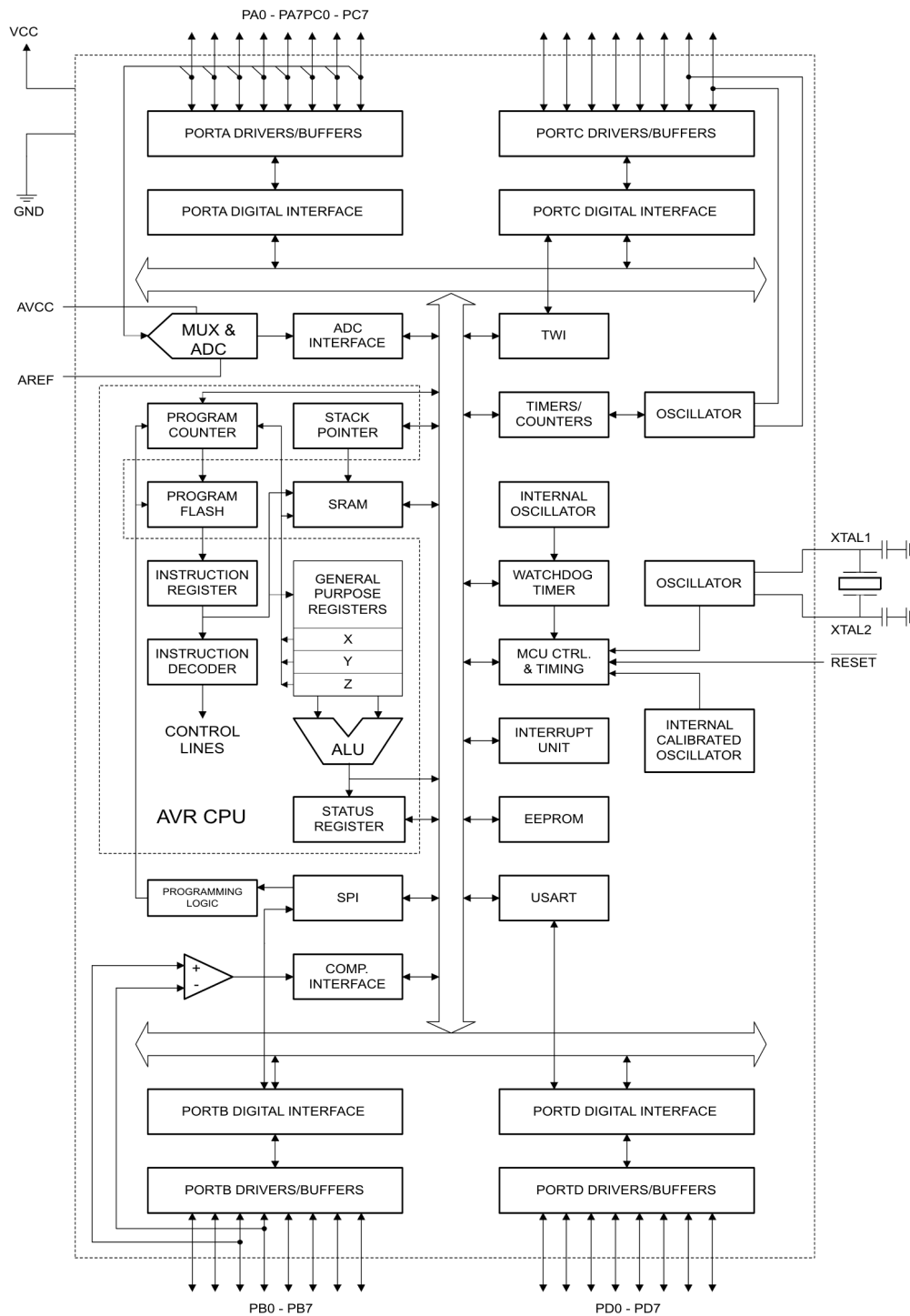


## **LABORATORIUM MIKROSTEROWNIKI I MIKROSYSTEMY ROZPROSZONE**

# **Dokumentacja mikrokontrolera Atmega16 firmy Atmel**

**Gdańsk 2006**

## Charakterystyka mikrokontrolera ATmega16



Rys. 1. Schemat blokowy mikrokontrolera ATmega16

### 1. Cechy i parametry mikrokontrolera ATmega16:

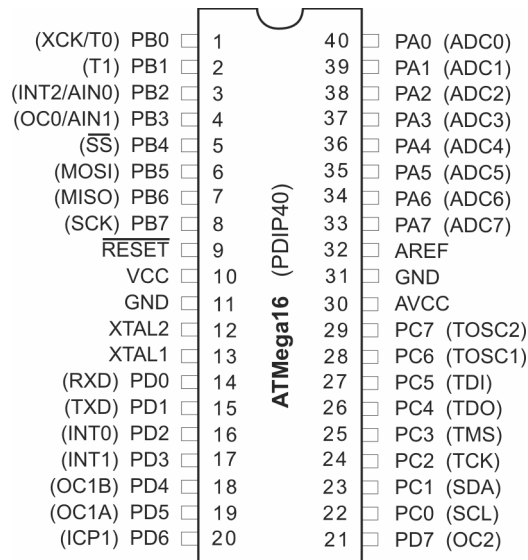
- Wysokowydajna architektura AVR.
- RISC, 131 instrukcji (większość 1 cyklowe), nastawione na język C, mnożenie 3 cyklowe.

- 16kB pamięci flash programowanej w systemie (ISP) z funkcją Read-While-Write, trwałość do 10k cykli kasuj/zapisz.
- 32 x8 rejestry robocze.
- 1kB pamięci SRAM.
- 512 pamięci EEPROM (100k cykli).
- Programowalne blokady bezpieczeństwa pamięci programu i eeprom.
- do 32 konfigurowalnych linii I/O.
- Interfejs JTAG (IEEE 1149.1): testowanie, debugowanie w układzie, programowanie pamięci w systemie.
- Trzy elastyczne timery/liczniki z trybami porównania (Input/Output Compare).
- Wewnętrzne i zewnętrzne programowalne przerwania.
- Szeregowy interfejs USART (praca synchroniczna i asynchroniczna).
- Interfejsy szeregowy TWI (kompatybilny z I2C) oraz SPI.
- 8kanałowy 10bitowy przetwornik ADC, z opcjonalnym trybem wejścia różnicowego wraz z programowalnym wzmocnieniem (tylko w wersji TQFP).
- Analogowy komparator w układzie.
- Programowalny Watchdog z własnym oddzielnym oscylatorem .
- Układ Power-On Reset (zapewnienie prawidłowego resetu po włączeniu zasilania).
- Wewnętrzny programowany generator RC (1, 2, 4 lub 8MHz), który pozwala w wielu przypadkach zrezygnować z podłączania zewnętrznego kwarcu.
- Programowalny próg spadku napięcia zasilania (Brown-out detection).
- 6 trybów oszczędzania energii.
- Zakresy napięć zasilania:
  - 2,7 – 5,5 V dla ATMega16L,
  - 4,5 – 5,5 V dla ATMega16.
- Dopuszczalna szybkość pracy:
  - 0 – 8 MHz dla ATMega16L,
  - 0 – 16 MHz dla ATMega16.
- Pobór mocy dla 3V przy 1MHz:
  - W stanie aktywnej pracy: 1,1mA,
  - Tryb Idle: 0,35mA,
  - Tryb Power-down: poniżej 1uA.
- Wersje obudowy: 40pin PDIP, 44pin TQFP, 44pad MLF.

Technologia Flash ISP pozwala na przeprogramowywanie pamięci „w systemie” poprzez szeregowy interfejs ISP, z wykorzystaniem konwencjonalnych programatorów, magistralę JTAG, a także przez program botujący<sup>1</sup> pracujący w samym układzie. Program botujący może wykorzystywać dowolny rodzaj interfejsu, aby załadować właściwy program do pamięci aplikacji. Program z sekcji botującej kontynuuje swoją pracę podczas programowania części aplikacyjnej, na co pozwala technika rzeczywistego Write-While-Read.

<sup>1</sup> Program botujący – podstawowa część programu uruchamiająca system, przed programem właściwym

Połączone możliwości 8-bitowej jednostki obliczeniowej RISC, programowania w systemie oraz samoreprogramowaniem pamięci flash w jednym układzie, pozwala na zastosowanie tego mikrokontrolera w elastycznych funkcjonalnie i oszczędnych cenowo rozwiązaniach, w wymagających kontroli działania aplikacjach wbudowanych (ang. embedded).



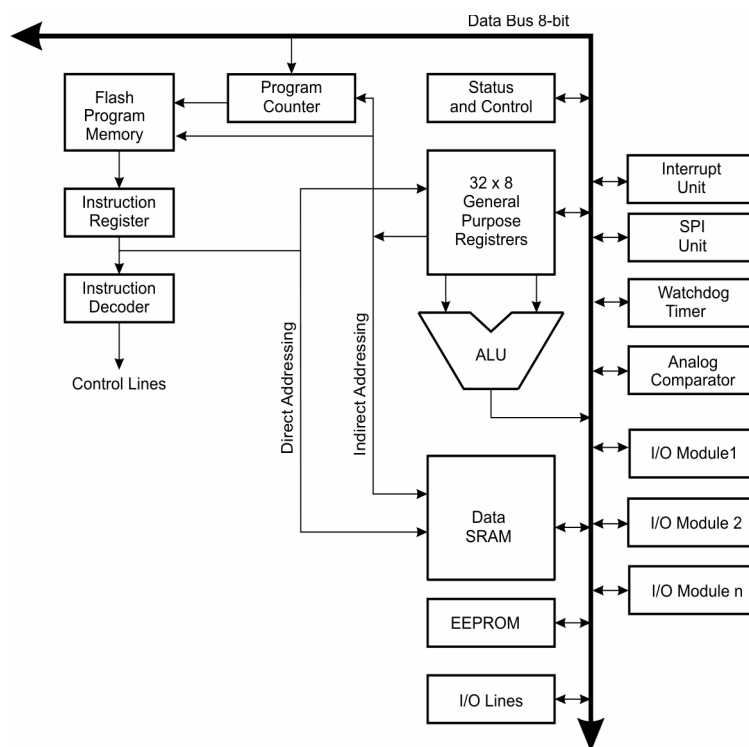
Rys. 2. Opis wyprowadzeń mikrokontrolera ATmega16 w obudowie DIP40

## 2. Zarys budowy rdzenia AVR

Głównym zadaniem rdzenia CPU (centralnej jednostki liczącej) jest zapewnienie poprawnego i szybkiego wykonywania kodu. Z tego powodu CPU musi mieć dostęp do pamięci, musi wykonywać operacje, sterować układami peryferyjnymi i obsługiwać przerwania.

Dla uzyskania maksymalnej wydajności AVR zbudowany jest w oparciu o architekturę harwardzką – z rozdzielonymi pamięciami i szynami dla programu i danych. Instrukcje w pamięci programu wykonywane są potokowo. Podczas gdy jedna instrukcja jest wykonywana, następna jest już pobierana z pamięci programu. Dzięki takiemu rozwiązaniu, możliwe jest wykonywanie całej instrukcji w każdym cyklu zegara.

Blok szybkiego dostępu zawiera 32 rejestry robocze o jednocyklowym czasie dostępu. To pozwala aby jednostka ALU (arytmetyczno logiczna) również mogła pracować w jednym cyklu. W typowej operacji, dwa argumenty ALU są wystawiane z bloku rejestrów, wykonywana jest operacja, a wynik jest powrotem umieszczany w bloku rejestrów, i to wszystko w jednym cyklu zegara.



Rys. 3. Schemat blokowy architektury AVR

Sześć spośród 32 rejestrów roboczych może być używane jako trzy 16-bitowe rejestry wskaźnikowe (X, Y, Z) w trybie adresowania pośredniego. Daje to możliwość wykonywania szybkich przeliczeń danych. Jeden z tych trzech wskaźników adresu (Z) może być użyty jako wskaźnik adresu w tzw. lookup tables w pamięci flash programu.

Jednostka ALU umożliwia wykonywanie operacji arytmetycznych i logicznych między rejestrami, między stałą a rejestrem, oraz także operacji na pojedynczym rejestrze. Po wykonaniu operacji uaktualniany jest rejestr statusowy dający informacje o rezultacie.

Działanie programu jest możliwe dzięki warunkowym i bezwarunkowym skokom i instrukcjom rozgałęziającym, dającym bezpośredni dostęp do całej przestrzeni adresowej. Większa część instrukcji AVR ma format pojedynczego 16-bitowego słowa. Każdy adres pamięci programu zawiera 16 lub 32-bitową instrukcję.

Pamięć programu może być podzielona na dwie sekcje: sekcję programu bootującego, oraz sekcję programu aplikacji. Obie części mają indywidualne bity bezpieczeństwa (Lock bits) do zabezpieczenia przed odczytem i nadpisaniem zawartości (np. przez programator). Program sekcji bootującej może służyć np. do aktualizacji oprogramowania systemu, gdyż tylko z tej części pamięci programu może być wykonywana instrukcja SPM nadpisująca pamięć flash aplikacji.

Podczas przerw i wywołań procedur podrzędnych adres powrotny licznika programu jest przechowywany w pamięci stosu. Stos zajmuje obszar pamięci SRAM, więc jego rozmiar jest ograniczony wielkością tej pamięci. Każdy program musi zainicjalizować wskaźnik stosu SP w procedurze obsługi resetu, zanim zostanie wywołana obsługa przerwania czy podprocedura.

Moduł przerwań ma swoje rejestry kontrolne w przestrzeni I/O z dodatkowym globalnym bitem odblokowującym przerwania w rejestrze statusowym SREG (opisany dalej). Wszystkie przerwania mają osobny wektor przerwania w tablicy wektorów przerwań. Przerwania mają priorytet zależny od pozycji wektora. Im niższy adres wektora przerwań, tym wyższy jest priorytet dla danego przerwania.

## Rejestr SREG

Rejestr statusowy SREG zawiera informacje o rezultacie ostatnio wykonywanej operacji arytmetycznej. Informacja ta może być użyta przez instrukcje warunkowe sprawdzające stan wyniku poprzedniej operacji. Rejestr statusowy jest aktualizowany przez operacje wykonane przez ALU. Nie zawsze musi być zmieniana cała zawartość SREG. Zależnie od instrukcji, może to być np. tylko jeden bit. Dzięki wykorzystaniu tych właściwości możliwe jest tworzenie bardziej zoptymalizowanego kodu pod względem szybkości i objętości. (jak zmieniane są flagi, patrz „Opis instrukcji”).

Zawartość rejestru SREG nie jest automatycznie zapamiętywana przy wejściu w kod obsługi przerwania i odzyskiwana po jego wykonaniu. Należy to wykonywać programowo.

### Rejestr SREG

Bit	7	6	5	4	3	2	1	0
	I	T	H	S	V	N	Z	C
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

- I - Globalne odblokowanie przerwań.  
Flaga globalnych przerwań musi zostać być ustawiona aby odblokować przerwania. Kontrola indywidualnych przerwań jest realizowana w osobnych rejestrach kontrolnych. Jeśli globalna flaga I jest wykasowana (równa 0) to żadne z przerwań nie nastąpi, pomimo iż mogą być niezależnie odblokowane. Bit I jest automatycznie kasowany po wystąpieniu przerwania i z powrotem ustawiany instrukcją RETI aby umożliwić wystąpienie kolejnym przerwaniami. Globalny bit przerwań może być też zmieniany programowo instrukcjami SEI oraz CLI.
- T - Przechowanie kopiowanego bitu.  
Instrukcje kopiowania bitu BLD i BST używają T jako bit źródłowy lub docelowy do operacji bitowej.
- H - Flaga przeniesienia połówkowego (half carry).  
Flaga H pojawia się w wyniku niektórych operacji arytmetycznych (gdy nastąpi przeniesienie na 4 bit). Użyteczna w arytmetyce BCD.
- S - Bit znaku  $S = N \oplus V$ .  
Wartość bitu S jest zawsze równy operacji XOR między bitami V oraz N.
- V - Flaga przepelnienia uzupełnienia do dwóch.  
Flaga wpierająca arytmetykę „uzupełnienia do dwóch”.
- N - Flaga wyniku ujemnego.

- Flaga N pojawia się w wyniku ujemnego wyniku operacji arytmetycznej lub logicznej.
- Z - Flaga zera.  
Flaga Z pojawia się, gdy w wyniku operacji arytmetycznej lub logicznej wynikiem jest wartość zero.
- C - Flaga przeniesienia.  
Flaga przeniesienia C (carry) pojawia się gdy w wyniku operacji arytmetycznej lub logicznej nastąpiło przeniesienie.

### 3. Źródła resetu

W mikrokontrolerze ATMega16 istnieje pięć źródeł resetu:

- Power-on Reset.
- Reset zewnętrzny.
- Watchdog reset.
- Brown-out reset.
- JTAG AVR reset.

Podczas resetu, wszystkie rejestry I/O są ustawiane na ich wartości początkowe a program rozpoczyna działanie od wektora resetu. Jeśli przerwania w układzie zostały włączone (np. instrukcją SEI), wówczas pod adresem 0x0000 powinna znaleźć się instrukcja JMP do procedury obsługi resetu. Jeśli program nigdy nie używa przerw, to od tego adresu może zacząć się normalny program.

Odczytując stan bitów rejestru MCUCSR można określić, które ze źródeł resetu spowodowało reset mikrokontrolera.

#### Power-on Reset

Mikrokontroler jest resetowany, gdy poziom napięcia zasilającego znajdzie się poniżej progu Power-on ( $V_{POT}$ ). Układ Power-on Reset ogranicza konieczność stosowania zewnętrznych specjalizowanych układów zapewniających poprawne uruchomienie mikrokontrolera, czyli przytrzymujące stan resetu (tutaj niski) przez określoną chwilę czasu po podłączeniu zasilania do mikrokontrolera. Wbudowany Power-On Reset generuje wewnętrzny impuls resetu o wymaganej długości. Takie rozwiązanie pozwala na bezpośrednie połączenie napięcia zasilającego mikrokontrolera  $V_{CC}$  do pinu  $\overline{RESET}$ . Dodatkowe układy mogą jednak przedłużyć czas uruchamiania.

#### Reset zewnętrzny

Zewnętrzny reset jest powodowany przez pojawienie się niskiego poziomu na pinie  $\overline{RESET}$ . Reset mikrokontrolera nastąpi, gdy impuls resetujący będzie trwał przynajmniej 1,5  $\mu s$ .

## Watchdog reset

Układ Watchdog służy wykluczeniu możliwości „zawieszania się” mikrokontrolera, np. na skutek zaistniałych zakłóceń w systemie czy poprzez wykonywanie nieskończonych niepożądanych pętli programu. Jeśli włączony licznik watchdog’a nie zostanie w określonym czasie wyzerowany, nastąpi wygenerowanie impulsu resetującego mikrokontroler. Taktowanie odbywa się poprzez niezależny wbudowany oscylator o częstotliwości pracy ok. 1MHz. Włączanie i konfigurowanie watchdog’a odbywa się przez rejestr WDTCR.

### Rejestr WDTCR

Bit	7	6	5	4	3	2	1	0
	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0
Dostęp	R	R	R	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

WDTOE - Bit odblokowania wyłączenia watchdog’a.

Ustawienie tego bitu na 1, tuż przed wpisaniem do WDE wartości 0, pozwala wyłączyć watchdog'a.

WDE - Bit włączania watchdog’a.

Wpisanie logicznej jedynki do bitu WDE powoduje włączenie watchdog’a, natomiast wpisanie 0 go wyłącza. Aby wyłączyć watchdoga, wcześniej bit WDTOE musi być ustawiony na 1, albo oba te bity zapisać równocześnie.

WDP2:0 - Bity preskalera licznika watchdog’a.

Za pomocą tych bitów dobierany jest czas, jaki upływa od zerowania watchdog’a do wygenerowania przez niego impulsu resetującego. Czas można dobierać w zakresie od ok. 16ms (stan 000) do ok. 2,2s (stan 111).

## Brown-out reset

Reset mikrokontrolera następuje, kiedy poziom napięcia zasilającego  $V_{CC}$  spadnie poniżej zaprogramowanego progu detektora Brown-out ( $V_{BOT}$ ) i jeśli ten detektor jest włączony. Funkcję tą konfiguruje się za pomocą bitów bezpiecznikowych (fuse BODEN).

## JTAG AVR reset

Mikrokontroler pozostaje tak długo w stanie Resetu, jak długo znajdować się będzie logiczna jedynka w rejestrze resetu (Reset Register), który jest częścią systemu JTAG.

## 4. Porty I/O

Wszystkie porty mikrokontrolerów AVR posiadają możliwość pracy jako cyfrowe porty wejścia/wyjścia. Pojedynczemu pinowi można zmienić kierunek pracy instrukcjami do zmian stanu bitów CBI i SBI.



Sterownik każdego pinu może dostarczyć wystarczającą ilość mocy na bezpośrednie wystawienie diody LED. Wydajność prądowa każdego z pinów sięga 20mA w stanie niskim. Wszystkie piny każdego z portów posiadają możliwość włączenia indywidualnie rezystora podciągającego do napięcia zasilania (pull-up).

Każdy z portów ma ściśle z nim związane 3 rejestry konfiguracyjne. Są to: rejestr danych PORTx, rejestr kierunkowy DDRx oraz port pinów wejściowych PINx. Dodatkowo ustawieniem jednego bitu PUD w rejestrze SFIOR można wyłączyć wszystkie rezystory pull-up we wszystkich portach jednocześnie.

Większość pinów posiada więcej niż jedną funkcję, do obsługi układów zintegrowanych w mikrokontrolerze. Uaktywnienie funkcji alternatywnej wybranym pinom portu nie blokuje pozostałym możliwości pracy jako cyfrowe piny I/O.

### Konfigurowanie cyfrowych portów I/O

Bit DDXn w rejestrze DDRx określa kierunek pracy pinu. Wpisana tam wartość 1 powoduje pracę pinu Pxn jako wyjściowy, 0 jako wejściowy.

Wpisanie 1 do PORTxn podczas, gdy pin skonfigurowany jest jako wejście, powoduje włączenie rezystora pull-up. Wartość 0 spowoduje jego wyłączenie. Skonfigurowanie pinu jako wyjście automatycznie odłączy rezystor pull-up. W czasie stanu resetu mikrokontrolera lub gdy nie pracuje zegar, piny są w trójstanowe.

Wpisanie 1 do PORTxn, podczas gdy pin skonfigurowany jest jako wyjście, spowoduje ustawienie stanu wysokiego. Wpisanie 0 ustawi stan niski.

Ustawienie bitu PUD w rejestrze SFIOR wyłącza rezystory pull-up we wszystkich portach, bez względu na ustawienie DDXn i PORTxn.

Tabela 1. Konfiguracja pinów portów

DDxn	PORTxn	PUD (SFIOR)	I/O	PullUp	Komentarz
0	0	X	Wejście	Nie	Trzy stany (Hi-Z)
0	1	0	Wejście	Tak	Pxn będzie źródłem prądu jeśli na zewnątrz połączony jest z masą.
0	1	1	Wejście	Nie	Trzy stany (Hi-Z)
1	0	X	Wyjście	Nie	Wyjście stan niski
1	1	X	Wyjście	Nie	Wyjście stan wysoki (źródło)

### Odczytywanie stanu pinów

Niezależnie od obranego kierunku pracy pinu w DDXn, wartość pinu portu może być odczytana poprzez bity PINxn. Jeśli zmiana stanu następuje z zewnątrz, to wartość PINxn jest opóźniona do momentu synchronizacji ze stanem niskim zegara. Wartość na pinie jest zapamiętywana na zboczu opadającym zegara a uaktualniona wartość PINxn pojawia się ze zboczem narastającym zegara. Potwierdzenie wpisanej programowo wartości pinu, poprzez odczyt PINxn, musi odbywać się więc co najmniej jeden cykl zegarowy po wpisaniu. Synchronizujące opóźnienie można zrealizować instrukcją NOP.

## Alternatywne funkcje portów

Większość pinów portów poza pracą jako cyfrowe wejście/wyjście posiada funkcje alternatywną. Funkcja alternatywna jednego z pinów portu nie blokuje pracy pozostałym jako cyfrowe piny wejścia/wyjścia.

### Port A

Alternatywną funkcją pinów portu A są kanały przetwornika ADC. Jeśli któreś z pinów są skonfigurowane jako wyjściowe, podczas gdy inne pracują jako kanały przetwornika, przełączanie wyjść podczas trwającej konwersji może pogorszyć jakość konwersji.

Tabela 2. Alternatywne funkcje pinów portu A

Pin portu	Funkcja alternatywna
PA7	ADC7 (wejście ADC, kanał 7)
PA6	ADC6 (wejście ADC, kanał 6)
PA5	ADC5 (wejście ADC, kanał 5)
PA4	ADC4 (wejście ADC, kanał 4)
PA3	ADC3 (wejście ADC, kanał 3)
PA2	ADC2 (wejście ADC, kanał 2)
PA1	ADC1 (wejście ADC, kanał 1)
PA0	ADC0 (wejście ADC, kanał 0)

### Port B

Tabela 3. Alternatywne funkcje pinów portu B

Pin portu	Funkcja alternatywna
PB7	SCK (zegar szyny szeregowej SPI)
PB6	MISO (szyna interfejsu SPI, Master Input/Slave Output)
PB5	MOSI (szyna interfejsu SPI, Master Output/Slave Input)
PB4	SS (interfejs SPI, wejście Slave Select)
PB3	AIN1 (ujemne wejście komparatora analogowego) OC0 (wyjście Output Compare Timera/Licznika 0)
PB2	AIN0 (dodatnie wejście komparatora analogowego) INT2 (wejście zewnętrznego przerwania 2)
PB1	T1 (zewnętrzne wejście licznika)
PB0	T0 (zewnętrzne wejście licznika) XCK (zewnętrzny zegar USART)

### Port C

Kilka pinów portu C może pełnić funkcje linii interfejsu JTAG. Jeśli JTAG jest włączony, to rezystory pull-up na pinach PC5, PC3, PC2 są aktywne nawet w stanie resetu. W

fabrycznie nowym układzie ATmega16 interfejs JTAG jest włączony, więc jego cztery piny PC2-PC5 nie będą pracować jako cyfrowe piny wejścia wyjścia.

Tabela 4. Alternatywne funkcje pinów portu C

Pin portu	Funkcja alternatywna
PC7	TOSC2, Oscylator licznika, Pin 2
PC6	TOSC1, Oscylator licznika, Pin 1
PC5	TDI, JTAG Test Data In
PC4	TDO, JTAG Test Data Out
PC3	TMS, JTAG Test Mode Select
PC2	TCK, JTAG Test Clock
PC1	SDA, linia wejścia/wyjścia danych interfejsu TWI (I2C)
PC0	SCL, linia zegarowa interfejsu TWI (I2C)

## Port D

Tabela 5. Alternatywne funkcje pinów portu D

Pin portu	Funkcja alternatywna
PD7	OC2, wyjście Output Compare Timera/Licznika 2
PD6	ICP1, pin wejściowy Input Capture Timera/Licznika 1
PD5	OC1A, wyjście Output Compare A, Timera/Licznika 1
PD4	OC1B, wyjście Output Compare B Timera/Licznika 1
PD3	INT1, wejście zewnętrznego przerwania 1
PD2	INT0, wejście zewnętrznego przerwania 0
PD1	TXD, pin wyjściowy USART
PD0	RXD, pin wejściowy USART

## Opis rejestrów dla portów I/O

### Rejestr danych portu A – PORTA

#### Rejestr PORTA

Bit	7	6	5	4	3	2	1	0
	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

### Rejestr kierunku danych portu A – DDRA

#### Rejestr DDRA

Bit	7	6	5	4	3	2	1	0
	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

### Rejestr pinów wejściowych danych portu A – PINA

**Rejestr PORTA**

Bit	7	6	5	4	3	2	1	0
	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0
Dostęp	R	R	R	R	R	R	R	R
Wartość początkowa	nd.	nd.	nd.	nd.	nd.	nd.	nd.	nd.

**Rejestr danych portu B – PORTB****Rejestr PORTB**

Bit	7	6	5	4	3	2	1	0
	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

**Rejestr kierunku danych portu B – DDRB****Rejestr DDRB**

Bit	7	6	5	4	3	2	1	0
	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

**Rejestr pinów wejściowych danych portu B – PINB****Rejestr PORTB**

Bit	7	6	5	4	3	2	1	0
	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
Dostęp	R	R	R	R	R	R	R	R
Wartość początkowa	nd.	nd.	nd.	nd.	nd.	nd.	nd.	nd.

**Rejestr danych portu C – PORTC****Rejestr PORTC**

Bit	7	6	5	4	3	2	1	0
	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

**Rejestr kierunku danych portu C – DDRC****Rejestr DDRC**

Bit	7	6	5	4	3	2	1	0
	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

**Rejestr pinów wejściowych danych portu C – PINC**

### Rejestr PORTC

Bit	7	6	5	4	3	2	1	0
	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
Dostęp	R	R	R	R	R	R	R	R
Wartość początkowa	nd.	nd.	nd.	nd.	nd.	nd.	nd.	nd.

### Rejestr danych portu D – PORTD

#### Rejestr PORTD

Bit	7	6	5	4	3	2	1	0
	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

### Rejestr kierunku danych portu D – DDRD

#### Rejestr DDRD

Bit	7	6	5	4	3	2	1	0
	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

### Rejestr pinów wejściowych danych portu D – PIND

#### Rejestr PORTD

Bit	7	6	5	4	3	2	1	0
	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
Dostęp	R	R	R	R	R	R	R	R
Wartość początkowa	nd.	nd.	nd.	nd.	nd.	nd.	nd.	nd.

## 1. Przerwania

System przerwania obecny jest prawdopodobnie w każdym mikrokontrolerze. Im bardziej jest on złożony, tym bardziej rozbudowany będzie w nim system przerwania. W mikrokontrolerze ATmega16 dostępnych jest 21 rodzajów przerwania, które mogą być wywoływane na skutek jeszcze większej liczby zdarzeń. Pozwala to na sprawne kontrolowanie pracą wbudowanych w układ urządzeń peryferyjnych, a dzięki przerwaniom zewnętrznym także urządzeń zewnętrznych.

Opis konfiguracji przerwania jest związany ściśle z układami peryferyjnymi i jest zawarty w opisie tych układów w kolejnych punktach. Globalna flaga przerwania znajduje się w głównym rejestrze kontrolnym SREG i została omówiona przy opisie tego rejestru przy końcu rozdziału „Zarys budowy rdzenia AVR”.

Zawartość rejestru SREG nie jest automatycznie zapamiętywana przy wejściu w kod obsługi przerwania i odzyskiwana po jego wykonaniu. Należy to wykonywać programowo.

Podczas przerwania i wywołań procedur podrzędnych adres powrotny licznika programu jest przechowywany w pamięci stosu.

Poniższa tabela zawiera zestawienie wektorów przerw dla ATmega16. Nie obejmuje przypadku korzystania z podziału pamięci na sekcję bootującą i aplikacyjną.

Tabela 6. Wektory przerw i resetu

Wektor, nr	Adres pamięci programu	Źródło	Opis przerwania
1	\$000	RESET	Pin zewnętrzny, Power-on reset, Brown-out reset, Watchdog reset, JTAG reset
2	\$002	INT0	Zewnętrzne przerwanie INT0
3	\$004	INT1	Zewnętrzne przerwanie INT1
4	\$006	TIMER2 COMP	Timer / Licznik 2 dopasowanie porównania
5	\$008	TIMER2 OVF	Timer / Licznik 2 przepełnienie
6	\$00A	TIMER1 CAPT	Timer / Licznik 1 zdarzenie przechwycenia
7	\$00C	TIMER1 COMPA	Timer / Licznik 1 dopasowanie porównania A
8	\$00E	TIMER1 COMPB	Timer / Licznik 1 dopasowanie porównania B
9	\$010	TIMER1 OVF	Timer / Licznik 1 przepełnienie
10	\$012	TIMER0 OVF	Timer / Licznik 0 przepełnienie
11	\$014	SPI, STC	Zakończona transmisja szeregową
12	\$016	USART, RXC	USART, Rx
13	\$018	USART, UDRE	USART, pusty rejestr danych
14	\$01A	USART, TXC	USART, Tx
15	\$01C	ADC	Zakończona konwersja ADC
16	\$01E	EE_RDY	EEPROM gotowy
17	\$020	ANA_COMP	Komparator analogowy
18	\$022	TWI	Interfejs szeregowy TWI (I2C)
19	\$024	INT2	Zewnętrzne przerwanie INT2
20	\$026	TIMER0 COMP	Timer / Licznik 0 dopasowanie porównania
21	\$028	SPM_RDY	Store Program Memory Ready

## Rejestr GICR

### Główny rejestr kontroli przerw

Dwa poniżej opisane bity rejestru GICR mają znaczenie w przypadku korzystania z dzielenia pamięci programu na część bootującą oraz część aplikacyjną. Opis pozostałych bitów i rejestrów konfiguracyjnych znajduje się przy opisach układów peryferyjnych które z nich korzystają.

#### Rejestr GICR

Bit	7	6	5	4	3	2	1	0
	INT1	INT0	INT2	-	-	-	IVSEL	IVCE
Dostęp	R/W	R/W	R/W	R	R	R	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

IVSEL - Bit wyboru początku wektora przerw.

Wykorzystywanie tego bitu wiąże się z korzystaniem z programów bootujących. Jego wartość zero oznacza że wektory przerw zaczynają się od początku obszaru pamięci flash. Wartość jeden przesuwą początek do miejsca wybranego przez konfigurację bitów bezpiecznikowych BOOTZS

IVCE - Bit odblokowujący wybór wektora przerw.

Zabezpiecza przed przypadkową zmianą stanu bitu IVSEL. Musi być ustawiony na 1 zanim zostanie zmieniony IVSEL. Bit ten jest sprzętowo kasowany po 4 cyklach zegara od ustawienia tego bitu, lub po zmianie IVSEL.

## Przerwania zewnętrzne

Przerwania zewnętrzne są wyzwalane pinami INT0, INT1 oraz INT2. Mogą być wyzwalane niezależnie od tego czy te piny są skonfigurowane jako wyjściowe czy wejściowe. Cecha ta może być wykorzystana do programowego generowania przerw. Zależnie od ustawień w rejestrach MCUCR oraz MCUCS, przerwanie zewnętrzne może być wyzwolone zboczem narastającym, opadającym, lub niskim poziomem (INT2 tylko zboczem).

**INT0 / INT1:** Przerwanie skonfigurowane jako wyzwalane poziomem będzie trwało tak długo, jak będzie ten stan się utrzymywał. Poziom jest wykrywany asynchronicznie i może służyć do wybudzenia mikrokontrolera z trybu Power-Down. Stan niski wówczas musi się utrzymać przez czas, jaki jest potrzebny na wybudzenie.

**INT2:** Przerwanie INT2 może być wyzwolone jednym ze zboczy, a zdarzenie to wykrywane jest asynchronicznie.

Aby zdarzenie synchroniczne zostało wykryte (np. z wejścia INT1), przy wykrywaniu zbocza impuls nie może być zbyt krótki, natomiast przy wykrywaniu poziomu, jego stan również nie może trwać zbyt krótko. Związane to jest z próbkowaniem stanu pinu w takt zegara Watchdog.

## Rejestr MCUCR

### Rejestr kontroli mikrokontrolera

Rejestr MCUCR zawiera między innymi bity kontrolne do wyboru sposobu wykrywania przerwania.

#### Rejestr MCUCR

Bit	7	6	5	4	3	2	1	0
	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

ISC10, ISC11 - Bity wyboru sposobu wykrywania przerwania 1.

Pin INT1 może być źródłem przerwania jeśli ustawiona jest flaga I w SREG oraz odpowiednia maska przerw w GICR.

ISC11	ISC10	Opis
0	0	Niski poziom pinu INT1 spowoduje żądanie przerwania
0	1	Dowolna zmiana stanu logicznego na INT1 wywoła żądanie przerwania
1	0	Zbocze opadające na INT1 wywoła żądanie przerwania
1	1	Zbocze narastające na INT1 wywoła żądanie przerwania

ISC00, ISC01 - Bity wyboru sposobu wykrywania przerwania 0.

Pin INT0 może być źródłem przerwania jeśli ustawiona jest flaga I w SREG oraz odpowiednia maska przerwania w GICR.

ISC01	ISC00	Opis
0	0	Niski poziom pinu INT0 spowoduje żądanie przerwania
0	1	Dowolna zmiana stanu logicznego na INT0 wywoła żądanie przerwania
1	0	Zbocze opadające na INT0 wywoła żądanie przerwania
1	1	Zbocze narastające na INT0 wywoła żądanie przerwania

## Rejestr MCUCSR

### Rejestr kontroli i statusu mikrokontrolera

#### Rejestr MCUCSR

Bit	7	6	5	4	3	2	1	0
	JTD	ISC2	-	JTRF	WDRF	BORF	EXTRF	PORF
Dostęp	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	<i>zobacz w opisie bitów</i>				

ISC2 - Bity wyboru sposobu wykrywania przerwania 2.

Pin INT2 (asynchroniczne przerwanie zewnętrzne) może być źródłem przerwania jeśli ustawiona jest flaga I w SREG oraz odpowiednia maska przerwania w GICR. Konfigurując ISC2 wybieramy rodzaj zbocza które wygeneruje przerwanie. Wpisanie 1 powoduje reakcję na zbocze narastające, wpisanie 0 na zbocze opadające. Aby zmiana została wychwycona, impuls powinien trwać typowo nie krócej niż 50ns.

Zmiana bitu ISC2 może wywołać przerwanie, więc aby można było go zmienić, najpierw należy wykasować bit odblokowujący w GICR, po czym można zmienić ISC2. Przed ponownym odblokowaniem przerwania INT2, flaga tego przerwania powinna być wykasowana przez wpisanie jedynek do INTF2 w GIFR.

## Rejestr GICR

### Główny rejestr kontroli przerwania

#### Rejestr GICR

Bit	7	6	5	4	3	2	1	0
	INT1	INT0	INT2	-	-	-	IVSEL	IVCE
Dostęp	R/W	R/W	R/W	R	R	R	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

INT1 - Bit odblokowujący zewnętrzne przerwanie 1.

Ustawienie tego bitu na 1 oraz ustawionym na 1 bicie I w SREG włącza zewnętrzne przerwanie. Sposób wykrywania przerwania wybiera się bitami



ISC11 i ISC10 w rejestrze MCUCR. Zdarzenie może być wykryte nawet gdy pin skonfigurowany jest jako wyjściowy.

INT0 - Bit odblokowujący zewnętrzne przerwanie 0.

Ustawienie tego bitu na 1 oraz ustawionym na 1 bicie I w SREG włącza zewnętrzne przerwanie. Sposób wykrywania przerwania wybiera się bitami ISC01 i ISC00 w rejestrze MCUCR. Zdarzenie może być wykryte nawet gdy pin skonfigurowany jest jako wyjściowy.

INT2 - Bit odblokowujący zewnętrzne przerwanie 2.

Ustawienie tego bitu na 1 oraz ustawionym na 1 bicie I w SREG włącza zewnętrzne przerwanie. Bitem ISC2 w rejestrze MCUCSR określa się rodzaj zbocza generującego przerwanie. Zdarzenie może być wykryte nawet gdy pin skonfigurowany jest jako wyjściowy.

## Rejestr GIFR

### Główny rejestr flag przerwania

#### Rejestr GIFR

Bit	7	6	5	4	3	2	1	0
	INTF1	INTF0	INTF2	-	-	-	-	-
Dostęp	R/W	R/W	R/W	R	R	R	R	R
Wartość początkowa	0	0	0	0	0	0	0	0

INTF1 - Flaga zewnętrznego przerwania 1.

Kiedy zdarzenie pinu INT1 wywoła przerwanie, flaga INTF1 jest ustawiana na 1. Jeśli ustawiony jest bit I w SREG oraz ustawiony bit INT1 w GICR, CPU skoczy do odpowiedniego wektora przerwania. Flaga INTF1 jest kasowana kiedy wykonywana jest procedura obsługi przerwania. Możliwe jest „ręczne” jej wykasowanie przez wpisanie do niej 1. Jeśli ustawionej jest wyzwalanie przerwania poziomem, ta flaga zawsze pozostaje wyczyszczona.

INTF0 - Flaga zewnętrznego przerwania 0.

Kiedy zdarzenie na pinie INT0 wywoła przerwanie, flaga INTF0 jest ustawiana na 1. Jeśli ustawiony jest bit I w SREG oraz ustawiony bit INT0 w GICR, mikrokontroler skoczy do odpowiedniego wektora przerwania. Flaga INTF0 jest kasowana kiedy wykonywana jest procedura obsługi przerwania. Możliwe jest „ręczne” jej wykasowanie przez wpisanie do niej 1.

INTF2 - Flaga zewnętrznego przerwania 2.

Kiedy zdarzenie na pinie INT2 wywoła przerwanie, flaga INTF2 jest ustawiana na 1. Jeśli ustawiony jest bit I w SREG oraz ustawiony bit INT2 w GICR, CPU skoczy do odpowiedniego wektora przerwania. Flaga INTF2 jest kasowana kiedy wykonywana jest procedura obsługi przerwania. Jeśli ustawione jest wyzwalanie przerwania poziomem, ta flaga zawsze pozostaje wyczyszczona.

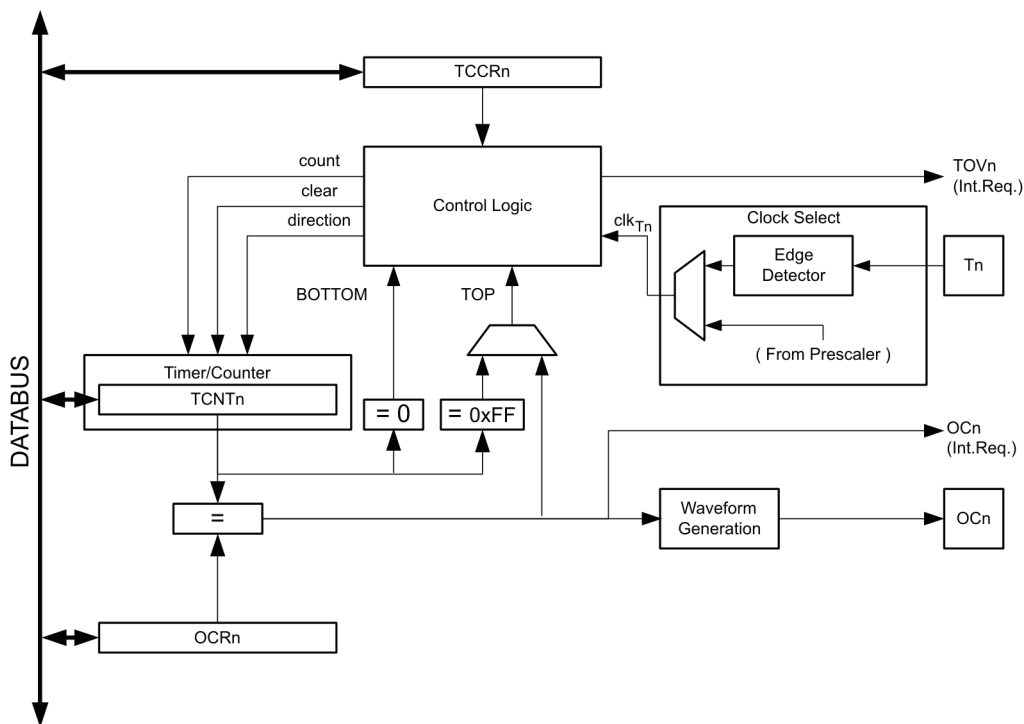
## 6. Timery/Liczniki

W mikrokontrolerze ATMega16 dostępne są trzy moduły liczników (timerów), dwa 8-bitowe i jeden 16-bitowy. Oprócz tych istnieje także specjalizowany licznik tzw. Watchdog Timer (opisany w punkcie „2.3. Źródła resetu”).

### 6.1. 8-bitowy licznik 0 z PWM

Główne właściwości oraz możliwości zastosowań licznika 0:

- Licznik jedno kanałowy.
- Automatyczne kasowanie i restart timera w trybie porównania.
- Generator częstotliwości.
- Generator przebiegu PWM.
- Licznik zdarzeń zewnętrznych.
- 10-bitowy programowalny preskaler.
- Źródło przerw (przepełnienie, zrównanie).



Rys. 4. Schemat blokowy licznika 8-bitowego

Rejestr licznika (TCNT0) oraz rejestr output compare (OCR0) są 8-bitowe. Licznik może być taktowany wewnętrznie poprzez układ preskalera lub przez zewnętrzne źródło podłączone do pinu T0.

Blok logiczny wyboru zegara, kontroluje wybór tego źródła oraz rodzaj zbrocza, które będzie licznik inkrementować lub dekrementować. Gdy źródło taktowania nie zostało

wybrane, licznik pozostaje nieaktywny. Zegar i podział preskalera wybiera się bitami CS02:0 w rejestrze TCCR0.

Preskaler jest współdzielony przez licznik 0 oraz licznik 1. W celu synchronizacji licznika z preskalerem (np. przy uruchomieniu licznika), możliwe jest zresetowanie preskalera bitem PSR10 w rejestrze SFIOR. Sygnał taktujący ze źródła zewnętrznego omija preskaler.

W trybie output compare zawartość rejestru OCR0 jest stale porównywana z wartością rejestru licznika TCNT0. Zdarzenie zrównania wartości licznika i rejestru OCR0 ustawi flagę OCF0, która może być źródłem żądania przerwania. Przerwanie output compare może zostać wygenerowane, jeśli ustawiona jest flaga przerwań OCIE0 oraz globalna flaga przerwań w SREG. Wynik porównania może służyć generowaniu przebiegów PWM lub sygnału o zmiennej częstotliwości na wyjściu pinu OC0. Praca w trybie z korygowaniem fazy zapobiega powstawaniu szpilek (glitch).

Najprostszy sposób pracy to tzw. tryb Normal, przy ustawieniu WGM01:0 = 0. W tym trybie licznik zawsze zlicza przez inkrementowanie, a po osiągnięciu maksymalnej wartości TOP = 0xFF licznik „przekręca się” i restartuje od 0. Flaga TOV0 zostaje ustawiona, gdy licznik osiągnie wartość 0. Normalnie ta flaga nie jest kasowana. Kasowanie automatyczne następuje przy wykorzystaniu przerwania przy przepełnieniu. W każdej chwili wartość licznika może być nadpisana inną liczbą. W trybie normalnym możliwe jest także generowanie przebiegu za pomocą Output Compare, jednak do tego celu zalecane jest skorzystanie z innych trybów.

## Rejestr TCCR0

### Rejestr kontroli licznika/timera 0

#### Rejestr TCCR0

Bit	7	6	5	4	3	2	1	0
	FOC0	WGM00	COM01	COM00	WGM01	SC02	CS01	CS00
Dostęp	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

FOC0 - Wymuszenie trybu output compare.

Bit ten jest aktywny tylko, gdy bity WGM określają tryb inny niż PWM. Dla zapewnienia kompatybilności z przyszłymi układami, ten bit musi być ustawiany na 0 przy zapisie do rejestru. Ustawienie tego bitu na 1 spowoduje natychmiastowe wymuszenie zdarzenia ‘compare’, a wyjście OC0 zostanie zmienione zgodnie z ustawieniami bitów COM01:0. Wymuszenie output compare nie spowoduje jednak ustawienia flagi OCF0 i wygenerowania przerwania.

WGM01:0 - Tryb generowania przebiegu.

Bity te kontrolują proces zliczania licznika. Obsługiwane są następujące tryby pracy licznika: normalny, kasowanie timera przy zrównaniu (compare match) oraz dwa tryby PWM.

WGM01	WGM00	Tryb pracy	TOP	Aktualizowanie OCR0	Ustawianie flagi TOV0
0	0	Normalny	0xFF	Natychmiast	MAX
0	1	PWM, korekcja fazy	0xFF	TOP	BOTTOM
1	0	CTC	OCR0	Natychmiast	MAX
1	1	Szybkie PWM	0xFF	TOP	MAX

TOP - wartość kończąca zliczanie (np. OCR0), MAX - osiągnięcie wartości (0xFF), BOTTOM - osiągnięcie minimum (0x00), CTC – kasowanie przy zrównaniu (clear timer on compare)

COM01:0 - Tryb pracy output compare (compare match output mode).  
Te bity kontrolują zachowanie pinu wyjściowego Output Compare (OC0). Jeśli któryś z tych bitów jest ustawiony, funkcja OC staje się nadrzędną nad innymi funkcjami IO tego pinu. Mimo to jego DDR musi być ustawione jako wyjściowe. Funkcja ustawiona przez bity COM01:0 jest uzależniona od ustawień WGM01:0.

Tryb nie-PWM (patrz WGM01:0):

COM01	COM00	Opis
0	0	Normalna praca portu, OC0 odłączone
0	1	Przełącza OC0 przy zrównaniu
1	0	Kasuje OC0 przy zrównaniu
1	1	Ustawia OC0 przy zrównaniu

Tryb szybkie PWM (patrz WGM01:0):

COM01	COM00	Opis
0	0	Normalna praca portu, OC0 odłączone
0	1	zarezerwowane
1	0	Kasuje OC0 przy zrównaniu, ustawia OC0 przy TOP
1	1	Ustawia OC0 przy zrównaniu, kasuje OC0 przy TOP

Tryb PWM, korygowana faza (patrz WGM01:0):

COM01	COM00	Opis
0	0	Normalna praca portu, OC0 odłączone
0	1	zarezerwowane
1	0	Kasuje OC0 przy zrównaniu podczas zliczania w górę, ustawia przy zrównaniu podczas zliczania w dół
1	1	Ustawia OC0 przy zrównaniu podczas zliczania w górę, kasuje przy zrównaniu podczas zliczania w dół

CS02:0 - Ustawienia źródła zegara dla licznika 0.

CS02	CS01	CS00	Opis
0	0	0	Nie wybrane źródło zegara, licznik zatrzymany
0	0	1	clk <sub>IO</sub> (bez preskalera)
0	1	0	clk <sub>IO</sub> / 8 (preskaler)
0	1	1	clk <sub>IO</sub> / 64 (preskaler)
1	0	0	clk <sub>IO</sub> / 256 (preskaler)
1	0	1	clk <sub>IO</sub> / 1024 (preskaler)
1	1	0	Zewnętrzne źródło zegara z pinu T0, zbocze opadające <sup>1)</sup>
1	1	1	Zewnętrzne źródło zegara z pinu T0, zbocze narastające <sup>1)</sup>

<sup>1)</sup> T0 może być źródłem taktowania nawet gdy jest ustawiony jako pin wyjściowy

## Rejestr TCNT0

### Rejestr licznika/timera 0

Rejestr pozwalający na odczyt stanu licznika lub jego zmianę. Modyfikacja zliczającego licznika może spowodować pominięcie momentu zrównania z OCR0.

#### Rejestr TCNT0

Bit	7	6	5	4	3	2	1	0
	TCNT0[7:0]							
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

## Rejestr OCR0

### Rejestr Output Compare

Zawartość rejestru OCR0 to 8-bitowa wartość, która stale jest porównywana z wartością rejestru licznika TCNT0. Moment ich zrównania może być wykorzystany do generowania przerwania, albo do generowania przebiegu na wyjściu pinu OC0.

#### Rejestr OCR0

Bit	7	6	5	4	3	2	1	0
	OCR0[7:0]							
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

## Rejestr TIMSK

### Rejestr maskowania przerwań liczników

#### Rejestr TIMSK

Bit	7	6	5	4	3	2	1	0
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

OCIE0 - Odblokowanie przerwania output compare licznika 0.  
Ustawienie bitu OCIE0 na 1 oraz flagi przerwania I w SREG odblokowuje przerwania output compare, które będzie generowane przez zrównanie się licznika 0 i rejestrem OCR0.

TOIE0 - Odblokowanie przerwania na przepełnienie licznika 0.  
Ustawienie bitu TOIE0 na 1 oraz flagi przerwania I w SREG odblokowuje przerwania na przepełnienie licznika 0 (overflow).

## Rejestr TIFR

### Rejestr flag przerwania liczników

#### Rejestr TIFR

Bit	7	6	5	4	3	2	1	0
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

OCF0 - Flaga output compare licznika 0.  
Bit OCF0 jest ustawiony gdy nastąpi zrównanie wartości rejestru OCR0 z licznikiem 0. Flaga jest sprzętowo kasowana przez wykonywanie kodu obsługi przerwania. W innym przypadku flaga może być kasowana przez wpisanie 1 (logicznej jedynki).

TOV0 - Flaga przepełnienia licznika 0.  
Bit TOV0 jest ustawiony gdy nastąpi przepełnienie licznika TCNT0. Flaga jest sprzętowo kasowana przez wykonywanie kodu obsługi przerwania, albo może być kasowana programowo przez wpisanie 0. W trybie PWM z korygowaniem fazy, bit TOV0 jest ustawiany przy zmianie kierunku zliczania przy wartości 0x00.

## Rejestr SFIOR

### Rejestr funkcji specjalnych

#### Rejestr SFIOR

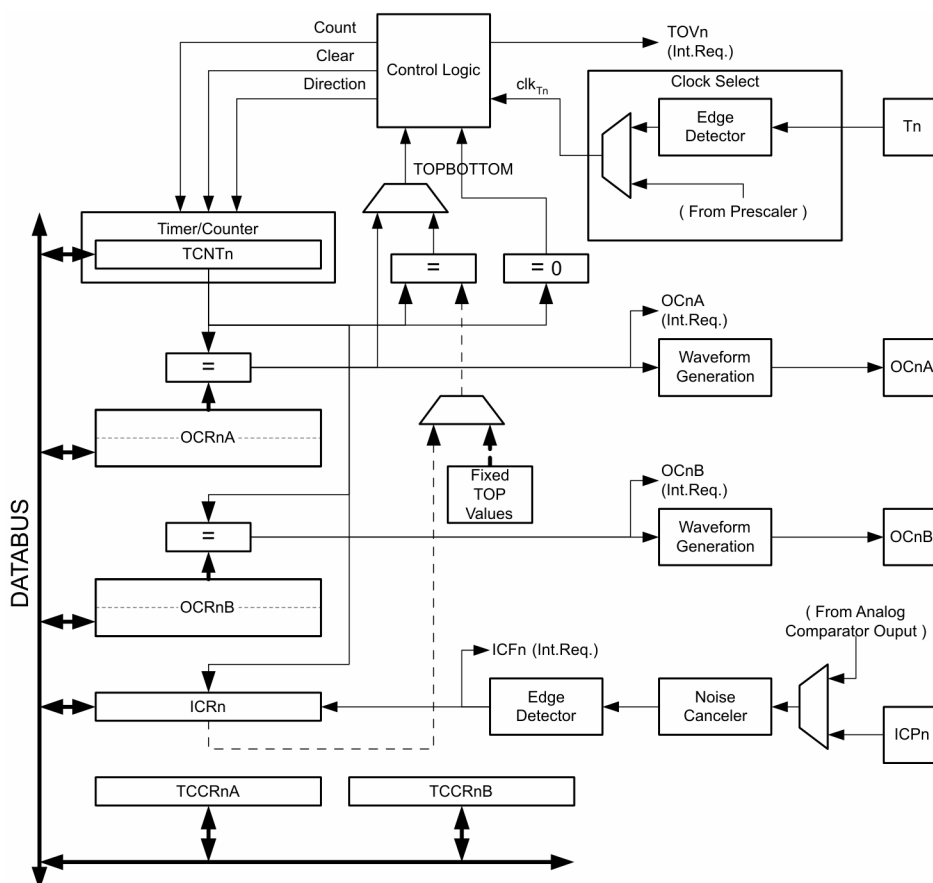
Bit	7	6	5	4	3	2	1	0
	ADTS2	ADTS1	ADTS0	-	ACME	PUD	PSR2	PSR10
Dostęp	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

PSR10 - Reset preskalera licznika 0 oraz licznika 1.  
Wpisanie 1 do tego bitu spowoduje reset preskalera. Należy pamiętać, że ten preskaler jest współdzielony przez liczniki 0 i 1, więc ta operacja wpłynie na oba liczniki. Po resetowaniu preskalera bit jest natychmiast kasowany sprzętowo.

## 6.2. 16-bitowy licznik 1

Główne właściwości oraz możliwości zastosowań licznika 1:

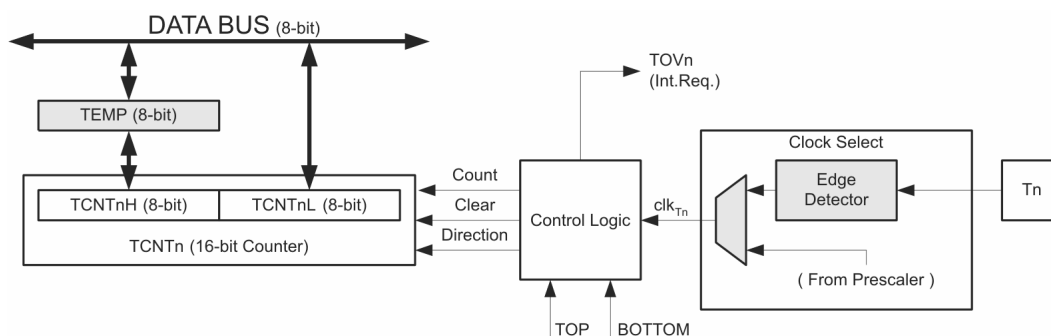
- Prawdziwa 16-bitowa konstrukcja (możliwy np. 16-bitowy PWM).
- Niezależne dwie jednostki Output Compare.
- Jednostka Input Capture z układem redukcji szumów.
- Automatyczne kasowanie i restart timera w trybie porównywania.
- Generator częstotliwości.
- Generator przebiegu PWM.
- Licznik zdarzeń zewnętrznych.
- 10-bitowy programowalny prescaler.
- 4 niezależne źródła przerwań.



Rys. 5. Schemat blokowy licznika 16-bitowego

Firma Atmel stale ulepsza swoje mikrokontrolery. Zmiany także dotyczą modułu licznika 16-bitowego. Obecny w ATmega16 różni się od wcześniejszych wersji ze starszych modeli AVR, więc starsze programy mogą nie działać poprawnie.

Dostęp do 16-bitowych rejestrów licznika1 (TCNT1, OCR1A/B, ICR1) musi się odbywać przez dwie operacje odczytu lub zapisu przez 8-bitową szynę danych. Przykładowo dla TCNT1, dostęp odbywa się poprzez dwa rejestry ośmiobitowe TCNT1L oraz TCNT1H. Dzięki buforowaniu części H (8-bitowy rejestr TEMP), odczyt lub modyfikacja mimo to odbywa się w jednym cyklu zegara (za wyjątkiem rejestrów OCR1A/B).



Rys. 6. Schemat blokowy jednostki zliczającej licznika 16-bitowego

Licznik może być taktowany wewnętrznie poprzez układ preskalera lub przez zewnętrzne źródło podłączone na pinie T1. Blok logiczny wyboru zegara, kontroluje wybór tego źródła oraz rodzaj zbocza, które będzie licznik inkrementować lub dekrementować.

Preskaler jest współdzielony przez licznik 0 oraz licznik 1. W celu synchronizacji licznika z preskalerem (np. przy uruchomieniu licznika), możliwe jest zresetowanie preskalera bitem PSR10 w rejestrze SFIOR. Sygnał taktujący ze źródła zewnętrznego omija preskaler.

W trybie output compare zawartość rejestrów OCR1A/B jest stale porównywana z wartością rejestru licznika TCNT1. Zdarzenie zrównania wartości licznika i rejestru OCR1A/B ustawi flagę OCF1A/B, która może być źródłem żądania przerwania. Wynik porównania może służyć generowaniu przebiegów PWM lub sygnału o zmiennej częstotliwości na wyjściu pinów OC1A/B.

Najprostszy sposób pracy to tzw. tryb Normal, przy ustawieniu WGM13:0 = 0. W tym trybie licznik zawsze zlicza przez inkrementowanie, a po osiągnięciu maksymalnej wartości (MAX = 0xFFFF) licznik „przekręca się” i restartuje od dolnej wartości (BOTTOM=0x0000). Flaga TOV1 zostaje ustawiona, gdy licznik osiągnie wartość 0. Normalnie flaga ta nie jest kasowana. Kasowanie automatyczne następuje przy wykorzystaniu przerwania przy przepełnieniu. W każdej chwili wartość licznika może być nadpisana inną liczbą. W trybie normalnym możliwe jest także generowanie przebiegu za pomocą Output Compare, jednak do tego celu zalecane jest skorzystanie z innych trybów.

Skorzystanie Input Compare najłatwiejsze jest w trybie Normal. Maksymalny odstęp między zdarzeniami na pinie ICP1 musi brać pod uwagę rozdzielczość licznika. W razie potrzeby należy użyć preskalera.

Rejestr ICR1 (Input Capture) może przechwycić wartość licznika na skutek zdarzenia zewnętrznego na pinie ICP1 albo na wyjściu komparatora analogowego (patrz opis bitu ACIC rejestru ACSR komparatora analogowego). W celu zredukowania wrażliwości na szumy, moduł Input Capture zawiera układ cyfrowej filtracji szumów.



Ustawienie źródła zegara dla licznika rozpoczyna jego pracę. Jeśli nie zostało wybrane, licznik pozostaje zatrzymany.

## Rejestr TCCR1A

### Rejestr A kontroli licznika/timera 1

#### Rejestr TCCR1A

Bit	7	6	5	4	3	2	1	0
	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10
Dostęp	R/W	R/W	R/W	R/W	W	W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

COM1A1:0 - Bity trybu pracy output compare dla kanału A (compare match output mode).

COM1B1:0 - Bity trybu pracy output compare dla kanału B .

Te bity kontrolują zachowanie pinów wyjściowych Output Compare (OC1A i OC1B). Jeśli któryś z tych bitów jest ustawiony, funkcja OC staje się nadrzędną nad innymi funkcjami IO tego pinu. Mimo to jego DDR musi być ustawione jako wyjściowe. Funkcje ustawione przez bity COM1x1:0 są uzależnione od ustawień WGM1x1:0.

Tryb nie-PWM (patrz WGM13:0):

COM1A1 / COM1B1	COM1A0 / COM1B0	Opis
0	0	Normalna praca portu, OC1A/OC1B odłączone
0	1	Przełącza OC1A/OC1B przy zrównaniu
1	0	Kasuje OC1A/OC1B przy zrównaniu (stan low)
1	1	Ustawia OC1A/OC1B przy zrównaniu (stan high)

Tryb szybkie PWM (patrz WGM13:0):

COM1A1 / COM1B1	COM1A0 / COM1B0	Opis
0	0	Normalna praca portu, OC1A/OC1B odłączone
0	1	Przy WGM13:0 = 15 przełącza OC1A przy zrównaniu, a OC1B odłączone. Pozostałe kombinacje WGM13:0 – normalna praca portu, OC1A/OC1B odłączone
1	0	Kasuje OC1A/OC1B przy zrównaniu, ustawia OC1A/OC1B przy TOP
1	1	Ustawia OC1A/OC1B przy zrównaniu, kasuje OC1A/OC1B przy TOP

Tryb PWM, korygowana faza (patrz WGM13:0):

COM1A1 / COM1B1	COM1A0 / COM1B0	Opis
0	0	Normalna praca portu, OC1A/OC1B odłączone

Tryb nie-PWM (patrz WGM13:0):

COM1A1 / COM1B1	COM1A0 / COM1B0	Opis
0	0	Normalna praca portu, OC1A/OC1B odłączone
0	1	Przy WGM13:0 = 9 lub 14 przełącza OC1A przy zrównaniu, a OC1B odłączone. Pozostałe kombinacje WGM13:0 – normalna praca portu, OC1A/OC1B odłączone
1	0	Kasuje OC1A/OC1B przy zrównaniu podczas zliczania w górę, ustawia przy zrównaniu podczas zliczania w dół
1	1	Ustawia OC1A/OC1B przy zrównaniu podczas zliczania w górę, kasuje przy zrównaniu podczas zliczania w dół

FOC1A - Wymuszenie trybu output compare dla kanału A.

FOC1B - Wymuszenie trybu output compare dla kanału B.

Bit ten jest aktywny tylko, gdy WGM13:0 określa tryb inny niż PWM. Dla zapewnienia kompatybilności z przyszłymi układami, ten bit musi być ustawiany na 0 podczas działania w trybie PWM. Ustawienie tego bitu na 1 spowoduje natychmiastowe wymuszenie zdarzenia 'compare', a wyjście OC1A/ OC1B zostanie zmienione zgodnie z ustawieniami bitów COM1x1:0.

Wymuszenie output compare nie spowoduje jednak ustawienia żadnej z flag OCF1A / B i wygenerowania przerwania.

WGM11:0 - Tryb generowania przebiegu.

W połączeniu z bitami WGM13:2 w rejestrze TCCR1B, bity te kontrolują proces zliczania licznika. Obsługiwane są następujące tryby pracy licznika: normalny, kasowanie timera przy zrównaniu (compare match - CTC) oraz trzy tryby PWM.

WGM13	WGM12	WGM11	WGM10	Tryb pracy licznika	TOP	Aktualizowanie OCR0	Ustawianie flagi TOV0
0	0	0	0	Normalny	0xFFFF	Natychmiast	MAX
0	0	0	1	PWM, korekcja fazy, 8-bit	0x00FF	TOP	BOTTOM
0	0	1	0	PWM, korekcja fazy, 9-bit	0x01FF	TOP	BOTTOM
0	0	1	1	PWM, korekcja fazy, 10-bit	0x03FF	TOP	BOTTOM
0	1	0	0	CTC	OCR1A	Natychmiast	MAX
0	1	0	1	Szybkie PWM, 8-bit	0x00FF	TOP	TOP
0	1	1	0	Szybkie PWM, 9-bit	0x01FF	TOP	TOP
0	1	1	1	Szybkie PWM, 10-bit	0x03FF	TOP	TOP
1	0	0	0	PWM, korekcja fazy i częstotliwości	ICR1	BOTTOM	BOTTOM
1	0	0	1	PWM, korekcja fazy i częstotliwości	OCR1A	BOTTOM	BOTTOM

WGM13	WGM12	WGM11	WGM10	Tryb pracy licznika	TOP	Aktualizowanie OCR0	Ustawianie flagi TOV0
1	0	1	0	PWM, korekcja fazy	ICR1	TOP	BOTTOM
1	0	1	1	PWM, korekcja fazy	OCR1A	TOP	BOTTOM
1	1	0	0	CTC	ICR1	Natychmiast	MAX
1	1	0	1	Zarezerwowane	-	-	-
1	1	1	0	Szybkie PWM	ICR1	TOP	TOP
1	1	1	1	Szybkie PWM	OCR1A	TOP	TOP

TOP - wartość kończąca zliczanie (np. z rejestru OCR1A, albo stałej), MAX - osiągnięcie wartości (0xff), BOTTOM - osiągnięcie minimum (0x00), CTC – kasowanie przy zrównaniu (clear timer on compare)

## Rejestr TCCR1B

### Rejestr B kontroli licznika/timera 1

#### Rejestr TCCR1B

Bit	7	6	5	4	3	2	1	0
	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
Dostęp	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

ICNC1 - Redukcja szumów pinu ICP1 (Input Capture).  
Ustawienie tego bitu na 1 włącza układ redukcji szumu na wejściu Input Capture. Zdarzenie zmiany stanu pinu następuje, gdy nowy poziom utrzymuje się przez kolejne czterokrotne próbkowanie tego pinu. Opóźnia to więc wykrycie zmiany na ICP1 o 4 cykle oscylatora.

ICES1 - Wybór zbocza zdarzenia na pinie ICP1.  
Bit ten służy do określenia rodzaju zbocza, które będzie interpretowane jako zdarzenie Input Capture na pinie ICP1. ICES1 = 0 – oznacza wybór zbocza opadającego, ICES1= 1 to zbocze narastające.

Wychwycenie zdarzenia powoduje że aktualna wartość licznika 1 jest kopiowana do rejestru ICR1. Ustawiana jest także flaga ICF1, co może być użyte do generowania przerwania.

Jeśli ICR1 jest użyte jako wartość TOP (zobacz opis bitów WGM13:0), to funkcja Input Capture na pinie ICP1 jest wyłączona.

Bit 5 - Zarezerwowany do przyszłych zastosowań, zapisując rejestr wpisywać 0.

WGM13:2 - Tryb generowania przebiegu, zobacz opis TCCR1A.

CS12:0 - Ustawienia źródła zegara dla licznika.

CS12	CS11	CS10	Opis
0	0	0	Nie wybrane źródło zegara, licznik zatrzymany
0	0	1	clk <sub>IO</sub> (bez preskalera)
0	1	0	clk <sub>IO</sub> / 8 (preskaler)
0	1	1	clk <sub>IO</sub> / 64 (preskaler)

CS12	CS11	CS10	Opis
1	0	0	clk <sub>IO</sub> / 256 (preskaler)
1	0	1	clk <sub>IO</sub> / 1024 (preskaler)
1	1	0	Zewnętrzne źródło zegara z pinu T1, zbocze opadające
1	1	1	Zewnętrzne źródło zegara z pinu T1, zbocze narastające

Pin T1 może być źródłem taktowania nawet gdy jest ustawiony jako pin wyjściowy.

## Rejestry TCNT1H i TCNT1L

### Rejestry licznika 1

#### Rejestr TCNT1

Bit	7	6	5	4	3	2	1	0
TCNT1H	TCNT1[15:8]							
TCNT1L	TCNT1[7:0]							
Dostęp	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

Dwie lokacje (TCNT1H, TCNT1L) dają bezpośredni dostęp do 16-bitowej jednostki licznika. Aby operacja odczytu lub zapisu była wykonywana jednocześnie na całych 16 bitach, dostęp do tego rejestru jest realizowany za pomocą 8-bitowego tymczasowego rejestru TEMP. Przez niego odbywa się dostęp do części High. Rejestr tymczasowy wykorzystywany jest w ten sposób przez wszystkie rejestry 16-bitowe.

Uwaga: Ze względu na 8-bitową szynę danych mikrokontrolera, dostęp do rejestrów odbywa się poprzez dwie operacje na jedno bajtowych częściach tego rejestru, najpierw starszej ICR1H, po czym młodszej ICR1L.

#### Rejestr OCR1A

##### Rejestr Output Compare 1A

#### Rejestr OCR1A

Bit	7	6	5	4	3	2	1	0
OCR1AH	OCR1A[15:8]							
OCR1AL	OCR1A[7:0]							
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

Patrz opis rejestru OCR1B.

#### Rejestr OCR1B

##### Rejestr Output Compare 1B

#### Rejestr OCR1B

Bit	7	6	5	4	3	2	1	0
OCR1BH	OCR1B[15:8]							
OCR1BL	OCR1B[7:0]							
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

Rejestry OCR1A i OCR1B zawierają 16-bitową wartość, która stale jest porównywana z wartością rejestru licznika 1 (TCNT1). Moment ich zrównania może być wykorzystany do generowania przerwania, albo do generowania przebiegu na wyjściu pinów OC1A OC1B.

Uwaga: Ze względu na 8-bitową szynę danych mikrokontrolera, dostęp do rejestrów odbywa się poprzez dwie operacje na jedno bajtowych częściach tego rejestru, najpierw starszej OCR1BH, po czym młodszej OCR1BL.

## Rejestr ICR1

### Rejestr Input Capture 1 (ICR1H, ICR1L)

#### Rejestr ICR1

Bit	7	6	5	4	3	2	1	0
ICR1H	ICR1[15:8]							
ICR1L	ICR1[7:0]							
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

Rejestr Input Capture jest aktualizowany wartością licznika 1 (TCNT1) za każdym wystąpieniem zdarzenia na pinie ICP1 (lub opcjonalnie na wyjściu komparatora analogowego, patrz opis rejestru ACSR komparatora analogowego). Rejestr ICR1 może być również użyty do określania górnej wartości (TOP) dla licznika.

Uwaga: Ze względu na 8-bitową szynę danych mikrokontrolera, dostęp do rejestrów odbywa się poprzez dwie operacje na jedno bajtowych częściach tego rejestru, najpierw starszej ICR1H, po czym młodszej ICR1L.

## Rejestr TIMSK

### Rejestr maskowania przerwań liczników

#### Rejestr TIMSK

Bit	7	6	5	4	3	2	1	0
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

- TICIE1 - Odblokowanie przerwania Input Capture licznika 1.  
Ustawienie bitu TICIE0 na 1 oraz flagi przerwań I w SREG odblokowuje przerwanie Input Capture.
- OCIE1A - Odblokowanie przerwania Output Compare A licznika 1.  
Ustawienie bitu OCIE1A na 1 oraz flagi przerwań I w SREG odblokowuje przerwanie output compare A, które będzie generowane przez zrównanie się licznika 1 i rejestrem OCR1A.
- OCIE1B - Odblokowanie przerwania Output Compare B licznika 1.  
Ustawienie bitu OCIE1B na 1 oraz flagi przerwań I w SREG odblokowuje przerwanie output compare B, które będzie generowane przez zrównanie się licznika 1 i rejestrem OCR1B.
- TOIE1 - Odblokowanie przerwania na przepełnienie licznika 1.

Ustawienie bitu TOIE1 na 1 oraz flagi przerwania I w SREG odblokowuje przerwanie na przepełnienie licznika 1.

## Rejestr TIFR

### Rejestr flag przerwania liczników

#### Rejestr TIFR

Bit	7	6	5	4	3	2	1	0
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

- ICF1 - Flaga Input Capture licznika 1.  
 Flaga ICF0 jest ustawiana, gdy nastąpi zdarzenie na pinie ICP1. Jeśli rejestr ICR1 jest używany jako górna wartość (TOP) poprzez bity WGM13:0, wówczas flaga ICF1 jest ustawiana, kiedy licznik osiągnie górną wartość. Flaga jest sprzętowo kasowana przez wykonywanie kodu obsługi przerwania. W innym przypadku flaga może być kasowana przez wpisanie 1 (logicznej jedynki).
- OCF1A - Flaga Output Compare A licznika 1.  
 Flaga OCF1A jest ustawiana, gdy nastąpi zrównanie wartości rejestru OCR1A z wartością licznika 1 (TCNT1). Flaga jest sprzętowo kasowana przez wykonywanie kodu obsługi przerwania. W innym przypadku flaga może być kasowana przez wpisanie 1 (logicznej jedynki).
- OCF1B - Flaga Output Compare B licznika 1.  
 Flaga OCF1B jest ustawiana gdy nastąpi zrównanie wartości rejestru OCR1B z wartością licznika 1 (TCNT1). Flaga jest sprzętowo kasowana przez wykonywanie kodu obsługi przerwania. W innym przypadku flaga może być kasowana przez wpisanie 1 (logicznej jedynki).
- TOV1 - Flaga przepełnienia licznika 1.  
 Ustawienie tej flagi jest zależne od ustawienia bitów WGM13:0. W trybie Normalnym lub CTC (clear timer on compare) flaga TOV1 jest ustawiana przy przepełnieniu licznika.  
 W innych przypadkach zachowanie flagi opisane jest w tabeli przy opisie rejestru TCCR1A (bity WGM). Flaga jest sprzętowo kasowana przez wykonywanie kodu obsługi przerwania. W innym przypadku flaga może być kasowana przez wpisanie 1 (logicznej jedynki).

### 6.3. 8-bitowy licznik 2 z PWM i pracą asynchroniczną

Główne właściwości oraz możliwości zastosowań 8-bitowego licznika 2:

- Licznik jedno kanałowy.
- Automatyczne kasowanie i restart timera w trybie porównania.

- Generator częstotliwości.
- Generator przebiegu PWM.
- Licznik zdarzeń zewnętrznych.
- 10-bitowy programowalny preskaler.
- Źródło przerwań (przepełnienie, zrównanie).
- Możliwe taktowanie z zewnętrznego źródła niezależnie od zegara wewnętrznego.

Rejestr licznika (TCNT2) oraz rejestr output compare (OCR2) są 8-bitowe. Licznik może być taktowany synchronicznie ze źródła wewnętrznego przez układ preskalera lub asynchronicznie z zewnętrznych pinów TOSC1/2. Asynchroniczne operacje są definiowane w rejestrze ASSR. Piny TOSC1/2 są przystosowane do bezpośredniego podłączenia zegarkowego oscylatora kwarcowego 32kHz bez dodatkowych kondensatorów.

Blok logiczny wyboru zegara, kontroluje wybór tego źródła oraz rodzaj zbocza, które będzie licznik inkrementować lub dekrementować. Gdy źródło taktowania nie zostało wybrane, licznik pozostaje nieaktywny. Zegar i preskaler określa się bitami CS22:0 w rejestrze TCCR2.

Licznik 2 posiada swój indywidualny preskaler, który daje możliwość wybrania jednego z siedmiu dzielników częstotliwości. W celu synchronizacji licznika z preskalerem (np. przy uruchomieniu licznika), możliwe jest zresetowanie preskalera bitem PSR2 w rejestrze SFIOR.

W trybie output compare zawartość rejestru OCR2 jest stale porównywana z wartością rejestru licznika TCNT2. Zdarzenie zrównania wartości licznika i rejestru OCR2 ustawi flagę OCF2, która może być źródłem żądania przerwania. Przerwanie output compare może zostać wygenerowane, jeśli ustawiona jest flaga przerwań OCIE2 oraz globalna flaga przerwań w SREG. Wynik porównania może służyć generowaniu przebiegów PWM lub sygnału o zmiennej częstotliwości na wyjściu pinu OC2. Praca w trybie z korygowaniem fazy zapobiega powstawaniu szpilek (glitch).

Najprostszy sposób pracy to tzw. tryb Normal, przy ustawieniu WGM21:0 = 0. W tym trybie licznik zawsze zlicza przez inkrementowanie, a po osiągnięciu górnej wartości TOP = 0xFF licznik „przekręca się” i restartując od 0. Flaga TOV2 zostaje ustawiona gdy licznik osiągnie wartość 0. Normalnie ta flaga nie jest kasowana. Kasowanie automatyczne następuje przy wykorzystaniu przerwania przy przepełnieniu. W każdej chwili wartość licznika może być nadpisana inną liczbą. W trybie normalnym możliwe jest także generowanie przebiegu za pomocą Output Compare, jednak do tego celu zalecane jest skorzystanie z innych trybów.

## Rejestr TCCR2

### Rejestr kontroli licznika/timera 2

#### Rejestr TCCR2

Bit	7	6	5	4	3	2	1	0
	FOC2	WGM20	COM21	COM20	WGM21	SC22	CS21	CS20
Dostęp	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

FOC2 - Wymuszenie trybu output compare.

Bit ten jest aktywny tylko, gdy bity WGM określają tryb inny niż PWM. Dla zapewnienia kompatybilności z przyszłymi układami, ten bit musi być ustawiany na 0 przy zapisie do rejestru. Ustawienie tego bitu na 1 spowoduje natychmiastowe wymuszenie zdarzenia 'compare', a wyjście OC2 zostanie zmienione zgodnie z ustawieniami bitów COM21:0. Wymuszenie output compare nie spowoduje jednak ustawienia flagi OCF2 i wygenerowania przerwania.

WGM21:0 - Tryb generowania przebiegu.

Bity te kontrolują proces zliczania licznika. Obsługiwane są następujące tryby pracy licznika: normalny, kasowanie timera przy zrównaniu (CTC) oraz dwa tryby pracy PWM.

WGM21	WGM20	Tryb pracy	TOP	Aktualizowanie OCR2	Ustawianie flagi TOV2
0	0	Normalny	0xFF	Natychmiast	MAX
0	1	PWM, korekcja fazy	0xFF	TOP	BOTTOM
1	0	CTC	OCR2	Natychmiast	MAX
1	1	Szybkie PWM	0xFF	TOP	MAX

TOP - wartość kończąca zliczanie (np. OCR2), MAX - osiągnięcie wartości (0xFF), BOTTOM - osiągnięcie minimum (0x00), CTC – kasowanie przy zrównaniu (clear timer on compare)

COM21:0 - Tryb pracy output compare (compare match output mode).

Te bity kontrolują zachowanie pinu wyjściowego OC2. Jeśli któryś z tych bitów jest ustawiony, funkcja OC staje się nadrzędną nad innymi funkcjami IO tego pinu. Mimo to jego DDR musi być ustawione jako wyjściowe. Funkcja ustawiona przez bity COM21:0 jest uzależniona od ustawień WGM21:0.

Tryb nie-PWM (patrz WGM21:0):

COM21	COM20	Opis
0	0	Normalna praca portu, OC2 odłączone
0	1	Przełączanie OC2 przy zrównaniu
1	0	Kasowanie OC2 przy zrównaniu
1	1	Ustawianie OC2 przy zrównaniu

Tryb szybkie PWM (patrz WGM21:0):

COM21	COM20	Opis
0	0	Normalna praca portu, OC2 odłączone
0	1	Zarezerwowane
1	0	Kasowanie OC2 przy zrównaniu, ustawia OC2 przy TOP



Tryb nie-PWM (patrz WGM21:0):

COM21	COM20	Opis
0	0	Normalna praca portu, OC2 odłączone
1	1	Ustawianie OC2 przy zrównaniu, kasuje OC2 przy TOP

Tryb PWM, korygowana faza (patrz WGM21:0):

COM21	COM20	Opis
0	0	Normalna praca portu, OC2 odłączone
0	1	zarezerwowane
1	0	Kasowane OC0 przy zrównaniu podczas zliczania w górę, ustawia przy zrównaniu podczas zliczania w dół
1	1	Ustawiane OC0 przy zrównaniu podczas zliczania w górę, kasuje przy zrównaniu podczas zliczania w dół

CS22:0 - Ustawienia źródła zegara dla licznika 2:

CS22	CS21	CS20	Opis
0	0	0	Nie wybrane źródło zegara, licznik zatrzymany
0	0	1	$\text{clk}_{T2S}$ (bez preskalera)
0	1	0	$\text{clk}_{T2S} / 8$ (preskaler)
0	1	1	$\text{clk}_{T2S} / 32$ (preskaler)
1	0	0	$\text{clk}_{T2S} / 64$ (preskaler)
1	0	1	$\text{clk}_{T2S} / 128$ (preskaler)
1	1	0	$\text{clk}_{T2S} / 256$ (preskaler)
1	1	1	$\text{clk}_{T2S} / 1024$ (preskaler)

## Rejestr TCNT2

### Rejestr licznika/timera 2

Rejestr pozwalający na odczyt lub zmianę stanu licznika 2. Modyfikacja zliczającego licznika może spowodować pominięcie momentu zrównania z OCR2.

#### Rejestr TCNT2

Bit	7	6	5	4	3	2	1	0
	TCNT2[7:0]							
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

#### Rejestr OCR2

### Rejestr Output Compare

Zawartość rejestru OCR2 stale jest porównywana z wartością rejestru licznika TCNT2. Moment ich zrównania może być wykorzystany do generowania przerwania, albo do generowania przebiegu na wyjściu pinu OC2.

### Rejestr OCR2

Bit	7	6	5	4	3	2	1	0
	OCR2[7:0]							
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

### Rejestr ASSR

#### Rejestr asynchronicznych operacji licznika

W zamyśle producenta, tryb asynchroniczny jest przygotowany do roli licznika/zegara czasu rzeczywistego (RTC). Do pinów TOSC1/2 mikrokontrolera można podłączyć bezpośrednio kwarc zegarkowy 32kHz, do tego należy ustawić bit AS2 w rejestrze ASSR, jednocześnie wyłączając inne funkcje I/O pinów TOSC1/2 (PC6, PC7).

### Rejestr ASSR

Bit	7	6	5	4	3	2	1	0
	-	-	-	-	AS2	TCN2UB	OCR2UB	TCR2UB
Dostęp	R	R	R	R	R/W	R	R	R
Wartość początkowa	0	0	0	0	0	0	0	0

### Rejestr TIMSK

#### Rejestr maskowania przerwań liczników

### Rejestr TIMSK

Bit	7	6	5	4	3	2	1	0
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

OCIE2 - Odblokowanie przerwania output compare licznika 2.

Ustawienie bitu OCIE2 na 1 oraz flagi przerwań I w SREG odblokowuje przerwanie output compare, które będzie generowane przez zrównanie się zawartości licznika 2 i wartości w rejestrze OCR2.

TOIE2 - Odblokowanie przerwania na przepełnienie licznika 2.

Ustawienie bitu TOIE2 na 1 oraz flagi przerwań I w SREG odblokowuje przerwanie na przepełnienie licznika 2.

### Rejestr TIFR

#### Rejestr flag przerwań liczników

### Rejestr TIFR

Bit	7	6	5	4	3	2	1	0
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

OCF2 - Flaga output compare licznika 2.

Bit OCF2 jest ustawiony, gdy nastąpi zrównanie wartości rejestru OCR2 z licznikiem 2. Flaga jest sprzętowo kasowana przez wykonywanie kodu obsługi przerwania. W innym przypadku flaga może być kasowana przez wpisanie 1 (logicznej jedynki).

TOV2 - Flaga przepełnienia licznika 2.

Bit TOV2 jest ustawiony, gdy nastąpi przepełnienie licznika TCNT2. Flaga jest sprzętowo kasowana przez wykonywanie kodu obsługi przerwania, albo może być kasowana programowo przez wpisanie 0. W trybie PWM z korygowaniem fazy, bit TOV2 jest ustawiany przy zmianie kierunku zliczania przy wartości 0x00.

## Rejestr SFIOR

### Rejestr funkcji specjalnych

#### Rejestr SFIOR

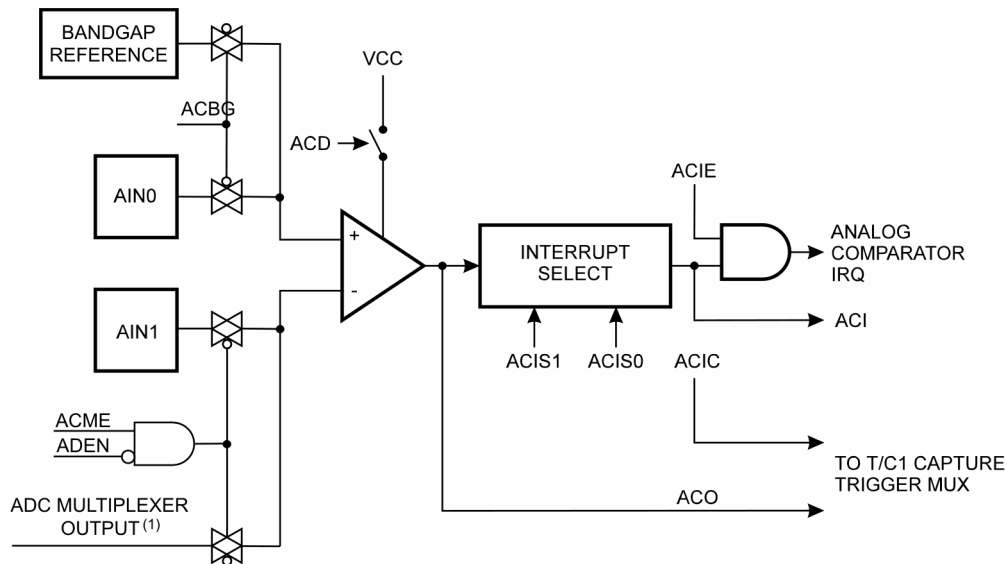
Bit	7	6	5	4	3	2	1	0
	ADTS2	ADTS1	ADTS0	-	ACME	PUD	PSR2	PSR10
Dostęp	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

PSR2 - Reset preskalera licznika 2.

Wpisanie 1 do tego bitu spowoduje reset preskalera licznika 2. Po resetie preskalera bit jest automatycznie ustawiany na 0.

## 7. Komparator analogowy

Komparator analogowy porównuje wartości wejściowe na pinie dodatnim AIN0 i ujemnym AIN1. Jeśli napięcie na pinie dodatnim jest większe niż na ujemnym, ustawiane jest wyjście ACO komparatora. Pojawienie się sygnału na wyjściu komparatora może służyć jako moment przechwycenia dla funkcji Input Capture licznika 1. Oprócz tego zmiana zbocza na wyjściu komparatora może być źródłem niezależnego przerwania komparatora. Za pomocą multipleksera ADC możliwe jest podłączenie jednego z pinów ACD7..0 do wejścia komparatora.



Rys. 7. Schemat blokowy komparatora analogowego

## Rejestr SFIOR

### Rejestr funkcji specjalnych

#### Rejestr SFIOR

Bit	7	6	5	4	3	2	1	0
	ADTS2	ADTS1	ADTS0	-	ACME	PUD	PSR2	PSR10
Dostęp	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

ACME - Odblokowanie multipleksera komparatora analogowego. Wpisanie 1 do tego bitu podczas gdy przetwornik ADC jest wyłączony, podłącza wyjście multipleksera ADC do ujemnego wejścia komparatora.

## Rejestr ACSR

### Rejestr kontroli i statusu komparatora analogowego

#### Rejestr ACSR

Bit	7	6	5	4	3	2	1	0
	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0
Dostęp	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	n.d.	0	0	0	0	0

ACD - Odłączenie komparatora analogowego. Wpisanie 1 do tego bitu odłącza zasilanie od komparatora analogowego. Bit może być ustawiony w dowolnej chwili, aby komparator został wyłączony, co obniża pobór mocy. Przed zmianą stanu bitu ACD, przerwanie komparatora musi być zablokowane, przez wykasowanie bitu ACIE (w ACSR), gdyż w przeciwnym razie nastąpi wygenerowanie przerwania.

ACBG - Wybranie źródła referencyjnego (Bandgap). Kiedy ten bit jest ustawiony na 1, do wejścia dodatniego komparatora, zamiast pinu AIN0, podłączone jest referencyjne źródło napięciowe bandgap

(skompensowane termicznie) o napięciu ok. 1.23V (min.1,15V max.1,35V). Jeśli bit ma wartość 0, to do dodatniego wejścia komparatora podłączony jest pin AIN0.

- ACO - Wyjście komparatora analogowego.  
Wyjście komparatora analogowego jest synchronizowane i bezpośrednio podłączone do ACO. Synchronizacja wprowadza opóźnienie ok. 1-2 cykli zegara. Bit ACO będzie miał wartość 1 gdy napięcie pinie AIN0 (dodatnie wejście komparatora) jest wyższe niż na AIN1 (ujemne wejście).
- ACI - Flaga przerwania komparatora analogowego.  
Bit ten jest ustawiany sprzętowo kiedy na wyjściu komparatora analogowego nastąpi zdarzenie określone jako źródło przerwania bitami ACIS1:0. Kod obsługi przerwania zostanie wykonany jeśli przerwanie od komparatora jest odblokowane (bit ACIE) oraz gdy ustawiona jest globalna flaga przerw. Flaga ACI jest kasowana automatycznie przy wykonywaniu kodu obsługi przerwania lub przez wpisanie logicznej jedynki do tego bitu.
- ACIE - Odblokowanie przerwania komparatora analogowego.  
Ustawienie tej flagi na 1 oraz globalnej flagi przerw, aktywuje przerwanie na zdarzenie na wyjściu komparatora analogowego (określone bitami ACIS). Wpisanie wartości 0 blokuje przerwanie.
- ACIC - Funkcja Input Capture z komparatora analogowego.  
Wpisanie do tego bitu 1, pozwala aby aktualizacja rejestru ICR1 (Input Capture licznika 1) nastąpiła na skutek zdarzenia na wyjściu komparatora analogowego. Wyjście komparatora zostaje bezpośrednio połączone do układu Input Capture (patrz schemat blokowy licznika 1).
- ACIS1:0 - Wybór sposobu wywołania przerwania komparatora analogowego.  
Tymi bitami określa się, jaki rodzaj zdarzenia na wyjściu komparatora analogowego może być źródłem przerwania od komparatora. Przed zmianą stanu bitów ACIS, przerwanie komparatora musi być zablokowane (bit ACIE w ACSR), gdyż w przeciwnym razie na skutek zmiany nastąpi wygenerowanie przerwania.

ACIS1	ACIS0	Tryb przerwania komparatora
0	0	Przerwanie od przełączenia wyjścia
0	1	Zarezerwowane
1	0	Przerwanie od zbocza opadającego na wyjściu
1	1	Przerwanie od zbocza narastającego na wyjściu

### Przełączane wejście komparatora

Do ujemnego wejścia komparatora analogowego, zamiast AIN1 może być podłączony jeden z pinów ADC7..0. Do tego używany jest multiplexer wejść przetwornika ADC, pod warunkiem że ADC jest wyłączony.

Należy odblokować multiplexer komparatora ustawieniem bitu ACME w rejestrze SFIOR oraz wyłączyć przetwornik ADC bitem ADEN w rejestrze ADCSRA. Od tej chwili bitami MUX2:0 w rejestrze ADMUX dokonuje się wyboru wejścia pinów ACDn.

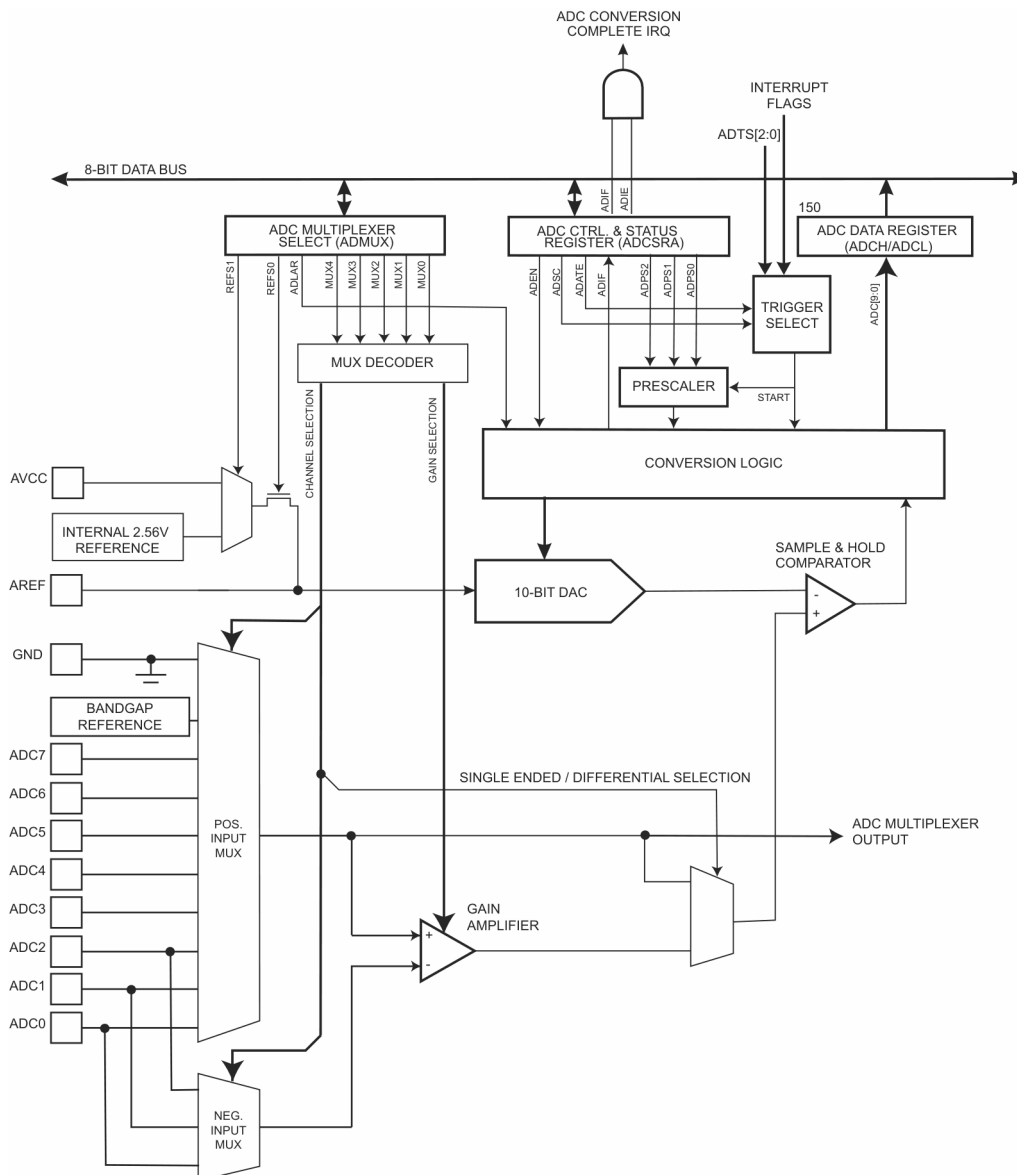
Tabela 2.7. Multipleksowane wejście komparatora

ACME	ADEN	MUX2:0	Wejście ujemne komparatora
0	x	xxx	AIN1
1	1	xxx	AIN1
1	0	000	ADC0
1	0	001	ADC1
1	0	010	ADC2
1	0	011	ADC3
1	0	100	ADC4
1	0	101	ADC5
1	0	110	ADC6
1	0	111	ADC7

## 8. Przetwornik ADC

Mikrokontroler ATmega16 wyposażony jest w 10-bitowy przetwornik ADC z sukcesywną aproksymacją. ADC podłączony jest do 8 kanałowego multiplexera pozwalając na przyłączenie ośmiu napięć wejściowych do każdego z pinów portu A.

Możliwe jest także 16 kombinacji podłączenia napięcia różnicowego. Dwa z różnicowych wejść (ADC1, ADC0 oraz ADC3, ADC2) są wyposażone w programowalny stopień wzmacniający, pozwalając na wzmocnienie 0dB (1x), 20dB (10x) i 46dB (200x) zanim sygnał wejściowy zostanie poddany konwersji. Tryb różnicowy nie będzie tu szczegółowo opisany, gdyż wspierany jest w układach ATmega16 w obudowach innych niż użyta w pracy PDIP40.



Rys. 8. Schemat blokowy przetwornika ADC

Do zasilania przetwornika został specjalnie wyprowadzony pin AVCC. Napięcie zasilające ADC nie może się różnić więcej niż o  $\pm 0,3$  od VCC. Dla lepszego zabezpieczenia przed szumami z zewnątrz, możliwe jest zasilanie przetwornika z filtra dolnoprzepustowego LC. Płytkę testową zestawu laboratoryjnego została w taki filtr wyposażona.

ADC wyposażony jest także w zintegrowane źródło napięcia referencyjnego o wartości 2,56V. Dodatkowo można podłączyć do pinu VREF kondensator filtrujący. Przez pin VREF można także podłączyć zewnętrzne napięcie referencyjne.

## Działanie

Minimalna przetwarzana wartość reprezentowana jest przez GND, a maksymalna przez napięcie na pinie VREF minus 1LSB. Źródłem napięcia referencyjnego może być AVCC, wewnętrzne źródło referencyjne 2,56V albo zewnętrzne napięcie podłączane do pinu VREF.

Przy korzystaniu z wewnętrznego źródła także możliwe też jest podłączenie do pinu VREF kondensatora odsprzęgającego, poprawiającego odporność na zakłócenia.

Przetwornik jest włączany przez ustawienie bitu ADEN w rejestrze ADCSRA. Wcześniejsze wybieranie źródła referencyjnego czy kanału nie przyniesie więc żadnego efektu. Zanim ADEN nie zostanie ustawiony, to przetwornik pozostaje wyłączony i nie pobiera prądu. Zalecane więc jest wyłączenie ADC zanim mikrokontroler będzie wprowadzony w tryb uśpienia.

ADC normalnie pracuje w trybie 10 bitowym, może też pracować w trybie 8 bitowym. Wynik konwersji przechowywany jest w rejestrach danych przetwornika ADCH i ADCL. Postać 10 bitowa jest reprezentowana w postaci „right adjusted” (wyrównanej do prawej) i odczytuje się dwuetapowo, najpierw z ADCL po czym ADCH. Odczytanie samego ADCL zablokuje aktualizowanie wyniku konwersji aż do odczytania ADCH. Jeśli wynik nie zostanie odczytany, to może zostać utracony poprzez nadpisanie wynikiem z kolejnej konwersji. Tryb „left adjusted” jest 8 bitowy, a do odczytania wyniku wystarczy odczytać rejestr ADCH.

Zakończona konwersja może być zdarzeniem generującym przerwanie.

### **Konwersja pojedyncza**

Pojedyncza konwersja rozpoczyna się wpisaniem logicznej jedynki do bitu ADSC, a po jej zakończeniu bit jest automatycznie kasowany.

### **Konwersja wyzwalana automatycznie**

Konwersja automatyczna może być wyzwalana przez różne źródła. Automatyczne wyzwalanie włączane jest przez ustawienie bitu ADATE w ADCSRA. Źródło wybierane jest za pomocą bitów ADTS w rejestrze SFIOR. Pojawienie się zdarzenia wyzwalającego powoduje zresetowanie preskalera i rozpoczęcie konwersji. Pozwala to na wyzwalanie konwersji w stałych odstępach czasu (np. gdy do wyzwalania zostanie skonfigurowany timer). Flaga przerwania będzie ustawiana bez względu na to czy globalne przerwanie w rejestrze statusu SREG są włączone czy nie. Automatyczne wyzwalanie nie musi generować przerwania, ale wówczas flaga przerwania i tak musi być wyzerowana przed kolejnym wyzwalającym zdarzeniem.

### **Preskaler**

Wbudowany przetwornik z sukcesywną aproksymacją wymaga aby zegar taktujący mieścił się w granicach 50-200kHz (ustawiane w ADCSRA). Gdy rozdzielczość przetwornika może być mniejsza, to można zwiększyć szybkość zegara i tym samym szybkość próbkowania.

Moduł przetwornika ADC posiada układ preskalera, który może podzielić zegar CPU, gdy jest szybszy niż 100kHz. Preskaler ustawia się bitami ADPS2:0 w rejestrze ADCSRA. Rozpoczyna on pracę od momentu włączenia ADC bitem ADEN w ADCSRA, a konwersja



może się rozpocząć od kolejnego narastającego zbocza zegara ADC (nie mylić włączenia przetwornika z rozpoczęciem konwersji).

Normalna konwersja w trybie ciągłym trwa 13 cykli zegara ADC. Pierwsza konwersja po ustawieniu ADRN wymaga natomiast 25 cykli, z których część potrzebna jest na zainicjowanie analogowej części przetwornika.

Po zakończeniu konwersji wynik zapisywany jest w rejestrach ADCL ADCH oraz ustawiany jest bit ADIF. W trybie wyzwalania pojedynczego bit ADSC za każdym razem jest kasowany. Kolejne programowe ustawienie tego bitu rozpoczyna kolejną konwersję.

W czasie pracy z wyzwalaniem automatycznym (auto triggered conversion) prescaler jest resetowany kiedy pojawia się zdarzenie. Zapewnia to stałe opóźnienie czasowe od zdarzenia wyzwalającego do rozpoczęcia konwersji.

### Wybór kanału i źródła referencyjnego

W rejestrze ADMUX znajdują się bity MUX4:0 oraz REFS1:0 odpowiedzialne za wybór kanału i źródła referencyjnego.

W trybie wyzwalania pojedynczego, kanał zawsze należy wybierać przed rozpoczęciem konwersji. Zmianę najłatwiej jest wykonywać po zakończeniu konwersji poprzedniej. Zmiana kanału czy źródła referencyjnego w czasie trwającej konwersji zostanie zapamiętana a multiplexer uaktualniony przed rozpoczęciem kolejnej konwersji.

Przy pracy z wyzwalaniem automatycznym należy zwrócić uwagę na przełączanie kanału czy źródła, gdyż moment wyzwolenia może być nieprzewidywalny.

Jeśli podłączone jest zewnętrzne źródło napięcia ref. do pinu VREF to nie można wybierać innego wewnętrznego, gdyż nastąpi zwarcie z tym podłączonym do zewnętrznego pinu. Jeśli natomiast zewnętrzne źródło nie jest podłączone, to można wybrać jako referencyjne AVCC, albo wewnętrzny wzorzec 2,56V. Pierwsza konwersja po zmianie źródła może być niedokładna i powinna być pominięta.

### Rejestr ADMUX

#### Rejestr konfiguracji multiplexera ADC

#### Rejestr ADMUX

Bit	7	6	5	4	3	2	1	0
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

REFS1:0 - Bity wyboru źródła referencyjnego.

Jeśli te bity zostaną zmienione podczas trwającej konwersji, to zmiana będzie uwzględniona dopiero przy kolejnej konwersji (ustawiony ADIF w rejestrze ADCSRA). Nie można używać wewnętrznych źródeł, jeśli podłączone jest napięcie do pinu VREF.

REFS1	REFS0	Wybrane źródło referencyjne
0	0	pin AREF, wewnętrzne Vref wyłączone

REFS1	REFS0	Wybrane źródło referencyjne
0	1	AVCC z zewnętrznym kondensatorem podłączonym do VREF
1	0	Kombinacja zarezerwowana
1	1	Wewnętrzne 2,56V z zewnętrznym kondensatorem przy pinie VREF

ADLAR - Wyrównanie wyniku konwersji do lewej.  
Ustawienie bitu ADLAR wpływa na reprezentację wyniku w rejestrach danych przetwornika. Wpisanie 1 powoduje wyrównanie do lewej (tryb 8-bitowy), natomiast 0 do prawej (tryb 10bit) (zobacz opis rejestrów ADCL i ADCH).

MUX4:0 - Wybór kanału (oraz wzmocnienia).  
Za pomocą kombinacji bitów MUX dokonuje się wyboru pinu, który będzie podłączony przez multiplekser do przetwornika ADC. Wzmocnienie ustala się dla trybu różnicowego, który w tej pracy nie jest omawiany.

MUX4:0 (bin)	MUX4:0 (dec)	Wybrany kanał
00 000	0	ADC 0
00 001	1	ADC 1
00 010	2	ADC 2
00 011	3	ADC 3
00 100	4	ADC 4
00 101	5	ADC 5
00 110	6	ADC 6
00 111	7	ADC 7

## Rejestr ADCSRA

### Rejestr kontrolny i statusowy przetwornika ADC

Pisząc program, w kodzie programu należy używać nazwy ADCSR, gdyż nazwa ADCSRA nie figuruje w oryginalnym pliku nagłówkowym od ATMega16.

#### Rejestr ADCSRA

Bit	7	6	5	4	3	2	1	0
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

ADEN - Odblokowanie przetwornika, wpisanie 1 włącza przetwornik, 0 wyłącza.

ADSC - Bit rozpoczęcia konwersji.  
W trybie pojedynczego wyzwiania, wpisanie 1 powoduje rozpoczęcie każdej konwersji. W trybie automatycznego wyzwiania, wpisanie jedynki uruchamia ten proces. Tryb automatyczny może być rozpoczęty równocześnie

z włączeniem przetwornika, przez wpisanie odpowiedniej wartości do ADCSRA.

- ADATE** - Tryb automatycznego wyzwalania.  
Ustawienie bitu na 1 powoduje włączenie trybu automatycznego wyzwalania konwersji. ADC rozpoczyna konwersję od narastającego zbocza wybranego sygnału wyzwalającego. Źródło wybiera się ustawieniem bitów ADTS w rejestrze SFIOR.
- ADIF** - Flaga przerwania ADC.  
Ten bit jest ustawiony na 1 gdy ADC zakończy konwersję i zaktualizuje rejestr danych konwersji ostatnim wynikiem. Przerwanie związane z zakończeniem jest wykonywane, gdy ustawione są bity ADIE oraz bit I w rejestrze SREG. Flaga jest sprzętowo kasowana przez wykonywanie kodu obsługi przerwania, lub należy kasować programowo przez wpisanie 1.
- ADIE** - Włączenie przerwania ADC.  
Kiedy bit zostanie ustawiony na 1 i ustawiona jest też flaga I w SREG, to zakończenie konwersji ADC wywoła przerwanie.
- ADPS2:0** - Bity wyboru preskalera.  
Przez odpowiednie ustawienie bitów aktywuje się dzielnik częstotliwości kwarcu aby dobrać odpowiedni zegar dla ADC. Dopuszczalny zakres z jakim może pracować to 50-200kHz (więc dla kwarcu 8Mhz będzie to 64 lub 128).

ADPS2	ADPS2	ADPS2	Dzielnik częstotliwości
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

## Rejestry ADCL i ADCH

### Rejestry danych przetwornika ADC

$ADLAR = 0$

Bit	15	14	13	12	11	10	9	8
<b>ADCH</b>	-	-	-	-	-	-	ADC9	ADC8
<b>ADCL</b>	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0

*ADLAR = 0*

Bit	7	6	5	4	3	2	1	0
Dostęp	R	R	R	R	R	R	R	R
	R	R	R	R	R	R	R	R
Wartość początkowa	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

*ADLAR = 1*

Bit	15	14	13	12	11	10	9	8
<b>ADCH</b>	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2
<b>ADCL</b>	ADC1	ADC0	-	-	-	-	-	-
Bit	7	6	5	4	3	2	1	0
Dostęp	R	R	R	R	R	R	R	R
	R	R	R	R	R	R	R	R
Wartość początkowa	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

Kiedy konwersja przez przetwornik ADC dobiegnie końca, te dwa rejestry zostają uaktualnione o wynik konwersji. Jeśli nastąpiło odczytanie ADCL rejestry nie są uaktualniane aż do odczytania ADCH. Jeśli wystarczy wynik 8 bitowy (ADLAR=1), to należy odczytać tylko ADCH. Bit ADLAR ustawia się w rejestrze ADMUX.

## Rejestr SFIOR

### Rejestr funkcji specjalnych

#### Rejestr SFIOR

Bit	7	6	5	4	3	2	1	0
	ADTS2	ADTS1	ADTS0	-	ACME	PUD	PSR2	PSR10
Dostęp	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

ADTS2:0 - Źródło automatycznego wyzwalania ADC.

Jeśli bit ADATE w rejestrze ADCSRA jest ustawiony, to wartość bitów ADT wskazuje na źródło automatycznego wyzwalania startu przetwarzania ADC. Wyzwalanie następuje na zboczu narastającym źródła.

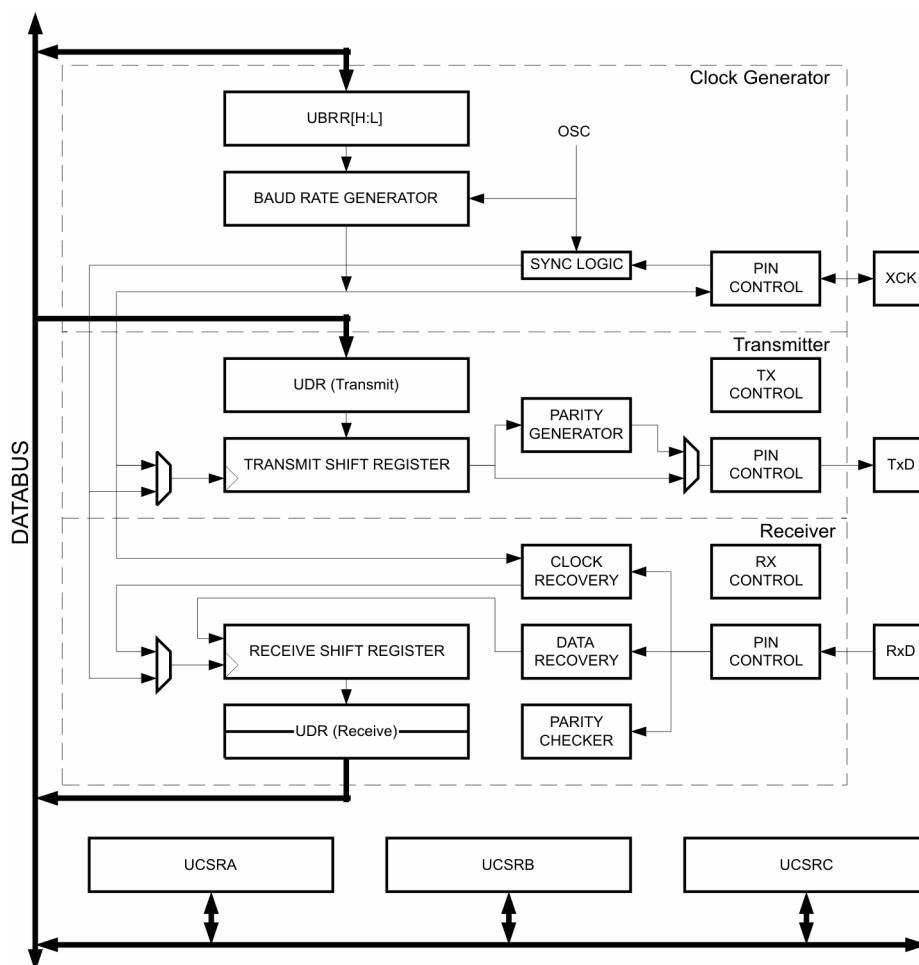
ADTS2	ADTS1	ADTS0	Źródło wyzwalania
0	0	0	Tryb Free Running (automatyczne wyzwalanie po zakończonej konwersji)
0	0	1	Komparator analogowy
0	1	0	Zewnętrzne żądanie przerwania
0	1	1	Licznik 0, przy zrównaniu
1	0	0	Licznik 0, przepełnienie
1	0	1	Licznik 1, przy zrównaniu B
1	1	0	Licznik 1, przepełnienie
1	1	1	Licznik 1, zdarzenie przechwycenia

## 9. Interfejs USART

Obecny w mikrokontrolerach szeregowy dwukierunkowy interfejs USART (lub UART) najczęściej służy do komunikacji z komputerem lub innym mikrokontrolerem. Moduły do tego typu transmisji integrowane w mikrokontrolerach, pozwalają często na zaawansowaną konfigurację i wygodną obsługę, a przy tym odciążają CPU.

Wybrane istotne cechy modułu USART:

- Pełny duplex (dzięki niezależnym rejestrom nadawania i odbioru).
- Praca synchroniczna lub asynchroniczna.
- Precyzyjny generator szybkości transmisji.
- Obsługa różnych konfiguracji ramek i kontroli parzystości.
- Wykrywanie błędów transmisji.
- Filtrowanie przeciwwzakłóceńowe.
- Trzy niezależne źródła przerwań.
- Tryb podwójnej szybkości w trybie asynchronicznym.
- Tryb komunikacji wieloprocessorowej.



Rys. 9. Schemat blokowy modułu USART

## USART kontra UART

Od układu UART, w którego często wyposażone są inne starsze mikrokontrolery Atmela, np. AT90S9515 różni się między innymi tym, że może pracować w trybie synchronicznym. W większości przypadków zachowana jest zgodność ze starszym UART, jednak dwa ulepszenia mogą zaburzyć kompatybilność. Dodany został drugi bufor odbiorczy, a rejestr przesuwany odbiornika może pełnić rolę trzeciego bufora. Przez to że dwa bufony odbiorcze pracują jako obrotowy FIFO, odczyt UDR wykonuje się tylko raz na każdy nowy odebrany pakiet danych. Bity statusu (FE, DOR) oraz 9 bit danych, muszą być odczytywane przed odczytem UDR, gdyż potem są tracone.

Pin XCK używany jest tylko w trybie pracy synchronicznej. W zestawie laboratoryjnym, który jest tematem niniejszej pracy, przewidziana jest tylko praca w popularnym trybie asynchronicznym.

## Generator szybkości transmisji

Rejestr szybkości transmisji UBRR wraz z licznikiem liczącym w dół stanowi programowalny preskaler lub generator szybkości transmisji. Licznik pracuje z częstotliwością zegara systemowego. Do określenia szybkości transmisji konieczne jest odpowiednie dobranie wartości rejestru UBRR. Do wyliczenia tej wartości należy skorzystać z następującej zależności:

$$UBRR = \frac{f_{osc}}{16 BAUD} - 1$$

gdzie:  $f_{osc}$  – częstotliwość oscylatora [Hz] (pracy mikrokontrolera),  $BAUD$  – pożądana wartość szybkości transmisji [bps]

Poniższa tabela zawiera zestawienie wartości UBRR dla kilku wybranych szybkości transmisji przy częstotliwości oscylatora z możliwie małym błędem (0,2 %).

Tabela 8. Zestawienie przykładowych wartości UBRR

<b>BAUD</b>	<b>UBRR</b> ( $f_{osc} = 8\text{MHz}$ )	<b>UBRR</b> ( $f_{osc} = 12\text{MHz}$ )
2400	207	312
4800	103	155
9600	51	77
19200	25	38

## Inicjalizacja transmisji

Przed jakąkolwiek transmisją, moduł USART musi zostać zainicjalizowany. Inicjalizacja obejmuje ustawienie szybkości transmisji, wybór formatu ramki oraz odblokowanie układu odbiorczego i/lub nadawczego. Jeśli operacje USART mają być źródłem przerw, to inicjalizacja powinna być wykonana z wykasowaną globalną flagą przerw.

## Rejestr UDR

### Rejestr danych IO USART

#### Rejestr UDR

Bit	7	6	5	4	3	2	1	0
	RXB[7:0] (odczyt)							
	TXB[7:0] (zapis)							
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

Rejestr nadawczy i odbiorczy USART współdzielą ten sam adres przestrzeni adresowej IO. Zapis do rejestru spowoduje umieszczenie danych w buforze nadawczym (TXB), natomiast poprzez odczyt spod adresu tego rejestru spowoduje odczytanie danych rejestru odbiorczego (RXB).

Bufor nadawczy może być nadpisany tylko wówczas, gdy jest ustawiona flaga UDRE w rejestrze UCSRA, w innym przypadku próba wpisania jest ignorowana. Wpisanie danych do bufora nadawczego, podczas gdy moduł nadawczy (transmitter) jest odblokowany, spowoduje przeniesienie danych do rejestru przesuwne (jeśli rejestr przesuwany jest pusty), a to jest równoznaczne z rozpoczęciem transmisji na pinie TXD.

Bufor odbiorczy składa się z dwóch buforów FIFO. Ze względu na budowę, nie należy korzystać z instrukcji bitowych pod adresem rejestru, gdyż zmieniają one stan buforu.

## Rejestr UCSRA

### Rejestr A kontroli i statusu USART

#### Rejestr UCSRA

Bit	7	6	5	4	3	2	1	0
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM
Dostęp	R	R/W	R	R	R	R	R/W	R/W
Wartość początkowa	0	0	1	0	0	0	0	0

RXC - Odbiór zakończony.

Flaga jest ustawiona, jeśli w buforze odbiorczym znajdują się nie odczytane dane. Flaga jest wykasowana, jeśli bufor jest pusty. Wyłączenie układu odbiornika kasuje zawartość buforu, więc i bit RXC jest także kasowany. Flaga RXC może generować przerwanie zakończenia odbioru (zob. opis RXCIE).

TXC - Transmisja zakończona.

Flaga jest ustawiana, gdy cała ramka zostanie wysłana z nadawczego rejestru przesuwne, oraz bufor nadawczy jest pusty. Flaga jest automatycznie kasowana przy wykonywaniu instrukcji spod wektora przerwań związanego z zakończeniem transmisji, lub może być skasowana przez wpisanie jedynek. Flaga TXC może generować przerwanie zakończenia transmisji (zob. opis bitu TXCIE).

UDRE - Pusty rejestr danych.

Flaga UDRE pojawia się jeśli bufor UDR jest gotowy na przyjęcie nowych danych. Flaga UDRE może generować przerwanie, gdy rejestr transmisji danych będzie pusty (zob. opis bitu UDRIE). Po resecie mikrokontrolera flaga jest ustawiana na 1.

- FE - Błędna ramka.  
 Bit ten jest ustawiony jeśli nadchodząca ramka w buforze odbiorczym jest błędna np. bit stopu w nadchodzącej ramce ma wartość 0. Bit jest ważny do odczytania buforu odbiorczego UDR. Bit FE ma wartość 0, gdy bit stopu odebranych danych ma wartość 1. Zapisując do UCSRA należy zawsze przypisać temu bitowi wartość 0.
- DOR - Przepelnienie danych.  
 Ten bit ma wartość 1, gdy zostało wykryte zdarzenie przepelnienia danych. Zdarzenie to następuje, gdy bufor odbiorczy jest pełny, a został wykryty kolejny bit startu. Bit jest ważny aż do odczytania buforu odbiorczego (UDR). Zapisując do UCSRA należy zawsze wpisywać temu bitowi wartość 0.
- PE - Błąd parzystości.  
 Bit ten ma wartość 0, gdy zawartość rejestru odbiorczego ma błąd parzystości oraz jest włączona kontrola parzystości (UPM1=1). Bit traci ważność z chwilą odczytania rejestru odbiorczego. Zapisując do UCSRA należy wpisywać 0.
- U2X - Podwójna szybkość transmisji.  
 Bit ma znaczenie tylko w transmisji asynchronicznej. Przy korzystaniu z trybu synchronicznego należy go zawsze zerować. Ustawienie bitu na 1 zmniejsza dzielnik szybkości transmisji z 16 na 8, co włącza tryb podwójnej szybkości dla transmisji asynchronicznej.
- MPCM - Tryb komunikacji między procesorami.  
 Jeśli MPCM zostanie ustawiony na 1, wszystkie nadchodzące ramki odbierane przez odbiornik USART, które nie zawierają informacji o adresie, zostaną zignorowane. MPCM nie ma wpływu na pracę nadajnika.

## Rejestr UCSRB

### Rejestr B kontroli i statusu USART

#### Rejestr UCSRB

Bit	7	6	5	4	3	2	1	0
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

- RXCIE - Odblokowanie przerwania zakończenia odbioru.  
 Ustawienie tego bitu zezwala na generowanie przerwania na pojawienie się flagi RXC. Aby przerwanie zostało wygenerowane, muszą być ustawione flagi RXCIE oraz globalna flaga przerwania I.
- TXCIE - Odblokowanie przerwania zakończenia nadawania.



- Ustawienie tego bitu zezwala na generowanie przerwania na pojawienie się flagi TXC. Aby przerwanie zostało wygenerowane, muszą być ustawione flagi TXCIE oraz globalna flaga przerwania I.
- UDRIE - Odblokowanie przerwania na pusty rejestr danych.  
Ustawienie tego bitu zezwala na generowanie przerwania na pojawienie się flagi UDRE. Aby przerwanie zostało wygenerowane, muszą być ustawione flagi UDRIE oraz globalna flaga przerwania I.
- RXEN - Odblokowanie odbiornika.  
Ustawienie tego włącza odbiornik USART. Włączenie odbiornika powoduje, że funkcja RxD staje się nadrzędną nad normalnymi funkcjami IO tego pinu. Wyłączenie odbiornika czyści zawartość rejestru odbiorczego oraz unieważnia flagi FE, DOR i PE.
- TXEN - Odblokowanie nadajnika.  
Ustawienie tego włącza nadajnik USART. Włączenie nadajnika powoduje, że funkcja TxD staje się nadrzędną nad normalnymi funkcjami IO tego pinu. Wyłączanie nadajnika (wpisywanie 0 do TXEN) nie będzie dawało rezultatu, jeśli nie nastąpi zakończenie trwającego nadawania.
- UCSZ2 - Rozmiar znaku.  
Bit UCSZ2 wraz z bitami UCSZ1:0 stanowi kombinację, która konfiguruje liczbę bitów danych w ramce.
- RXB8 - 8 bit danych odbieranych.  
RXB8 stanowi 9 bit danych odbieranego znaku, jeśli operacje transmisji odbywają się w ramach o takim formacie. Odczyt tego bitu musi nastąpić przed odczytaniem pozostałej części z rejestru UDR.
- TXB8 - 8 bit danych nadawanych.  
TXB8 stanowi 9 bit danych znaku do nadania, jeśli operacje transmisji odbywają się w ramach o takim formacie. Ustawienie tego bitu musi nastąpić przed zapisem pozostałej części do rejestru UDR.

## Rejestr UCSRC

### Rejestr C kontroli i statusu USART

Rejestr UCSRC współdzieli ten sam adres przestrzeni adresowej IO wraz z rejestrem UBRRH. Zobacz opis bitu URSEL.

#### Rejestr UCSRC

Bit	7	6	5	4	3	2	1	0
	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

- URSEL - Wybór rejestru.  
Bit ten służy do przełączania dostępu między rejestrami UBRRH i UCSRC. Ma wartość 1 jeśli odczytany został rejestr UCSRC. Chcąc zapisać do rejestru

UCSRC należy do tego bitu wpisywać wartość 1. Porównaj z opisem dla rejestru UBRRH.

UMSEL - Wybór trybu USART.

Bit ten służy do wyboru między pracą synchroniczną a asynchroniczną. Do pracy asynchronicznej bit musi mieć wartość 0, natomiast do synchronicznej należy wpisać ustawić na 1.

UPM1:0 - Tryb parzystości.

Za pomocą tych bitów ustawia się rodzaj parzystości, która jest generowana przy nadawaniu oraz sprawdzana odbieraniu każdej ramki:

UPM1	UPM0	Tryb parzystości
0	0	Parzystość wyłączona
0	1	zastrzeżone
1	0	Włączony, tryb nieparzystości
1	1	Włączony, tryb parzystości

USBS - Wybór bitu stopu.

Ustawienie tych bitów decyduje o liczbie bitów stopu ustawianych przy nadawaniu ramki. Ustawienie to nie ma wpływu na pracę odbiornika.

USBS	Liczba bitów stopu
0	1 bit
1	2 bity

UCSZ1:0 - Liczba bitów danych w ramce.

Bit UCSZ2 (rejestr UCSZB) wraz z bitami UCSZ1:0 stanowi kombinację, która konfiguruje liczbę bitów danych w ramce i jest używana przez nadajnik i odbiornik.

UCSZ2	UCSZ1	UCSZ0	Rozmiar znaku
0	0	0	5 bitów
0	0	1	6 bitów
0	1	0	7 bitów
0	1	1	8 bitów
1	0	0	zastrzeżone
1	0	1	zastrzeżone
1	1	0	zastrzeżone
1	1	1	9 bitów

UCPOL - Polaryzacja zegara.

Bit ten jest używany tylko w trybie pracy synchronicznej. Przy pracy asynchronicznej należy go ustawiać na wartość 0.

## Rejestry UBRR1 i UBRRH

### Rejestr szybkości transmisji

Rejestr UBRRH współdzieli ten sam adres przestrzeni adresowej IO wraz z rejestrem UCSRC. Zobacz opis bitu URSEL.

## Rejestr UBRR

Bit	7	6	5	4	3	2	1	0
UBRRH	URSEL	-	-	-	UBRR[11:8]			
UBRRL	UBRR[7:0]							
Dostęp	R/W	R	R	R	R/W	R/W	R/W	R/W
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

URSEL - Wybór rejestru.

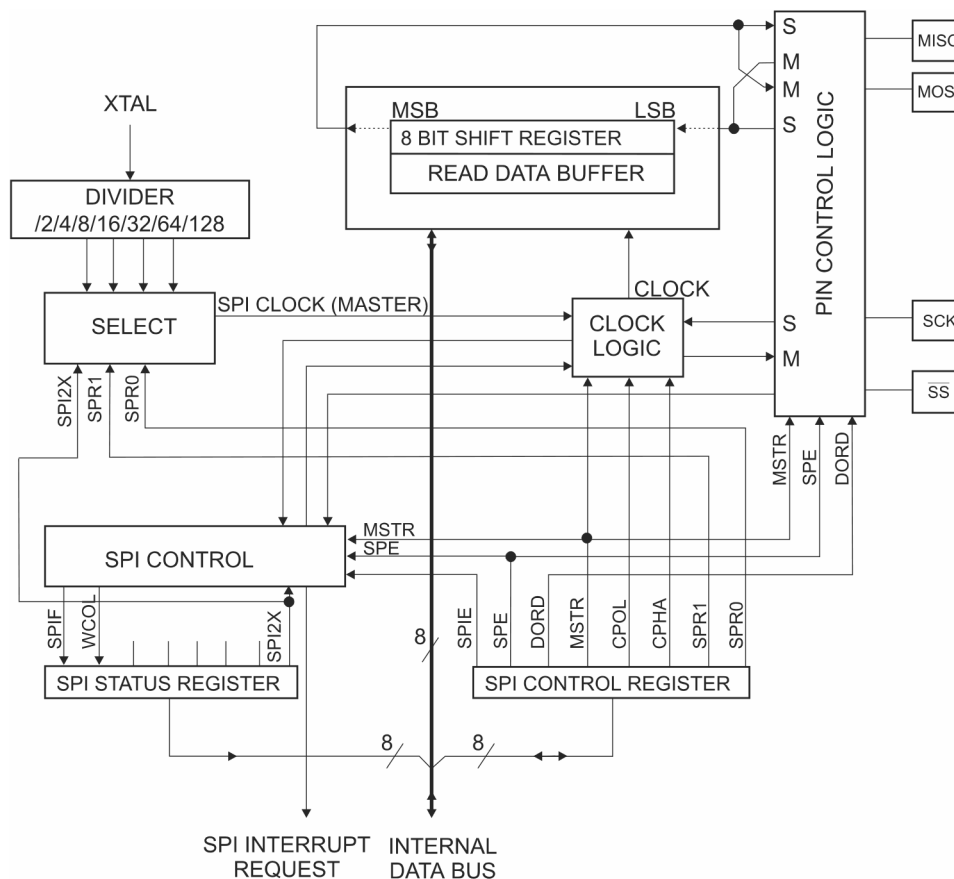
Bit ten służy do przełączania dostępu między rejestrami UBRRH i UCSRC. Ma wartość 0 jeśli odczytany został rejestr UBRRH. Zapisując do rejestru UBRRH należy do tego bitu wpisywać wartość 0. Porównaj z opisem dla rejestru UCSRC.

Bity 6:4 - Zarezerwowane do przyszłych zastosowań, zapisując ustawiać na wartość 0.

UBRR11:0 - Rejestr szybkości transmisji.

Te bity stanowią 12 bitowy rejestr wyboru szybkości transmisji modułu USART. UBRRH zawiera 4 najbardziej znaczące bity, natomiast UBRRL 8 mniej znaczących. Nadpisanie bitów UBRR wprowadza natychmiastową zmianę preskalera w generatorze szybkości transmisji. Jak obliczać wartość UBRR napisano w punkcie „Generator szybkości transmisji”.

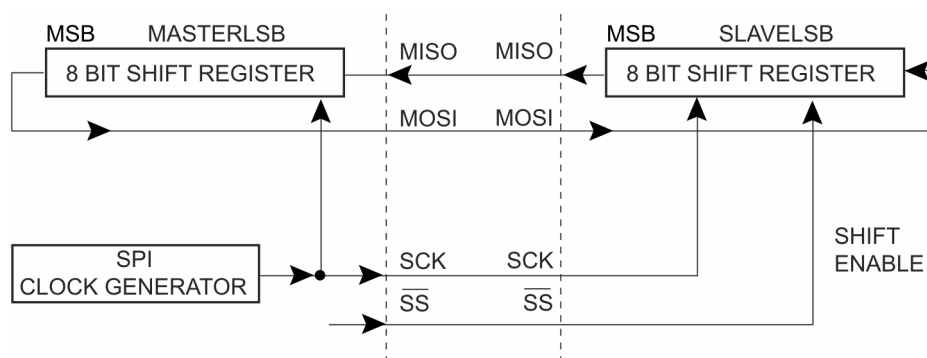
## 10. Interfejs SPI



Rys. 10. Schemat blokowy modułu SPI

Interfejs SPI (Serial Peripheral Interface) umożliwia synchroniczny transfer (full-duplex) danych między ATmega16 i układami peryferyjnymi lub między kilkoma układami AVR. Układami peryferyjnymi są układy o specjalizowanych funkcjach, jak np. pamięci szeregowo EEPROM, potencjometry cyfrowe, a także bardziej złożone układy sterowane cyfrowo jak np. zintegrowany w jeden układ system audio wraz ze filtrami, wzmacniaczami, mieszaczami.

Jest to interfejs 3-przewodowy (lub 4), składający się z dwóch linii danych w obu kierunkach (MISO, MOSI) oraz linii synchronizującej (SCK). W przypadku połączenia dwóch układów przez SPI, jeden z nich pełni rolę nadrzędną (master) a drugi podrzędną (slave). W przypadku komunikacji między dwoma urządzeniami, linię  $\overline{SS}$  (Slave Select) można pominąć podłączając ją w urządzeniu podrzędnym do masy.



Rys.11. Schemat połączeń między układami master (z lewej) i slave (z prawej)

## Rejestr SPCR

### Rejestr kontroli SPI

#### Rejestr SPCR

Bit	7	6	5	4	3	2	1	0
	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

SPIE - Odblokowanie przerwania SPI.

SPE - Bit włączenia interfejsu SPI.

Ustawienie tego bitu na 1 powoduje odblokowanie / włączenie SPI.

DORD - Kolejność danych.

Kiedy bit DORD jest ustawiony na 1, dane przesyłane są od najmniej znaczącego bitu (LSB). DORD = 0 oznacza wysyłanie od bitu najbardziej znaczącego (MSB).

MSTR - Wybór master/slave.

Przez wpisanie 1, tym bitem włącza się tryb master (rolę układu jako nadrzędnego). SPI będzie pracować jako slave jeśli ten bit będzie miał wartość logicznego zera. Do pracy układu jako master, pin  $\overline{SS}$  powinien być wcześniej ustawiony jako wyjściowy.

CPOL - Polaryzacja zegara.

Jeśli do CPOL jest wpisane 1, to SCK w spoczynku jest w stanie wysokim, jeśli 0, to w stanie niskim.

CPHA - Faza zegara.

CPHA=1 – pierwsze zbocze sygnału zegarowego próbkuje dane wejściowe a drugie przesuwa dane w rejestrze, CPHA=0 – pierwsze zbocze przesuwa dane w rejestrze a drugie próbkuje dane wejściowe.

SPR1:0 - Szybkość zegara SPI.

Bity kontrolujące szybkość taktowania zegara SCK dla urządzenia skonfigurowanego jako Master (nadrzędny), w stosunku do częstotliwości oscylatora mikrokontrolera. Dla pracy jako slave, ustawienie nie ma znaczenia.

SPI2X	SPR1	SPR0	Częstotliwość SCK
0	0	0	$f_{osc} / 4$
0	0	1	$f_{osc} / 16$
0	1	0	$f_{osc} / 64$
0	1	1	$f_{osc} / 128$
1	0	0	$f_{osc} / 2$
1	0	1	$f_{osc} / 8$
1	1	0	$f_{osc} / 32$
1	1	1	$f_{osc} / 64$

## Rejestr SPSR

### Rejestr statusowy SPI

#### Rejestr SPSR

Bit	7	6	5	4	3	2	1	0
	SPIF	WCOL	-	-	-	-	-	SPI2X
Dostęp	R	R	R	R	R	R	R	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

SPIF - Flaga przerwania SPI.

Flaga jest ustawiana po zakończeniu transmisji. Przerwanie jest generowane jeśli ustawiony jest bit SPIE w rejestrze SPCR oraz ustawiona globalna flaga przerwania I. Flaga jest ustawiana także, gdy pin  $\overline{SS}$  jest ustawiony jako wejściowy i w pojawi się na nim stan niski, a SPI ustawiony jest jako master. Flaga jest sprzętowo kasowana przy wykonywaniu programu spod wektora przerwania SPI. Flaga kasowana sprzętowo jest także, gdy po odczytaniu rejestru SPSI, nastąpi dostęp (odczyt lub zapis) do rejestru danych SPDR.

WCOL - Flaga kolizji zapisu.

Bit jest ustawiony, gdy nastąpi zapis do SPDR podczas trwającej transmisji. Flaga jest kasowana gdy po odczytaniu rejestru SPSI nastąpi dostęp do rejestru danych SPDR.

SPI2X - Podwójna szybkość SPI.

Ustawienie tego bitu na 1 podwaja szybkość pracy SPI (częstotliwość SCK).  
Zobacz też opis bitów SPR1:0 w rejestrze SPCR.

## Rejestr SPDR

### Rejestr danych SPI

Rejestr danych SPI służy do odczytu i zapisu transmitowanych danych. Zapis do rejestru inicjuje transmisję. Odczytywanie rejestru powoduje odczyt buforu odbiorczego rejestru przesuwanego.

### Rejestr SPSR

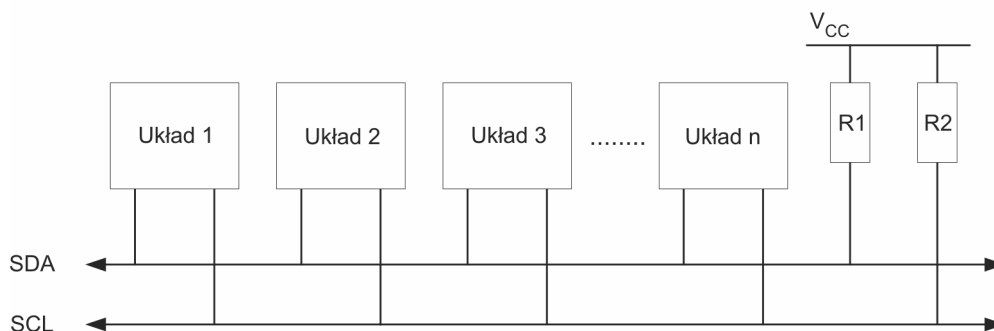
Bit	7	6	5	4	3	2	1	0
	MSB							LSB
Dostęp	R	R	R	R	R	R	R	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

## 11. Interfejs TWI

Interfejs TWI (two wire interface) jest w pełni zgodny z popularnym I2C firmy Philips, a Atmel używa innej nazwy aby nie łamać praw patentowych.

Interfejs I2C podobnie jak SPI, został stworzony do komunikacji między układami scalonymi w obrębie jednego urządzenia. Komunikacja odbywa się dwuprzewodowo, linią danych SDA oraz linią zegara SCL, które są mogą być wspólne dla wszystkich układów w systemie. Pomimo jednej linii danych, transmisja może odbywać się dwukierunkowo, jednak ze stosunkowo niewielką transmisją. W większości przypadków spotyka się możliwość pracy przy częstotliwości linii SCL do 100kHz, a rzadziej do 400kHz. Moduł TWI wbudowany w ATmega16 umożliwia pracę w obu tych trybach.

Współpraca wielu układów po tych samych przewodach jest możliwa dzięki temu, iż każdemu z urządzeń przypisuje się indywidualny adres. Liczba urządzeń podłączonych jednocześnie jest ograniczona jedynie maksymalną pojemnością linii 400pF oraz 7-bitową przestrzenią adresową dla urządzeń podrzędnych. Projektując magistralę TWI (I2C) nie należy zapomnieć o rezystorach podciągających.



Rys. 12. Schemat połączeń w magistrali TWI (I2C)

## 12. Interfejs JTAG

Właściwości interfejsu JTAG w mikrokontrolerze ATMega16:

- Interfejs JTAG zgodny ze standardem IEEE 1149.1.
- Możliwości skanowania wyprowadzeń zgodnie ze standardem JTAG.
- Dodatkowe instrukcje diagnostyczne IDCODE, AVR\_RESET.
- Instrukcje zapewniające możliwość programowania pamięci flash.
- Funkcje wspomagające debugowanie i śledzenie działania programu w układzie.

Idea ścieżki obserwacyjno-testującej pozwala rozwiązać problem testowania coraz bardziej skomplikowanych systemów w możliwie prosty i tani sposób, co ma szczególne znaczenie przy produkcji wielkoseryjnej.

Ścieżka diagnostyczna w systemie w oparciu o magistralę IEEE 1149.1 pozwala na wyizolowanie fragmentu obwodu i przetestowaniu go poprzez doprowadzenie wymuszenia i obserwacji reakcji. Daje na przykład możliwość wysterowania i obserwacji poziomów logicznych na cyfrowych pinach wejścia/wyjścia.

Z poziomu systemu, wszystkie układy wyposażone w funkcje JTAG są połączone szeregowo przez linie danych TDI/TDO tworząc długi rejestr przesuwany. Zewnętrzny kontroler poprzez niego konfiguruje piny wyjściowe układów i obserwuje wartości wejściowe otrzymane od innych układów. Kontroler porównuje otrzymane dane ze spodziewanymi wynikami i w ten sposób skanowanie krawędziowe daje możliwość testowania połączeń i poprawność działania.

Fizyczny dostęp do interfejsu JTAG umożliwiają cztery wyprowadzenia mikrokontrolera. Piny te stanowią dostępowy port testowania TAP (Test Acces Port). Tymi pinami są: TMS (test mode select), TCK (test clock), TDI (test data in), TDO (test data out). Zależnie od bitu bezpiecznika JTAGEN, piny te mogą pełnić rolę normalnych pinów portów I/O mikrokontrolera lub działać jako piny portu TAP. Jeśli JTAGEN jest zaprogramowany, normalną pracę IO można wymusić bitem JTD w rejestrze MCUCSR.

Oprócz obowiązkowych instrukcji definiowanych standardem JTAG, w mikrokontrolerze ATMega16 obecne są dodatkowe instrukcje wspomagające diagnostykę, są to IDCODE oraz AVR\_RESET. Za pomocą IDCODE możemy zidentyfikować układ w systemie. Instrukcja AVR\_RESET może się przydać np. do przetestowania systemu, gdy znajdzie potrzeba wprowadzenia mikrokontrolera w stan resetu.

Poza możliwościami diagnostycznymi, za pomocą interfejsu JTAG możliwe jest programowanie pamięci mikrokontrolera w systemie (poza programowaniem ISP). Dzięki zestawowi instrukcji wspierających programowanie otrzymujemy możliwość programowania i odczyt zawartości pamięci flash, EEPROM, oraz bitów zabezpieczających i konfiguracyjnych.

Magistrala JTAG zaimplementowana w ATMega16 umożliwia debugowanie programu w zaprogramowanym mikrokontrolerze będącym fizycznie częścią systemu. Zastosowane rozszerzenia pozwalają na śledzenie programu krok po kroku, ustawianie pułapek czy obserwowanie lub modyfikowanie zawartości rejestrów. Aby było możliwe korzystanie z tych

udogodnień, konieczne jest skorzystanie ze specjalizowanego interfejsu „JTAG ICE” (lub zgodnego) do połączenia komputera z mikrokontrolerem. Programowym środowiskiem na komputerze jest darmowe środowisko programistyczne AVRStudio dostarczane przez firmę Atmel i zawarty w nim symulator.

## Rejestry związane z JTAG

### Rejestr MCUCSR

#### Rejestr MCUCSR

Bit	7	6	5	4	3	2	1	0
	JTD	ISC2	-	JTRF	WDRF	BORF	EXTRF	PORF
Dostęp	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0		<i>zobacz w opisie bitów</i>			

JTD - Odłączenie interfejsu JTAG .

Jeśli bezpiecznik JTAGEN jest zaprogramowany, wówczas ustawienie bitu JTD na wartość 1 umożliwia wyłączenie interfejsu JTAG i możliwość pracy pinom tego interfejsu jako normalne piny IO mikrokontrolera. W celu nieprzypadkowego wyłączenia JTAG’a, bit JTD musi być zapisany dwukrotnie w ciągu 4 cykli zegarowych.

Bezpiecznik JTAGEN w fabrycznie nowym układzie ATMega16 jest zaprogramowany. Udostępniony w zestawie laboratoryjnym egzemplarz ma JTAGEN nie zaprogramowany, więc nie jest konieczne wyłączenie bitem JTD.

JTRF - Flaga resetu JTAG.

Bit ten jest ma wartość 1, gdy reset mikrokontrolera został spowodowany instrukcją AVR\_RESET przez JTAG.

### Tryby oszczędzania energii

Różnorodne tryby oszczędzania energii, pozwalają na znaczne wydłużenie czasu działania urządzeń zasilanych bateryjnie, a także wpływają na redukcję emitowanych zakłóceń w środowiskach, gdzie ma to szczególne znaczenie.

#### Idle

Tryb Idle (bezczynności) zatrzymuje jednostkę CPU pozwalając na pracę układów interfejsów USART, TWI, SPI, przetwornika ADC, systemu przerwań, watchdog’a, komparatora analogowego. W tym trybie zegary  $clk_{CPU}$  i  $clk_{FLASH}$  są zatrzymane podczas gdy inne mogą pracować. Wybudzenie możliwe jest przez wewnętrzne lub zewnętrzne przerwanie. Jeśli układ nie musi być wzbudzany przez komparator, to także komparator można wyłączyć dla zwiększenia oszczędności energii. Jeśli ADC jest włączone, to po wejściu w tryb Idle nastąpi automatyczne wystartowanie konwersji.



## **Power-down**

Tryb Power-down (wyłączonego zasilania) wyłącza mikrokontroler. W tym trybie zewnętrzny oscylator jest zatrzymany dotąd, aż pojawi się przerwanie zewnętrzne, dopasowania adresu TWI, watchdog'a, reset zewnętrzny lub Brown-out. Wyłączone zostają wszystkie zegary pozostawiając działające moduły asynchroniczne. Ponowne uruchomienie następuje z pewnym opóźnieniem, aby rozpocząć ponowne generowanie sygnału zegarowego i aby osiągnął on stabilność. Czas wybudzania jest określany przez te same bity zabezpieczające CKSEL, które definiują opóźnienie resetu.

## **Power-save**

Tryb Power-save (oszczędności zasilania) jest podobny do Power-down. Jedyną różnicą jest możliwość pracy Timera/Licznika2 w sposób asynchroniczny (np. ustawiony bit AS2 w rejestrze ASSR). Może nastąpić przerwanie wybudzające spowodowane przez timer/licznik 2, jeśli odpowiedni bit włączający to przerwanie jest ustawiony w TIMSK oraz ogólny bit przerwania i w SREG jest ustawiony. Jeśli timer asynchroniczny nie będzie pracował z zegara asynchronicznego, wówczas zalecane jest zamiast Power-save użycie trybu Power-down.

## **ADC Noise Reduction**

Tryb ADC Noise Reduction (redukcji szumów przetwornika ADC) zatrzymuje CPU pozwalając na pracę i wybudzenie przez ADC, zewnętrzne przerwania, monitorowanie adresu TWI, timera/licznika2 i watchdog'a. Tryb ten pozwala zminimalizować szum przełączania podczas trwania konwersji przetwornika ADC dla zapewnienia większej dokładności. Jeśli przetwornik jest włączony, to wejście w stan uśpienia spowoduje automatyczne rozpoczęcie konwersji. Tryb ten powoduje zatrzymanie zegarów  $clk_{CPU}$ ,  $clk_{I/O}$  i  $clk_{FLASH}$ , podczas gdy inne mogą pracować.

## **Standby**

W trybie Standby (czuwania) rezonator kwarcowy pracuje, podczas gdy reszta układu jest w uśpieniu. Tryb ten może być użyty tylko jeśli układ skonfigurowany jest do pracy z zewnętrznym źródłem zegarowym. Podobny jest do trybu Power-down, z tą różnicą że w połączeniu z bardzo niskim poborem mocy pozwala na bardzo szybkie wybudzenie układu.

## **Extended Standby**

Tryb Extended Standby (rozszerzonego czuwania) od trybu Power-save różni się tym że oscylator pozostaje włączony. Wybudzenie następuje w ciągu sześciu cykli zegarowych.

## Rejestr MCUCR

Aby wprowadzić układ w któryś ze stanów oszczędzania energii, należy ustawić bit SE w rejestrze MCUCR, oraz wykonać instrukcję SLEEP. Wykonywanie rozkazów zostaje wstrzymane w miejscu wywołania SLEEP i jest kontynuowane z tego miejsca po wybudzeniu. Jeśli wybudzenie następuje na skutek przerwania, to zachowanie się układu jest identyczne, jak by w tym momencie nastąpiło to przerwanie podczas normalnej pracy. Pożądaný tryb wybiera się konfigurując bity SMn w MCUCR.

### Rejestr MCUCR

Bit	7	6	5	4	3	2	1	0
	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00
Dostęp	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	0	0	0	0	0	0	0	0

SE - Odblokowanie oszczędzania energii (sleep enable).

Ustawienie tego bitu w stan 1 umożliwia wprowadzenie mikrokontrolera w jeden z 6 trybów oszczędzania energii. W celu zapobiegnięcia przypadkowego wprowadzenia w stan uśpienia, zalecane jest ustawienie tego bitu na 1 bezpośrednio przed instrukcją SLEEP oraz wyczyszczenie SE na 0 tuż po wybudzeniu.

SM2:0 - Bity wyboru trybu uśpienia:

SM2	SM1	SM0	Tryb
0	0	0	Idle
0	0	1	ADC Noise Reduction
0	1	0	Power-down
0	1	1	Power-save
1	0	0	Konfiguracja zastrzeżona
1	0	1	Konfiguracja zastrzeżona
1	1	0	Standby <sup>(1)</sup>
1	1	1	Extended Standby <sup>(1)</sup>

<sup>(1)</sup> tryby dostępne tylko z zewnętrznym źródłem zegara

Tabela 9. Zestawienie źródeł wybudzania zależnie od trybów oszczędzania energii:

Tryb uśpienia	Źródła wybudzania					
	INT0 INT1 INT2	Pasujący adres TWI	Timer 2	SPM/ EEPROM Ready	ADC	Pozostałe I/O
Idle	X	X	X	X	X	X
ADC Noise Reduction	X <sup>(3)</sup>	X	X	X	X	
Power Down	X <sup>(3)</sup>	X				
Power Save	X <sup>(3)</sup>	X	X <sup>(2)</sup>			
Standby	X <sup>(3)</sup>	X				
Extended Standby	X <sup>(3)</sup>	X	X <sup>(2)</sup>			

<sup>(2)</sup> jeśli jest ustawiony bit AS2 w rejestrze ASSR; <sup>(3)</sup> tylko INT2 lub poziom przerwania INT1 i INT0;

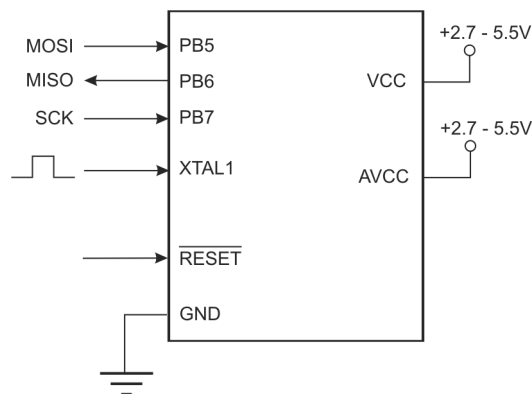
Poza wprowadzaniem mikrokontrolera w tryb uśpienia, dodatkowo można zmniejszać pobór mocy innymi metodami, jak np. wyłączanie niepotrzebnych podukładów.

### 13. Programowanie w systemie (ISP)

Zarówno pamięć flash jak i EEPROM, a także bity konfiguracyjne i zabezpieczające mogą być programowane za pomocą szeregowego interfejsu w systemie (In-System Programming - ISP). Programowanie w systemie oznacza, że zaprogramowanie i weryfikacja zawartości pamięci układu jest możliwa w mikrokontrolerze będącym fizycznie na docelowej płytce drukowanej wraz z innymi układami. Jest to charakterystyczna cecha wszystkich mikrokontrolerów z szerokiej rodziny AVR firmy Atmel. Programowanie szeregowe w systemie może odbywać się za pomocą szyny SPI jak i JTAG.

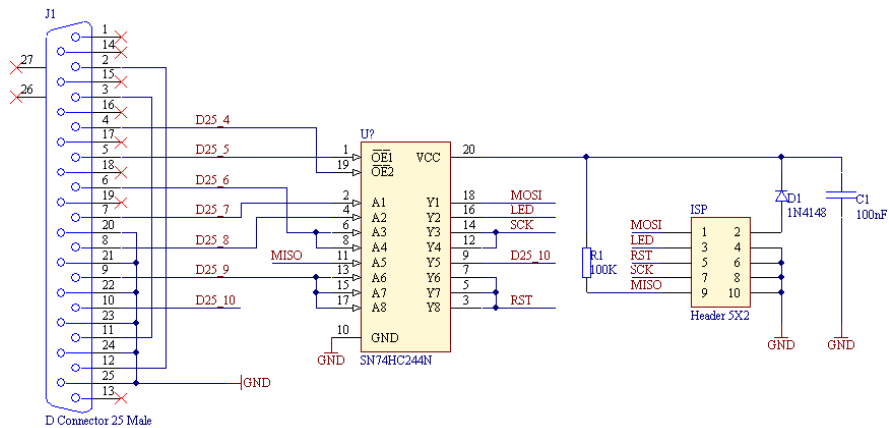
Jeśli w systemie magistrala JTAG nie jest wykorzystywana, to używanie tego sposobu wyłącznie do programowania jest mało praktyczne. Do tego celu konieczne by było zrezygnowanie z 4 pinów mikrokontrolera normalnie działających jako piny portów IO. W przypadku programowania przez SPI nie ma takiej konieczności. Jest to możliwe, gdyż odbywa się to, gdy pin  $\overline{\text{RESET}}$  jest wysterowany niskim poziomem (zwarły do masy), a wówczas (w stanie resetu) wszystkie piny IO mikrokontrolera są w stanie wysokiej impedancji.

W niniejszym układzie uruchomieniowym przewidziana została możliwość programowania ISP za pośrednictwem szyny SPI.



Rys. 13. Sposób podłączenia programatora szeregowego

Do połączenia z komputerem, wykorzystany będzie programator typu „STK200 PonyProg” (nazwany również AVR-ISP), podłączany do portu równoległego. W połączeniu z darmowym programem PonyProg 2000, z poziomu układu będzie zgodny z zalecanym przez firmę Atmel programatorem STK200. Konstrukcja samego programatora nie jest skomplikowana i przedstawia ją rysunek 14.



Rys.14. Schemat programatora „SKT200 PonyProg”

## 14. Opis instrukcji asemblera

W poniższym zestawieniu instrukcji, w argumentach używane są następujące oznaczenia:

- Rd R0-R31, rejestr docelowy operacji (także źródłowy), czasem np.  $16 \leq d \leq 31$ ,  
 Rr R0-R31, rejestr źródłowy, czasem np.  $16 \leq r \leq 31$ ,  
 b stała o wartości 0-7, do oznaczania numeru lub nazwy bitu, np. OCIE0,  
 s stała o wartości 0-7, bit rejestru statusowego SREG, np. V, Z,  
 P stała (0-31/63) określająca adres z przestrzeni adresowej I/O, np. PORTA,  
 K stała (0-31/63) określająca adres z przestrzeni adresowej I/O, np. R16, PORTA,  
 k stała z przedziału wartości zależnego od instrukcji,  
 q stała (0-63) określająca wartość przesunięcia przy adresowaniu pośrednim,  
 X rejestr adresowania pośredniego,  $X=R27:26$ ,  
 Y rejestr adresowania pośredniego,  $Y=R29:28$ ,  
 Z rejestr adresowania pośredniego,  $Z=R31:30$ .

Opis oznaczeń flag rejestru statusowego SREG (szczegóły w rozdziale „Zarys budowy rdzenia AVR”):

- C flaga przeniesienia (carry),  
 Z flaga zera,  
 N flaga wyniku ujemnego,  
 V wskaźnik przepełnienia uzupełnienia do dwóch,  
 S  $S = N \oplus V$ , do testów znaku,  
 H flaga przeniesienia połówkowego (half carry),  
 T bit transferowy, używany przez instrukcje BLD i BST,  
 I globalna flaga przerwań.

Opis kolumn:

- Mnemonic – składnia instrukcji asemblera,  
 Opis – skrócony opis operacji realizowanej przez instrukcję,  
 Operacja – logiczny zapis operacji wykonywanej przez instrukcję,

Clk – liczba taktów zegara systemowego potrzebnych na wykonanie operacji.

Pozostałe oznaczenia używane w zestawieniu instrukcji:

SP wskaźnik stosu (stack pointer),

STACK stos adresów powrotnych i przechowywanych rejestrów.

W przypadku używania złożonych instrukcji jak np. mnożenie (MULS, FMULS itd.), zalecane jest skorzystanie z opisu w pomocy programu AVRStudio lub dokumentacji instrukcji mikrokontrolerów AVR [4].

### Instrukcje arytmetyczne i logiczne

Mnemonic	Opis	Operacja	Flags	Clk
<b>ADD</b> Rd, Rr	Dodaj dwa rejestry	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
<b>ADC</b> Rd, Rr	Dodaj dwa rejestry z CARRY	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
<b>ADIW</b> Rdl,K	Dodaj stałą do słowa	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z,C,N,V,S	2
<b>SUB</b> Rd, Rr	Odejmij rejestr od rejestru	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
<b>SUBI</b> Rd, K	Odejmij stałą od rejestru ( $16 \leq d \leq 31$ )	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
<b>SBC</b> Rd, Rr	Odejmij rejestr od rejestru z CARRY	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
<b>SBCI</b> Rd, K	Odejmij stałą od rejestru z CARRY ( $16 \leq d \leq 31$ )	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
<b>SBIW</b> Rdl,K	Odejmij stałą od słowa	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V,S	2
<b>AND</b> Rd, Rr	Logiczny AND, dwa rejestry	$Rd \leftarrow Rd \bullet Rr$	Z,N,V	1
<b>ANDI</b> Rd, K	Logiczny AND, rejestr i stała	$Rd \leftarrow Rd \bullet K$	Z,N,V	1
<b>OR</b> Rd, Rr	Logiczny OR, dwa rejestry	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
<b>ORI</b> Rd, K	Logiczny OR, rejestr i stała ( $16 \leq d \leq 31$ )	$Rd \leftarrow Rd \vee K$	Z,N,V	1
<b>EOR</b> Rd, Rr	Exclusive OR, dwa rejestry	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
<b>COM</b> Rd	Dopełnienie do \$FF	$Rd \leftarrow \$FF - Rd$	Z,C,N,V	1
<b>NEG</b> Rd	Dopełnienie do \$00	$Rd \leftarrow \$00 - Rd$	Z,C,N,V,H	1
<b>SBR</b> Rd,K	Ustaw bit(y) w rejestrze ( $16 \leq d \leq 31$ )	$Rd \leftarrow Rd \vee K$	Z,N,V	1
<b>CBR</b> Rd,K	Wyczyść bit(y) w rejestrze ( $16 \leq d \leq 31$ )	$Rd \leftarrow Rd \bullet (\$FF - K)$	Z,N,V	1
<b>INC</b> Rd	Incrementuj	$Rd \leftarrow Rd + 1$	Z,N,V	1
<b>DEC</b> Rd	Dekrementuj	$Rd \leftarrow Rd - 1$	Z,N,V	1
<b>TST</b> Rd	Test czy zero lub minus	$Rd \leftarrow Rd \bullet Rd$	Z,N,V	1
<b>CLR</b> Rd	Wyczyść rejestr	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
<b>SER</b> Rd	Ustaw rejestr ( $16 \leq d \leq 31$ )	$Rd \leftarrow \$FF$	Żadne	1
<b>MUL</b> Rd, Rr	Mnożenie bez znaku	$R1:R0 \leftarrow Rd * Rr$	Z,C	2
<b>MULS</b> Rd, Rr	Mnożenie ze znakiem ( $16 \leq d,r \leq 31$ )	$R1:R0 \leftarrow Rd * Rr$	Z,C	2
<b>MULSU</b> Rd, Rr	Mnożenie liczby ze znakiem i bez ( $16 \leq d,r \leq 23$ )	$R1:R0 \leftarrow Rd * Rr$	Z,C	2
<b>FMUL</b> Rd, Rr	Ułamkowe mnożenie bez znaku ( $16 \leq d,r \leq 23$ )	$R1:R0 \leftarrow (Rd * Rr) \ll 1$	Z,C	2
<b>FMULS</b> Rd, Rr	Ułamkowe mnożenie ze znakiem ( $16 \leq d,r \leq 23$ )	$R1:R0 \leftarrow (Rd * Rr) \ll 1$	Z,C	2
<b>FMULSU</b> Rd, Rr	Ułamkowe ze znakiem lub bez ( $16 \leq d,r \leq 23$ )	$R1:R0 \leftarrow (Rd * Rr) \ll 1$	Z,C	2

## Instrukcje organizujące rozgałęzienia

Mnemonik	Opis	Operacja	Flagi	Clk
JMP k	Skok bezpośredni	$PC \leftarrow k$	Żadne	2
RJMP k	Skok względny	$PC \leftarrow PC + k + 1$	Żadne	2
IJMP	Skok pośredni do (Z)	$PC \leftarrow Z$	Żadne	2
RCALL k	Skok względny ze śladem	$PC \leftarrow PC + k + 1$	Żadne	3
ICALL	Skok pośredni ze śladem	$PC \leftarrow Z$	Żadne	3
CALL k	Skok bezpośredni	$PC \leftarrow k$	Żadne	3
RET	Powrót z procedury	$PC \leftarrow \text{STACK}$	Żadne	4
RETI	Powrót z obsługi przerwania	$PC \leftarrow \text{STACK}$	I	4
CPSE Rd,Rr	Porównaj, pomiń jeśli równe	if (Rd = Rr) $PC \leftarrow PC + 2$ or 3	Żadne	1 / 2 / 3
CP Rd,Rr	Porównaj	Rd - Rr	Z, N,V,C,H	1
CPC Rd,Rr	Porównaj z CARRY	Rd - Rr - C	Z, N,V,C,H	1
CPI Rd,K	Porównaj rejestr ze stałą ( $16 \leq d \leq 31$ )	Rd - K	Z, N,V,C,H	1
SBRC Rr, b	Pomiń jeśli bit w rejestrze równy 0	if (Rr(b)=0) $PC \leftarrow PC + 2$ or 3	Żadne	1 / 2 / 3
SBRS Rr, b	Pomiń jeśli bit w rejestrze równy 1	if (Rr(b)=1) $PC \leftarrow PC + 2$ or 3	Żadne	1 / 2 / 3
SBIC P, b	Pomiń jeśli bit w rejestrze I/O równy 0	if (P(b)=0) $PC \leftarrow PC + 2$ or 3	Żadne	1 / 2 / 3
SBIS P, b	Pomiń jeśli bit w rejestrze I/O równy 1	if (P(b)=1) $PC \leftarrow PC + 2$ or 3	Żadne	1 / 2 / 3
BRBS s, k	Skok gdy flaga STATUS równa 1	if (SREG(s) = 1) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRBC s, k	Skok gdy flaga STATUS równa 0	if (SREG(s) = 0) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BREQ k	Skok gdy równe	if (Z = 1) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRNE k	Skok gdy nierówne	if (Z = 0) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRCS k	Skok gdy CARRY równy 1	if (C = 1) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRCC k	Skok gdy CARRY równy 0	if (C = 0) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRSH k	Skok gdy większe lub równe	if (C = 0) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRLO k	Skok gdy mniejsze	if (C = 1) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRMI k	Skok gdy minus	if (N = 1) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRPL k	Skok gdy plus	if (N = 0) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRGE k	Skok gdy więk. lub równe, ze znakiem	if ( $N \oplus V = 0$ ) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRLT k	Skok gdy mniej niż zero, ze znakiem	if ( $N \oplus V = 1$ ) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRHS k	Skok gdy flaga HALF CARRY = 1	if (H = 1) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRHC k	Skok gdy flaga HALF CARRY = 0	if (H = 0) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRTS k	Skok gdy flaga T = 1	if (T = 1) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRTC k	Skok gdy flaga T = 0	if (T = 0) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRVS k	Skok gdy flaga OVERFLOW = 1	if (V = 1) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRVC k	Skok gdy flaga OVERFLOW = 0	if (V = 0) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRIE k	Skok gdy przerwania włączone	if (I = 1) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2
BRID k	Skok gdy przerwania wyłączone	if (I = 0) then $PC \leftarrow PC + k + 1$	Żadne	1 / 2

## Instrukcje transferu danych

<b>Mnemonic</b>	<b>Opis</b>	<b>Operacja</b>	<b>Flagi</b>	<b>Clk</b>
<b>MOV</b> Rd, Rr	Prześlij z rejestru do rejestru	$Rd \leftarrow Rr$	Żadne	1
<b>MOVW</b> Rd, Rr	Kopiuj słowo rejestru	$Rd + 1:Rr \leftarrow Rr + 1:Rr$	Żadne	1
<b>LDI</b> Rd, K	Ładuj stałą do rejestru ( $16 \leq d \leq 31$ )	$Rd \leftarrow K$	Żadne	1
<b>LD</b> Rd, X	Ładuj pośrednio	$Rd \leftarrow (X)$	Żadne	2
<b>LD</b> Rd, X+	Ładuj pośrednio i inkrementuj indeks	$Rd \leftarrow (X), X \leftarrow X + 1$	Żadne	
<b>LD</b> Rd, -X	Dekrementuj indeks i ładuj pośrednio	$X \leftarrow X - 1, Rd \leftarrow (X)$	Żadne	2
<b>LD</b> Rd, Y	Ładuj pośrednio	$Rd \leftarrow (Y)$	Żadne	2
<b>LD</b> Rd, Y+	Ładuj pośrednio i inkrementuj indeks	$Rd \leftarrow (Y), Y \leftarrow Y + 1$	Żadne	2
<b>LD</b> Rd, -Y	Dekrementuj indeks i ładuj pośrednio	$Y \leftarrow Y - 1, Rd \leftarrow (Y)$	Żadne	2
<b>LDD</b> Rd, Y+q	Ładuj pośrednio z przesunięciem	$Rd \leftarrow (Y + q)$	Żadne	2
<b>LD</b> Rd, Z	Ładuj pośrednio	$Rd \leftarrow (Z)$	Żadne	2
<b>LD</b> Rd, Z+	Ładuj pośrednio i inkrementuj indeks	$Rd \leftarrow (Z), Z \leftarrow Z + 1$	Żadne	2
<b>LD</b> Rd, -Z	Dekrementuj indeks i ładuj pośrednio	$Z \leftarrow Z - 1, Rd \leftarrow (Z)$	Żadne	2
<b>LDD</b> Rd, Z+q	Ładuj pośrednio z przesunięciem	$Rd \leftarrow (Z + q)$	Żadne	2
<b>LDS</b> Rd, k	Ładuj bezpośrednio z SRAM-u	$Rd \leftarrow (k)$	Żadne	2
<b>ST</b> X, Rr	Zachowaj pośrednio	$(X) \leftarrow Rr$	Żadne	2
<b>ST</b> X+, Rr	Zachowaj pośrednio i inkrementuj indeks	$(X) \leftarrow Rr, X \leftarrow X + 1$	Żadne	2
<b>ST</b> -X, Rr	Dekrementuj indeks i zachowaj pośrednio	$X \leftarrow X - 1, (X) \leftarrow Rr$	Żadne	2
<b>ST</b> Y, Rr	Zachowaj pośrednio	$(Y) \leftarrow Rr$	Żadne	2
<b>ST</b> Y+, Rr	Zachowaj pośrednio i inkrementuj indeks	$(Y) \leftarrow Rr, Y \leftarrow Y + 1$	Żadne	2
<b>ST</b> -Y, Rr	Dekrementuj indeks i zachowaj pośrednio	$Y \leftarrow Y - 1, (Y) \leftarrow Rr$	Żadne	2
<b>STD</b> Y+q, Rr	Zachowaj pośrednio z przesunięciem	$(Y + q) \leftarrow Rr$	Żadne	2
<b>ST</b> Z, Rr	Zachowaj pośrednio	$(Z) \leftarrow Rr$	Żadne	2
<b>ST</b> Z+, Rr	Zachowaj pośrednio i inkrementuj indeks	$(Z) \leftarrow Rr, Z \leftarrow Z + 1$	Żadne	2
<b>ST</b> -Z, Rr	Dekrementuj indeks i zachowaj pośrednio	$Z \leftarrow Z - 1, (Z) \leftarrow Rr$	Żadne	2
<b>STD</b> Z+q, Rr	Zachowaj pośrednio z przesunięciem	$(Z + q) \leftarrow Rr$	Żadne	2
<b>STS</b> k, Rr	Zachowaj bezpośrednio w SRAM-ie	$(k) \leftarrow Rr$	Żadne	2
<b>LPM</b>	Ładuj pamięć programu	$R0 \leftarrow (Z)$	Żadne	3
<b>LPM</b> Rd, Z	Ładuj pamięć programu	$R0 \leftarrow (Z)$	Żadne	3
<b>LPM</b> Rd, Z+	Ładuj pamięć programu i postinkrementuj	$R0 \leftarrow (Z), Z \leftarrow Z + 1$	Żadne	3
<b>SPM</b>	Zachowaj pamięć programu	$(Z) \leftarrow R1:R0$	Żadne	-
<b>IN</b> Rd, P	Czytaj port	$Rd \leftarrow P$	Żadne	1
<b>OUT</b> P, Rr	Wyślij na port	$P \leftarrow Rr$	Żadne	1
<b>PUSH</b> Rr	Wyślij na stos	$STACK \leftarrow Rr$	Żadne	2
<b>POP</b> Rd	Pobierz ze stosu	$Rd \leftarrow STACK$	Żadne	2

## Instrukcje operacji bitowych

<b>Mnemonic</b>	<b>Opis</b>	<b>Operacja</b>	<b>Flagi</b>	<b>Clk</b>
<b>SBI</b> P,b	Ustaw bit w rejestrze I/O	$I/O(P,b) \leftarrow 1$	Żadne	2
<b>CBI</b> P,b	Wyczyść bit w rejestrze I/O	$I/O(P,b) \leftarrow 0$	Żadne	2
<b>LSL</b> Rd	Logiczne przesunięcie w lewo	$Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0$	Z,C,N,V	1
<b>LSR</b> Rd	Logiczne przesunięcie w prawo	$Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0$	Z,C,N,V	1
<b>ROL</b> Rd	Rotacja w lewo przez CARRY	$Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7)$	Z,C,N,V	1
<b>ROR</b> Rd	Rotacja w prawo przez CARRY	$Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0)$	Z,C,N,V	1
<b>ASR</b> Rd	Arytmetyczne przesunięcie w prawo	$Rd(n) \leftarrow Rd(n+1), n=0..6$	Z,C,N,V	1
<b>SWAP</b> Rd	Zamień zawartości starszej i młodszej połowy rejestru	$Rd(3..0) \leftarrow Rd(7..4), Rd(7..4) \leftarrow Rd(3..0)$	Żadne	1
<b>BSET</b> s	Ustaw flagę	$SREG(s) \leftarrow 1$	SREG(s)	1
<b>BCLR</b> s	Wyczyść flagę	$SREG(s) \leftarrow 0$	SREG(s)	1
<b>BST</b> Rr, b	Przepisz bit z rejestru do T	$T \leftarrow Rr(b)$	T	1
<b>BLD</b> Rd, b	załaduj bit rejestru stanem T	$Rd(b) \leftarrow T$	Żadne	1
<b>SEC</b>	Ustaw flagę CARRY	$C \leftarrow 1$	C	1
<b>CLC</b>	Wyczyść flagę CARRY	$C \leftarrow 0$	C	1
<b>SEN</b>	Ustaw flagę N ( flaga NEGATIVE )	$N \leftarrow 1$	N	1
<b>CLN</b>	Wyczyść flagę N ( flaga NEGATIVE )	$N \leftarrow 0$	N	1
<b>SEZ</b>	Ustaw flagę ZERO	$Z \leftarrow 1$	Z	1
<b>CLZ</b>	Wyczyść flagę ZERO	$Z \leftarrow 0$	Z	1
<b>SEI</b>	Włącz przerwania	$I \leftarrow 1$	I	1
<b>CLI</b>	Wyłącz przerwania	$I \leftarrow 0$	I	1
<b>SES</b>	Ustaw flagę S ( flaga SIGNED TEST )	$S \leftarrow 1$	S	1
<b>CLS</b>	Wyczyść flagę S ( flaga SIGNED TEST )	$S \leftarrow 0$	S	1
<b>SEV</b>	Ustaw flagę V (TWOS COMPLEMENT OVERFL)	$V \leftarrow 1$	V	1
<b>CLV</b>	Wyczyść flagę V (TWOS COMPLEMENT OVER)	$V \leftarrow 0$	V	1
<b>SET</b>	Ustaw T w SREG	$T \leftarrow 1$	T	1
<b>CLT</b>	Wyczyść T w SREG	$T \leftarrow 0$	T	1
<b>SEH</b>	Ustaw flagę H (flaga HALF CARRY) w SREG	$H \leftarrow 1$	H	1
<b>CLH</b>	Wyczyść flagę H (flaga HALF CARRY) w SREG	$H \leftarrow 0$	H	1
<b>NOP</b>	Nic nie rób	----	Żadne	1
<b>SLEEP</b>	Wejść w stan Sleep	zobacz opis trybów pracy	Żadne	3
<b>WDR</b>	Resetuj Watchdog'a	zobacz opis watchdog'a	Żadne	1