SPI řešené se dvěma mikroprocesory AT89C2051

Doc.Ing.Rudolf Jalovecký,CSc. (vyvoj@jalsoft.iol.cz)

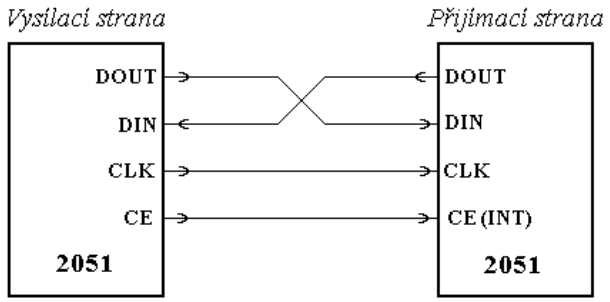
Úvod aneb něco o SPI

Komunikační rozhraní SPI (Serial Peripheral Interface) je velmi rozšířené jednosměrné nebo obousměrné komunikační rozhraní pro připojení celé řady periferních obvodů k mikroprocesoru. Ve své podstatě jde o synchronní jednosměrné (dvouvodičové) nebo obousměrné (třívodičové) rozhraní, které bývá ještě navíc vybaveno třetím (čtvrtým) vodičem pro výběr periferního obvodu (CE - Chip Enable). V současnosti se s pomocí SPI mohou k mikroprocesoru připojit sériové paměti (RAM, EEPROM), zobrazovače (jak LCD tak LEDkové), ovladače krokových motorů, generátory pulsně šířkové modulace (PWM) apod. Pro svou „jednoduchost“ je velmi oblíbené a často používané. V praxi existuje několik variant SPI, princip přenosu dat je v podstatě stejný. Na rozdíl od paralelního přenosu dat se zde používá jen jeden vodič pro přenos dat a druhý pro potvrzení platnosti těchto dat na straně výstupu. Znamená to, že tento přenos se musí uskutečnit bit po bitu sériově. Zásadní vlastností tohoto přenosu je také skutečnost, že vysílací strana nikdy nečeká, zda přijímací strana všechna data bezchybně přijala. To musí být řešeno většinou volbou vhodného přenosového protokolu.

V praxi se občas vyskytne potřeba přenášet data mezi dvěma nebo i více mikroprocesory a klasické sériové rozhraní je již obsazené, případně přenos přes toto rozhraní může být pomalý. Uvedený příspěvek řeší tuto problematiku programovými prostředky na obou stranách (vysílací a přijímací).

Technické řešení

Elektrické zapojení konkrétního obousměrného SPI je uvedeno na obr. 1. Za zmínku snad stojí upřesnění „překřížení“ datových bitů DOUT a DIN na vysílací a přijímací straně (oba mikroprocesory jsou ve své podstatě zdroji signálu) a spíš jen upozornění že signály CLK a CE jsou na přijímací straně **vstupními** signály.



obr. 1. Elektrické schéma zapojení pro SPI se dvěma procesory 2051

Význam jednotlivých vodičů je, z hlediska vysílací strany, následující:

- Výstupní datový vodič DOUT obsahuje v určitých časových okamžicích platná vysílaná data (jeden bit).
- Vstupní datový vodič DIN zase obsahuje platná přijímaná data (jeden bit).
- Potvrzovací vodič CLK definuje svou nástupní nebo sestupnou hranou platnost dat (bitu) na datovém vodiči pro vysílaná data a současně vyzývá druhou komunikační stranu na vložení dat na příjmové straně (platného bitu).
- Výběr obvodu CE (CS) slouží pro aktivaci periferního obvodu, zvláště při připojení více periferních obvodů.

Komentář funkce obou úseků programů

Vysílací programová rutina (IO_SPIV) není nikterak složitá a je možné ji zařadit kamkoliv do programu. Na prvním řádku je výzva k přenosu dat (signál CE jde do „L“), další tři řádky pouze uschovají obsah používaných registrů. Smyčka ***IO_SPV2: ... DJNZ...*** provede malé zpoždění, které objasníme později. Vlastní vysílání a příjem dat provádí rutina ***IO_SPIV1 ... DJNZ ...***, která se provede 8x pro vyslání (a příjem) celého bytu. V této smyčce se vyšle bit z registru Cy, potvrdí se jeho platnost (CLK jde na „L“) a po čtyřech NOPech se zase nastaví CLK na „H“. Hned další příkaz přečte přijímaný bit do Cy. Běžná rotace přes akumulátor zajistí současně vysílání i příjem všech osmi bitů. Po ukončení smyčky vysílání dat se přijatý byte z registru A uloží do příjmového registru, vrátí se uschovaná data a ukončí se výběr periferního obvodu signálem CE do úrovně „H“.

Přijímací programová rutina (IO_SPIP) **musí** pracovat pod přerušením (v tomto případě IEX1) a v době příjmu nesmí být přerušena jiným zdrojem přerušením. Je to dáno tím, že tato sekvence příkazů musí stihnout příjem a vysílání bite ve stejném taktu, jak jej bude taktovat signál CLK z vysílací strany. Po přijmu přerušení (sestupná hrana signálu IEX1) se musí program co nejrychleji dostat do rutiny obsluhy SPI, zakázat veškerá přerušení, uložit obsahy používaných registrů, připravit první bite z vysílaného byte PA_DOUT přes registr A a Cy na výstupní port a nastavit smyčku pro příjem 8 bitů. Příkaz s návěštím *A:* ***JB IO_CLK,$*** zajišťuje čekání na sestupnou hranu signálu CLK. Je to totiž okamžik, kdy vysílací strana nastavila platný bit na vodiči PA_DIN (pro vysílací stranu PA_DOUT). Příkaz s návěštím ***B: CPL P1.6*** není výkonný, ale velmi dobře nám poslouží pro ladění celého přenosu. Ten totiž způsobí změnu stavu na zvoleném portu a tím také určuje, že vyslaný bite byl přijat. Další příkazy již provedou čtení platného bitu a přes rotaci s akumulátorem vymění další bit na vysílacím portu. Po ukončení smyčky příjmu dat se uloží přijatý byte z A do příjmového registru, obnoví se používané registry a povolí se přerušení. Posledním příkazem rutiny musí být příkaz RETI jako návrat z přerušovací rutiny.

A nyní k čekací smyčce ve vysílací rutině. Na přiložených obrázcích jsou vidět časové průběhy sejmuté logickým analyzátorem. Čtyři signály linky SPI jsou popsány výše, signál ICLK je sejmut právě z pinu P1.6, kdy každý průběh příkazem s návěštím ***B: CPL P1.6*** provede změnu stavu pinu a proto každá hrana tohoto impulsu označuje načtení jednoho bitu. Na obr.2. je vidět, že po příchodu sestupné hrany signálu CE a příliš brzy započatém vysílání dat (signál CLK) nestíhá přijímací strana první bity (v tomto případě dokonce 2 bity) přečíst. Je to dáno tím, že činnost na straně příjmu je velmi časově náročná. Obr. 3. již ukazuje, že pokud signál CLK (a tím i vysílání dat) vhodně zpozdíme, stihne se přijímací strana na tento příjem včas přichystat. Obr. 2. současně ukazuje, že vysílaná data mají hodnotu 055H a přijímaná dat hodnotu 0AAH. Časové hodnoty přenosu dat jsou odvozeny od krystalu o kmitočtu 11,0592MHz. Doba zpoždění od signálu CE po počátek vysílání signálu CLK je cca 35µs a celková doba přenosu jednoho byte je 145µs.

Je tu i prostor pro experimenty. Především na straně příjmu, kde lze jednak zvýšit kmitočet krystalu a tím lze samozřejmě zkrátit kritickou dobu zpoždění mezi signály CE a CLK a také po odladění konkrétního přenosu SPI je možné odstranění příkazu s návěštím ***B: CPL P1.6*** a pak i na vysílací straně postupným odebíráním NOPů (nejméně jednoho za již zmíněný příkaz). Další možnou změnou je „vytknutí“ signálu CE před vysílací rutinu. Pak je možné vlastní rutinu volat vícekrát a tím získat 16-ti bitový přenos a ve vzniklém přenosovém „protokolu“ určit např. první byt je příkazový (command) a druhý jsou data.

Vysílací strana

```
;     vystupni signaly

IO_DOUT     BIT P1.0          ;data vysilam
IO_CE       BIT P1.1          ;CE
IO_CLK      BIT P3.7          ;hodiny


;     vstupní signaly

IO_DIN      BIT P1.2          ;data ctu


;     pametove promenne

PA_DOUT     DATA  030H        ;vysilajici data
PA_DIN      DATA  031H        ;prijimana data


;#####################################################
;       SPI vysilaci cast
;#####################################################
IO_SPIV:
            CLR  IO_CE        ;vyber obvodu
            PUSH ACC          ;uklid registru
            PUSH B
            MOV  B,#6         ;musime pockat
IO_SPV2:
            NOP               ;pro stranu prijmu
            DJNZ B,IO_SPV2

            MOV  B,#8         ;8 bitu
            MOV  A,PA_DOUT    ;data pro vysilani
            RLC  A            ;do CY
IO_SPV1:
            MOV  IO_DOUT,C    ;jeden bit ven
            CLR  IO_CLK       ;data plati po sestupne hrane
            NOP               ;casovani
            NOP
            NOP
            NOP
            SETB IO_CLK       ;cteme data po nastupni hrane
            MOV  C,IO_DIN     ;cteme bit
            RLC  A            ;rotujeme
            DJNZ B,IO_SPV1    ;opakujeme 8x

            MOV  PA_DIN,A     ;ulozime nactena data
            POP B             ;vratime registry
            POP ACC
            SETB  IO_CE       ;konec prenosu
            RET
;#####################################################
```

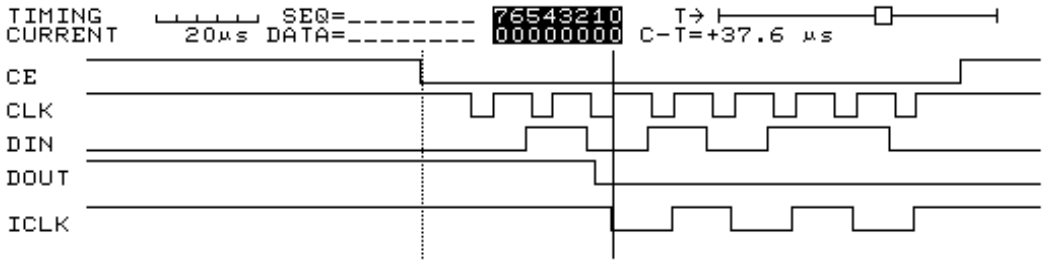
Přijímací strana

```
;     vstupni signaly

IO_DIN      BIT P1.0          ;data ctu
IO_CE       BIT P3.2          ;CE - INT1 (nebo INT0)
IO_CLK      BIT P3.7          ;hodiny

;     vystupní signaly

IO_DOUT     BIT P1.2          ;data vysilam


;     pametove promenne

PA_DOUT     DATA  030H        ;vysilajici data
PA_DIN      DATA  031H        ;prijimana data


;=====================================================
;       INTERUPT adresa pro IRQ externí interupt IEX1
;*****************************************************

            ORG  ADR+0013H
            JMP IO_SPIP
            ;                 ;RETI  je tam
```

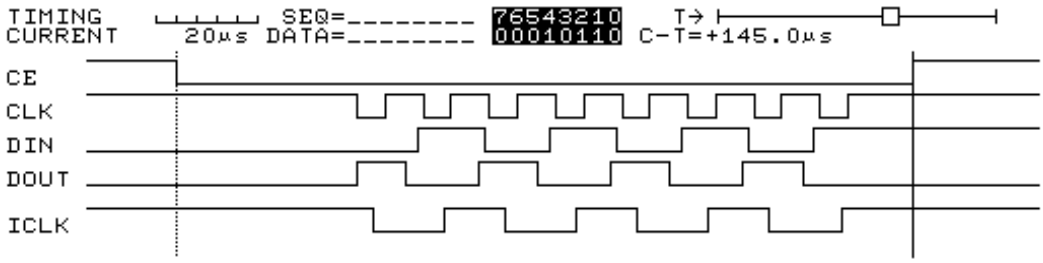
..... a někde v programu IRQ povolit a umístit tam rutinu pro obsluhu IRQ

```
;#####################################################
;       SPI vysilaci cast - volano pres IRQ !
;#####################################################
;

IO_SPIP:
            CLR  EA           ;zakaz IRQ
            PUSH ACC          ;uklid registru
            PUSH B
            MOV  B,#8         ;8 bitu
            MOV  A,PA_DOUT    ;data pro vyslani
            RLC  A            ;do CY
IO_SPP1:
            MOV  IO_DOUT,C    ;data ven
A:          JB   IO_CLK,$     ;skok pro "H"
B:          CPL  P1.6         ;pro testy = ICLK
            MOV  C,IO_DIN     ;nactem data
            RLC  A            ;posunem do A
            DJNZ B,IO_SPP1    ;opakujeme 8x
IO_RET:
            MOV  PA_DIN,A     ;ulozime
            POP  B            ;navrat registru
            POP  ACC
            SETB EA           ;povolime IRQ
            RETI
;#####################################################
```

Získané výsledky z logického analyzátoru:



obr. 2. Časový průběh situace, kdy přijímací strana „nestihne“ signál CLK.



obr. 3. Časový průběh pro vysílání byte 055H a příjem byte 0AAH