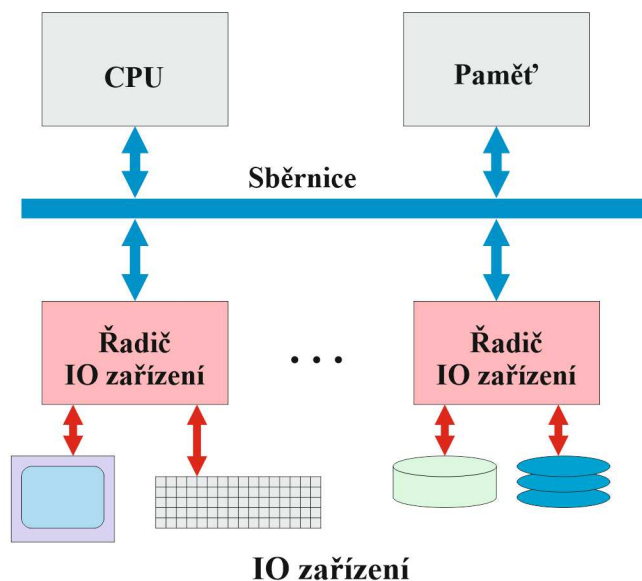


Příloha 1

SELEKČNÍ KANÁL

Koncepce připojení IO zařízení

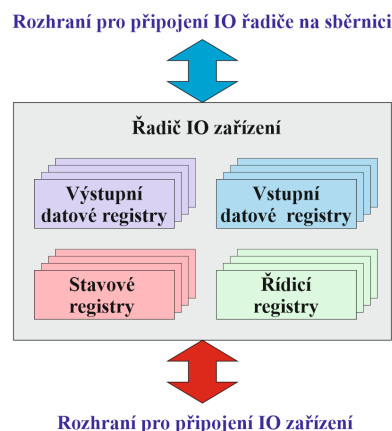
- IO zařízení jsou připojena na sběrnici pomocí řadičů.
- Řadiče
 - specializované (např. řadič disku, grafická karta, ...) – lze k nim připojit jen zařízení určitého typu.
 - univerzální (např. sériové rozhraní USB, ...) – lze k nim připojit zařízení různých typů.



Rozhraní IO řadiče

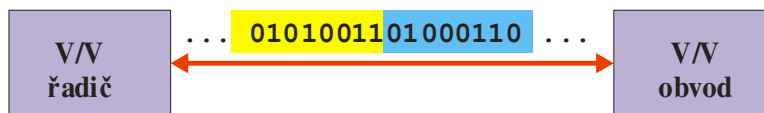
Rozhraní IO řadiče:

- Rozhraní pro připojení na sběrnici
 - obvykle navrženo univerzálně pro snadné připojení na různé systémové sběrnice (nebo naopak jen pro určitou sběrnici, např. PCI),
 - zahrnuje datové vodiče, několik adresních vodičů (výběr registru), řídicí signály (/CS, /RD, /WR, Reset, ...).
- Rozhraní pro připojení IO zařízení
 - u specializovaných řadičů přizpůsobeno připojenému IO zařízení (disk, ...),
 - u univerzálních je buď programovatelné nebo přímo odpovídá standardu některého univerzálního rozhraní (SCSI, USB, ...).

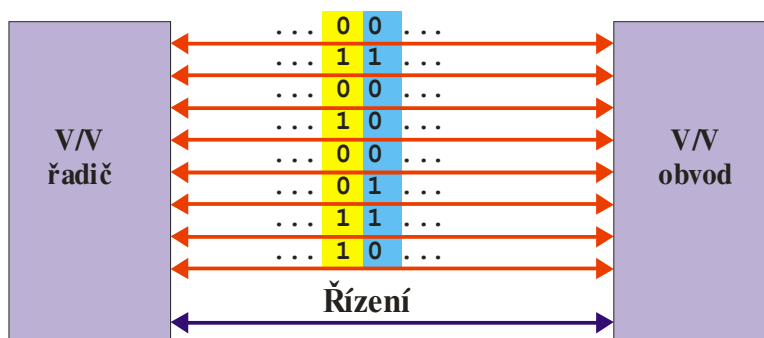


Základní typy přenosu dat z/do IO zařízení (1)

- **Sériový přenos dat** – data (a často i strobovací signál) se přenáší po jednom vodiči „bit po bitu“.

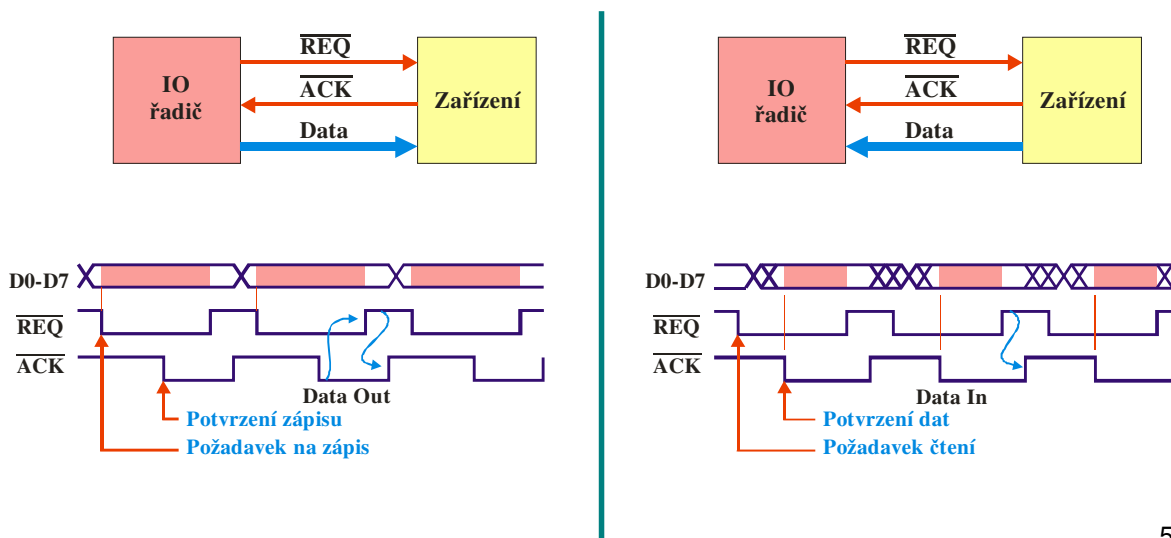


- **Paralelní přenos dat** – data se přenáší po více vodičích, obvykle po bytech, slovech (16 bitů) apod.



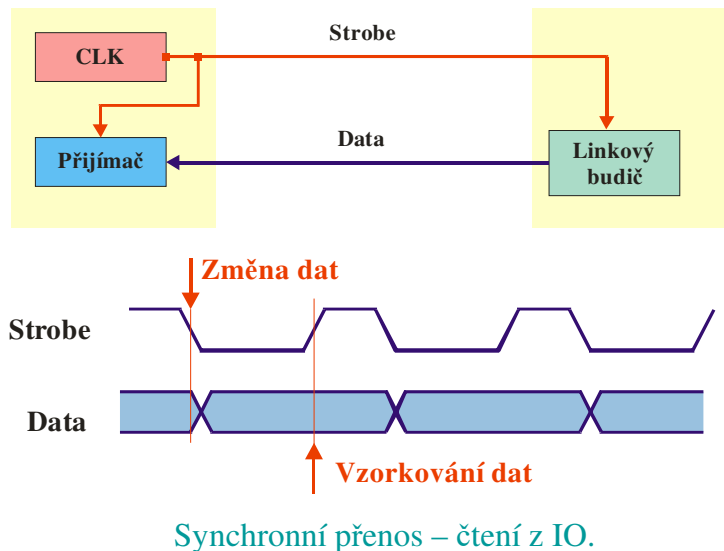
Základní typy přenosu dat z/do IO zařízení (2)

- **Asynchronní přenos** – Řadič komunikuje se zařízením způsobem požadavek – potvrzení (**Request – Acknowledge**), tzv. handshake.
 - Umožňuje snadné přizpůsobení rychlosti počítače (řadiče) a zařízení.
 - Obousměrná komunikace omezuje maximální dosažitelnou rychlost.



Základní typy přenosu dat z/do IO zařízení (3)

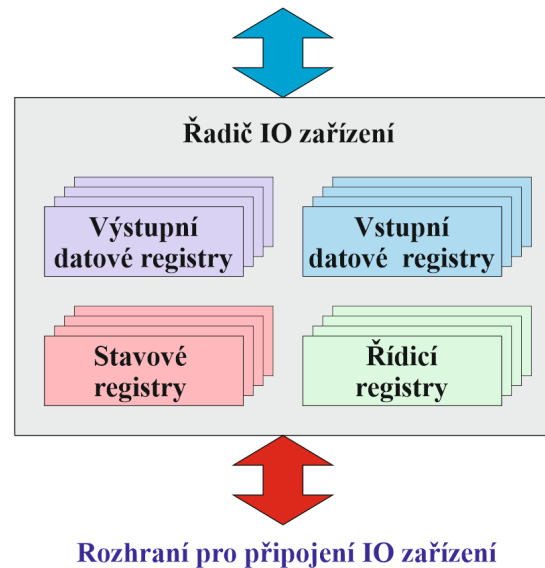
- **Synchronní přenos** – Řadič (nebo zařízení) vysílá taktovací signál. V každé periodě proběhne jeden dílčí přenos. Provedení přenosu se nepotvrzuje.
 - Umožňuje dosažení vysoké přenosové rychlosti.



IO řadič z pohledu programátora (1)

- IO řadič je reprezentován sadou registrů:
 - **Datové registry** – pro vstup a výstup dat.
 - **Řídicí registry** – nastavení parametrů IO řadiče resp. připojeného IO zařízení.
 - **Stavové registry** – zjištění stavu IO řadiče resp. připojeného IO zařízení.

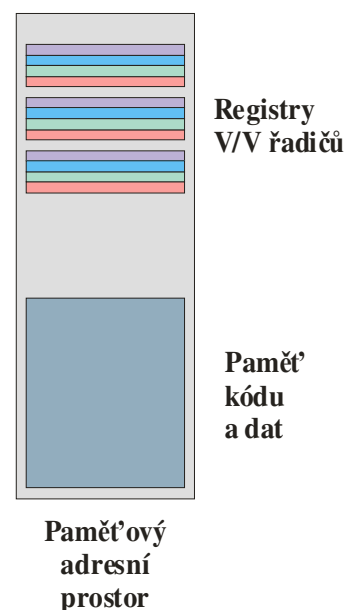
Rozhraní pro připojení IO řadiče na sběrnici



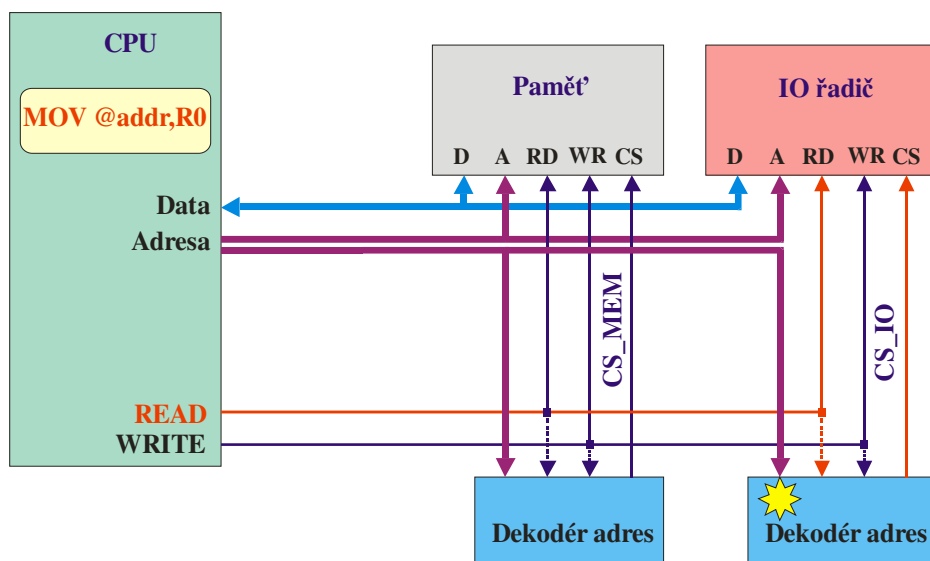
IO řadič z pohledu programátora (2)

Jednotlivé registry IO řadiče mohou být mapovány v paměťovém nebo ve IO adresním prostoru.

- **IO řadič v paměťovém adresním prostoru (například procesor H8S):**
 - pro přístup k registrům IO řadiče se používají stejné instrukce jako pro práci s pamětí (MOV, ...),
 - lze používat i další instrukce (pokud jsou v instrukčním souboru).



IO řadič v paměťovém adr. prostoru

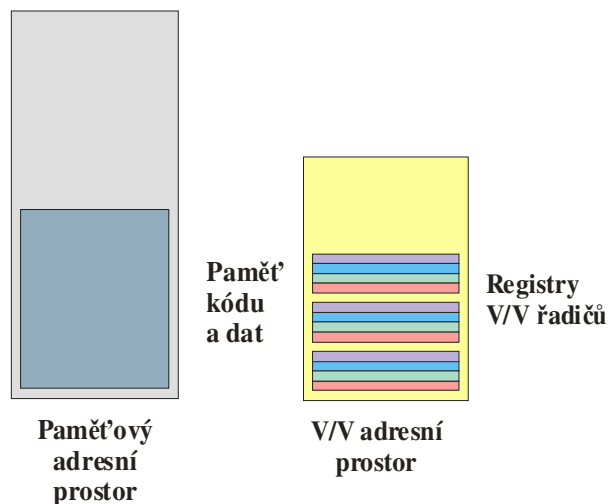


IO řadič z pohledu programátora (3)

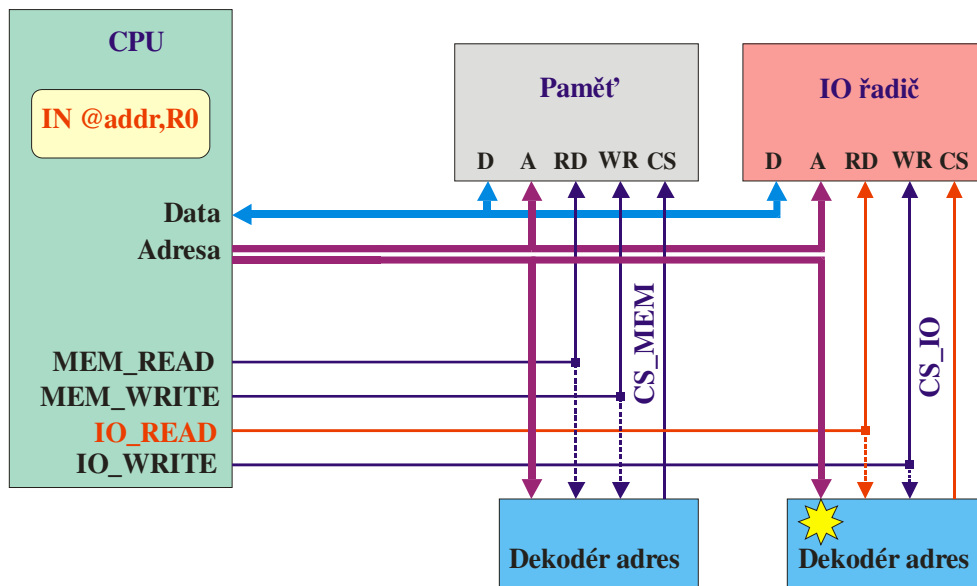
Jednotlivé registry IO řadiče mohou být mapovány v paměťovém nebo ve IO adresním prostoru.

- **IO řadič ve IO adresním prostoru (např. IA-32):**

- procesor rozlišuje adresní prostor paměti a adresní prostor IO zařízení,
- pro práci se zařízeními ve IO adresním prostoru slouží zvláštní instrukce (IN, OUT),
- na sběrnici se vyskytují kromě cyklů „čtení z paměti“ a „zápis do paměti“ také cykly „čtení z IO zařízení“ a „zápis do IO zařízení“,
- adresní prostor IO zařízení má obvykle menší velikost než adresní prostor paměti.



IO řadič v odděleném IO adr. prostoru

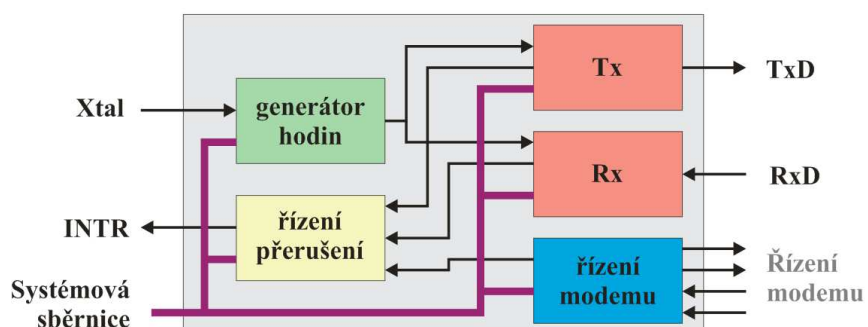


Univerzální řadič sériové linky PC16550D

- „Klasický“ řadič pro sériovou linku RS232.
- Řada výrobců, dostupný též jako syntetizovatelné jádro pro ASIC nebo FPGA.
- Funkční bloky v PC16550D:
 - vysílač sériových dat (Tx),
 - přijímač sériových dat (Rx),
 - obvody pro řízení modemu,
 - generátor vysílací/přijímací frekvence (BRG – Baud Rate Generator),
 - obvody pro řízení přerušování.

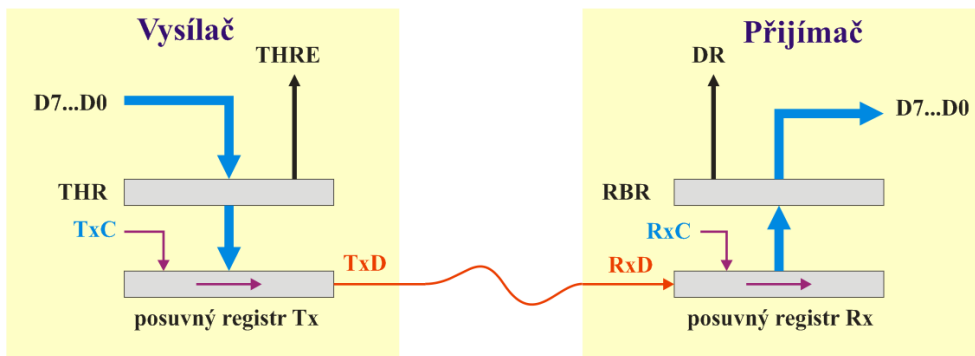
Podrobnosti např.

<https://www.ti.com/product/TL16C550C>



Sériové rozhraní RS232 (1)

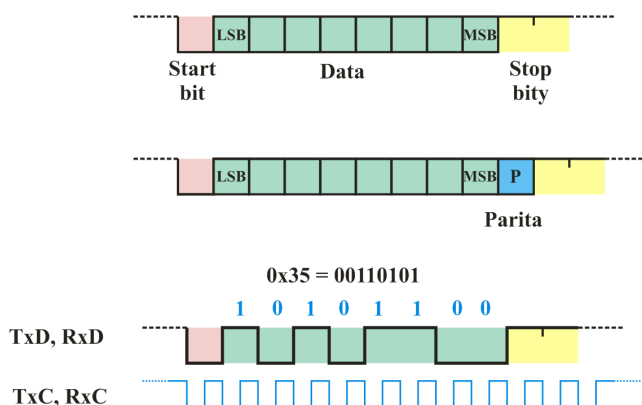
- Data se přenáší po jednom vodiči „bit po bitu“.
- Pro každý směr samostatný vodič.
- Vysílač (Tx) a přijímač (Rx) musí mít nastavenou stejnou přenosovou rychlost (frekvenci hodinového signálu TxC, RxC).



Základní uspořádání sériové linky pro jeden směr přenosu.

Sériové rozhraní RS232 (2)

- Na začátku každého znaku je Start bit (vždy 0).
- Data mají obvykle 8 bitů (lze použít i 5, 6 nebo 7 bitů), LSB se vysílá první.
- Na konci znaku je 1 nebo 2 Stop bity (vždy 1).
- Data mohou být volitelně doplněna lichou nebo sudou paritou.

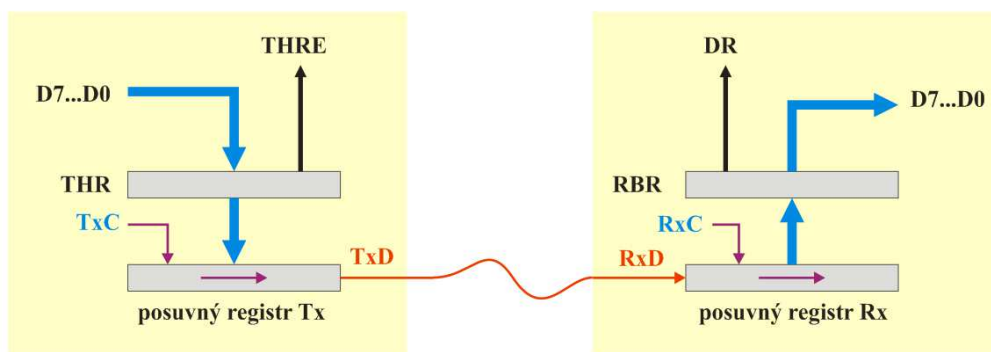


Poznámka:

Rozhraní RS232 používá pro TxD a RxD 12V logiku (0 = +12V, 1 = -12V).

Sériové rozhraní RS232 (3)

- Data určená k vysílání se zapisují do THR (Transmitter Holding Register).
 - Vyprázdnění THR indikuje signál THRE (THR Empty).
- Přijatý znak se uloží do RBR (Receiver Buffer Register).
 - Naplnění RBR indikuje signál DR (Data Ready).
 - THRE a DR jsou ve stavovém registru PC16550D.



Registry PC16550D

Adresa registru	Registr	Kategorie
0 (W)*	THR	Datové
0 (R)*	RBR	
1 (RW)*	IER	Řídicí
2 (W)	FCR	
3 (RW)	LCR	
4 (RW)	MCR	
0 (RW)**	DLL	Stavové
1 (RW)**	DLM	
2 (R)	IIR	
5 (R)	LSR	
6 (RW)	MSR	
7 (RW)	SCR	

* je-li DLAB v LCR = 0
** je-li DLAB v LCR = 1

- **THR** – data pro vysílání.
- **RBR** – přijatá data.
- **IER (Interrupt Enable Register)** – povolení přerušení.
- **FCR (FIFO Control Register)** – řízení fronty FIFO.
- **LCR (Line Control Register)** – řízení přenosu.
- **MCR (Modem Control Register)** – řízení modemu.
- **DLL, DLM** – nastavení přenosové rychlosti.
- **IIR – (Interrupt Identification Register)** – identifikace příčiny přerušení.
- **LSR (Line Status Register)** – stav přenosu.
- **MSR (Modem Status Register)** – stav modemu.

Detaily k některým registrům - LCR

7	6	5	4	3	2	1	0	
DLAB	SB	SP	EPS	PEN	STB	WLS1	WLS0	LCR

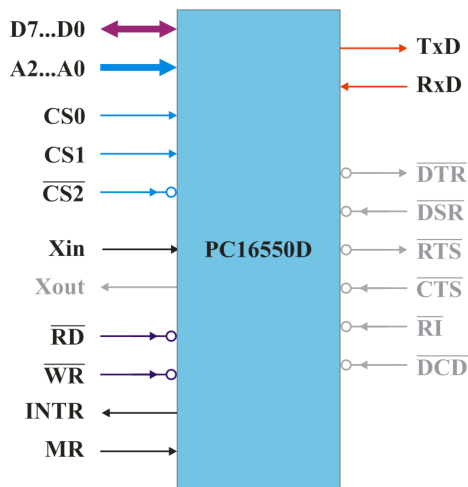
- DLAB** 0 = adresování THR/RBR,
1 = adresování DLL/DLM.
- SB** 0 = normální vysílání,
1 = vysílání BREAK.
- SP** 0 = normální vysílání,
1 = vysílání 9. bitu.
- EPS** 0 = lichá parita,
1 = sudá parita.
- PEN** 0 = bez parity,
1 = s paritou.
- STB** 0 = 1 Stop bit,
1 = 2 Stop bity.
- WLS1,WLS0** 00 = 5 bitů,
01 = 6 bitů,
10 = 7 bitů,
11 = 8 bitů.

Detaily k některým registrům - LSR

7	6	5	4	3	2	1	0	
ERF	TEMT	THRE	BI	FE	PE	OE	DR	LSR

- ERF** 0 = bez chyb,
1 = chyba ve FIFO přijímače.
- TEMT** 0 = probíhá vysílání,
1 = vysílací registr prázdný.
- THRE** 0 = THR není prázdný,
1 = THR je prázdný.
- BI** 0 = normální stav linky,
1 = stav BREAK na lince.
- FE** 0 = bez chyby,
1 = Framing Error.
- PE** 0 = bez chyby,
1 = Parity Error.
- OE** 0 = bez chyby,
1 = Overrun Error.
- DR** 0 = RBR prázdný,
1 = RBR obsahuje přijatá data.

Vývody PC16550D



Vývody obvodu PC16550D
(zjednodušeno)

- D7 ... D0 – datová sběrnice.
- CS0, CS1, /CS2 – výběr obvodu.
- A2 ... A0 – výběr registru.
- Xin, Xout – vstup hodinového signálu pro odvození TxC, RxC.
- /RD, /WR – čtecí a zápisový signál.
- INTR – výstup přerušení.
- MR – reset.
- TxD, RxD – sériová data.
- /DTR, /DSR, /RTS, /CTS, /RI, /DCD – signály pro řízení modemu.

Připojení PC16550D na sběrnici procesoru H8S – 1. verze

- Adresní vodiče A0, A1 a A2 zavedeme přímo na příslušné vstupy pro výběr registru uvnitř PC16550D.
- Ostatní adresní vodiče zavedeme do externího dekodéru adres, který bude generovat /CS2 pro PC16550D.
 - Adresa obvodu bude dána konstrukcí dekodéru.
- Signály /RD a /HWR lze použít přímo, /RESET je nutno invertovat.
 - Čtecí a zápisový cyklus PC16550D má podobné časování jako paměti RAM.

A23 A22 A21 A20 ... A3 A2 A1 A0

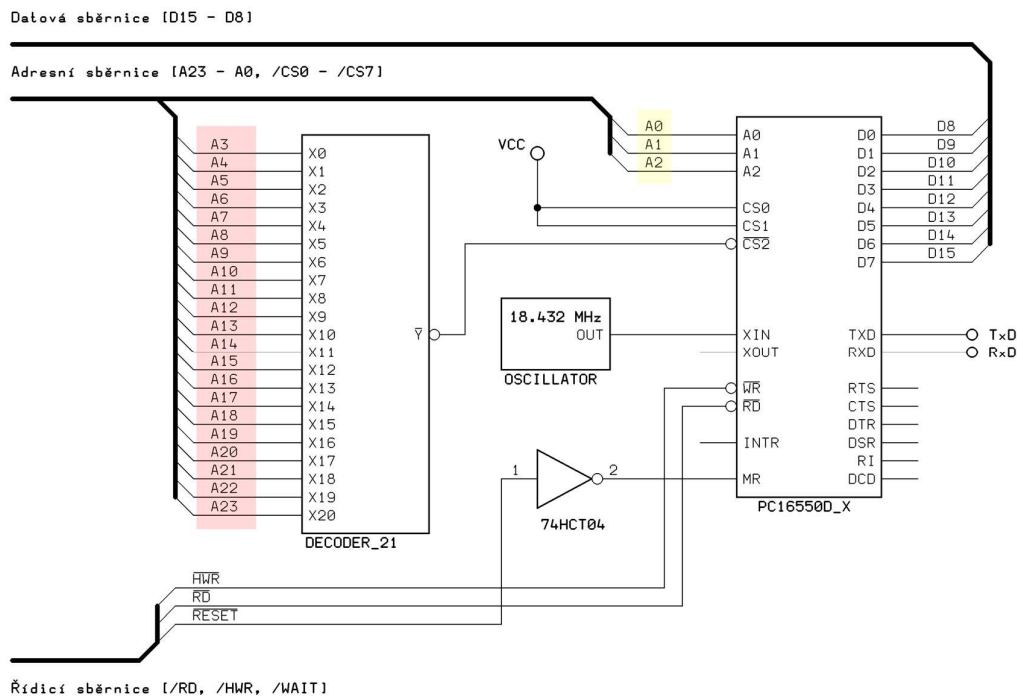


Externí dekodér



Adresa registru v PC16550D

Připojení PC16550D na sběrnici procesoru H8S – 1. verze



Připojení PC16550D na sběrnici procesoru H8S – 2. verze

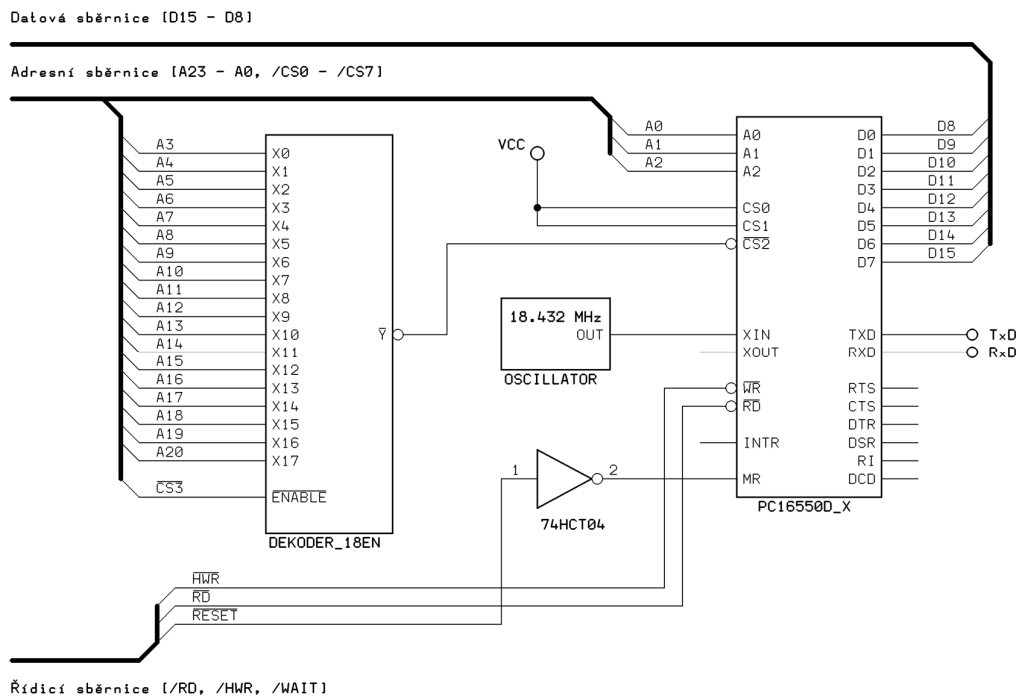
– Využijeme interní dekodér adres v H8S.

- V nejjednodušší verzi budeme předpokládat dekodér s 18 vstupy pro kombinaci vstupů X17 ... X0 = L, /ENABLE = L).

A23 A22 A21 A20 ... A8 A7 A6 A5 A4 A3 A2 A1 A0

- Dekódováno v H8S
- Externí dekodér
- Adresa registru v PC16550D

Připojení PC16550D na sběrnici procesoru H8S – 2. verze



Připojení PC16550D na sběrnici procesoru H8S – 2. verze

A23 A22 A21 A20 ... A8 A7 A6 A5 A4 A3 A2 A1 A0

- Dekódováno v H8S
- Externí dekodér
- Adresa registru v PC16550D

- Důsledek předchozího zapojení:
 - RBR, THR leží na adrese **0110 0000 0000 0000 0000 0000** (binárně), tj.600 000.
 - IER leží na adrese **0110 0000 0000 0000 0000 0001** (binárně), tj.600 001.
 - FCR, IIR leží na adrese **0110 0000 0000 0000 0000 0010** (binárně), tj.600 002.
 - LCR leží na adrese **0110 0000 0000 0000 0000 0011** (binárně), tj.600 003.
 - atd.
 - MSR leží na adrese **0110 0000 0000 0000 0000 0110** (binárně), tj.600 006.

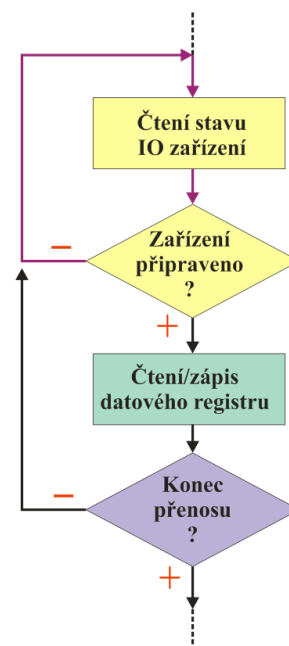
Řízení IO operací

Komunikace s IO zařízením může být řízena podle některé ze základních koncepcí:

- Programové řízení IO operací procesorem.
- Řízení IO operací s využitím přerušení.
- Řízení IO operací DMA řadičem.
- Použití specializovaného IO procesoru.

Programové řízení V/V zařízení procesorem (1)

1. **Procesor čte (programově) stavové registry řadiče a testuje připravenost zařízení.**
2. **Není-li zařízení připraveno opakuje čtení a test (bod 1).**
3. Je-li zařízení připraveno, čte nebo zapisuje data z/do datových registrů.
4. Postup se opakuje až do konce přenosu.



Programové řízení V/V zařízení procesorem (2)

1. Procesor čte (programově) stavové registry řadiče a testuje připravenost.
2. Není-li zařízení připraveno opakuje čtení a test (bod 1).
3. Je-li zařízení připraveno, čte nebo zapisuje data z/do datových registrů.
4. Postup se opakuje až do konce přenosu.

Vlastnosti:

- Procesor je trvale zatížen čtením a testováním stavu.
- Maximální rychlost je omezena rychlostí provedení instrukcí pro čtení stavu – test – zápis/čtení dat.
- Jednoduchý hardware.

Příklad: programové řízení vysílání dat na sériovém rozhraní s PC16550D.

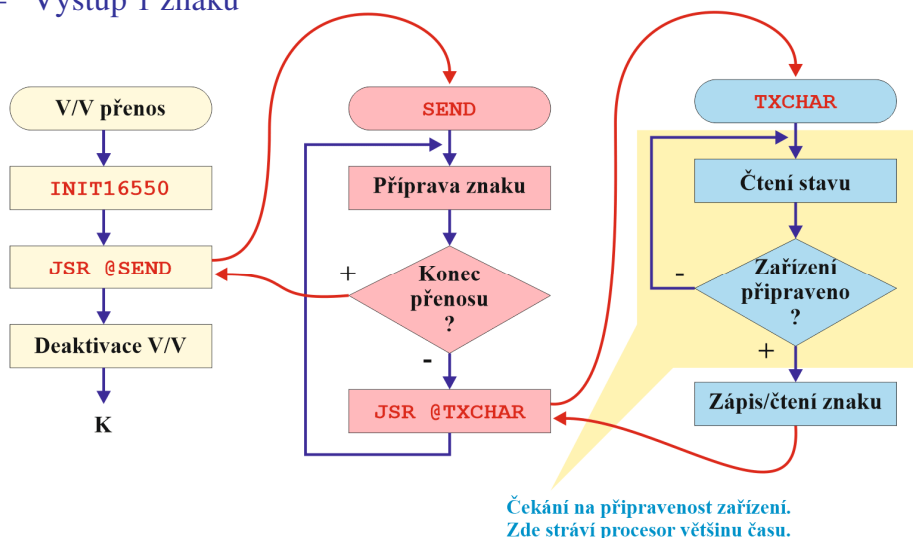
- Vysílání řetězce znaků zakončeného znakem 0x00.

Příklad: programová obsluha IO zařízení

- Příklad: jednoduchý ovladač pro vyslání řetězce znaků sériovým rozhráním.
 - Řetězec je uložen v paměti a je ukončen znakem 0x00.
 - Sériové rozhraní je řízeno obvodem PC16550D.
 - Formát dat bude nastaven na 9k6,n,8,2 (frekvence 9600 bitů/sec., bez parity, 8 bitů, 2 Stop bity).
 - Vyslání řetězce bude řízeno programově.

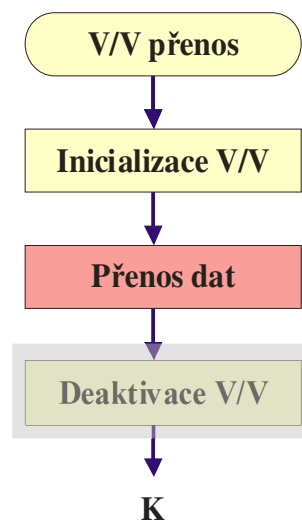
Příklad: programová obsluha IO zařízení

- Ovladač bude složen ze tří procedur:
 - Inicializace PC16550D
 - Výstup řetězce
 - Výstup 1 znaku



Programová obsluha IO zařízení

- Inicializace PC16550D :
 - nastavení přenosové rychlosti,
 - nastavení režimu (počet bitů, parita),
 - zákaz přerušení.
- Přenos dat
 - adresa textu je programu předána v ER1,
 - text zakončen 0x00.



Programová obsluha – inicializace PC16550D (1)

```

.equ  BASE16550, 0x600000    ;bazova adresa PC16550D
.equ  THR,      BASE16550    ;transmit register
.equ  IER,      BASE16550+1  ;interrupt control
.equ  LCR,      BASE16550+3  ;line control register
.equ  LSR,      BASE16550+5  ;line status register
.equ  DLL,      BASE16550    ;nast. frekvence low
.equ  DLM,      BASE16550+1  ;nast. frekvence high
...
TEXT:  .asciz „AHOJ“        ;text pro tisk + 0x00
...
START: JSR      @INIT16550    ;inicializace PC16550D
      MOV.L    #TEXT,ER1     ;adresa textu do ER1
      JSR      @SEND         ;procedura pro vysilani
...

```

Programová obsluha – inicializace PC16550D (2)

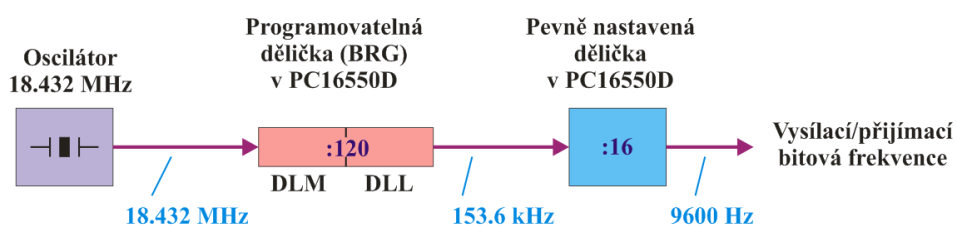
```

INIT16550: MOV.B    @LCR,R0L    ;puvodni hodnota LCR
          OR.B     #0x80,R0L    ;DLAB nastaven na 1
          MOV.B    R0L,@LCR    ;zapis do LCR
          MOV.W    #120,R0     ;delici pomer do R0
          MOV.B    R0L,@DLL    ;zapis dolni casti
          MOV.B    R0H,@DLM    ;zapis horni casti

```

pokračování na další stránce

$$\frac{18432000}{153600} = 120$$

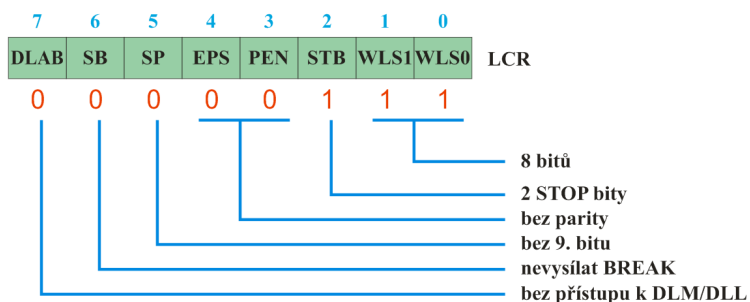


Programová obsluha – inicializace PC16550D (3)

pokračování z předchozí stránky

```

MOV.B    #0x07,R0L    ;8bitu, 2stop, bez parity
MOV.B    R0L,@LCR     ;nastaveni LCR, DLAB = 0
XOR.B    R0L,R0L     ;0x00 -> R0L
MOV.B    R0L,@IER     ;zakaz vseh preruseni
RTS      ;navrat
    
```



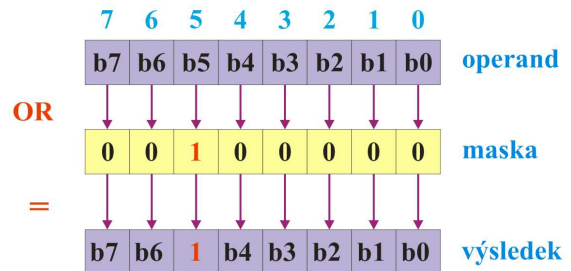
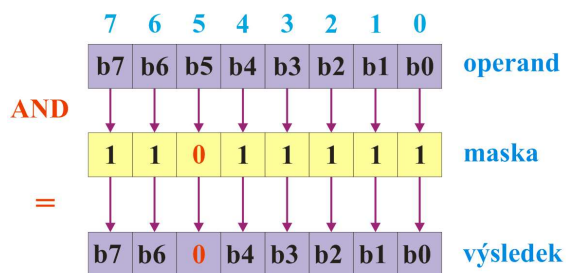
Poznámka: bitové operace (1)

- H8S má pro bitové operace speciální instrukce:
 - **BSET** nastavení bitu na 1
 - **BCLR** nastavení bitu na 0
 - **BTST** testování bitu
 - ...
- Například:
 - **BSET #5,@0x00FF4020 ; nastavení bitu č.5 v bytu na adrese 0x00FF4020**
- Instrukce se provádí způsobem **READ – MODIFY – WRITE**
 - přečtení bytu z paměti → nastavení zvoleného bitu → zápis bytu do paměti

BSET #xx:3,@ERd	R:W 2nd	R:B EA	R:W NEXT	W:B EA		
BSET #xx:3,@aa:8	R:W 2nd	R:B EA	R:W NEXT	W:B EA		
BSET #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W NEXT	W:B EA	
BSET #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W NEXT	W:B EA
BSET Rn,Rd	R:W NEXT					
BSET Rn,@ERd	R:W 2nd	R:B EA	R:W NEXT	W:B EA		
BSET Rn,@aa:8	R:W 2nd	R:B EA	R:W NEXT	W:B EA		

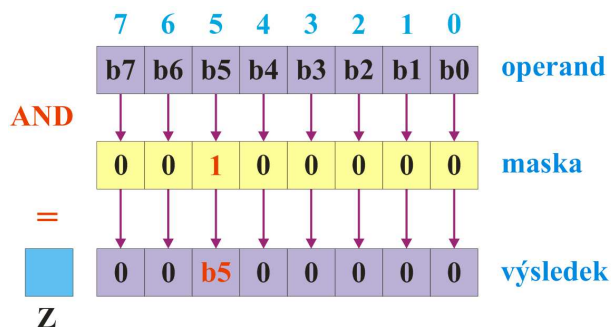
Poznámka: bitové operace (2)

- Bitové operace lze provádět s využitím instrukcí pro bitový součin (AND) nebo bitový součet (OR):
- Nastavení zvolených bitů na 0:
 - Proveďte se součinem s vhodně zvolenou maskou.
 - Ve výsledku jsou zvolené bity nastavené na 0, ostatní jsou nezměněné.
- Nastavení zvolených bitů na 1:
 - Proveďte se součinem s vhodně zvolenou maskou.
 - Ve výsledku jsou zvolené bity nastavené na 1, ostatní jsou nezměněné.



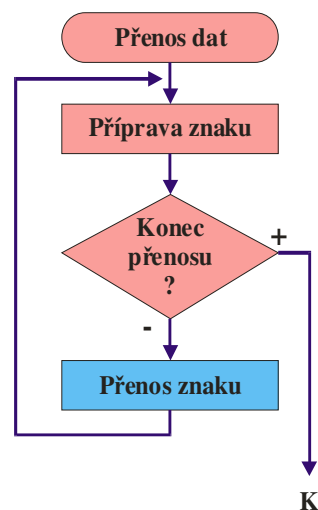
Poznámka: bitové operace (3)

- Bitové operace lze provádět s využitím instrukcí pro bitový součin (AND) nebo bitový součet (OR):
- Testování hodnoty určitého bitu:
 - Proveďte se součinem s vhodně zvolenou maskou.
 - Podle výsledku operace se nastaví příznak Z (Zero):
 - Je-li (bit == 1), nastaví se Z = 0.
 - Je-li (bit == 0), nastaví se Z = 1.



Programová obsluha – přenos dat (1)

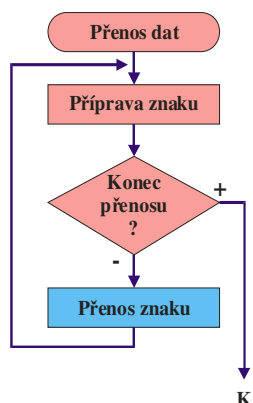
- Příprava znaku
 - přečtení znaku z paměti do registru,
 - posun pointeru.
- Kontrola konce přenosu
 - znak == 0x00 ?



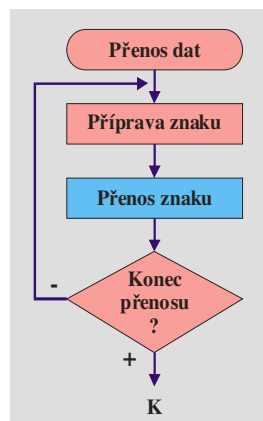
Ukončení přenosu

- Nalezením koncového znaku.
 - Např. znak == 0x00 ?
- Přenesením všech znaků (vyčerpání délky přenosu).
 - Např. sektor na disku = 512 bytů.

Nalezení
koncového
znaku –
koncový znak
se netiskne



Vyčerpání délky
nebo nalezení
koncového znaku
– koncový znak
se tiskne



Programová obsluha – přenos dat (2)

```

...
SEND:  MOV.B  @ER1,R0L      ;znak do R0L
        BEQ   LAB_END      ;je-li znak 0x00 (Z == 1)
        INC.L  ER1         ;posun pointeru
        JSR   @TXCHAR      ;vysilani znaku
        BRA   SEND        ;opakovani
LAB_END:RTS                ;navrat - znak byl 0x00
...

```

Programová obsluha – vysílání znaku (1)

- Čtení stavu
 - čtení LSR,
 - test připravenosti
 - (THRE == 1) ⇒ připraveno.
- Zápis znaku
 - zápis do THR
 - při zápisu do THR se automaticky vynuluje THRE.



Programová obsluha – přenos znaku (2)

```

...
TXCHAR:  MOV.B  @LSR, R0H      ;LSR do R0H
          AND.B  #0x20, R0H    ;maskovani bitu 5
          BEQ   TXCHAR        ;THRE==0 -> cekame
;
          MOV.B  R0L, @THR     ;zapis do THR
          RTS                    ;hotovo
...
    
```

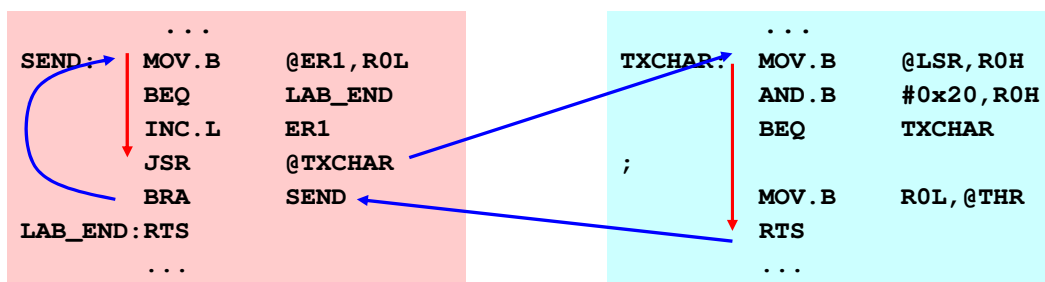


Zde program čeká na vyprázdnění datového registru

Při přenosové rychlosti 9k6 trvá vysílání 1 bytu cca. 1 ms. Trvá-li provedení 1 instrukce cca. 1 μs, pak se při čekání provede čekací cyklus (tj. 3 instrukce) přibližně 300x.

Maximální rychlost přenosu

- Pro přenos každého znaku se musí provést minimálně 10 instrukcí ⇒ rychlostí jejich provedení je určena maximální rychlost přenosu.
 - Trvá-li provedení 1 instrukce přibližně 1 μs, je maximální dosažitelná přenosová rychlost přibližně 100 kB/sec.



1 průchod cyklem ≈ 10 μs