

Systemy komputerowe (W07)

Sytuacje wyjątkowe i systemy obsługi urządzeń wejścia/wyjścia
A.Pruszkowski

Sytuacje wyjątkowe

SYKO Sytuacje wyjątkowe

- Sytuacje wyjątkowe
 - CPU zawiesza wykonywanie aktualnego kodu i wykonuje skok do procedur obsługi tej sytuacji
 - Zdarzenia niezamierzone
 - Błąd adresowania lub rozkazu
 - Błąd argumentu
 - np.: dzielenie przez zero podczas instrukcji procesora DIV
 - Zdarzenia zamierzone - wewnętrzne
 - Praca w trybie śledzenia
 - Po każdym rozkazie wykonywana jest obsługa procedury obsługi
 - Praca w trybie pułapki
 - Program działa normalnie do napotkania rozkazu pułapki (umieszczonego celowo)
 - Przerwanie programowe
 - Metoda przekazywania sterowania do systemu operacyjnego (np.: INT21)

SYKO Sytuacje wyjątkowe

- Sytuacje wyjątkowe
 - Zdarzenia zewnętrzne
 - Błąd urządzeń peryferyjnych
 - Błąd na magistrali adresowej danych (w tym parzystość czy ECC)
 - Zanik zasilania
 - Przerwania związane ze zmianą stanu sprzętu
 - Przerwanie zegarowe, przerwanie odebrania pakietu kartą sieciową, ...
- Przerwania
 - Oprogramowanie może decydować czy CPU ma reagować na określone przerwania
 - Mamy dwa typy przerwań
 - Maskowalne
 - Niemaskowalne
 - W x86 przerwania niemaskowalne także można było maskować

SYKO Przerwania

- Przerwania dlaczego w ogóle potrzebne?
 - Dane z urządzeń peryferyjnych nie muszą być dostępne natychmiast bo ich wypracowanie:
 - Wymaga czasu
 - dysk magnetyczny (HDD) musi dotrzeć (obrócić talerz, przesunąć głowicę) do żądanych informacji
 - Pewne dane wypracowuje otoczenie
 - system komputerowy może czekać na wprowadzenie danych przez użytkownika
 - dane mogą napływać z innych komputerów, np.: poprzez sieć Internet
 - Bez przerwania kod wyglądałby (tzw. polling) - oczekiwanie na naciśnięcie klawisza w PC

```
MAIN:    ...
Wait:    IN  AL, 0x64      ;odczytaj „stan klawiatury” (poprzez port 0x64)
          AND AL, 1       ;maska wyłączająca inne przyczyny zmiany stanu
          JZ  Wait        ;nie naciśnięto klawisza - skacz do Wait
          IN  AL, 0x60     ;odczytaj kod naciśnietego klawisza
          ...             ;obróbka otrzymanego numeru klawisza
```

SYKO Przerwania

- Przerwania dlaczego w ogóle potrzebne, cd.?
 - Podejście poprawne, ale nie efektywne
 - program nie robi nic poza sprawdzaniem stanu określonego portu (instrukcja IN)
 - Z obsługą przerwania napisalibyśmy

```
MAIN:          ...
               CALL Enable_IRQ_KEYBOARD
               ...
IRQ_KEYBOARD:  PUSH AL
               IN  AL, 0x64          ;odczytaj „stan klawiatury”
               AND AL, 1             ;maska wyłączająca inne przyczyny zmiany stanu
               JNZ  SKIP_IRQ         ;nie naciśnięto klawisza – opuść przerwanie
               IN  AL, 0x60          ;odczytaj kod naciśnietego klawisza
               ...                   ;obróbka otrzymanego numeru klawisza
SKIP_IRQ:     POP AL
               IRET                  ;powrót z przerwania
```

- W powyższym w czasie działania programu głównego przerwanie będzie w tle obsługiwać klawiaturę – oszczędzamy moc obliczeniową
 - Koszt: komplikacja oprogramowania
 - Konieczność buforowania danych – w przerwaniu nie przetwarzamy danych!
 - Program główny i kod obsługi przerwania mogą działać współbieżnie – wymagana synchronizacja danych

SYKO Przerwania

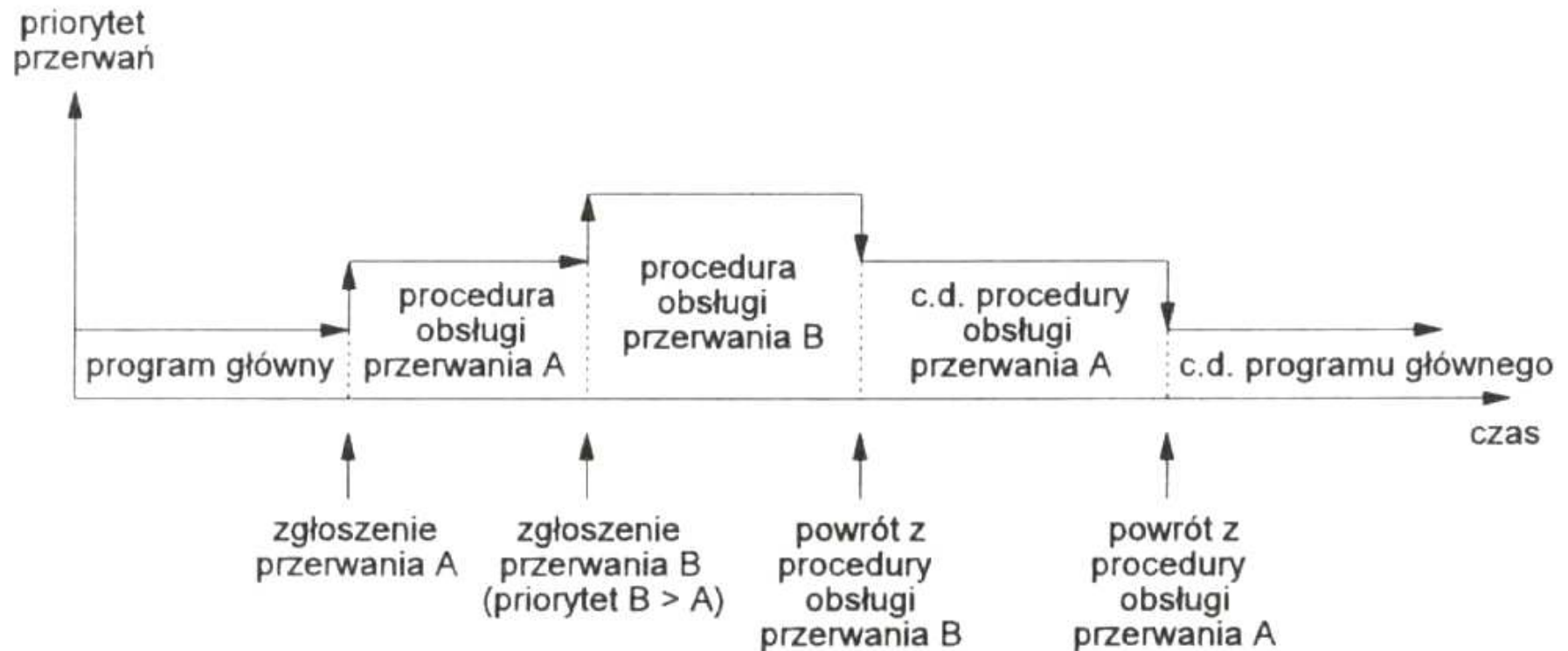
- Przerwania - Jak CPU obsługuje przerwania
 - 1)pod koniec fazy wykonania operacji maszynowych sprawdzany jest stan wejść
 - INT - przerwania sprzętowe (ich może być wiele)
 - NMI – przerwania niemaskowalne (są CPU nie posiadające tego typu wejścia)
 - 2)jeżeli stan nie wymaga wejścia w procedurę obsługi przerwania - CPU przechodzi do fazy pobrania
 - 3)zawartość rejestrów EPC, EFLAGS odkładana jest na stos
 - 4)generowany jest adres pod który wykonany będzie skok i wpisywany jest on do EPC, i przechodzi CPU do fazy pobrania
 - Taki sposób realizacji jest optymalny - przerwanie nie dzieli żadnej z faz pracy CPU

SYKO Przerwania

- Przerwania – Co programista musi zrobić w kodzie obsługi przerwania
 - 1) Jeżeli to możliwe wyłączyć przerwania
 - niekontrolowana „rekurencja przerwań” może skończyć się katastrofą (!)
 - przepełnienie stosu może występować w „dziwnych momentach”, trudnych w diagnostyce
 - decydujące mogą być: kolejność wywołań przerwań, stopień zapisania stosu
 - niektóre procesory domyślnie wyłączają możliwość przerywania wykonywanych właśnie przerwań
 - jeżeli programista życzy sobie - może odblokować możliwość przyjmowania przerwań
 - 2) Zapamiętać na stosie wszystkie rejestry używane w funkcji obsługi przerwania
 - PUSH EAX; PUSH EBX; ... lub prościej (mniej efektywnie) PUSHA
 - czasami wcześniej trzeba przełączyć stos
 - 3) Wykonać zadania możliwie **najkrócej jak to możliwe!**
 - 4) Odtworzyć w odwrotnej kolejności ze stosu zawartości rejestrów
 - POP ...; POP EBX; POP EAX lub prościej POPA
 - 5) Wykonać instrukcję IRET / RETI
 - w innych procesorach zwykłą instrukcją RET lub jej odpowiednik

SYKO Przerwania

- Priorytety przerwania
 - Przerwania przerywające inne przerwania
 - „rekurencja przerwania”



SYKO Przerwania

- Przerwania – w x86
 - W trybie rzeczywistym tablica przerwania umiejscowiona jest pod adresem 0
 - Struktura - jeden wpis zajmuje 4B i jest adresem procedury obsługi przerwania
 - wybrane przykładowe lokacje w tablicy przerwania x86
 - 0x00 Dzielenie stałoprzecinkowe przez zero
 - 0x01 Przerwanie programowe (gdy TF=1 przerwanie wywoływane po każdym rozkazie)
 - 0x02 Przerwanie NMI związane z błędem parzystości pamięci lub koprocatora
 - 0x03 Pułapka programowa (generowana przez INT 3)
 - 0x06 Niedozwolony kod rozkazu
 - 0x0B Segment nieobecny w pamięci głównej
 - 0x0C/0x0D Przekroczenie segmentu stosu
 - 0x0E Błąd strony
 - W trybie chronionym przerwania opisuje tablica deskryptorów przerwania IDT
 - umiejscowienie tej pamięci wyznacza rejestr IDTR

SYKO Przerwania

- Przerwania – w x86
 - Wybrane przerwania sprzętowe w x86 (obsługiwane przez układ 8259)
 - IRQ0 Zegar systemowy (lokacja w tablicy przerwań: 0x08)
 - IRQ1 Obsługa klawiatury (0x09)
 - IRQ2 Kaskada dla podrzędnego kontrolera IRQ (0x0A)
 - IRQ3/IRQ4 Obsługa portu szeregowego COM2/COM1 (0x0B, 0x0C)
 - IRQ6 Obsługa sterownika napędu dysków elastycznych (0x0E)
 - IRQ5/IRQ7 Obsługa portu równoległego LPT2/LPT1 (0x0D, 0x0F)
 - IRQ8 Obsługa zegara czasu rzeczywistego (0x70)
 - IRQ9 Wywołanie przerwania IRQ2 gdy połączone kaskadowe (0x71)
 - IRQ14 Obsługa sterownika twardego dysku (0x76)
 - Nie przypisane: IRQ10, IRQ11, IRQ12, IRQ15
 - Karty rozszerzeń mogły z nich korzystać

SYKO Przerwania

- Tablica wektorów przerwania innych procesorów – 80C51
 - Wycinek obok przedstawia listę przerwania z nazwami i adresami
 - Procedury obsługi przerwania rozmieszczone co 8B
 - w tym MCU zakłada się, że są przerwania których kod może zmieścić się w 8B
 - przykład procedury obsługi zmieniającej stan bitu 1 portu 3 np.:

```
ORG 000BH ;gdzie umieścić kod
          ;tu pobudzenie to T0
CPL P3.1 ;P3.1=!P3.1
RETI     ;powrót z przerwania
```

Źródło: Tomasz STARECKI, „Mikrokontrolery jednocukładowe rodziny 51”

Tabela 4.1. Zestawienie przerwania występujących w wybranych mikrokontrolerach rodziny 51

Źródło przerwania	Wskaźniki przerwania	Wektor przerwania	Uwagi
przerwanie zewnętrzne INT0	IE0	0003H	
przerwanie zewnętrzne INT1	IE1	0013H	za wyjątkiem CE558 i CE559
przerwanie zewnętrzne INT1 i licznik sekundowy	IE1+SECINT	0013H	CE558, CE559
przerwanie zewnętrzne INT2	IQ2	003BH	mikrokontrolery odmiany CL
przerwanie zewnętrzne INT2	IEX2	004BH	C515, C515A, C517, C517A
przerwanie zewnętrzne INT2	IE2	0053H	C51GB
przerwanie zewnętrzne INT2	IE2	0043H	C310, C320, C323, C520, C530
przerwanie zewnętrzne INT3	IQ3	0043H	mikrokontrolery odmiany CL
przerwanie zewnętrzne INT3	IEX3	0053H	C515, C515A, C517, C517A
przerwanie zewnętrzne INT3	IE3	005BH	C51GB
przerwanie zewnętrzne INT3	IE3	004BH	C310, C320, C323, C520, C530
przerwanie zewnętrzne INT4	IQ4	004BH	mikrokontrolery odmiany CL
przerwanie zewnętrzne INT4	IEX4	005BH	C515, C515A, C517, C517A
przerwanie zewnętrzne INT4	IE4	0063H	C51GB
przerwanie zewnętrzne INT4	IE4	0053H	C310, C320, C323, C520, C530
przerwanie zewnętrzne INT5	IQ5	0053H	mikrokontrolery odmiany CL
przerwanie zewnętrzne INT5	IEX5	0063H	C515, C515A, C517, C517A
przerwanie zewnętrzne INT5	IE5	006BH	C51GB
przerwanie zewnętrzne INT5	IE5	005BH	C310, C320, C323, C520, C530
przerwanie zewnętrzne INT6	IQ6	005BH	mikrokontrolery odmiany CL
przerwanie zewnętrzne INT6	IEX6	006BH	C515, C515A, C517, C517A
przerwanie zewnętrzne INT6	IE6	0073H	C51GB
przerwanie zewnętrzne INT7	IQ7	0063H	mikrokontrolery odmiany CL
przerwanie zewnętrzne INT8	IQ8	006BH	mikrokontrolery odmiany CL
przerwanie zewnętrzne INT9	IQ9	0073H	odmiana CL, za wyjątkiem CL580
układ licznikowy T0	TF0	000BH	za wyjątkiem C748–C752

SYKO Przerwania

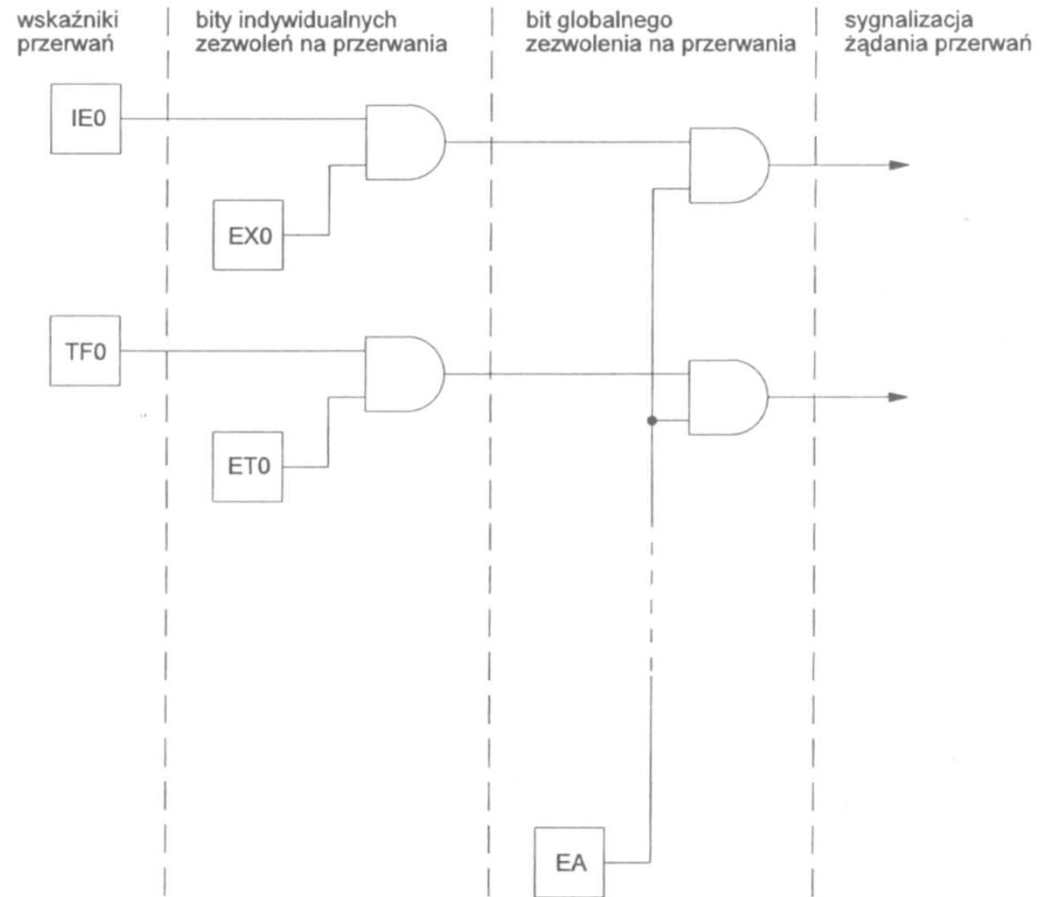
- 80C51 – kompilator SDCC:
 - Sposób zapisu funkcji obsługi przerw

```
void timer_isr (void) __interrupt (1) __using (1){  
    ...  
}
```

- Specjalne słowa kluczowe pozwalają odpowiednio
 - ustalić numer wektora przerw „__interrupt (numer_przerw)”
 - ustalić bank rejestrów używanych przez przerwanie „__using (numer_banku)”
 - metoda szybkiego przełączenia kontekstu – eliminuje konieczność zapamiętywania rejestrów na stosie

SYKO Przerwania

- Obsługa przerwania w 80C51 (wycinek)
 - Włączenie przerwania wymaga ustawienia wielu tzw. bitów sterujących
 - sbit w strukturze SFR
 - na rysunku
 - EA – włącza wszystkie przerwania
 - EX0, ET0 – włącza przerwanie od określonych źródeł
 - TF0 – przepełnienie licznika jako źródło przerwania



Źródło: Tomasz STARECKI, „Mikrokontrolery jednoukładowe rodziny 51”, W-wa 1996

Rys. 4.1. Fragment struktury układu przerwania przedstawiający sposób funkcjonowania wskaźników przerwania i bitów zezwoleń na przerwanie.

SYKO Przerwania

- Tablica wektorów przerwań innych procesorów – AVR
 - Jeden wpis w tablicy mieści 4B (opcode są 2B)
 - Tablica to spis instrukcji JMP <ADDRES_WLASCIVEGO_HANDLERA>
 - JMP to zajmuje 4B

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

Źródło: 8-bit AVR Microcontroller
ATmega128A DATASHEET
COMPLETE

SYKO Przerwania

- Tablica wektorów przerwania innych procesorów – AVR, cd.
 - Przykład kodu startowego dla procesora AVR

```
$0000      jmp RESET      ; Reset Handler
$0002      jmp EXT_INT0   ; IRQ0 Handler
$0004      jmp EXT_INT1   ; IRQ1 Handler
$0006      jmp EXT_INT2   ; IRQ2 Handler
$0042      jmp TWI      ; Two-wire Serial Interface Interrupt Handler
$0044      jmp SPM_RDY    ; SPM Ready Handler (re-programowanie
...                ; pamięci FLASH)
...
$0046 RESET: ldi r16, high(RAMEND) ; Main program start
$0047      out SPH,r16      ; Set stack pointer to top of RAM
$0048      ldi r16, low(RAMEND)
$0049      out SPL,r16
$004A      sei              ; Enable interrupts
$004B      <instr> xxx     ; Main code
...
```

RAMEND – określa ile pamięci
posada dany typ AVR MCU
W AVR położeniu czegoś na
stosie wskaźnik stosu SP jest
zmniejszany

SYKO Przerwania

■ Przerwania w AVR-GCC:

```
#include <avr/interrupt.h>
#include <util/atomic.h>

volatile char my_flag=0;

ISR(INT0_vect) {
    my_flag=1; //ustawienie naszej flagi
    ...      //jak najmniej operacji
}

int main(void) {
    ...      //Konfiguracja przerwań
    char local_copy_my_flag;
    for(;;){
        ATOMIC_BLOCK(ATOMIC_RESTORESTATE) {
            local_copy_my_flag = my_flag;
        }
        if(local_copy_my_flag!=0){
            ATOMIC_BLOCK(ATOMIC_RESTORESTATE){ my_flag=0; } //skasowanie flagi
            ...      //tu wiemy że przerwanie zostało wywołane
            ...      //tu możemy dokonywać przetwarzania danych „złapanych” w przerwaniu
        }
    }
    ...
}
```

ATOMIC_BLOCK - tworzy blok kodu jako sekcja atomowa

ATOMIC_RESTORESTATE - przywrócenie stanu systemu z przed wejściem do sekcji atomowej (tu rej. SREG)

ISR - funkcja obsługi przerwania

volatile - zmienna nie będzie podlegać optymalizacji (przydatne gdy kod funkcji przerwania i kod główny korzystają z takich zmiennych)

INT0_vect - wektor przerwania wspierany przez dane MCU (np.: wejście INT0 czyli „External Interrupt 0”)

SYKO Przerwania

■ Przerwania w Arduino:

```
const byte ledPin = 13;
const byte interruptPin = 2; //wejście przerwania
volatile byte state = LOW;
void setup() {
    pinMode(ledPin, OUTPUT);
    pinMode(interruptPin, INPUT_PULLUP);
    attachInterrupt (digitalPinToInterrupt(interruptPin), blink, CHANGE);
}
void loop() {
    digitalWrite(ledPin, state);
}
void blink() {
    state = !state;
}
```

digitalPinToInterrupt - zamiana numeru pinu w notacji Arduino na notacje wymagana przez m.in. `attachInterrupt`

attachInterrupt - podłącz funkcję „`blink()`” pod przerwanie związane ze zmianą stanu pinu `interruptPin (2)`, nie wszystkie piny potrafią „generować” przerwania

■ Brak sekcji atomowej – bo kod jest dość prosty

SYKO Przerwania

- Tablica wektorów przerwania procesorów zgodnych z ARM – STM32F0xxx Cortex-M0
 - Obszar wektorów przerwania od IRQ podzielono na wpisy o wielkości 4B
 - Numery IRQ ujemne dla zaznaczenia, że są to wyjątki (exceptions) a nie przerwania sprzętowe

Exception number ⁽¹⁾	IRQ number ⁽¹⁾	Exception type	Priority	Vector address or offset ⁽²⁾	Activation
1	-	Reset	-3, the highest	0x00000004	Asynchronous
2	-14	NMI	-2	0x00000008	Asynchronous
3	-13	Hard fault	-1	0x0000000C	Synchronous
4-10	-	Reserved	-	-	-
11	-5	SVCall	Configurable ⁽³⁾	0x0000002C	Synchronous
12-13	-	Reserved	-	-	-
14	-2	PendSV	Configurable ⁽³⁾	0x00000038	Asynchronous
15	-1	SysTick	Configurable ⁽³⁾	0x0000003C	Asynchronous
16 - 47	0 - 31	Interrupt (IRQ)	Configurable ⁽³⁾	0x00000040 and above ⁽⁴⁾	Asynchronous

Źródło: STM32F0xxx Cortex-M0 programming manual

SYKO Przerwania

- Deklarowanie funkcji obsługi przerwania w GCC (bez wsparcia specjalistycznych bibliotek)

- dla CPU ARM:

```
void f() __attribute__((interrupt ("IRQ"))); //lub: FIQ, SWI, ABORT, UNDEF, ...
void f(void) {
    ...
}
```

- Dla CPU RISC-V:

```
void f(void) __attribute__((interrupt ("user"))); //lub: supervisor, machine
void f(void) {
    ...
}
```

- Dla X86:

```
struct interrupt_frame; //deklaracja zgodna z manuałem CPU
__attribute__((interrupt)) void f(struct interrupt_frame* frame){
    ...
}
```

SYKO Przerwania

- Przypomnienie - Konwencja wywołania funkcji (ang. Calling convention)
 - Cdecl
 - argumenty: przekazywane tylko przez stos – stos w porządku: RTL (C)
 - stos czyści kod wołany („Caller”)
 - Pascal
 - argumenty: przekazywane tylko przez stos – stos w porządku: LTR (Pascal)
 - stos czyści kod wywołujący („Callee”)
 - System V (dla x86-64)
 - argumenty: RDI, RSI, RDX, RCX, R8, R9, [XYZ]MM0–7 – w porządku: RTL (C)
 - wprowadza 128B tzw. „red zone”
 - obszar o stałej wielkości umieszczony w ramce funkcji – domyślnie alokowany
 - stos czyści kod wołany („Caller”)

RTL – arg. on stack from right to left
LTR – arg. on stack from left to right

Obsługa urządzeń wejścia/wyjścia

SYKO Obsługa urządzeń wejścia/wyjścia

- Urządzenia wejścia/wyjścia
 - Procesor x86 widzi je w specjalnej przestrzeni
 - Jej wielkość to 65536 lokacji (pierwotnie tylko 1024)
 - Dostęp do przestrzeni za pomocą instrukcji IN, OUT, INS, OUTS
 - Mapa przestrzeni I/O – wyciąg
 - 0x0000-0x001F/0x00C0-0x00DF
 - podstawowy/drugi sterownik DMA 8237A – dawniej używany był 82237 dziś PCI-DMA
 - 0x0020-0x0021/0x00A0-0x00A1
 - podstawowy/podrzędny kontroler przerwań 8259A
 - 0x0040-0x0047
 - programowalny timer – 8253/8254
 - 0x0060-0x0064
 - sterownik klawiatury – 8042
 - 0x0070-0x0071
 - sterownik RTC i układu zapamiętywania konfiguracji (CMOS)
 - 0x0080-0x008F
 - rejestry stron – wsparcie dla 16bitowego sterownika DMA, zbudowany w oparciu o układ scalony 74612
 - zapewnia możliwość adresowania 20/24bitowego podczas transferu
 - obecnie mechanizm wyparty przez PCI-DMA

SYKO Obsługa urządzeń wejścia/wyjścia

- Urządzenia wejścia/wyjścia
 - Mapa przestrzeni I/O – wyciąg, cd.
 - 0x01F0-0x01F7/0x0170-0x0177
 - podstawowy/drugi sterownik dysków zgodnych z standardem ATA
 - 0x0278-0x027A/0x0378-0x037A
 - porty drukarkowe LPT2/LPT1
 - 0x02F8-0x02FF/0x03F8-0x03FF
 - porty komunikacji szeregowej COM2/COM1, oparte na układzie 8250, obecnie wyparty przez jego unowocześnioną wersję 16450/16550(FIFO)
 - 0x0300-0x330
 - przestrzeń dla testów
 - 0x03B0-0x03DF
 - obsługa konfiguracji kart graficznych

SYKO Obsługa urządzeń wejścia/wyjścia

- Urządzenia wejścia/wyjścia 80C51
 - Procesory zgodne z 80C51 mają dla I/O specjalną przestrzeń SFR o wielkości 128B

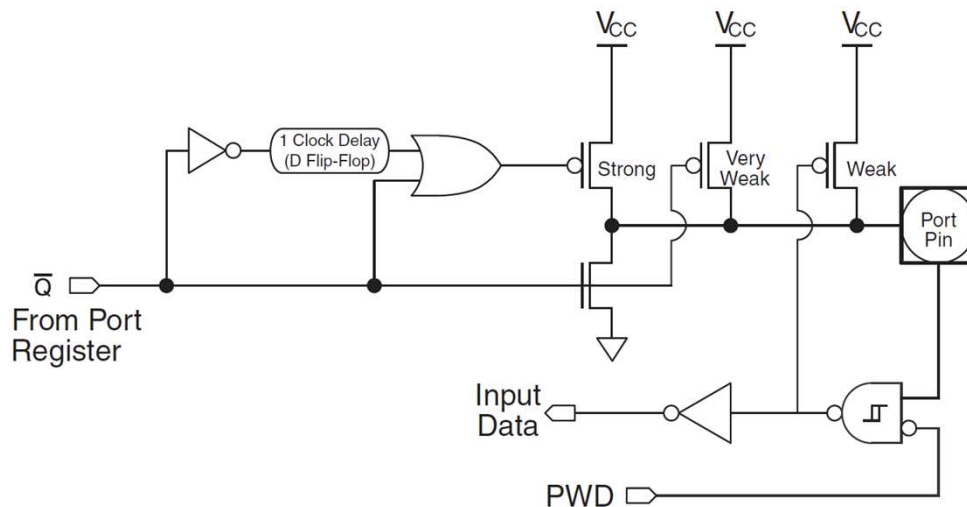
- Część portów dostępna bitowo (o adresach podzielnych przez 8)

	8	9	A	B	C	D	E	F	
0F8H		CH 0000 0000	CCAP0H 0000 0000	CCAP1H 0000 0000	CCAP2H 0000 0000	CCAP3H 0000 0000	CCAP4H 0000 0000		0FFH
0F0H	B 0000 0000		RL0 0000 0000	RL1 0000 0000	RH0 0000 0000	RH1 0000 0000	PAGE 0000 0000	BX 0000 0000	0F7H
0E8H		CL 0000 0000	CCAP0L 0000 0000	CCAP1L 0000 0000	CCAP2L 0000 0000	CCAP3L 0000 0000	CCAP4L 0000 0000	SPX xxxx x000	0EFH
0E0H	ACC 0000 0000	AX 0000 0000	DSPR 0000 0000	FIRD 0000 0000	MACL 0000 0000	MACH 0000 0000	P0M0 (2)	P0M1 0000 0000	0E7H
0D8H	CCON 00x0 0000	CMOD 00xx x000	CCAPM0 x000 0000	CCAPM1 x000 0000	CCAPM2 x000 0000	CCAPM3 x000 0000	CCAPM4 x000 0000		0DFH
0D0H	PSW 0000 0000	FCON xxxx 0000	EECON 0000 0000		DPLB 0000 0000	DPHB 0000 0000	P1M0 (2)	P1M1 0000 0000	0D7H
0C8H	T2CON 0000 0000	T2MOD 0000 0000	RCAP2L 0000 0000	RCAP2H 0000 0000	TL2 0000 0000	TH2 0000 0000	P2M0 (2)	P2M1 0000 0000	0CFH
0C0H	P4 1111 1111			SPCON 0001 0100	SPSTA 0000 0000	SPDAT xxxx xxxx	P3M0 (2)	P3M1 0000 0000	0C7H
0B8H	IPL0 xx00 0000	SADEN 0000 0000				AREF 0000 0000	P4M0 (2)	P4M1 0000 0000	0BFH
0B0H	P3 1111 1111	IEN1 xxxx 0000	IPL1 xxxx 0000	IPH1 xxxx 0000				IPH0 xx00 0000	0B7H
0A8H	IEN0 0x00 0000	SADDR 0000 0000		ACSRB 0000 0000	DADL 0000 0000	DADH 0000 0000	CLKREG 0101 xxxx	CKCON1 xxxx xxx0	0AFH
0A0H	P2 1111 1111	DPCF 0000 xxxx	AUXR1 0000 00x0	ACSRA 0000 0000	DADC 0000 0000	DADI 0000 0000	WDTRST (write-only)	WDTPRG 0000 00x0	0A7H
98H	SCON 0000 0000	SBUF xxxx xxxx	BRL 0000 0000	BDRCON xxx0 0000	KBLS 0000 0000	KBE 0000 0000	KBF 0000 0000	KBMOD 0000 0000	9FH
90H	P1 1111 1111	TCONB 0010 0100	BMSEL xxxx xxx0	SSCON 0000 0000	SSCS 1111 1000	SSDAT 1111 1111	SSADR 1111 1110	CKRL 1111 1111	97H
88H	TCON 0000 0000	TMOD 0000 0000	TL0 0000 0000	TL1 0000 0000	TH0 0000 0000	TH1 0000 0000	AUXR 0000 0000	CKCON0 0000 0000	8FH
80H	P0 1111 1111	SP 0000 0111	DPL 0000 0000	DPH 0000 0000		CKSEL xxxx xxx0	OSCCON xxxx x001	PCON 000x 0000	87H
	0	1	2	3	4	5	6	7	

Źródło: 8-bit Microcontroller with 4K Bytes In-System Programmable Flash AT89LP51RD2 AT89LP51ED2 AT89LP51ID2

SYKO Obsługa urządzeń wejścia/wyjścia

- Urządzenia wejścia/wyjścia 80C51
 - Konfigurowalne porty GPIO – konstrukcja uproszczona

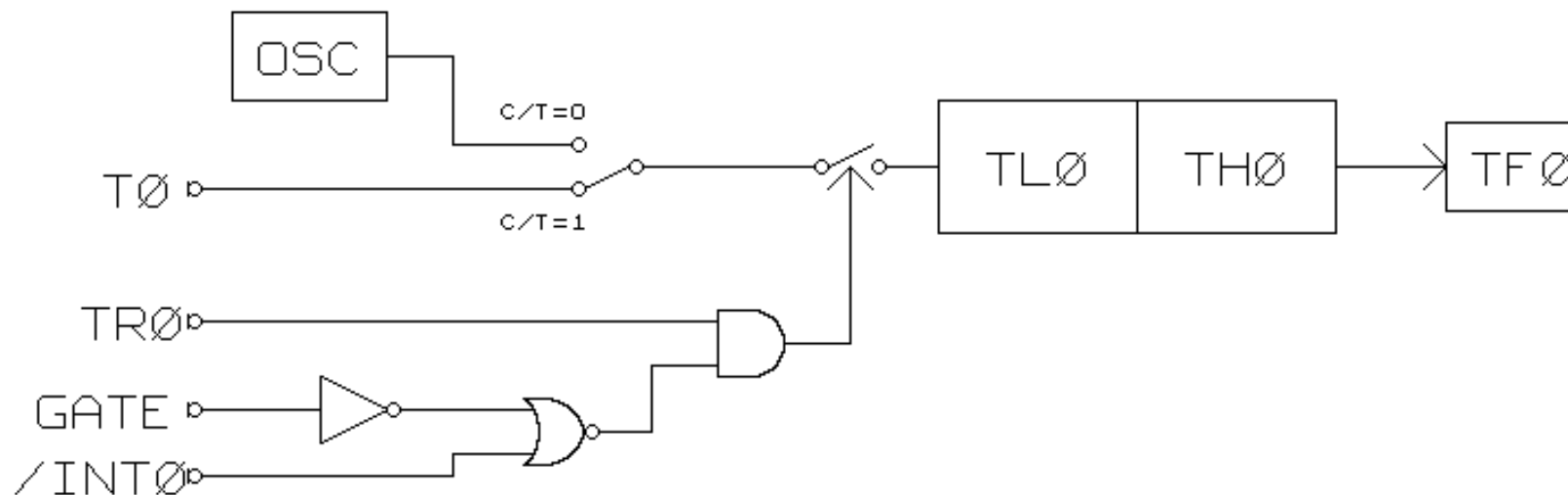


PxM0.y	PxM1.y	Port Mode
0	0	Quasi-bidirectional
0	1	Push-pull Output
1	0	Input Only (High Impedance)
1	1	Open-Drain Output

Port	Port Data	Port Configuration
0	P0 (80H)	P0M0 (D4H), P0M1 (D5H)
1	P1 (90H)	P1M0 (E6H), P1M1 (E7H)
2	P2 (A0H)	P2M0 (D6H), P2M1 (D7H)
3	P3 (B0H)	P3M0 (DEH), P3M1 (CFH)
4	P4 (C0H)	P4M0 (BEH), P4M1 (BFH)

SYKO Obsługa urządzeń wejścia/wyjścia

- Urządzenia wejścia/wyjścia 80C51
 - Licznik T0 - przykładowy element peryferyjny
 - Tryb 1, zliczane impulsy z wejścia T0 (TR0, GATE, /INT0 zatrzymały zliczanie)
 - Układ posiada dwa takie liczniki, nowsza wersja 80c52 - trzy



SYKO Obsługa urządzeń wejścia/wyjścia

- Urządzenia wejścia/wyjścia
 - Procesor AVR (i wiele innych) widzi urządzenia we/wy je jako określone komórki pamięci
 - Teoretycznie wielkość tej przestrzeni jest wielkością pamięci danych
 - Z reguły mały ułamek – w AVR 64(peryferia podstawowe)+160(peryferia rozszerzone) lokacji
 - Wycinek mapy pamięci

Data Memory	
32 Registers	\$0000 - \$001F
64 I/O Registers	\$0020 - \$005F
160 Ext I/O Reg.	\$0060 - \$00FF
Internal SRAM (4096 x 8)	\$0100
	\$10FF
External SRAM (0 - 64K x 8)	\$1100

0x18 (0x38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
0x17 (0x37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
0x16 (0x36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
0x15 (0x35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0
0x14 (0x34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
0x13 (0x33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
0x12 (0x32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
0x11 (0x31)	DDRD	DDd7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
0x10 (0x30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
0x0F (0x2F)	SPDR	SPI Data Register							
0x0E (0x2E)	SPSR	SPIF	WCOL	-	-	-	-	-	SPI2X
0x0D (0x2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0

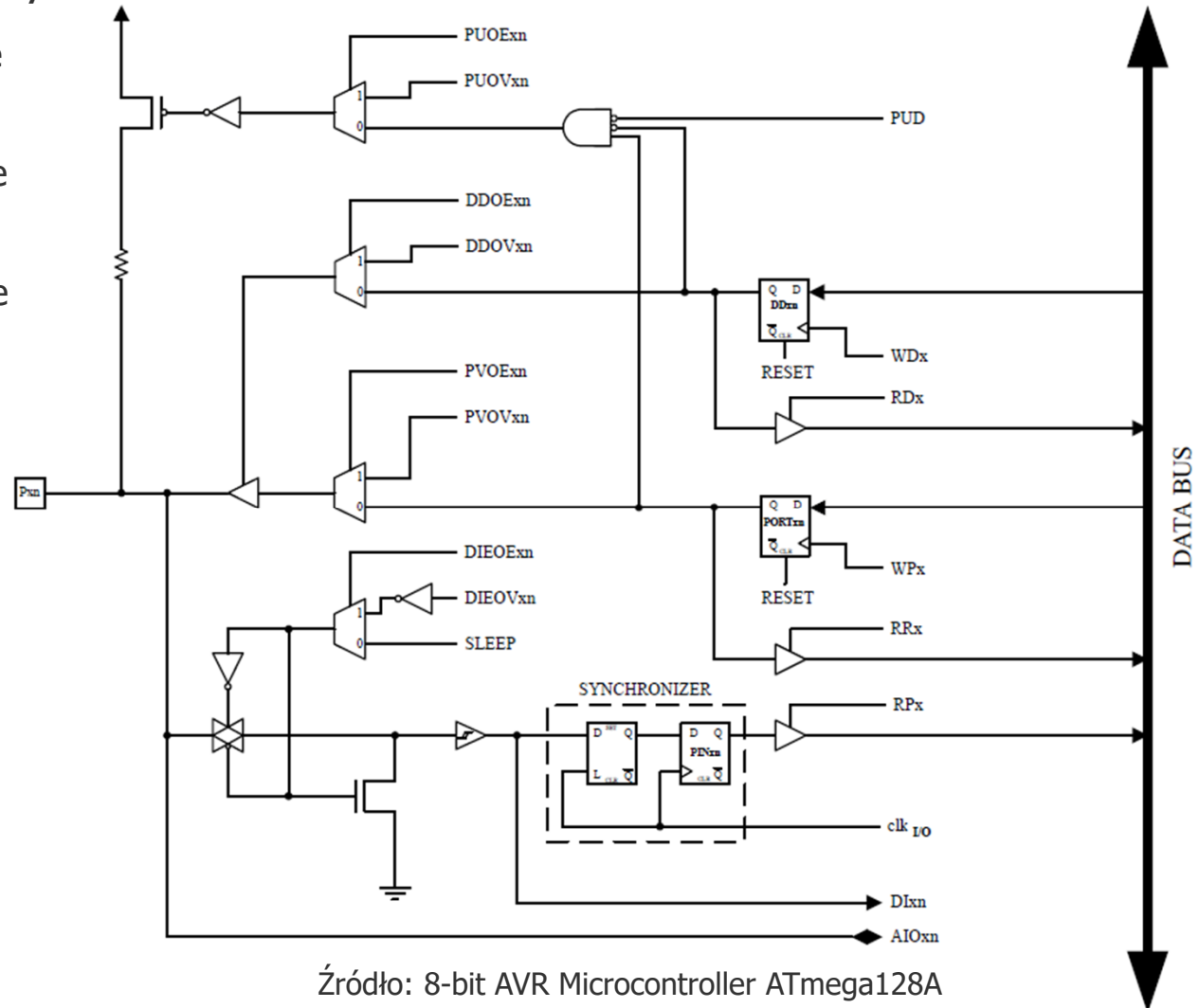
Źródło: 8-bit AVR Microcontroller ATmega128A DATASHEET COMPLETE

SYKO Obsługa urządzeń wejścia/wyjścia

- Urządzenia wejścia/wyjścia AVR

- Konfigurowalne porty GPIO

- PUOE Pull-up Override Enable
- PUOV Pull-up Override Value
- DDOE Data Direction Override Enable
- DDOV Data Direction Override Value
- PVOE Port Value Override Enable
- PVOV Port Value Override Value
- DIEOE Digital Input Enable Override Enable
- DIEOV Digital Input Enable Override Value
- DI Digital Input
- AIO Analog Input/Output



Źródło: 8-bit AVR Microcontroller ATmega128A

SYKO Obsługa urządzeń wejścia/wyjścia

- Urządzenia wejścia/wyjścia AVR

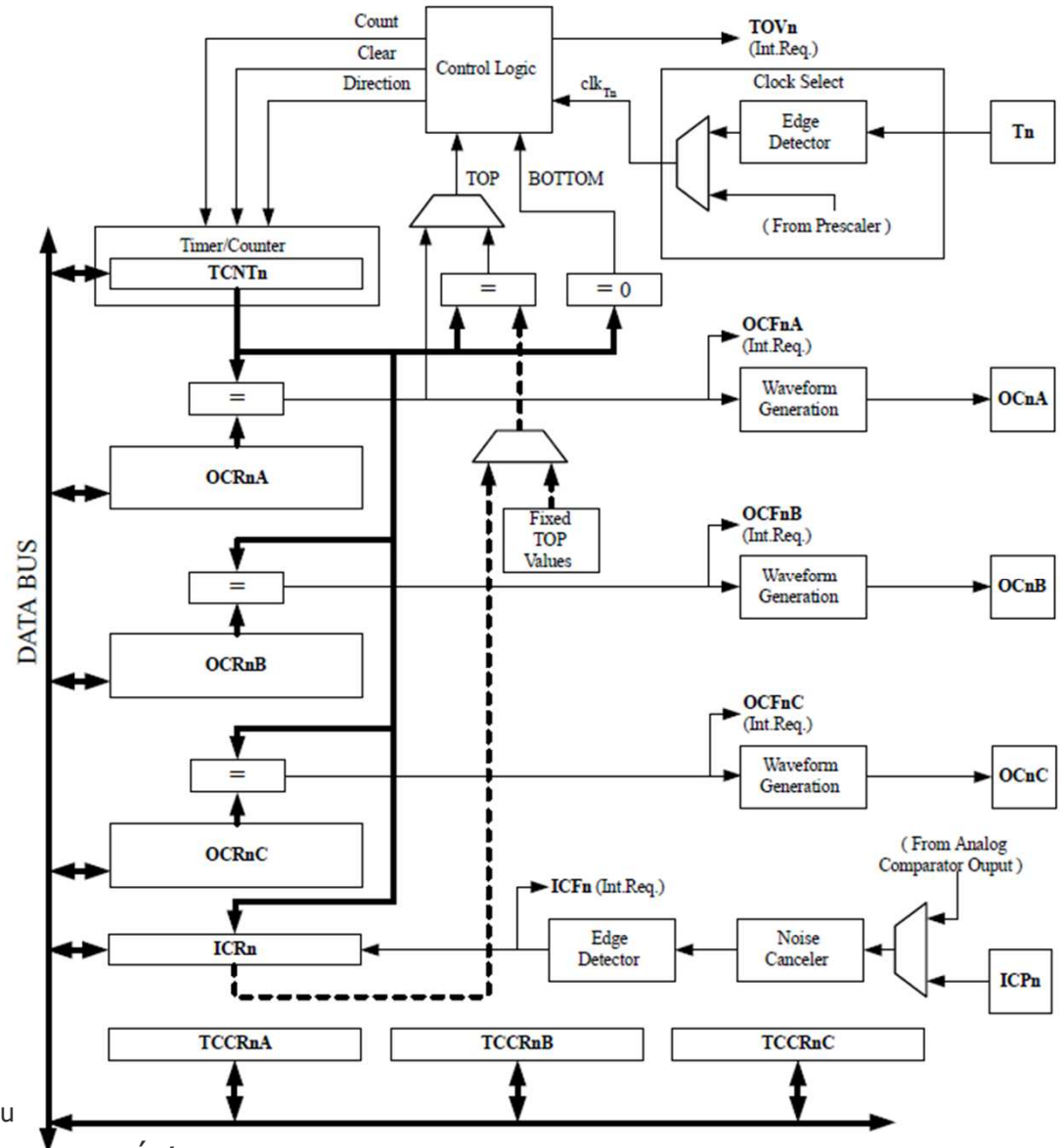
- Liczniki – przykładowy element peryferyjny

peryferyjny

- ATmega128 ma trzy takie liczniki - jeden „liczy” 16bitowo, dwa – 8 bitowo

- Zdarzenia generowane przez liczniki

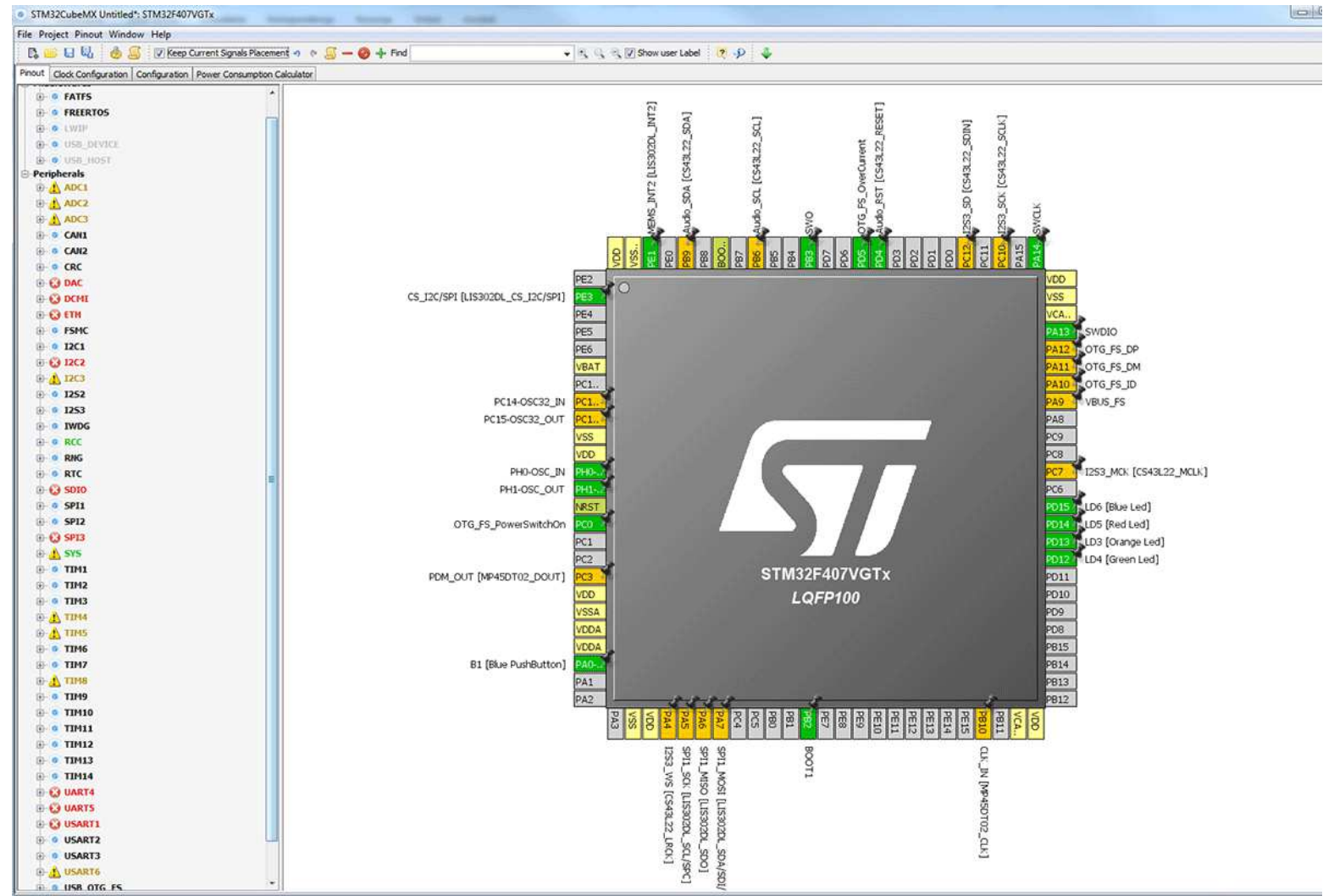
- Timer Overflow (TOVx)
 - Zwiększa stan do war. MAX resetuje się – generuje zdarzenie i liczy dalej
 - Compare Match
 - Stan licznika porównuje się z wartością rej. OCRx – gdy równe generuje zdarzenie
 - Input Capture
 - Stan licznika może być „zatrzaskiwany” w momencie pojawienia się zewnętrznego sygnału



Źródło: Atmel ATmega128A

SYKO Systemy dołączania urządzeń wejścia/wyjścia

- A co gdy mnogość peryferii staje się przeszkodą dla programisty?
 - STM32Cube – aplikacja do „wyklikania” kodu obsługi peryferii



Systemy dołączania urządzeń wejścia/wyjścia

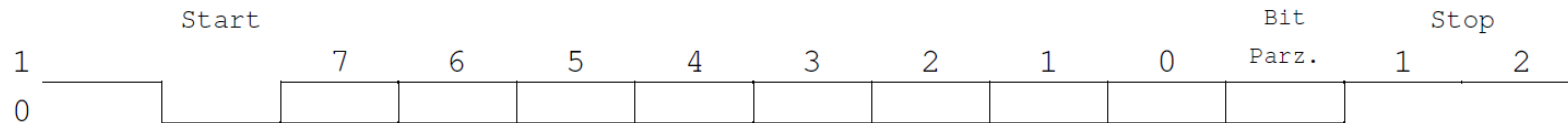
SYKO Systemy dołączania urządzeń wejścia/wyjścia

- Metody łączenia peryferii
 - GPIO
 - Poprzez interfejsy
- Interfejsy
 - UART/RS232/RS485
 - I2C
 - SPI
 - USB
 - PCI i PCI express

SYKO Systemy dołączania urządzeń wejścia/wyjścia

- UART/RS232

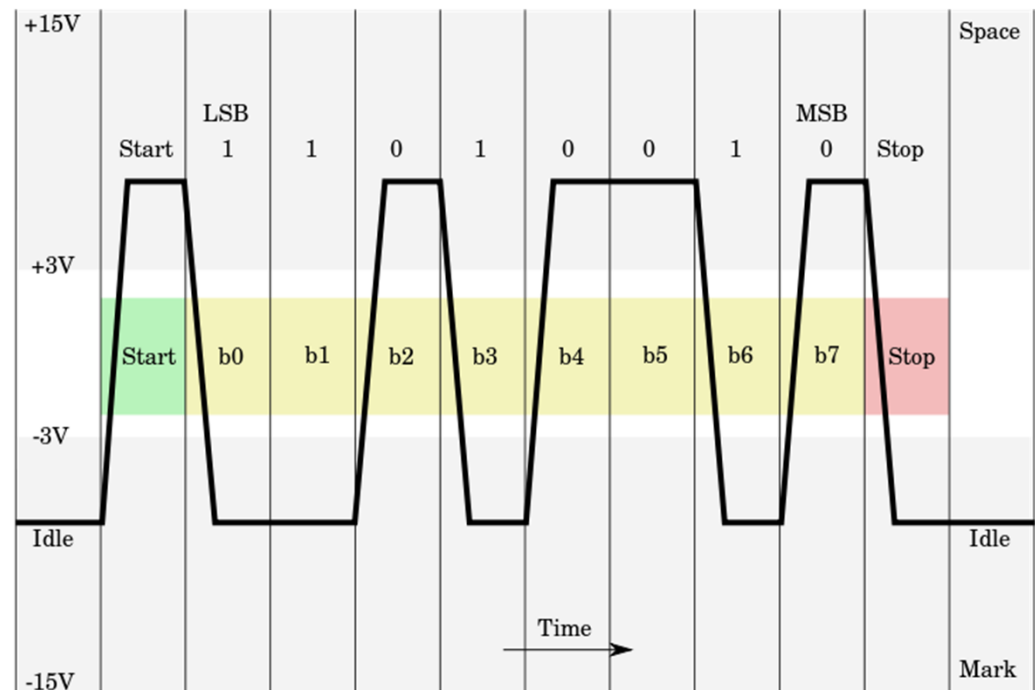
- UART: Universal Asynchronous Receiver and Transmitter
- Tryby: asynchroniczny, synchroniczny (mniej używany)
- Ramka



- Prędkości [bit/s]: 9600, 19200, 38400, 57600, 115200, rzadziej używane: 56000, 76800, 230400, 256000, 460800, 921600, historyczne: 50...4800

SYKO Systemy dołączania urządzeń wejścia/wyjścia

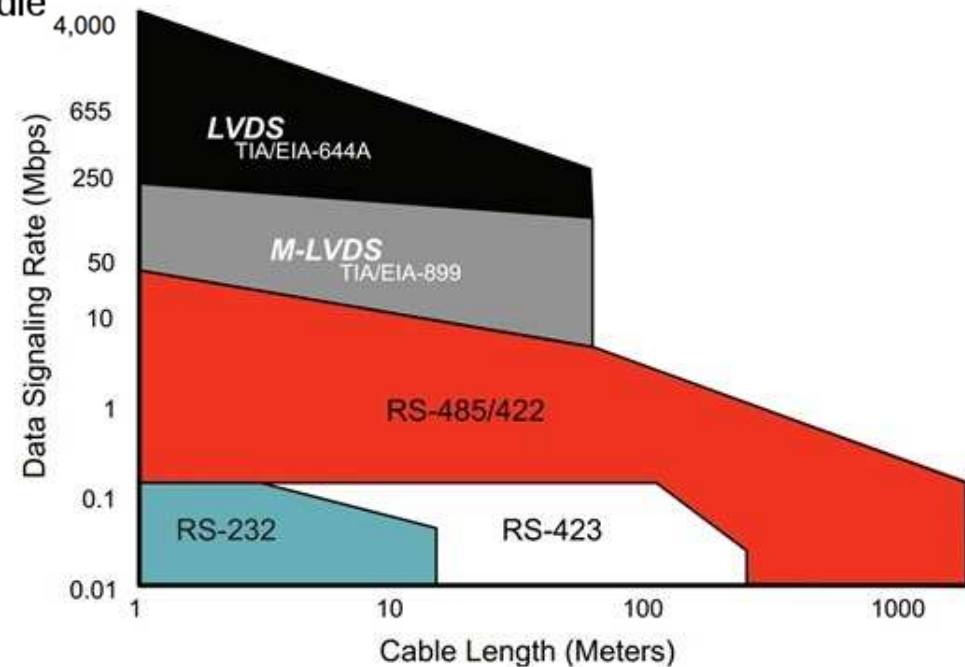
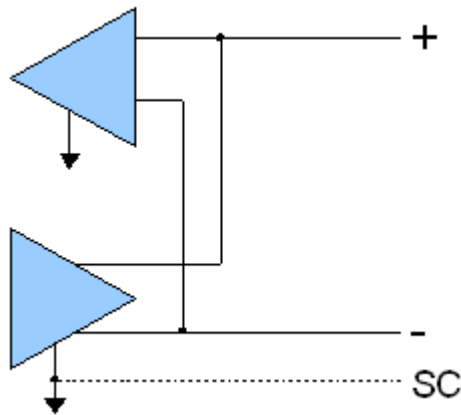
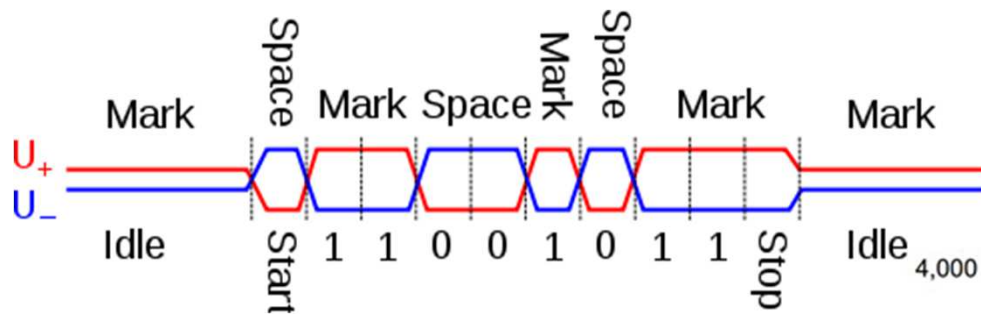
- UART/RS232, cd.
- RS232 to opis sposobu łączenia urządzeń i charakterystyka elektryczna tych połączeń
 - DTE (ang. Data Terminal Equipment) czyli urządzeń końcowych danych (np. komputer) oraz urządzeń DCE (ang. Data Communication Equipment)
 - „0” to: +3...+15V
 - „1” to: -15...-3V
 - Mimo wprowadzenia USB nadal UART jest dostępny w wielu mikrokontrolerach
 - Interfejsy USB i BlueTooth wspierają emulacje połączeń UART



Źródło: wikimedia.org

SYKO Systemy dołączania urządzeń wejścia/wyjścia

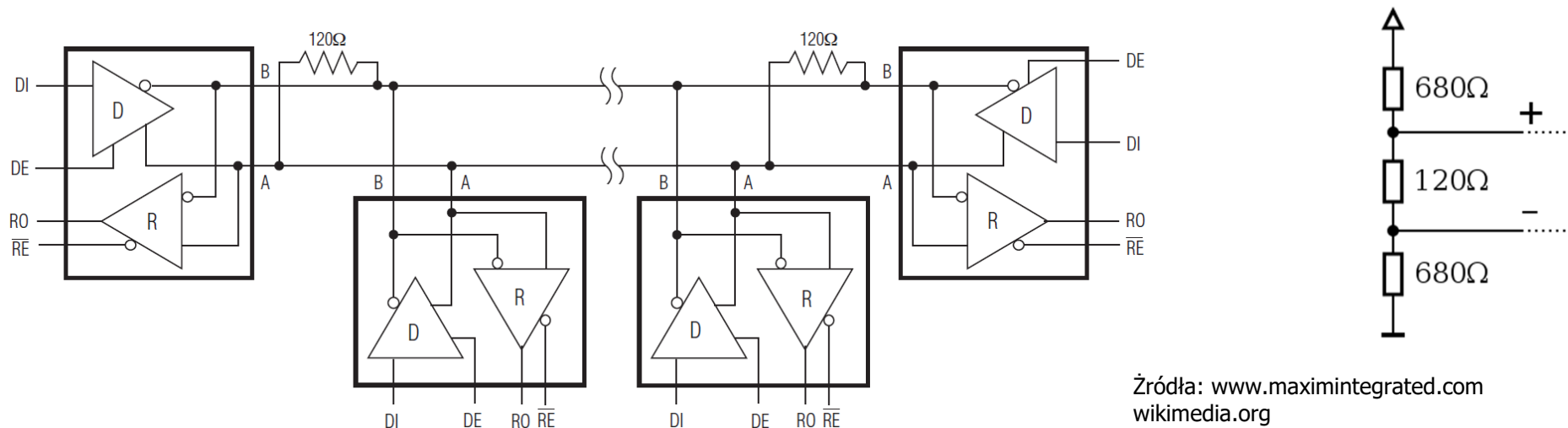
- RS485
 - Podstawy podobne do RS232 - główna różnica to warstwa fizyczna, dane przesyłane są w sposób różnicowy



SYKO Systemy dołączania urządzeń wejścia/wyjścia

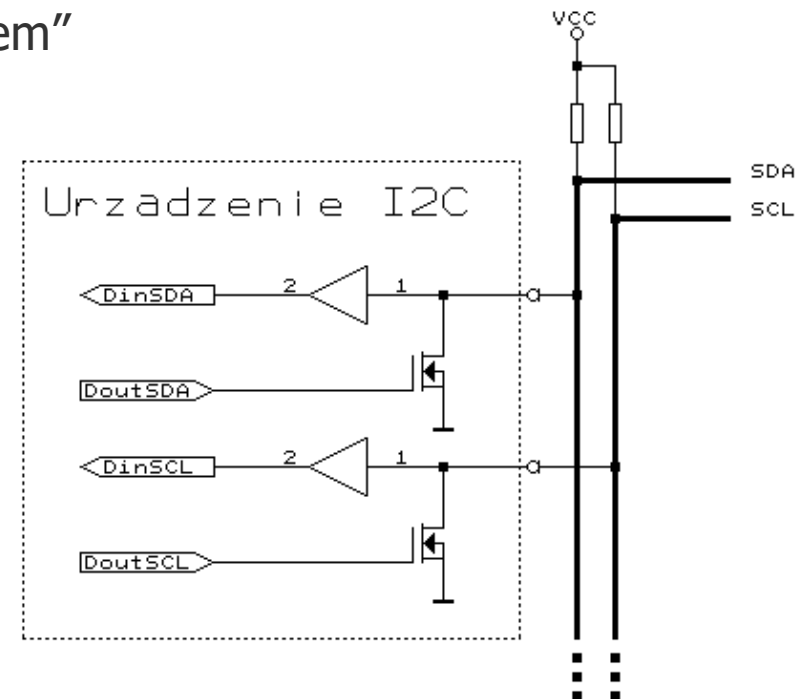
■ RS485 - magistrala

- <32 urządzeń, <1200m długości, <10Mbit/s
- Konieczność stosowania „terminatorów”
- Standard nie opisuje co robić z kolizjami - konieczne wprowadzenie odpowiednich mechanizmów CSMA/CA (Carrier-sense multiple access with collision avoidance)
- Wiele mikrokontrolerów pozwala stosować komunikację UART z 9 bitami
 - Działanie - gdy 9bit w ramce ustawiony: to wszystkie MCU podpięte do magistrali zwracają na pozostałe bity ramki uwagę, gdy wyzerowany to tylko zainteresowani odbiorcy przetwarzają dane w ramkach



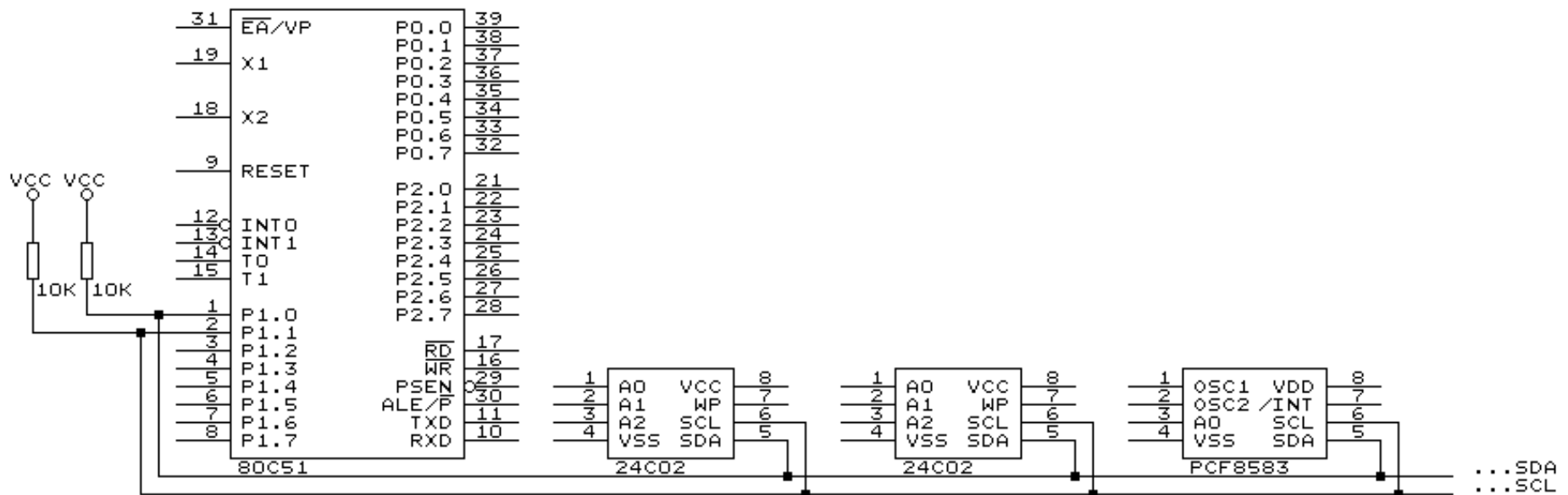
SYKO Systemy dołączania urządzeń wejścia/wyjścia

- I2C (Inter-Integrated Circuit)
 - Produkt będący znakiem towarowym firmy Phillips/NXP
 - Interfejs znany również jako: TWI (Two Wire Interface) lub TWISI (Two-Wire Serial Interface)
 - SMB/SMBus (System Management Bus) rozszerza I2C
 - I2C tworzy magistrale z zegarami zapewniającymi szybkość od 100kbit/s do 3,4Mbit/s
 - Podstawą są bufory z „otwartym kolektorem”



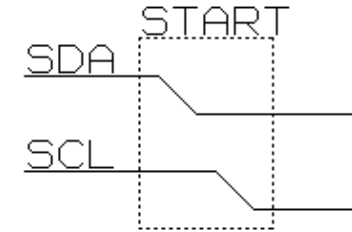
SYKO Systemy dołączania urządzeń wejścia/wyjścia

- I2C (Inter-Integrated Circuit), cd.
 - Urządzenia tworzą magistrale
 - Zarządcą może być każde urządzenie - w danej transakcji tylko jedno
 - Urządzenia adresuje się 7 lub 10bitami
 - adresy nie są unikatowe, nadawane w procesie produkcji i związane są z klasą danego urządzenia

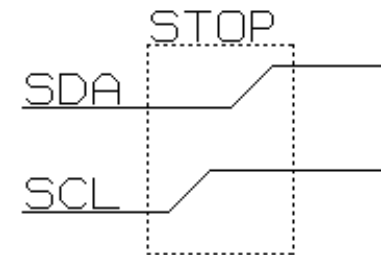


SYKO Systemy dołączania urządzeń wejścia/wyjścia

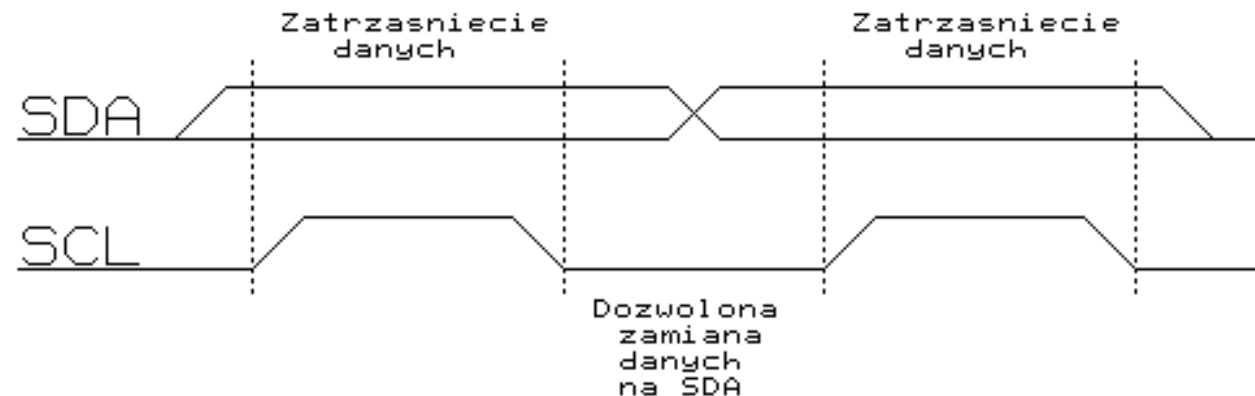
- I2C (Inter-Integrated Circuit), cd.
 - SDA - Serial Data, SCL - Serial Clock
 - Początek komunikacji - sekwencja START



- Koniec transmisji - sekwencja STOP

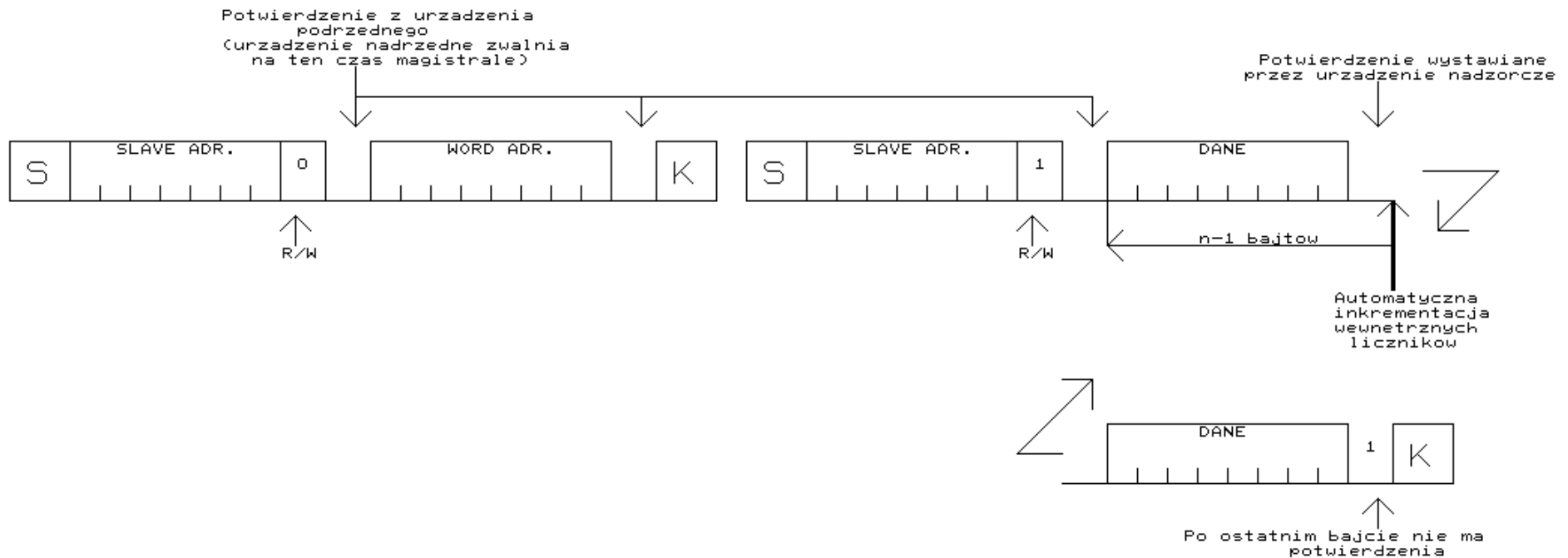


- Zmiana danych na SDA (przed wysłaniem kolejnego bitu informacji) możliwe tylko gdy SCL=0



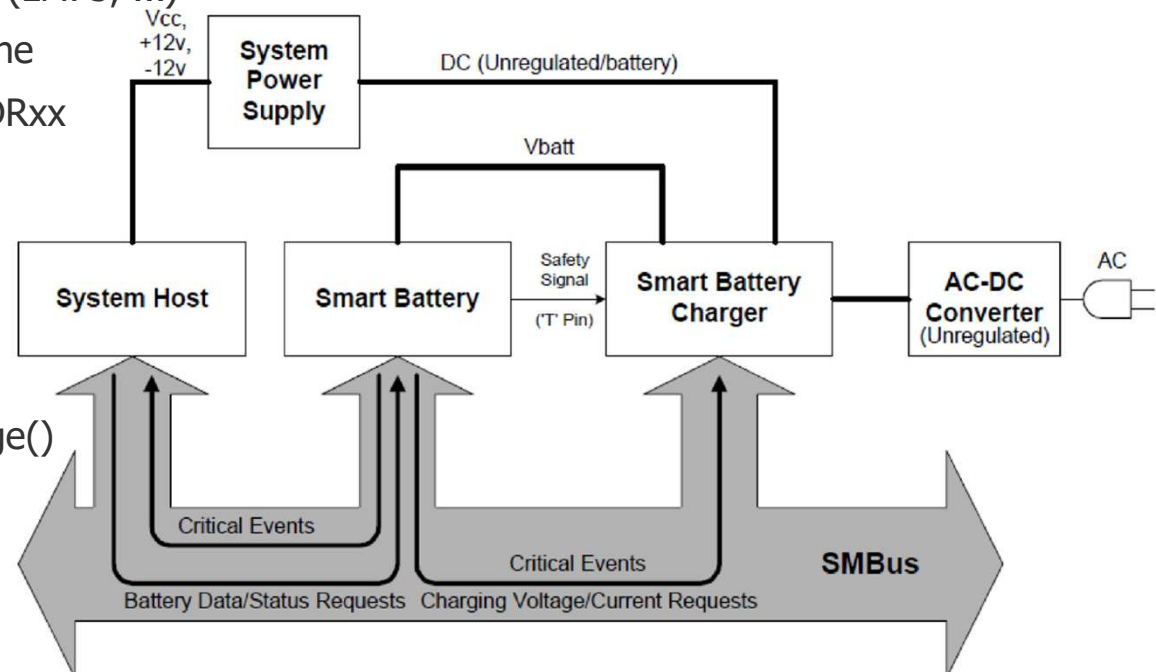
SYKO Systemy dołączania urządzeń wejścia/wyjścia

- I2C (Inter-Integrated Circuit), cd.
 - Typowa komunikacja zakłada potwierdzanie przez SLAVE każdego odebranego bajta informacji



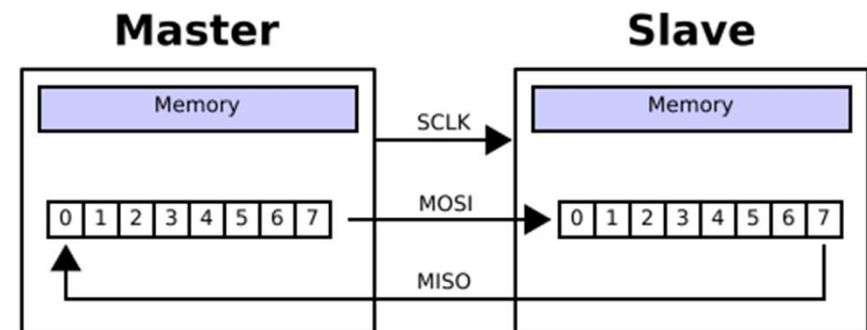
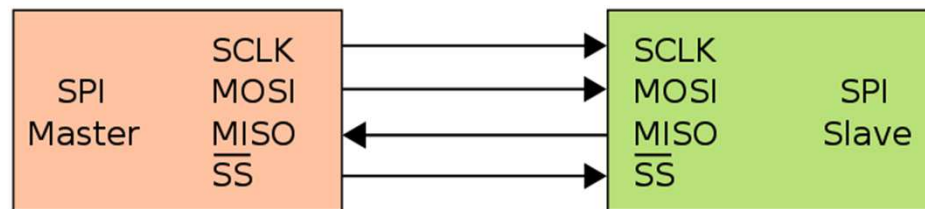
SYKO Systemy dołączania urządzeń wejścia/wyjścia

- I2C - wykorzystanie w systemach komputerowych
 - W systemach komputerowych I2C istnieje jako system SMBus
 - Pobranie informacji dostarczanych przez komponenty (m.in.)
 - Numer modelu i części (dane od ich producenta)
 - Sterowanie stanem (np.: przejście w uśpienie)
 - Raport o stanie/błędach działania komponentu
 - Komponenty to (m.in.)
 - Czujniki temperatury (LM75, ...)
 - Pamięci konfiguracyjne modułów pamięci DDRxx
 - Karty PCI/PCIe
 - Inteligentne baterie
 - Polecenia, np.:
 - AverageCurrent()
 - RelativeStateOfCharge()
 - Voltage()
 - Temperature()



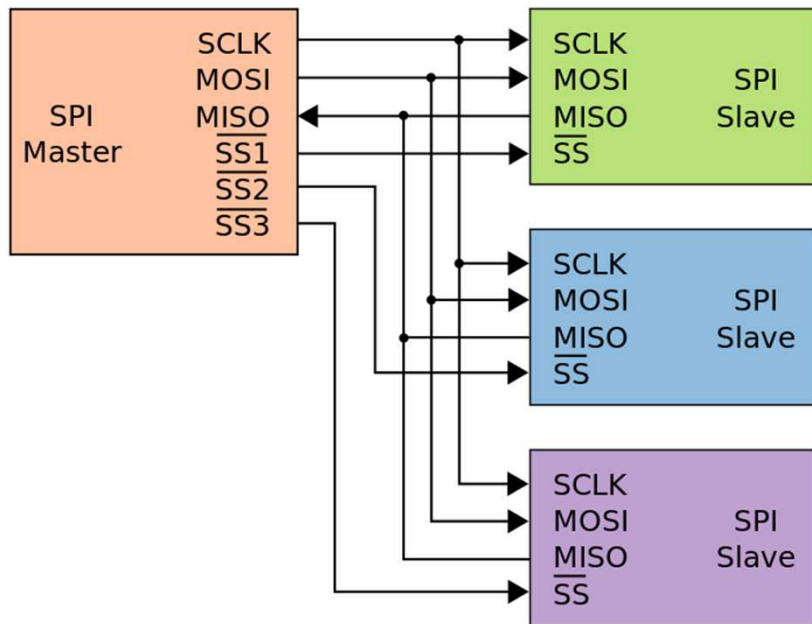
SYKO Systemy dołączania urządzeń wejścia/wyjścia

- SPI (Serial Peripheral Interface)
 - Magistrala używana do komunikacji wewnątrz urządzeń
 - Prędkość - zegar transmisji to maks. dziesiątki MHz
 - Nadzorca wybiera urządzenie podrzędne (z dołączonego zestawu) przez specjalny sygnał fizycznym: slave select - SS
 - Sygnały
 - SCLK: Serial Clock - steruje nadzorca
 - MOSI: Master Output Slave Input - dane wysyłane przez nadzorcę
 - MISO: Master Input Slave Output - dane otrzymywane od urządzeń podrzędnych
 - Typowe połączenie zarządcy z jednym urządzeniem podrzędnym

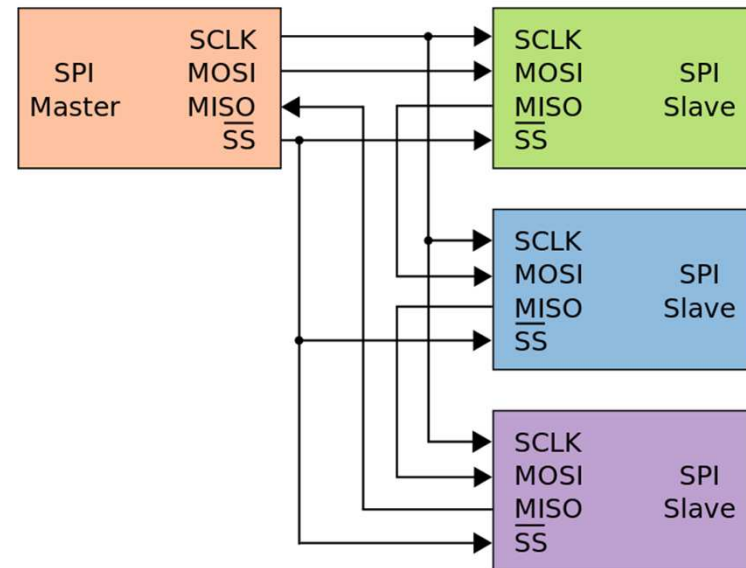


SYKO Systemy dołączania urządzeń wejścia/wyjścia

- SPI, cd.
 - Równoczesne podłączenie wielu urządzeń podrzędnych przez pojedynczy interfejs SPI
 - niektóre MCU posiadaj parę kontrolerów SPI



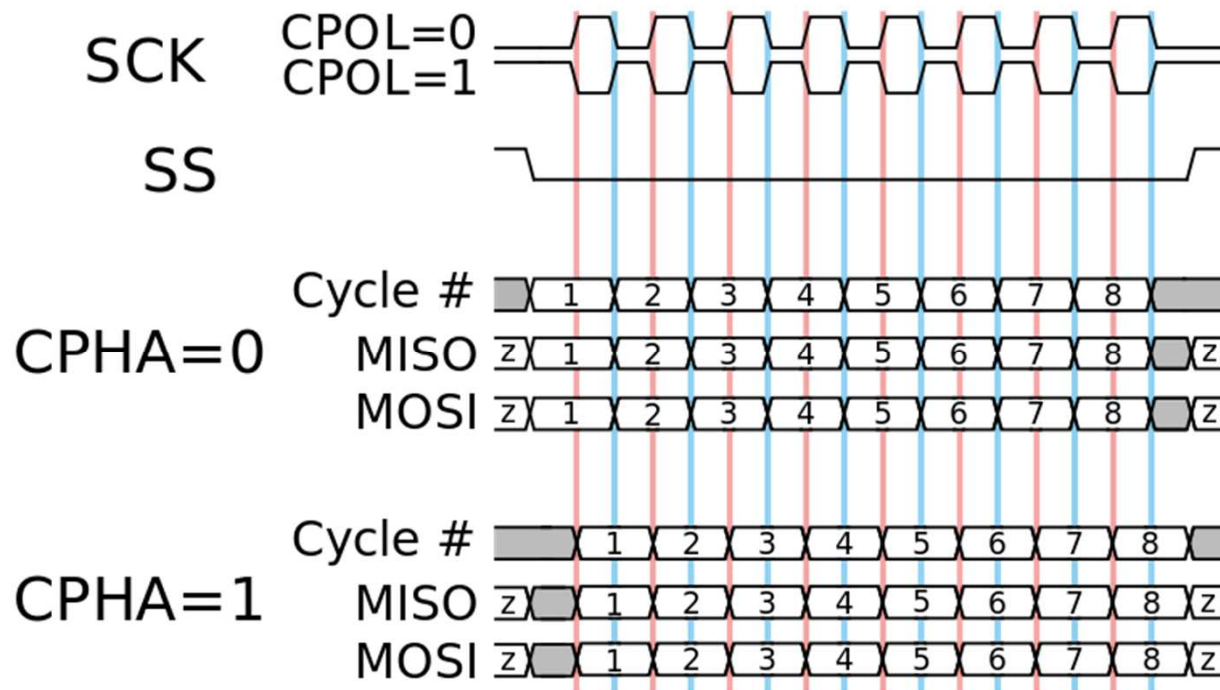
Urządzenia wybierane przez „SS”



Urządzenia pracujące w łańcuchu

SYKO Systemy dołączania urządzeń wejścia/wyjścia

- SPI, cd.
 - Sygnały zegarowe
 - polaryzacja CPOL={0,1}
 - faza zegara CPHA={0,1}



SYKO Systemy dołączania urządzeń wejścia/wyjścia

- SPI wykorzystanie w systemach komputerowych
 - Moduły pamięci masowych
 - Karty SD/MicroSD
 - Pamięci trwałe zawierające tzw. BIOS/UEFI

SYKO Systemy dołączania urządzeń wejścia/wyjścia

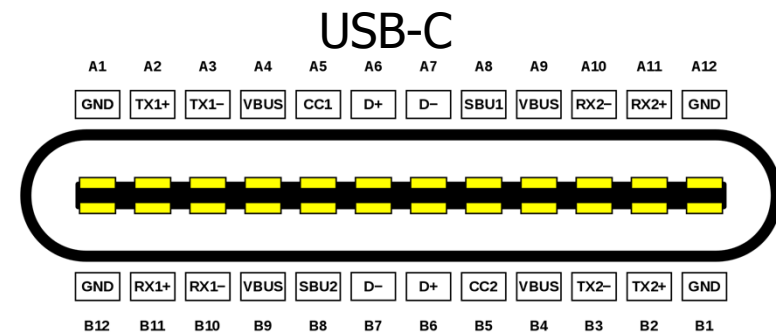
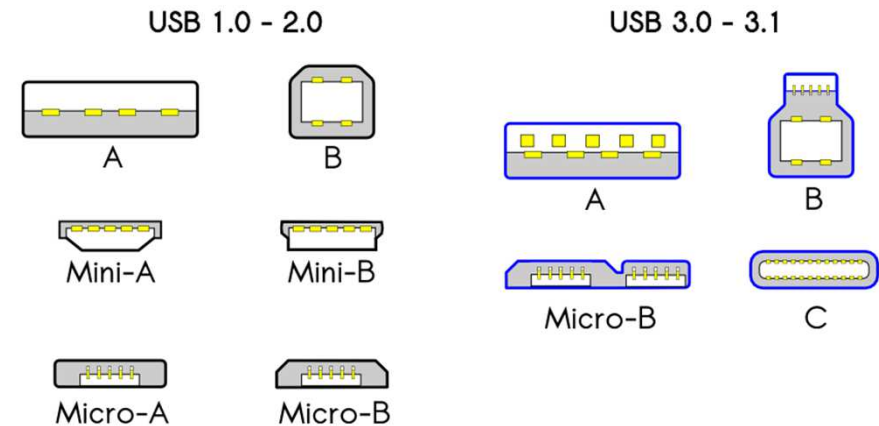
- USB (Universal Serial Bus)
 - Magistrala gdzie komputer jest częścią główną (Host centric)
 - Maksymalnie można podłączyć do 127 urządzeń
 - Urządzenia mogą działać jako
 - część główna (Host USB) - w danej sieci może być tylko jedno takie urządzenie w całym systemie (z reguły komputer)
 - część podrzędna (Device USB)
 - Urządzenia USB-OTG to rozszerzenie USB2.0 - pozwala bardziej inteligentnym urządzeniom zmieniać w „locie” swoją rolę
 - gdy w systemie jest obecna część główna - urządzenie OTG może być tylko częścią podrzędną
 - Urządzenia podrzędne realizują zadania odpowiedniej klasy:
 - CDC-konwertery transmisji szeregowej, HID-interfejsy komunikacji z użytkownikiem, MassStorage-pamięci masowe, ...
 - Każde urządzenie posiada swój unikatowy identyfikator producenta i dostawcy PID i VID - nabywane od konsorcjum USB

SYKO Systemy dołączania urządzeń wejścia/wyjścia

■ USB, cd.

■ Transfery vs. Typ

- USB 1.0 (Low-speed (LS))
 - 1,5 Mbit/s
- USB 1.1 (Full-speed (FS))
 - 1,5 Mbit/s
- USB 2.0 (High-speed (HS))
 - 480Mb/s
- USB 3.0 (SuperSpeed (SS))
 - 5Gbit/s
- USB 3.1 (SuperSpeed+ (SS+))
 - 10 Gbit/s
- USB 4 (wprowadzony w 2019)
 - 40 Gbit/s



■ USB-C

- Nowe złącze pomagające przekazywać większe energie zasilanym urządzeniom
- Może przenosić od standardu USB 2.0 do USB 4

SYKO Systemy dołączania urządzeń wejścia/wyjścia

- USB, cd.
 - Zasilanie przez interfejs
 - Maksymalny prąd bez negocjacji $\leq 100\text{mA}$
 - Po negocjacji (m.in. wymiana deskryptora urządzenia)
 - USB ≥ 2.0
 - $\leq 500\text{mA}$ przy 5V, do 2,5W
 - USB $\geq 3.x$
 - $\leq 900\text{mA}$ przy 5V, do 4,5W
 - USB BC (battery charging)
 - 500–1500mA przy 5V, do 7,5W (wersja BC 1.1) lub do 25W (wersja BC 1.2)
 - USB PD (power delivery)
 - 2500–5000mA przy negocjowanym napięciu w zakresie 5..20V, do 100W
 - Odłączenie obciążenia po przekroczeniu limitu może być wyłączone lub nie zaimplementowane!

SYKO Systemy dołączania urządzeń wejścia/wyjścia

- USB, cd.
 - Tworzenie urządzeń podrzędnych
 - Z wykorzystaniem konwerterów USB-UART (możliwa realizacja wyłącznie klasy CDC)
 - FTDI
 - PL2303
 - CH304G
 - ACM
 - ...
 - Używając MCU ze wsparciem sprzętowym dla USB (np. ATmega32u2)
 - Takie podejście pozwala utworzyć urządzenie USB dowolnej klasy
 - Przydatna do tego celu jest biblioteka LUFA [www.fourwalledcubicle.com]
 - Bez wsparcia sprzętowego MCU - może być urządzeniem podrzędnym tylko za pomocą specjalnej bibliotek V-USB [www.obdev.at]
 - Jest ona utworzona dla ściśle określonego sprzętu (np.: MCU klasy AVR)
 - Zawiera fragmenty zaimplementowane z zachowaniem wysoko czasowo krytycznych operacji - sterowanie w stylu bit-bang z szybkością działań poniżej 1us

SYKO Systemy dołączania urządzeń wejścia/wyjścia

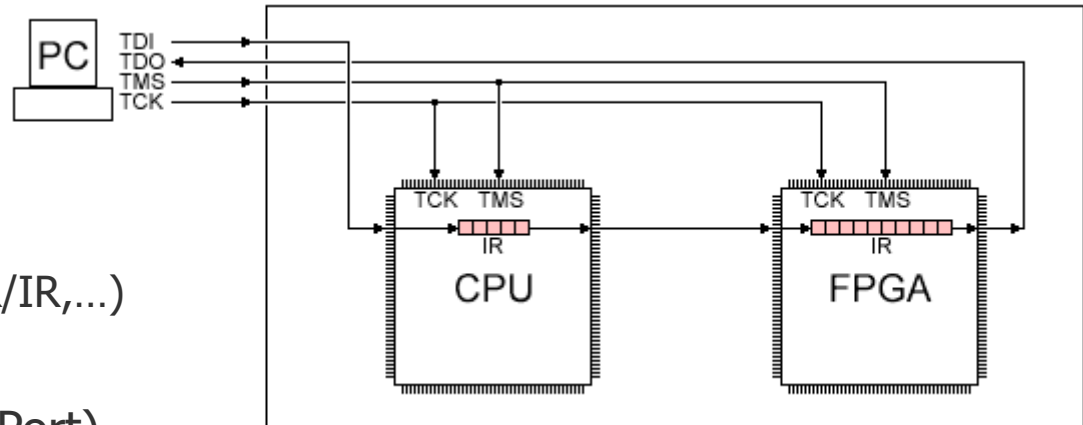
- USB, cd.
 - Budowa urządzenia nadrzędnego USB
 - Wymagane jest wysoko wydajne wsparcie sprzętowe
 - Transmisja z urządzeniami podrzędnymi realizowana jest z dużymi prędkościami
 - Istnieje wiele układów np.: MAX3421E
 - UWAGA!!! od programisty tworzącego kod wymagane jest napisanie własnych procedur obsługi dołączanych urządzeń podrzędnych USB (np.: kamery)
 - patrz: https://github.com/felis/USB_Host_Shield_2.0

SYKO Systemy dołączania urządzeń wejścia/wyjścia

■ JTAG – szeregową specjalizowaną magistrala testowania układów

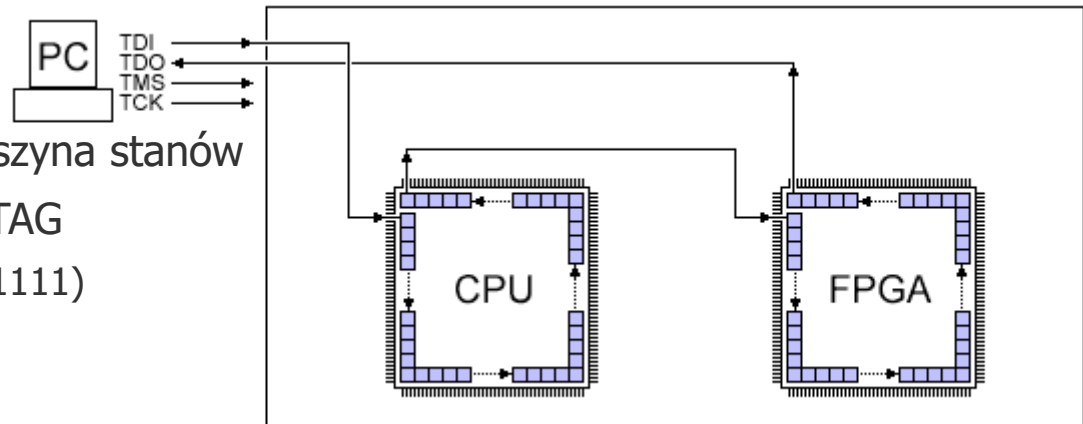
■ Sygnały

- TCK – sygnał zegary
- TDI - wejście testowe
- TDO - wyjście testowe
- TMS - tryb testowania (DR/IR,...)
- TRST# - inicjowanie testu



■ Kontroler TAP (Test Access Port)

- Wbudowany w testowane elementy
- Implementowany jako maszyna stanów
- Obowiązkowe instrukcje JTAG
 - BYPASS (np.: dla CPU 11111)
 - EXTEST
 - SAMPLE/PRELOAD
 - IDCODE



- Opcjonalna, ale zawsze występuje i zwraca 32bitowy identyfikator producenta i typu danej części utrzymywane w JEDEC Standard Manufacturer's Identification Code standard

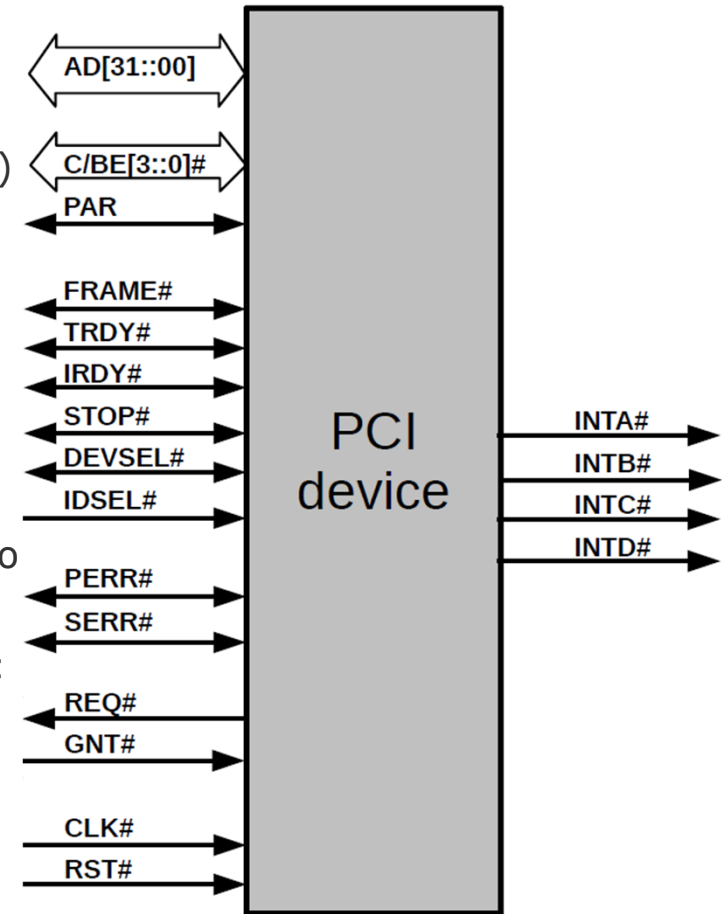
SYKO Systemy dołączania urządzeń wejścia/wyjścia

- Magistrala PCI
 - Nowoczesna magistrala pozwalająca aby wiele urządzeń mogło współpracować podczas transferu
 - W danej chwili tylko jedno inicjuje transfer (Master)
 - W tej samej chwili wiele urządzeń może odbierać dane
 - PCI definiuje trzy przestrzenie do których może pisać i z których może czytać
 - Przestrzeń pamięci
 - Przestrzeń we/wy
 - Przestrzeń konfiguracyjnej
 - Urządzenia mogą informować system jaka jest ich budowa, możliwości i wymagania
 - Udostępnia m.in. pola
 - Vendor ID (2B) i Device ID (2B) – identyfikacja urządzenia (patrz PID i VID w USB)
 - Polecenie (2B)
 - BAR0 (4B) – adres pod jakim urządzenie PCI ma „odpowiadać” w przestrzeni, dla danego urządzenia PCI tych elementów może być wiele

SYKO Systemy dołączania urządzeń wejścia/wyjścia

■ Magistrala PCI - Sygnały

- AD – adresy multipleksowane z danymi
- C/BE – polecenie/określenie wielkości transferu (1B/.../4B)
- PAR – parzystość dla: AD (w 1 takcie), C/BE (po 1 takcie)
- FRAME# - początek ramki
- IRDY# - (Initalizer Ready) inicjator gotowy do przyjęcia danych
- TRDY# - (Target Ready) urządzenie docelowe zgłasza gotowość
- DEVSEL# - zgłoszenie od docelowej jednostki poprawnego rozpoznania swojego adresu
- STOP# - docelowa jednostka chce natychmiast zakończyć transmisje
- IDSEL# - sygnał konfigurowania magistrali PCI
- REQ#/GNT# - zgłoszenie zapotrzebowanie na magistralę/arbiter zezwała na przejecie magistrali
- PERR# - wykrycie błędu przez: inicjator - w danych zapisywanych/ urządzenie docelowe – w danych odczytywanych
- SERR# - błąd parzystości adresu lub systemowy
- RST# - zerowanie wszystkich urządzeń magistrali, CLK - sygnał zegara taktującego magistralę
- INTA-INTD - 4 przerwania sprzętowe (jedna linia dla każdego urządzenia)

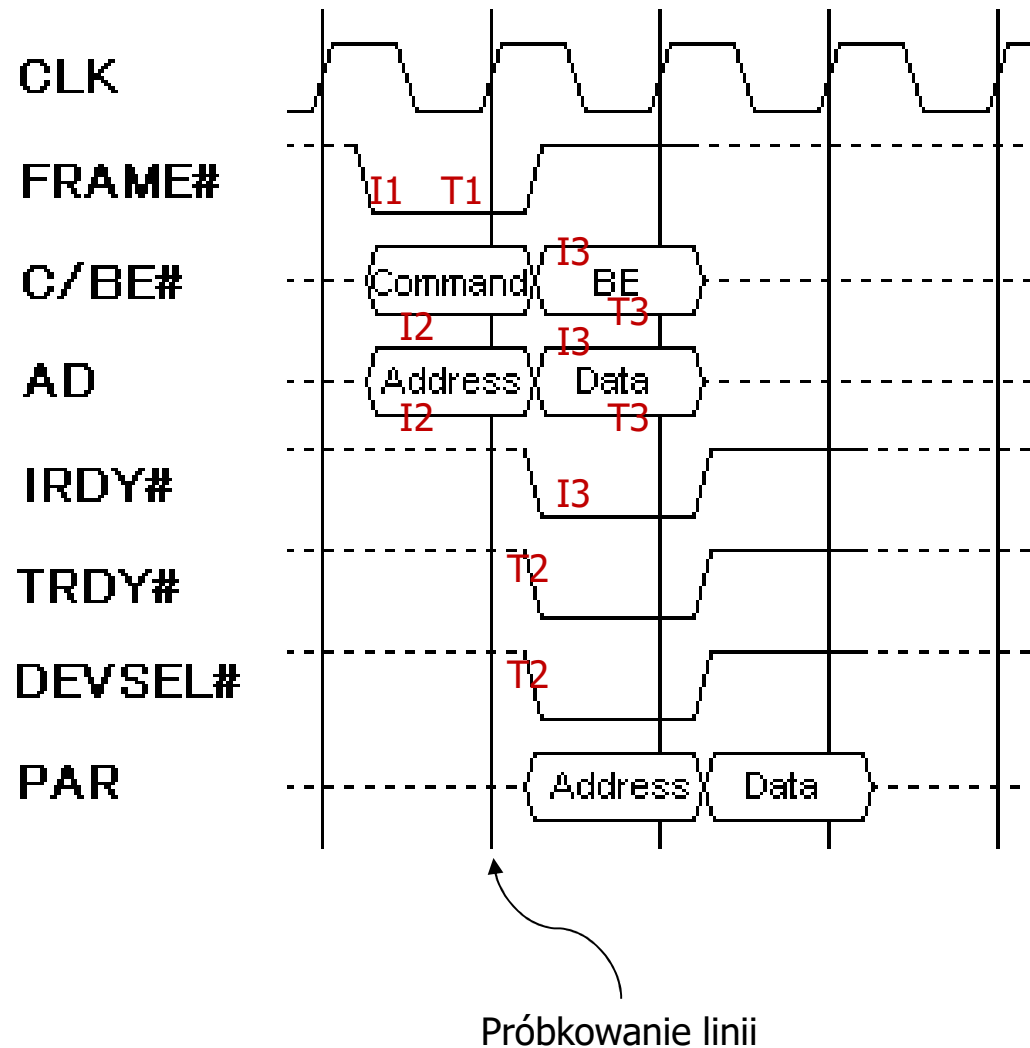


SYKO Systemy dołączania urządzeń wejścia/wyjścia

- Magistrala PCI
 - Rozkazy magistrali – sygnały **C/BE[0...3]** – faza przekazywania rozkazu
 - (0000) INTA sequence
 - (0001) Special Cycle
 - (0010) I/O Read, (0011) I/O Write
 - (0110) Memory Read, (1110) Memory Read Line, (1100) Memory Read Multiple
 - (1010) Configuration Read, (1011) Configuration Write – tylko 11 dolnych bitów adresowych
 - (1101) Dual Address Cycle (gdy potrzebny transfer z użyciem adresu o więcej niż 32 bitach)
 - (1111) Memory Write and Invalidate, (0111) Memory Write
 - Sygnał **C/BE[0...3]** – w fazie transferu
 - wielkość transferowanych danych – które bity linii AD w transferze niosą ważne dane
 - Moment zmian sygnałów na magistrali
 - Zmiany sygnału są w okolicach **zboczy opadających** linii zegarowej
 - Próbkowanie stanu linii - następuje **w czasie zbocza narastającego** – ciągłe linie na następnych rysunkach

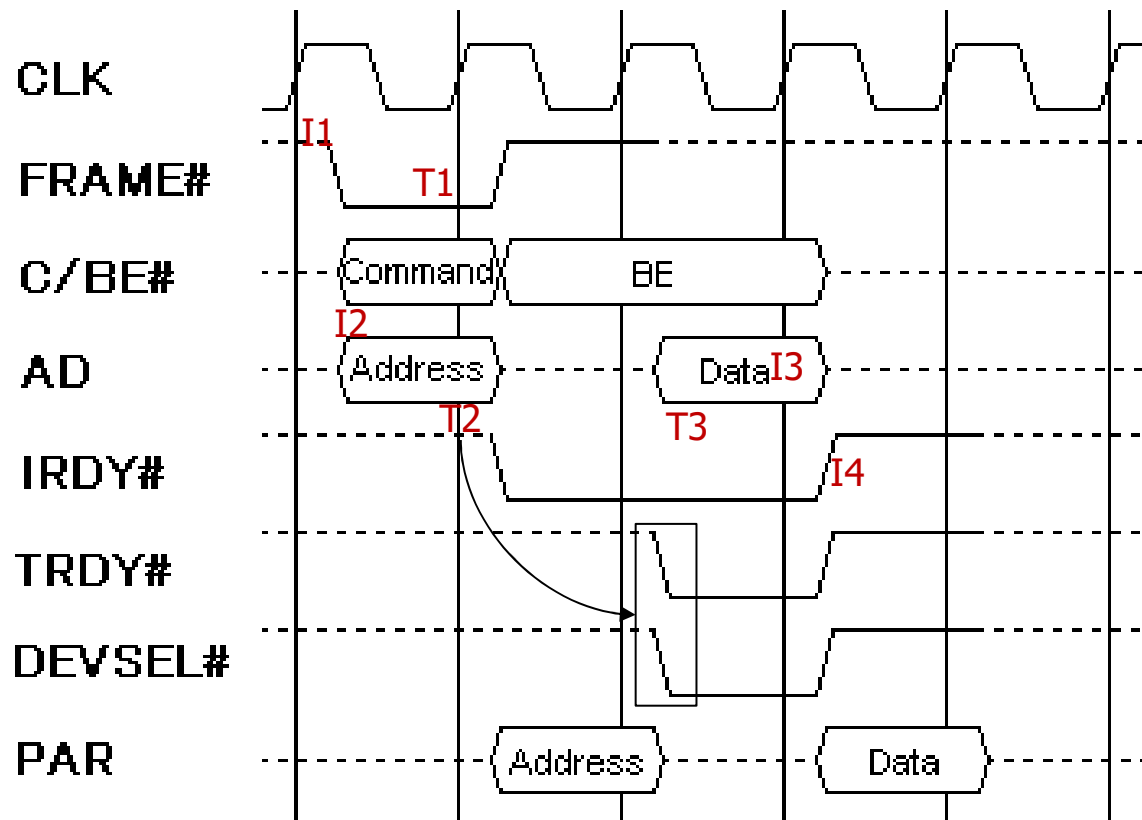
SYKO Systemy dołączania urządzeń wejścia/wyjścia

- Magistrala PCI - Cykl zapisu
- I1) Inicjator przejmuje magistralę FRAME#
- I2) Inicjator wystawia Polecenie i Adres
- T1) Urządzenie docelowe (TARGET) wykrywa start cyklu
- T2) Jeżeli TARGET wykryje swój adres wystawia DEVSEL# i jeżeli gotowy wystawia TRDY#
- I3) Inicjator wystawia wielkość danych i same DANE oraz IRDY#
- T3) Urządzenie docelowe wczytuje dane magistrali



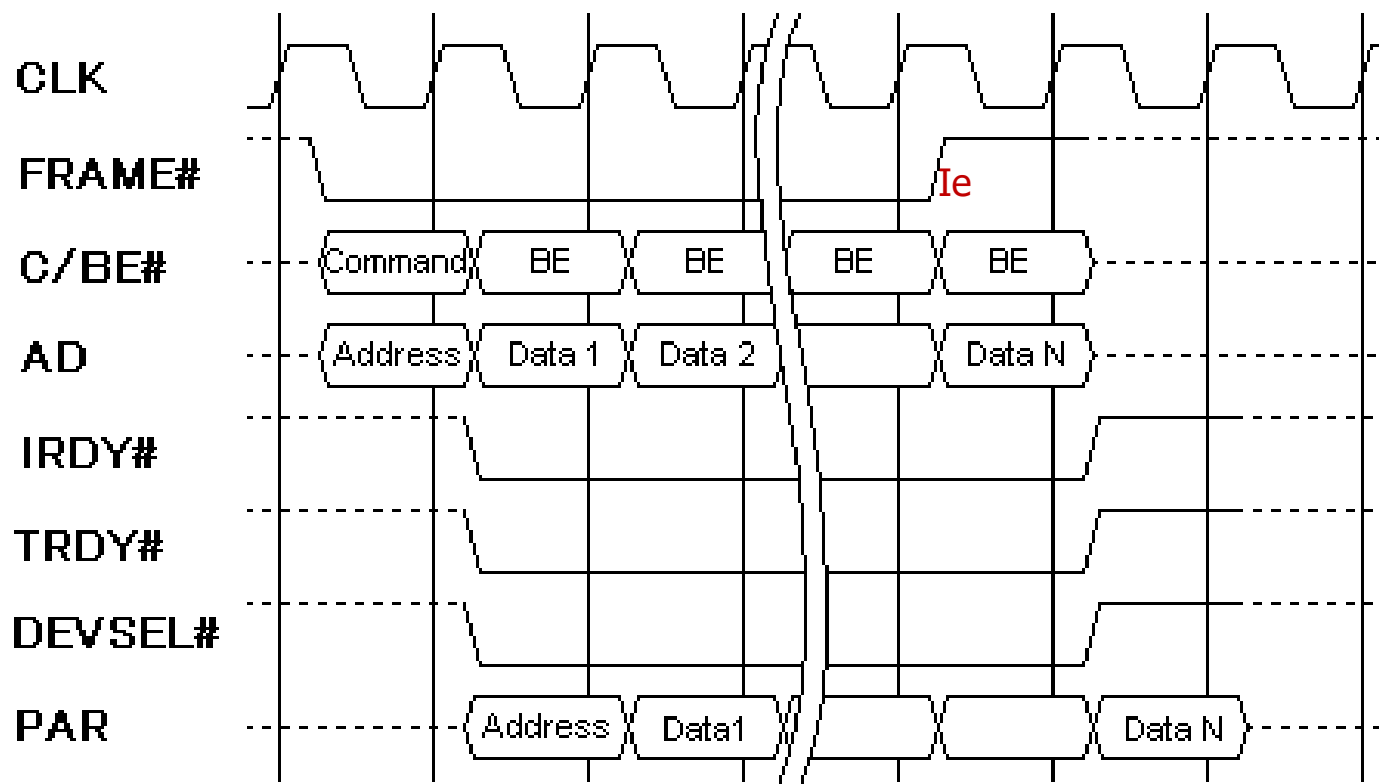
SYKO Systemy dołączania urządzeń wejścia/wyjścia

- Magistrala PCI - Cykl odczytu
- I1) Inicjator przejmuje magistralę FRAME#
- I2) Inicjator wystawia Polecenie i Adres
- T1) Urządzenie docelowe (TARGET) wykrywa start cyklu
- T2) Jeżeli TARGET wykryje swój adres wystawia DEVSEL# i jeżeli gotowy wystawia TRDY#
- T3) Jeżeli Inicjator gotowy (stan IRDY#) Target wystawia dane
- I3) Inicjator próbuje dane
- I4) Jeżeli to koniec to inicjator zdejmie swoją gotowość (IRDY_i i FRAME#)
- T4) Target w odpowiedzi na I4 zdejmie sygnały (TRDY# i DEVSEL#)
- Sygnały na PAR to parzystość adresu lub danych



SYKO Systemy dołączania urządzeń wejścia/wyjścia

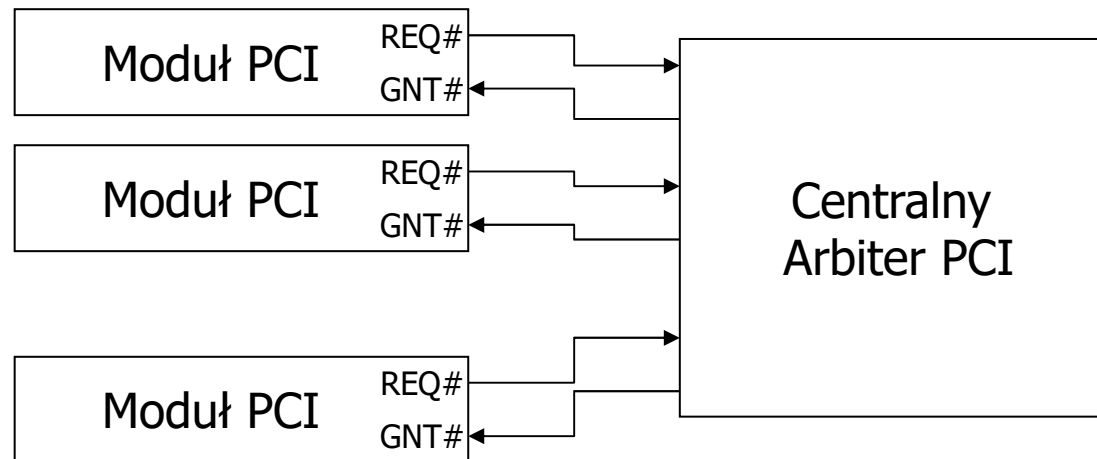
- Magistrala PCI - Cykl zapisu seryjnego (Burst Transfer)
 - Zasadniczo stanowi rozwinięcie pojedynczego cyklu zapisu
 - Urządzenie docelowe kontynuuje zapisy zgodnie z poleceniem – transfer kończy zdjęcie sygnału FRAME# (Ie)



- Inne tryby transferu to „Fast Back to Back Transaction”

SYKO Systemy dołączania urządzeń wejścia/wyjścia

- Magistrala PCI – arbitraż magistrali
 - W systemie jest centralny układ arbitrażu
 - Wszystkie urządzenia PCI podłączone są do tego układu arbitrażu sygnałami REQ# i GNT#



- Moduł pragnący zainicjować transfer – wystawia sygnał REQ# i czeka na sygnał GNT#
 - Jeżeli transfer będzie trwał dłużej niż jeden cykl – wymaga utrzymania sygnału REQ# i obserwowania czy nadal układ arbitrażu przydziela kontrolę (czy sygnał GNT# jest aktywny)
 - Gdy żaden inny moduł nie wystawia swojego żądania dostępu do magistrali – stan ten może trwać do momentu gdy moduł który obecnie wykonuje transfery nie zakończy ich

SYKO Systemy dołączania urządzeń wejścia/wyjścia

- PCI - warstwy niższe wykonane jako PCI Express

- Złącze PCI Express kart tzw. „x1”

Pin	Strona A	Strona B	Opis
1	+12V	PRSNT1#	Pin obecności karty (B)
2	+12V	+12V	
3	+12V	+12V	
4	GND	GND	
5	SMCLK	TCK	SMBus(A) i JTAG(B)
6	SMDAT	TDI	SMBus(A) i JTAG(B)
7	GND	TDO	JTAG(B)
8	+3,3V	TMS	JTAG(B)
9	TRST#	+3,3V	JTAG(A)
10	+3,3V	+3,3V	AUX(A)
11	WAKE#	PERST#	Ponowienie aktywności złącza (A), Główny Reset magistrali (B)
-----klucz-mechaniczny-----			
12	Rezerwa	GND	
13	GND	REFCLK+	Referencyjny sygnał zegarowy, pin + (B)
14	HSOp(0)	REFCLK-	Linia różnicowa transmitera - „Lane 0” (A) Referencyjny sygnał zegarowy, pin - (B)
15	HSOp(1)	GND	Linia różnicowa transmitera - „Lane 0” (A)
16	GND	HSIp(0)	Linia różnicowa odbiornika - „Lane 0” (B)
17	PRSNT2#	HSIp(1)	Pin obecności karty (A), Linia różnicowa odbiornika - „Lane 0” (B)
18	GND	GND	

SYKO Systemy dołączania urządzeń wejścia/wyjścia

- PCI Express – warstwowy model transmisji
 - Warstwa Transakcji (Transaction Layer)



- Warstwa Łącza Danych (Data Link Layer)
 - Odebrane pakiety są potwierdzane: ACK DDLP – gdy odebrano je poprawnie a w przeciwnym przypadku wysyłane jest Nack DDLP



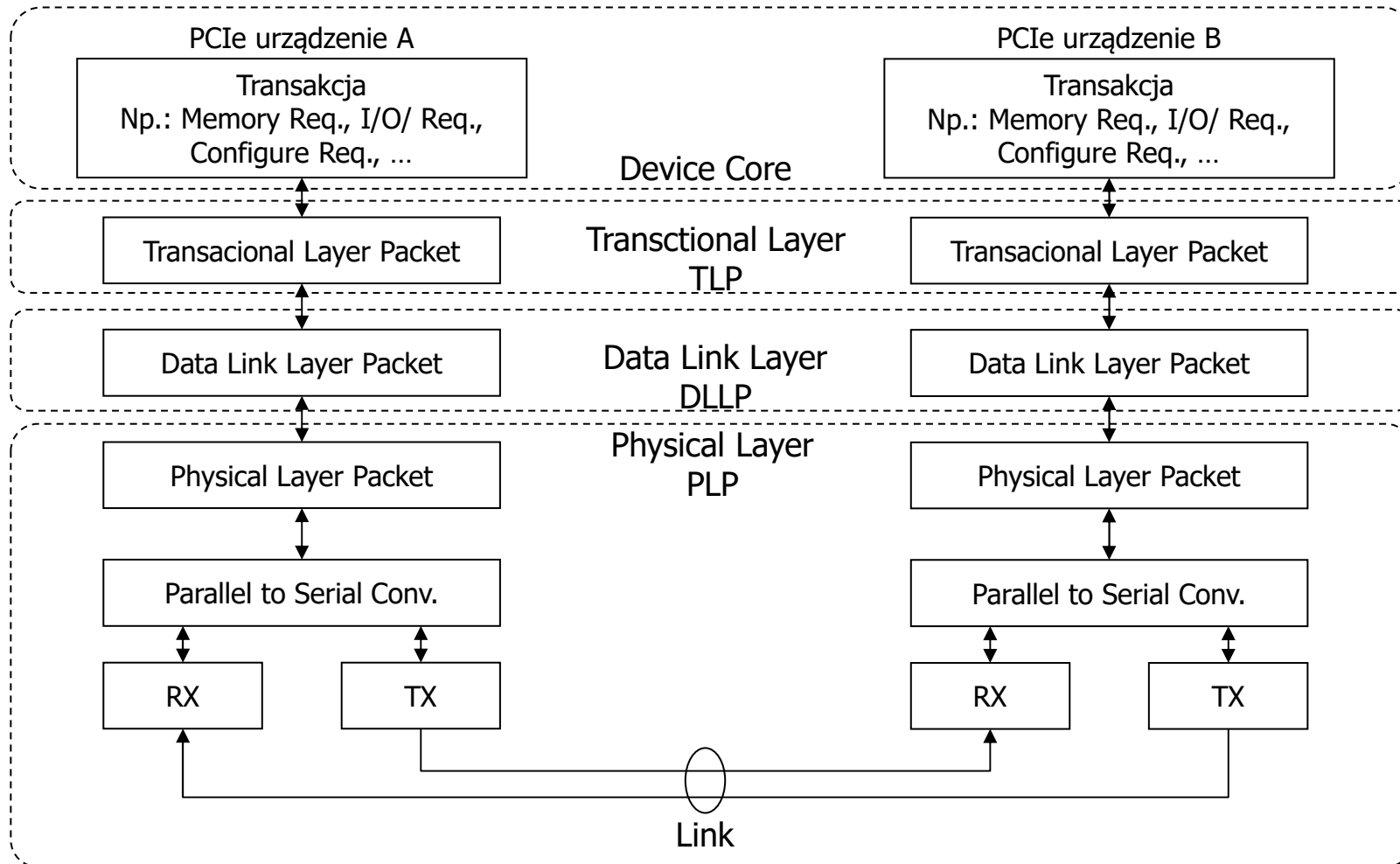
- Warstwa fizyczna (Physical Layer)



- Modulacja nadmiarowa - skramblowanie: 8b/10b lub 128b/130b (nowsza wersja standardu)

SYKO Systemy dołączania urządzeń wejścia/wyjścia

- PCI Express – warstwowy model transmisji, cd.



SYKO Systemy dołączania urządzeń wejścia/wyjścia

- PCI Express – przepustowości i połączenia

■ Liczba łączy (lane)	Pasmo (GB/sec)
■ x1	0,5
■ x2	1
■ x4	2
■ ...	
■ x16	8

- PCIe link czy PCIe lane

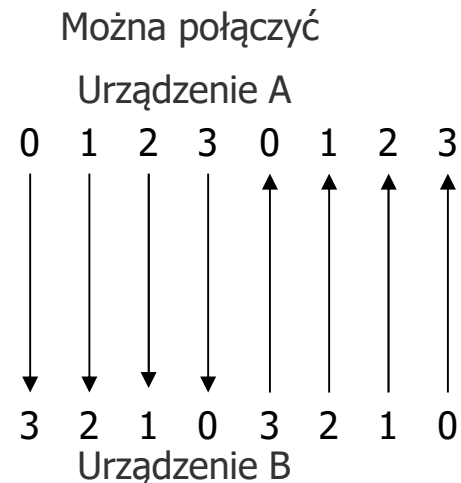
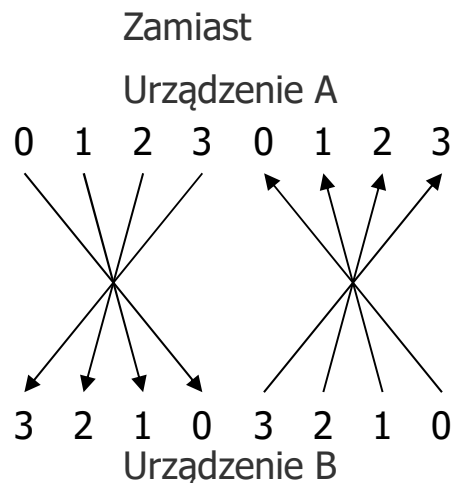
- Link może zawierać wiele łączy (lane)
 - xN – oznacza że link zawiera N łączy (lane)
 - Jeden link obsługuje jeden port
- Łącze (Lane) – jedna para różnicowa do transmisji a druga do odbioru

- Połączenia są realizowane w stylu point-to-point

- Tworzą je połączenia szeregowo w ramach każdej z par różnicowych
 - Poziom sygnałów: różnicowe 800mV...1,2V
 - Składowa stała między 0 a 3,6V – a raczej użytkowo między 800mV a 2,4V

SYKO Systemy dołączania urządzeń wejścia/wyjścia

- PCI Express – połączenia
 - Konfigurowalność połączeń
 - Połączenia nie muszą być pełne
 - Połączenie „do góry” może mieć więcej łączy (lane) niż połączenie „do dołu”
 - Odwracalność połączeń i polaryzacji (Lane Reversal i Polarity Inversion)
 - Zdolność do zmiany odpowiadających sobie połączeń (Lane) - upraszcza tworzenie PCB
 - Zamiast łączyć lane 01 urządzenia A z lane 01 urządzenia B można je połączyć z lane 32 urządzenia B
 - Jak również D+ i D- zamiast łączyć wprost z D+ i D- można po inwersji polaryzacji podłączyć z D- i D+



Dziękuję za uwagę