

Magistrala SPI

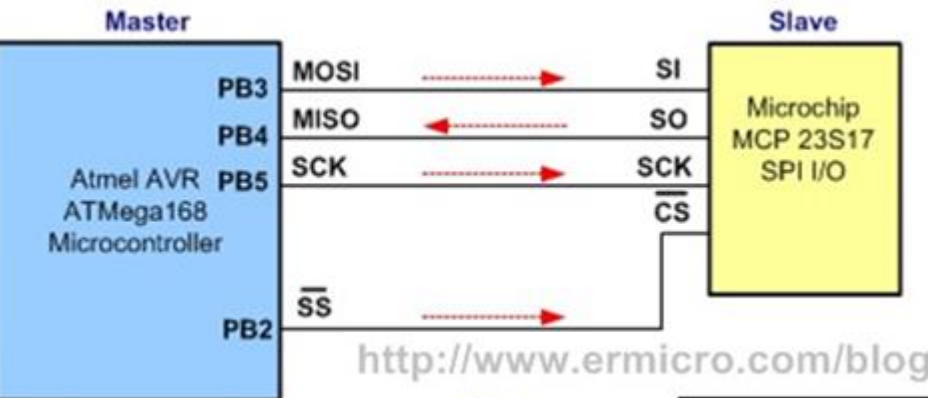
Magistrala SPI typu Master-Slave składa się z czterech linii:

- **MOSI** Master Output Slave Input zwany też **SI** – Slave Input
- **MISO** Master input Slave Output zwany też **SO** – Slave Output
- **SCK** Clock
- **/SS** Slave select zwany też **/CS** – Chip Select)

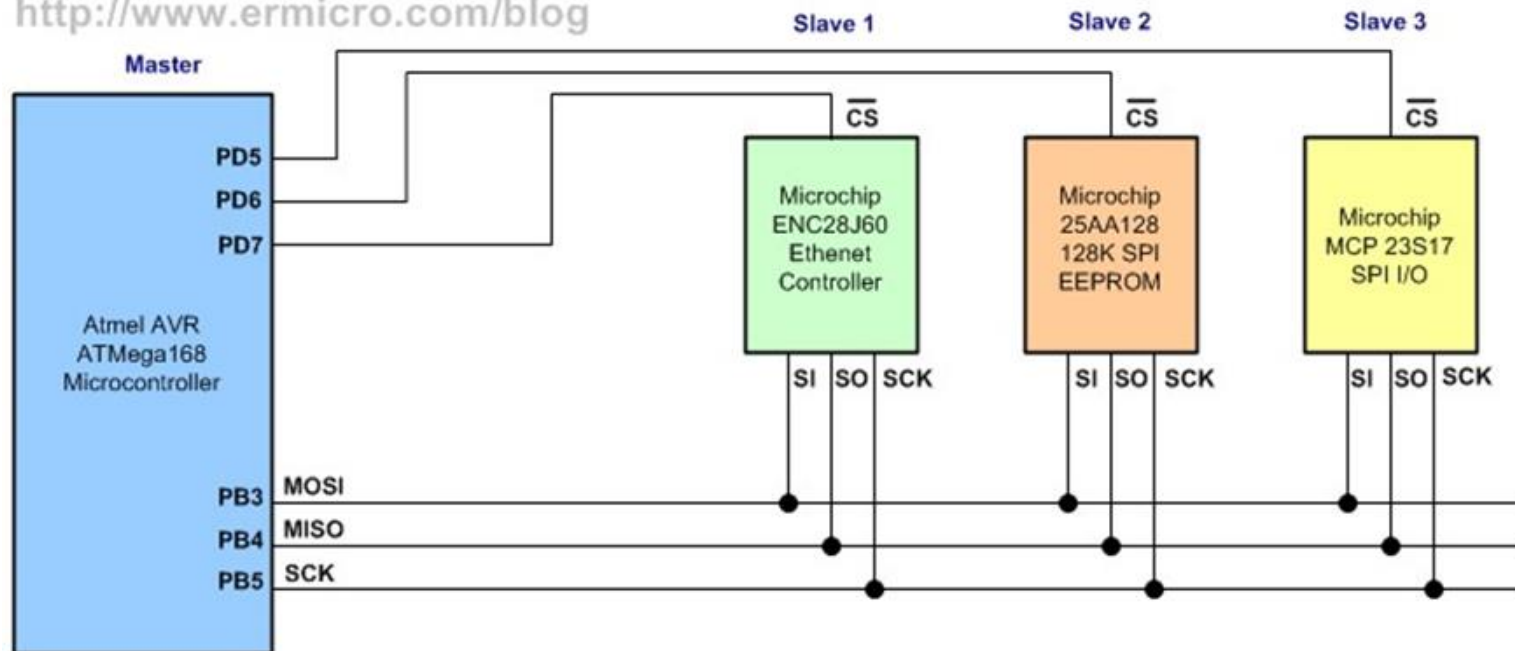
stan „idle” jest logiczną jedynką, a stan aktywny logicznym zerem (istnieją jednak układy cyfrowe, które wymagają odwrotnej polaryzacji na linii /SS)

Linie MOSI i MISO są wspólne dla wszystkich układów na magistrali, linia SS jest prowadzona do każdego układu Slave oddzielnie.

Master wybiera Slave do transmisji danych stanem niskim na linii /SS – wówczas pozostałe Slave’y przechodzą w stan wielkiej impedancji (High-Z).



<http://www.ermicro.com/blog>

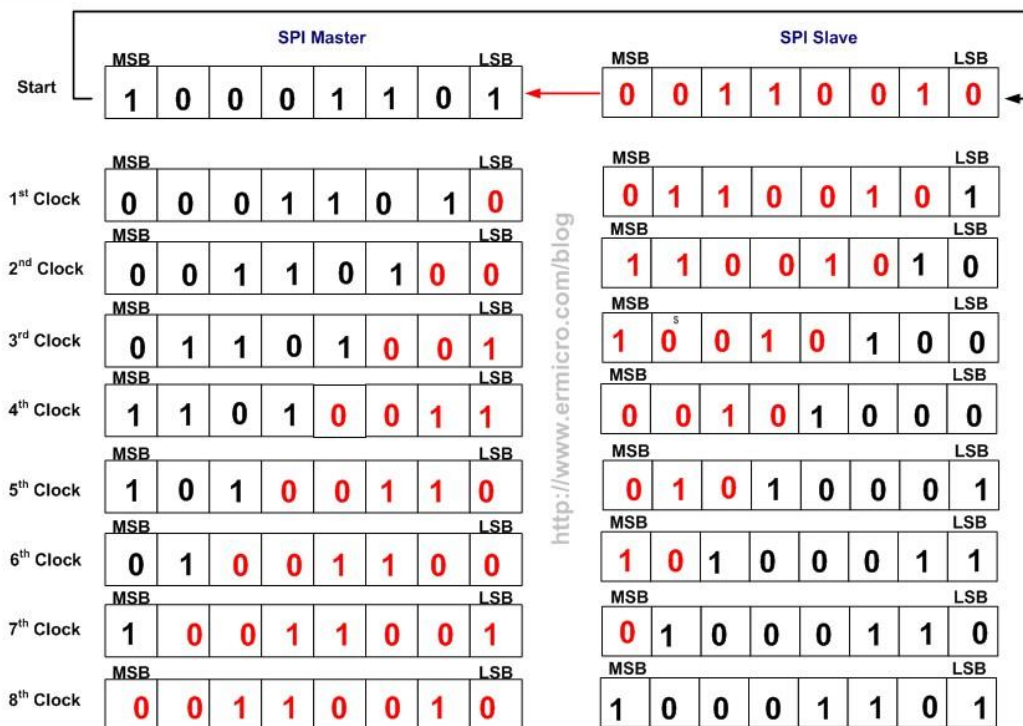
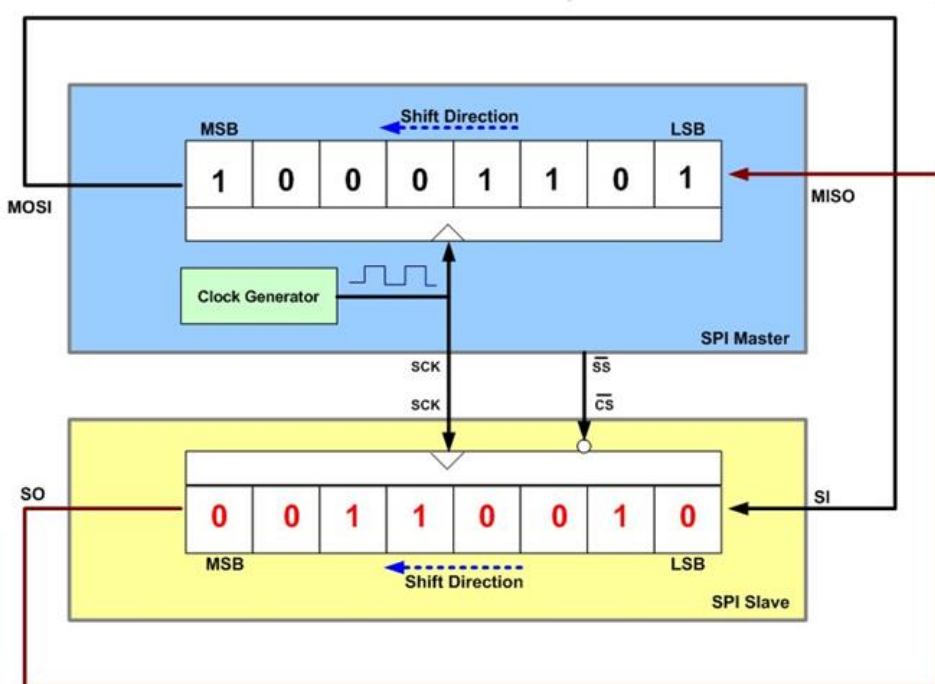


Transmisja bitów na magistrali SPI

Generatorem sygnału taktującego transmisję jest Master mikrokontroler.

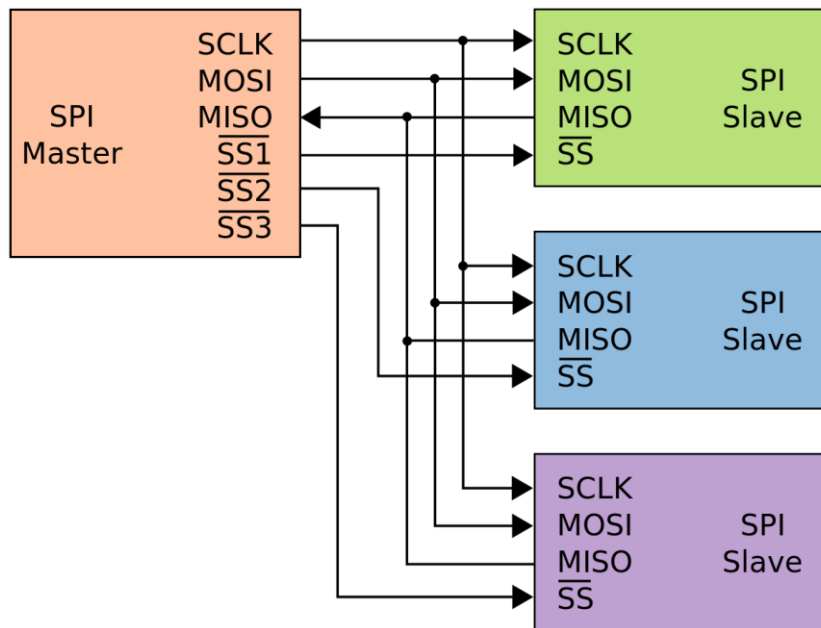
Transmisja po magistrali jest dookólna – urządzenia Master i Slave zachowują się jak rejestry przesuwne.

Po ośmiu taktach Master i Slave wymieniają się bajtami.

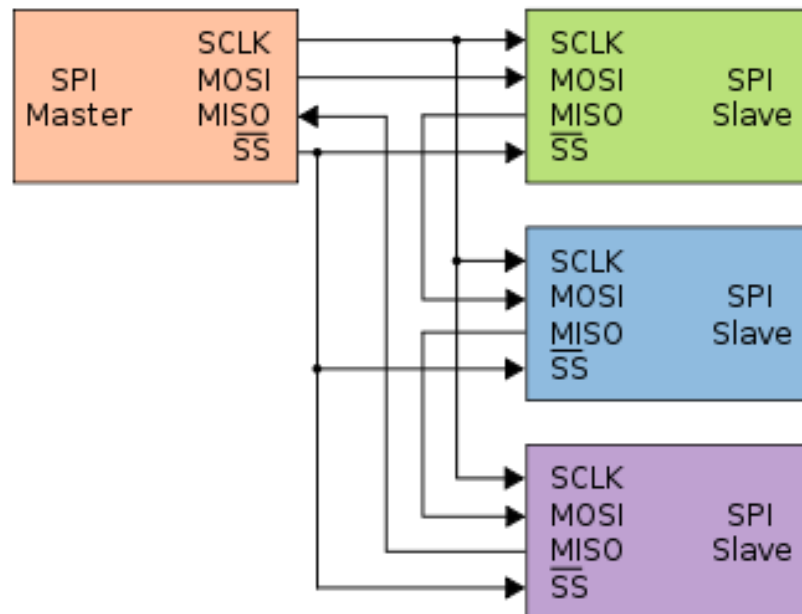


SPI Daisy chain

Niezależne (równoległe) podłączenie układów Slave



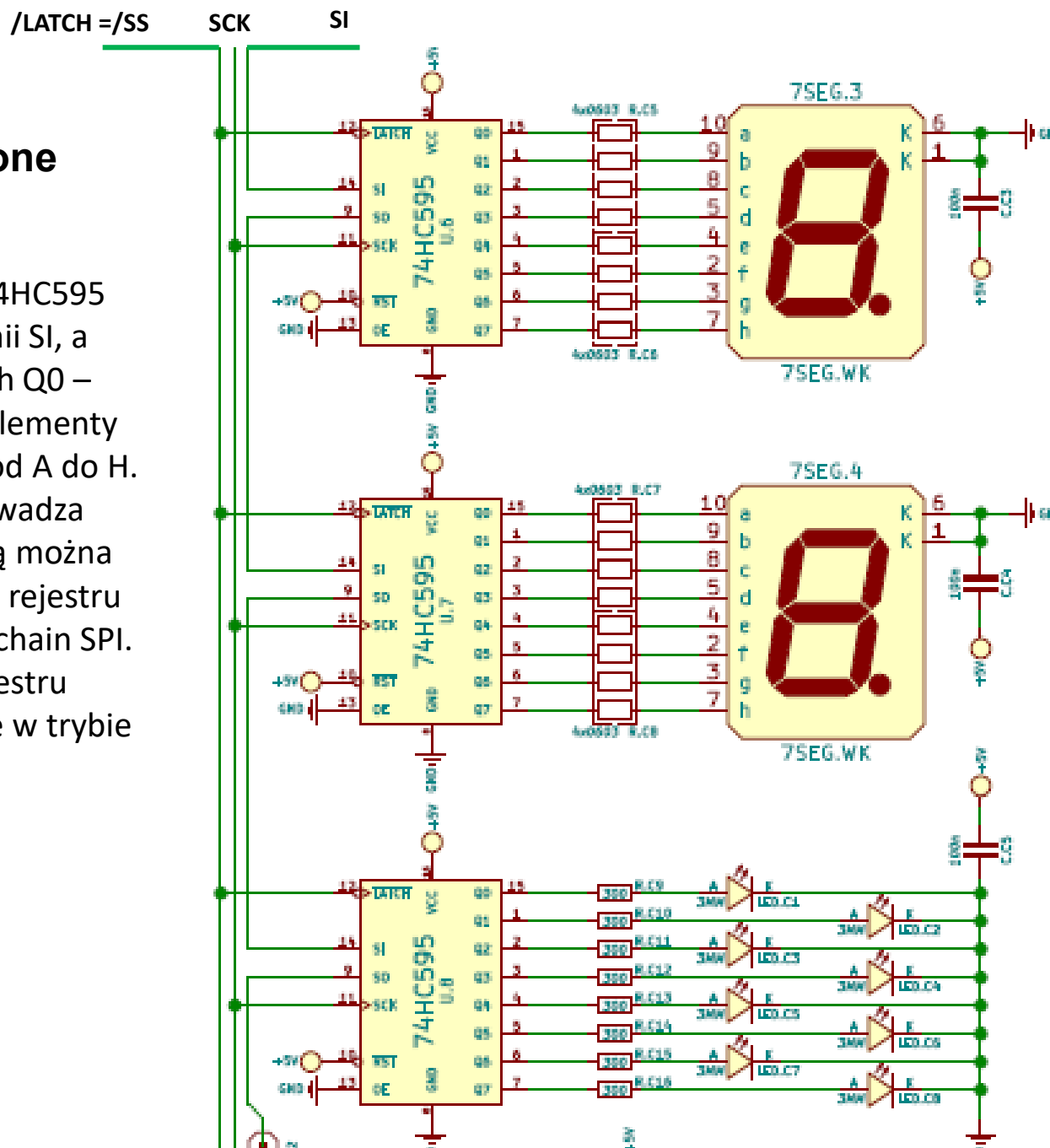
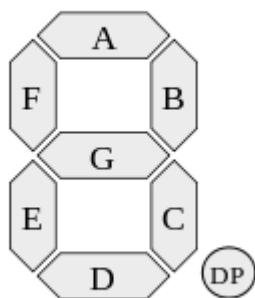
Układy Slave połączone w Daisy chain (szeregowo)



W konfiguracji Daisy chain wyjście pierwszego układu Slave jest połączone z wejściem drugiego itd. Każdy Slave jest „przezroczysty” dla przesyłanych danych czyli wysyła dalej identyczną kopię danych działając jak rejestr przesuwany. Konfiguracja Daisy chain wymaga tylko jednej linii $\overline{SS}$ dla wszystkich układów Slave.

Układy 74HC595 połączone w daisy chain

Rejestr szeregowo-równoległy 74HC595 przyjmuje dane szeregowo po linii SI, a wystawia je równoległe na liniach Q0 – Q7, gdzie można podłączyć np. elementy wyświetlacza 7-segmentowego od A do H. Ponadto rejestr 74HC595 wyprowadza dane szeregowo na linii SO, którą można połączyć z wejściem SI kolejnego rejestru 74HC595 tworząc łańcuch daisy chain SPI. Z analizy impulsów na liniach rejestru wynika, że magistrała SPI pracuje w trybie zero (CPOL = 0, CPHA = 0)



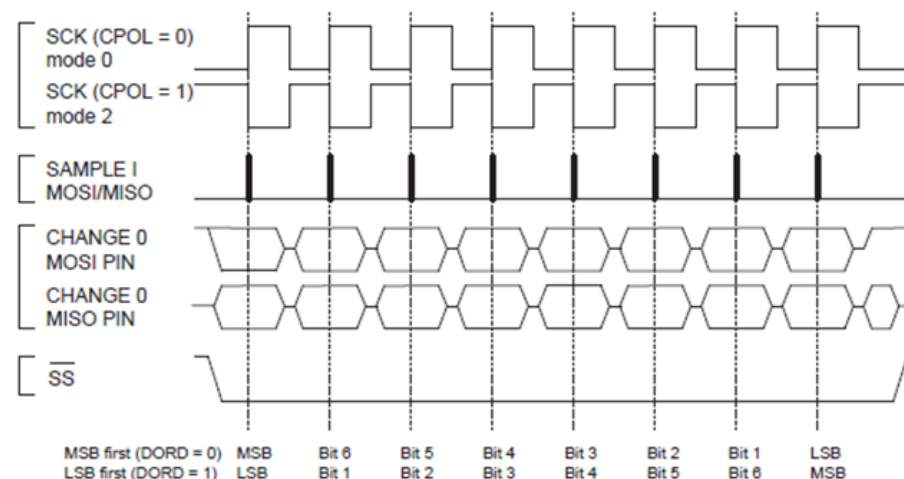
Tryby pracy interfejsu SPI

Bity konfiguracyjne CPOL i CPHA decydują o tym, na którym zboczach impulsu dane na magistrali są stabilne i można je odczytać (*sample*), a na którym następuje zmiana danych na linii (*setup*).

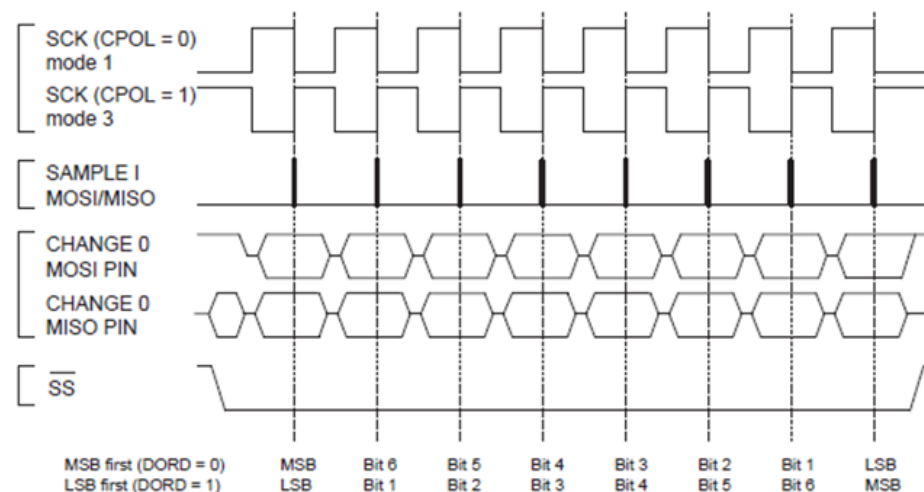
Możliwe są 4 kombinacje związane z fazą i polaryzacją impulsu zegarowego

Tryby SPI	SPI Modes	Przednie zbocze Impulsu <i>leading edge</i>	Tylne zbocze Impulsu <i>trailing edge</i>
SPI Mode	Conditions	Leading Edge	Trailing eDge
0	CPOL=0, CPHA=0	Sample (Rising)	Setup (Falling)
1	CPOL=0, CPHA=1	Setup (Rising)	Sample (Falling)
2	CPOL=1, CPHA=0	Sample (Falling)	Setup (Rising)
3	CPOL=1, CPHA=1	Setup (Falling)	Sample (Rising)

SPI Transfer Format with CPHA = 0



SPI Transfer Format with CPHA = 1



Rejestry interfejsu SPI

Rejestr kontrolny interfejsu SPI - SPCR

Bit	7	6	5	4	3	2	1	0
Nazwa	SPIE	SPE	DODR	MSTR	CPOL	CPHA	SPR1	SPR0
Odczyt/zapis	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

- Bit 7 – SPIE: maska przerwania interfejsu SPI
- Bit 6 – SPE: włączenie interfejsu
- Bit 5 – DODR: kolejność przesyłania bitów: 0 – MSB pierwszy, 1 – LSB pierwszy
- Bit 4 – MSTR: tryb Master/Slave (tylko jeden mikrokontroler na magistrali może być Master, kolejne muszą być Slav'ami)
- Bit 3 – CPOL: polaryzacja sygnału taktującego
- Bit 2 – CPHA: faza sygnału taktującego
- Bit 1 – SPR1, bit 0 – SPR0: częstotliwości taktująca transmisję (w stosunku do zegara systemowego f_{osc})

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

Rejestry interfejsu SPI

Rejestr statusowy interfejsu SPI - SPSR

Bit	7	6	5	4	3	2	1	0
Nazwa	SPIF	WCOL	-	-	-	-	-	-
Odczyt/zapis	R	R	R	R	R	R	R	R
Wartość początkowa	0	0	0	0	0	0	0	0

- Bit 7 – SPIF: flaga zakończenia transferu 1-go bajtu danych
- Bit 6 – WCOL: flaga kolizji

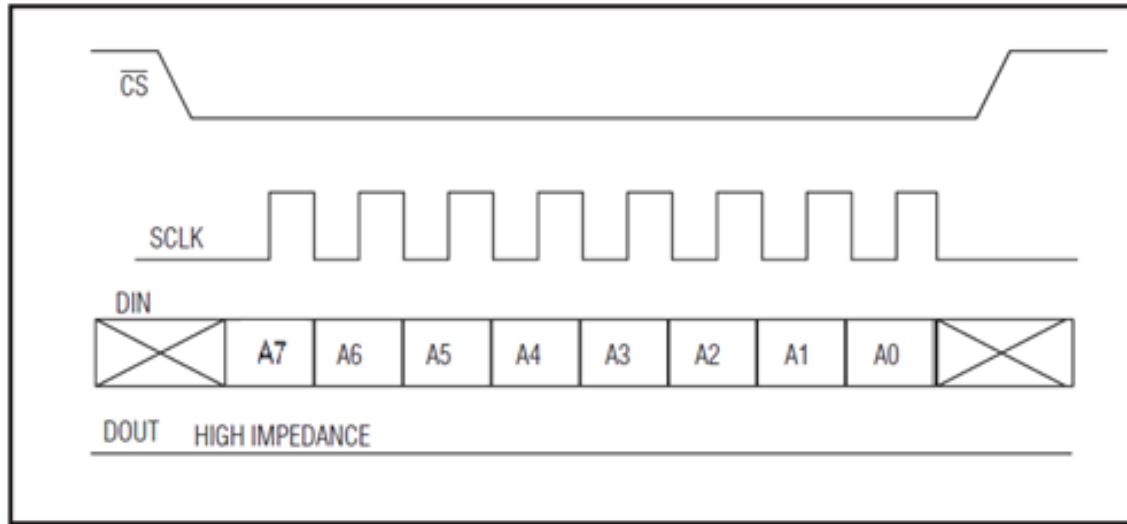
Rejestr danych interfejsu SPI - SPDR

Bit	7	6	5	4	3	2	1	0
Nazwa	MSB	-	-	-	-	-	-	LSB
Odczyt/zapis	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Wartość początkowa	X	X	X	X	X	X	X	X

Rejestr SPDR służy zarówno do zapisu jak i odczytu danych z interfejsu SPI

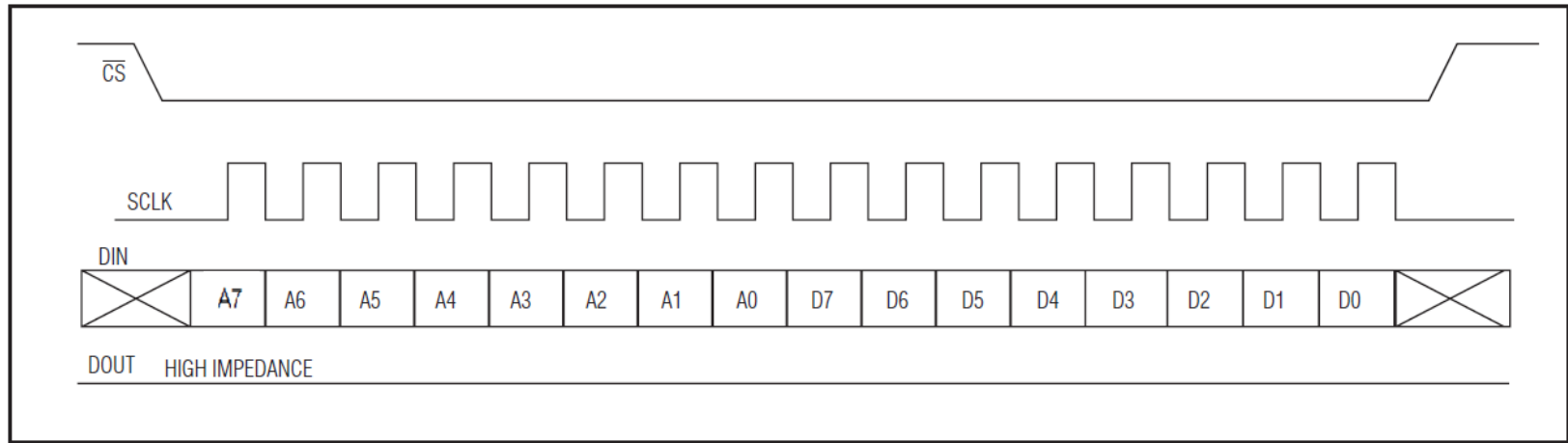
Przykładowa transmisja po magistrali SPI (mode 0: CPOL=0, CPHA=0)

Zapis jednego bajtu po linii MOSI bez odczytu

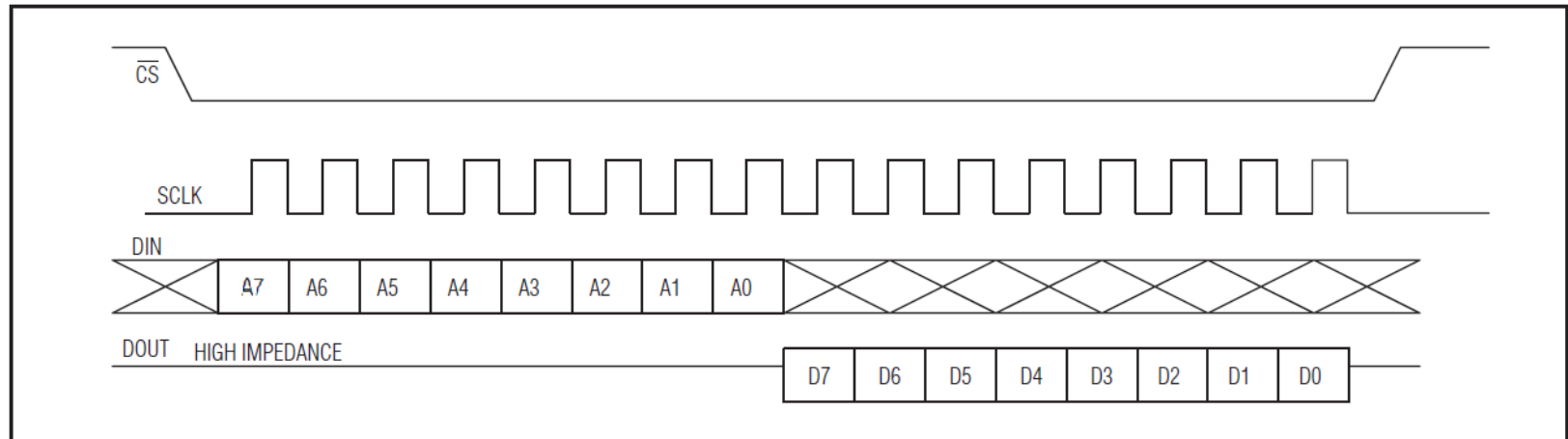


Przykładowa transmisja po magistrali SPI (mode 0: CPOL=0, CPHA=0)

Zapis dwóch bajtów po linii MOSI

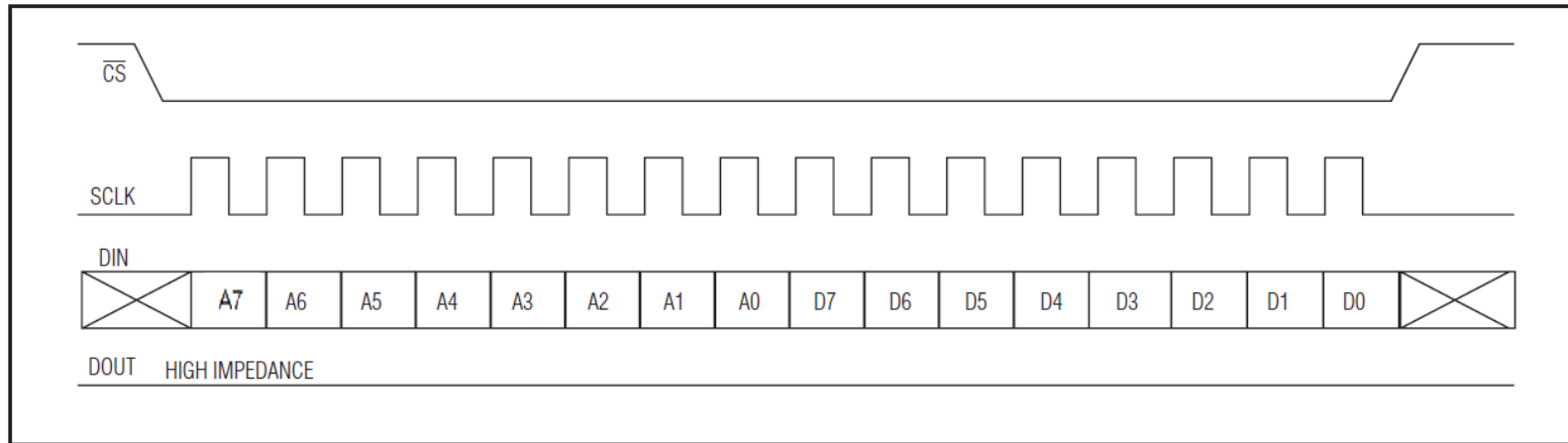


Zapis jednego bajtu po linii MOSI i odczyt jednego bajtu po linii MISO

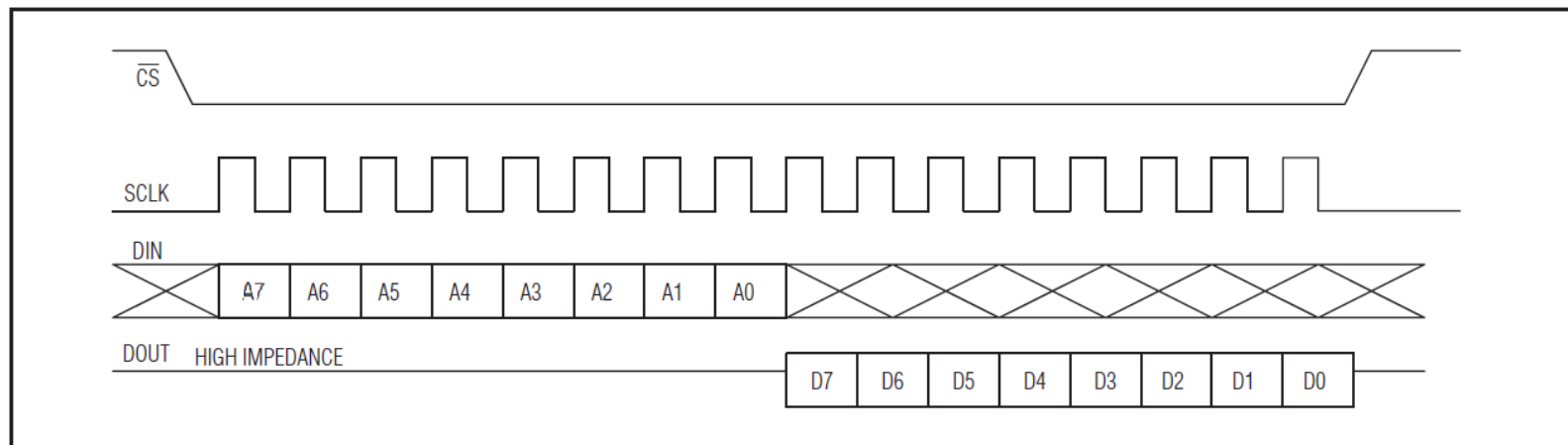


Przykładowa transmisja po magistrali SPI (mode 1: CPOL=0, CPHA=1)

Zapis dwóch bajtów po linii MOSI



Zapis jednego bajtu po linii MOSI i odczyt jednego bajtu po linii MISO



Przykładowe procedury ini oraz transferu po SPI

```
void setup()
{
  DDRB |= (1<<SS)|(1<<MOSI)|(1<<SCK); //kierunek output na liniach MOSI, /SS i SCK
  PORTB &= ~(1<<SCK); //SCK startuje od logicznego 0
  PORTB |= (1<<SS); //SS startuje od logicznej jedynki
  SPCR=(1<<SPE) | (1<<MSTR)|(1<<SPR1); //interfejs SPI włączony w trybie Master,
                                     tryb 0 (CPOL = 0, CPHA = 0, prędkość wybrana (SPR1 =1)
}
```

Procedura transferu bajtu umożliwia
jednoczesne przesłanie i odbiór bajtu ze
względem na dookólny charakter
transmisji.

```
char SPI_transfer(char data_out)
{
  char data_in;
  PORTB &= ~(1<<SS);
  SPDR = data_out;
  while (!(SPSR & (1<<SPIF)));
  PORTB |= (1<<SS);
  return data_in;
}
```

Interfejs mikrokontrolera ATmega 328

