Ústav radioelektroniky
Vysoké učení technické v Brně

Komunikace zařízení po sběrnici

Mikroprocesorová technika a embedded systémy

Přednáška 4

doc. Ing. Tomáš Frýza, Ph.D.

11. října 2011

Obsah přednášky

Watchdog časovač

Základní pojmy a terminologie z mikroprocesorové techniky

Obvody s tří-stavovým výstupem

Komunikace po sběrnici

Funkce, struktura a provedení pomocných obvodů
- Napájení, hodinový signál, obvod reset

Programovatelné propojky (Fuse)

Ukázka programu v JSA a v jazyce C pro ATmega16
- Časovač watchdog

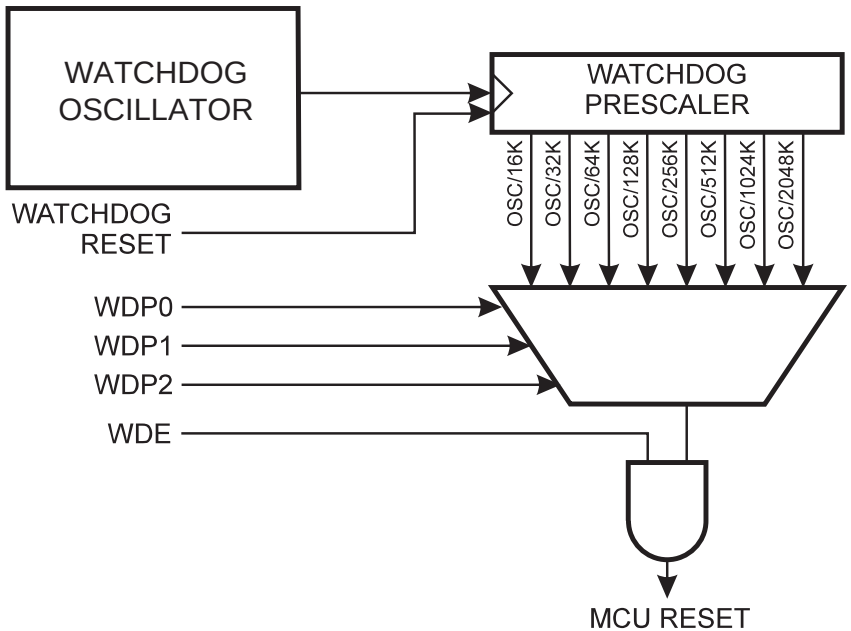
Obsah přednášky

Obvod watchdog

- Obvod watchdog je "bezpečnostní časovač". Jeho význam je vyvolat reset (a tím i znovu-nastartování) mikrokontroléru v případě, že se procesor "ztratí", příp. nevykonává program, který by měl.
- Nazávislý časovač s odděleným hodinovým systémem (frekvence je závislá na napájení; $V_{CC} = 5\,\mathrm{V} \Leftrightarrow f_{WDT} = 1\,\mathrm{MHz}$).
- Běžný chod programu: program opakovaně znovunačítá, příp. resetuje stav časovače (např. v nekonečné smyčce programu) a tím nedojde k přetečení jeho hodnoty.
- V případě "ztráty" programu, zacyklení, apod. dojde k přetečení a k vyvolání resetu.

Watchdog periférie

- Funkce watchdog časovače je běžně vypnuta, tj. není potřeba se obávat nechtěného resetu systému.
- Povolení generace resetu MCU a nastavení předděličky je řízeno řídicím registrem WDTCR (Watchdog Timer Control Register).



Obrázek: Struktura Watchdogu.

Tabulka: Příklad změny rychlosti čítání watchdogu (reg. WDTCR)

WDP2:WDP0	Přetečení ($V_{CC} = 5\,V$)
0b000	16 ms
0b001	33 ms
0b010	65 ms
0b011	130 ms
0b100	260 ms
0b101	520 ms
0b110	1 s
0b111	2,1 s

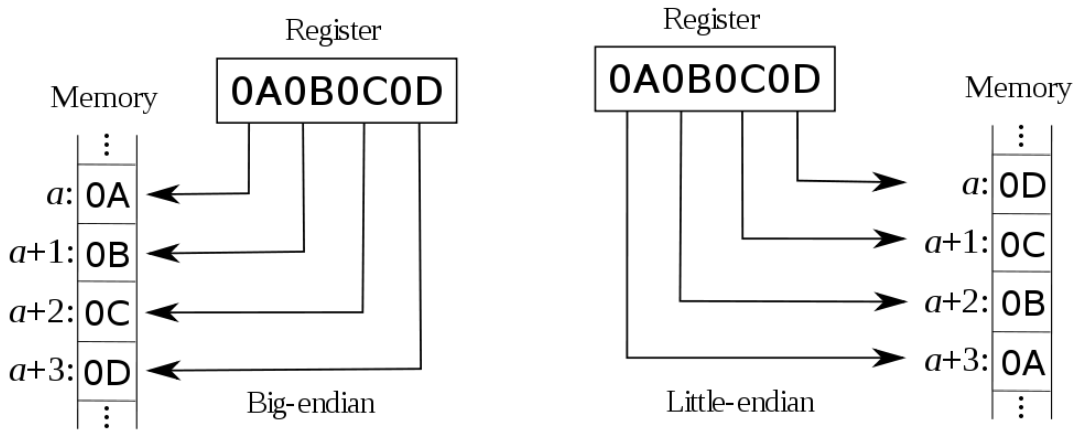
Obsah přednášky

Základní pojmy mikroprocesorové techniky

- Bit – zkratka z "Binary Digit", tj. jeden symbol v binární soustavě:
  - označení "b". Např. 8 b,
  - nastavení bitu (set bit) hodnota bitu = 1 (vysoká úroveň),
  - nulování bitu (clear bit) hodnota bitu = 0 (nízká úroveň).
- Paměťová buňka zařízení nebo elektrický obvod pro uchování hodnoty 1 bitu (klopný obvod, stav tranzistoru, jeden "spot" na CD, magnetický náboj, ...).
- Paměťové slovo množina paměťových buněk určité velikosti (registr z osmi klopných obvodů, ...); typická velikost od 4 do 64 bitů.
- Paměť součástka, umožňující uložení programu nebo dat v binární podobě. Soubor určitého počtu paměťových slov.
- Registr – skupina paměťových buněk k uložení binární informace. Kratší přístupová doba než u ostatních paměťových slov.
- Šířka registru – počet bitů v registru.
- Registrový pár – dvojice registrů.
- Pin – vodič vyvedený vně součástky.
- Port – skupina pinů (nejčastěji 8), umožňující vstupně/výstupní komunikaci mikrokontroléru s okolím.
- Sběrnice – skupina paralelních vodičů umožňující propojení vstupů a výstupů několika zařízení, registrů, apod.

Základní pojmy a terminologie

- Byte [bajt] – slovo o šířce 8 bitů; jedno z nejpoužívanějších paměťových slov v mikroprocesorové technice:
  - označení "B". Např. 16 B,
- Nibl [nibl] – slovo o šířce 4 bitů; polovina bytu; dříve hojně využíváno.
- MSB (Most Significant Bit/Byte) – nejvýznamnější bit/byte.
- LSB (Least Significant Bit/Byte) – nejméně významný bit/byte.
- Big-endian, Little-endian – pořadí ukládání bytů do paměti.



Obrázek: [Wiki].

- Paměťová kapacita – celkový počet bitů v paměti, nebo v systému. Udává se ve tvaru počet slov × šířka slova:
  - např. 8k × 16, 1k × 8, 512 × 8.
- Pozn.: Pracujeme ve dvojkové soustavě, proto:
  - 1k [kilo] $= 2^{10} = 1\,024$ (pozor: 1k $\neq 1\,000$),
  - 1M [mega] $= 2^{20} = 1\,048\,576$,
  - 1G [giga] $= 2^{30} = 1\,073\,741\,824$.

Základní pojmy a terminologie

Příklad

Co představuje zápis paměťové kapacity 4 096 × 20?

Řešení

- *Počet slov = 4 096 (4k),*
- *šířka slova, tj. počet bitů na slovo = 20 b.*

Příklad

Nechť je paměťová kapacita vyjádřena hodnotou 2k × 8

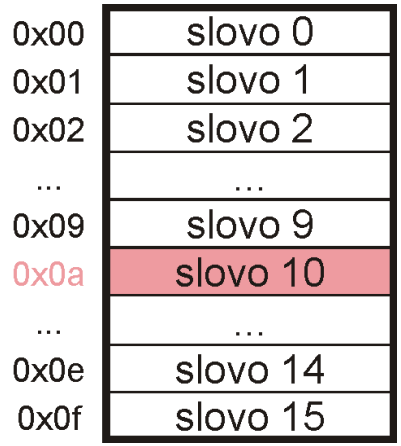
(a) *Kolik slov obsahuje tato součástka?*

(b) *Jaká je šířka jednoho slova?*

(c) *Jaký je celkový počet bitů v paměťové součástce?*

Základní pojmy a terminologie, paměti

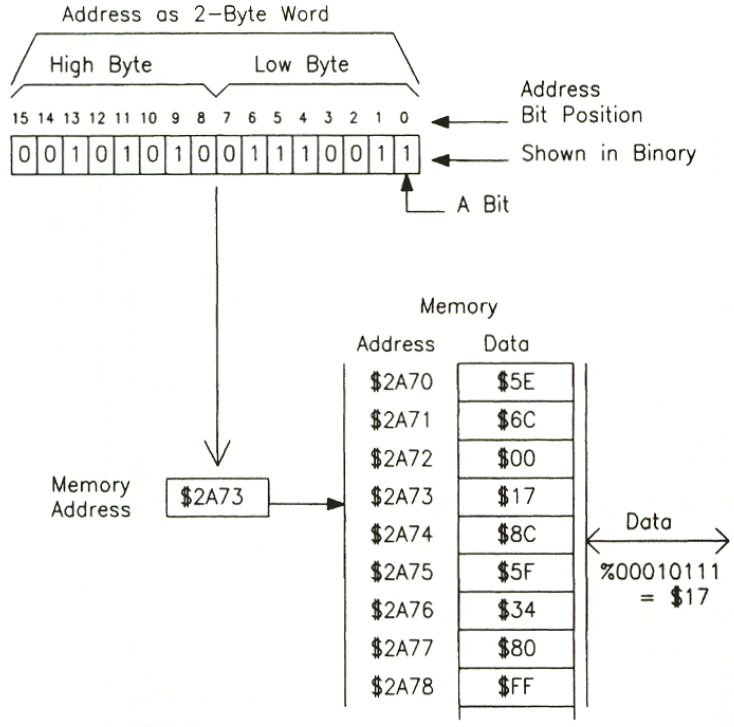
- Adresa – index, který identifikuje pozici slova v paměti; každé slovo má unikátní adresu; nejčastěji se udává v šestnáctkové soustavě (prefix 0x nebo $).
- Čtení z paměti (anglicky Fetch) – operace, při které je binární slovo (např. *slovo 10* na obrázku) vyzvednuto z konkrétní adresy v paměti (0x0A) a přeneseno na jiné místo.
- Zápis do paměti (anglicky Store) – operace, při které je nové slovo (např. *slovo 1* na obrázku) přeneseno na konkrétní pozici v paměti (0x01); původní informace je ztracena.



Obrázek: Uspořádání paměti.

Ukázka adresování paměťového slova

- Ukázka adresování 8bitového slova pomocí 16bitové adresy.
- MSB tvoří vyšší byte adresy, LSB tvoří nižší byte adresy
  - MSB: 0b0010 1010 @ $2A,
  - LSB: 0b0111 0011 @ $73.
- Všechna paměťová slova jsou stejně široká. Adresovaná data:
  - 0b0001 0111 @ $17.



Obrázek: Adresování paměťového slova.

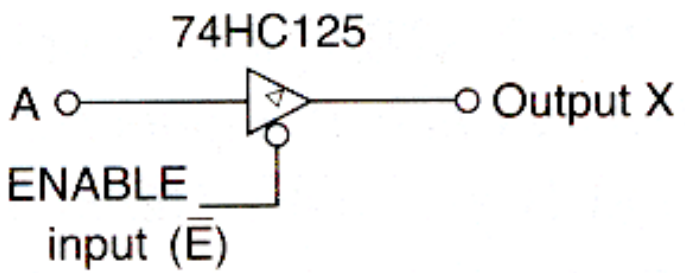
Obsah přednášky

Komunikace mezi zařízeními

- Typická komunikace uvnitř mikrokontroléru, ale i mezi mikrokontrolérem a zařízeními probíhá po paralelním vedení, tj. po sběrnici (u 8bitových systémů tvořena zpravidla 8 vodiči).
- Všechna zařízení obsahují 8bitový vstupně/výstupní port, pomocí něhož jsou připojena na jedinou skupinu vodičů sběrnici. Nejedná se tedy o spojení point-point.
- Aby byly eliminovány kolize na sběrnici, kdy několik zařízení vysílá současně – stav na sběrnici je dán kombinací více signálů; data jsou "nečitelná"– se používají systémy s tří-stavovými výstupy (Three State), příp. oddělovací bufery s třístavovým výstupem.
- Tří-stavový výstup je řízen povolovacím vstupem (Enable), který zajistí aktivaci vždy jen jednoho zařízení, výstupy ostatních jsou ve stavu vysoké impedance.

Tří-stavový výstup

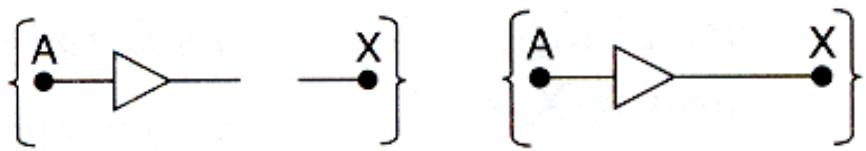
- Výstupy obsahují kromě stavu HIGH a LOW také stav vysoké impedance. Řízení režimu pomocí řídicího signálu ENABLE ($\overline{E}$).
- Ukázka: Symbolická značka části tří-stavového buferu 74HC125 (Philips):
  - pouzdro obsahuje celkem 4 takové buňky,
  - trojúhelníček znamená tří-stavový výstup,
  - malé kolečko u řídicího vstupu $\overline{E}$ informuje, že aktivní úroveň je nízká,
  - některé 3stavové buffery mají aktivní úroveň vysokou (74HC126), příp. invertují výstupní hodnotu.



Obrázek: Symbolická značka tří-stavového bufferu 74HC125.

Tří-stavový výstup, pokračování

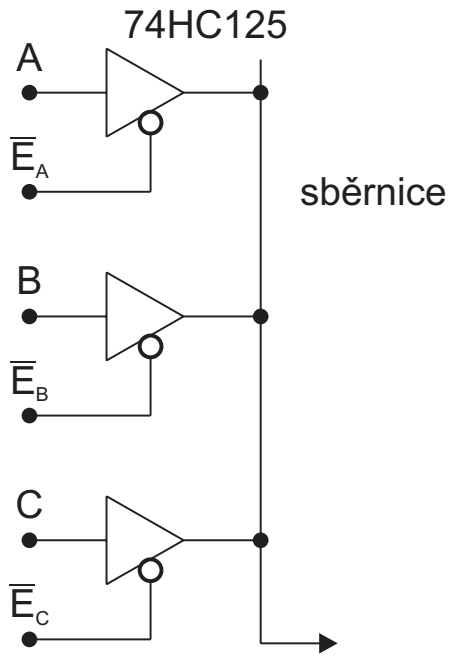
- Popis činnosti:
  - $\overline{E}$ =1; výstup ve stavu vysoké impedance – výstup se chová jakoby fyzicky odpojený od sběrnice,
  - $\overline{E}$ =0; buffer pracuje v normálním režimu, tj. výstup X je napěťový ekvivalent vstupu A.



- Tří-stavové obvody jsou obsaženy v klopných obvodech, registrech, paměťových součástkách, a téměř ve všech mikrokontrolérech.

Tří-stavový výstup, dokončení

- Tří-stavové buffery se běžně používají k připojení několika zařízení/signálů ke sběrnici.



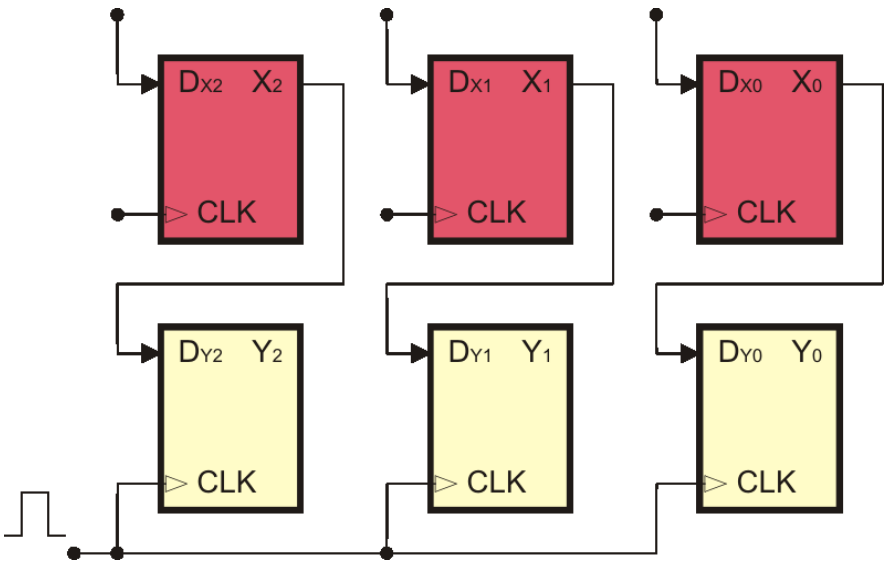
Obrázek: Tři odlišné buffery pro signály A, B a C.

- V případě vysílání více zařízení, dojde ke kolizi na sběrnici (nežádoucí). Řídicí signály $\overline{EN}$ musí zajistit, aby ke sběrnici bylo připojeno jen jedno zařízení.

Obsah přednášky

Paralelní přenos dat mezi registry

- Registry hrají významnou úlohu v mikroprocesorové technice. Ze softwarového pohledu je mikrokontrolér systémem registrů, mezi kterými je binární informace přesunována a zpracovávána.
- Existují dva způsoby přesunu dat mezi registry:
  - paralelní, sériový.

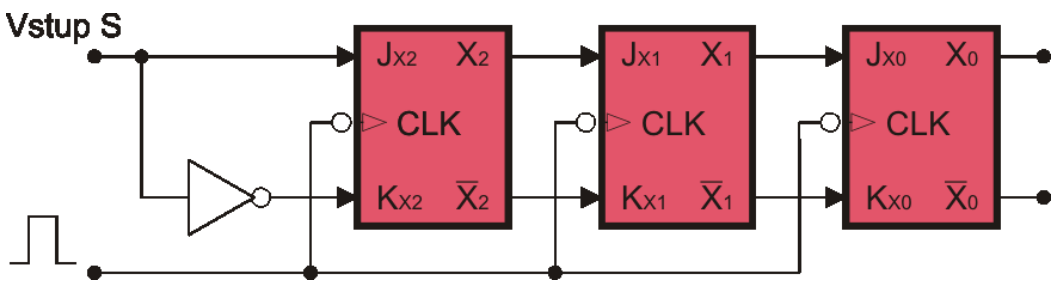


Obrázek: Paralelní přenos dat.

- Ukázka: Dvojice 3bitových registrů X a Y, které jsou tvořeny klopnými obvody typu D, řízených hranou.
- Přesun dat je řízen hodinovým signálem CLK.

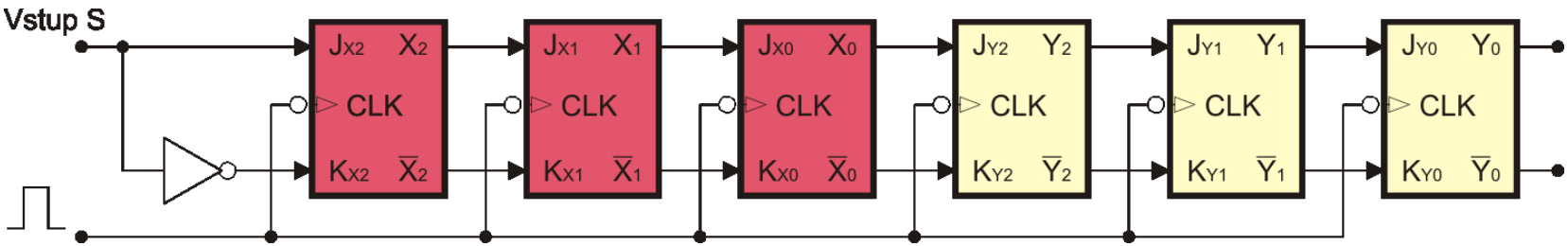
Paralelní a sériový přenos dat mezi registry

- Symbolický zápis přenosu dat: [X]→[Y] (přesun obsahu registru X do registru Y).
- Hodinový pulz aplikován na vstupy CLK způsobí přesun dat z výstupů X_2, X_1, X_0 (vstupy D_Y) na výstupy Y_2, Y_1, Y_0.
- Jedná se o paralelní přesun, protože celý obsah registru X se překopíruje do Y najednou (pomocí jedné hrany řídicího signálu).
- Ukázka: Sériový přenos dat prostřednictvím 3bitového posuvného registru (klopné obvody typu JK, řízené hranou).



Obrázek: Posuvný (shift) registr.

Sériový přenos dat mezi registry



Obrázek: Sériový přenos dat.

- Při aktivní hraně signálu CLK jsou hodnoty vstupních signálů klopných obvodů překopírovány na výstupy (tj. na vstupy obvodů vpravo).
- Data z výstupu posledního KO jsou ztracena. Sériový přenos vyžaduje více času (jeden bit vyžaduje jeden pulz), ale méně vodičů.

Sériový přenos dat mezi registry, příklad

Příklad

Přenos dat [X]→[Y]; nechť [X]=101, [Y]=011 a S = 0

Řešení

1. *pulz: [X]=010, [Y]=101*
2. *pulz: [X]=001, [Y]=010*
3. *pulz: [X]=000, [Y]=101*

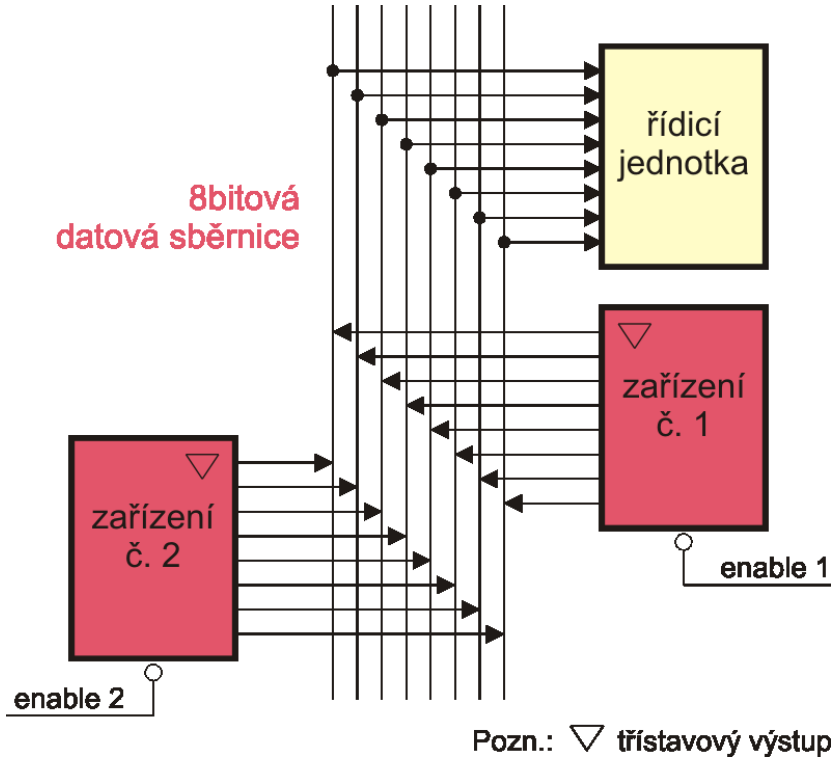
Tabulka: Postupné posouvání dat.

S	X_2	X_1	X_0	Y_2	Y_1	Y_0	
0	1	0	1	0	1	1	před 1. pulzem
0	0	1	0	1	0	1	po 1. pulzu
0	0	0	1	0	1	0	po 2. pulzu
0	0	0	0	1	0	1	po 3. pulzu

Komunikace mezi zařízeními

Příklad
Jakým způsobem zajistit bezpečný přenos dat ze zařízení č. 2 do řídicí jednotky? (Nechť řídicí jednotka v tomto případě shromažďuje informace od jednotlivých zařízení).

Příklad
Jaké napěťové úrovně budou na datové sběrnici v případě, že všechna zařízení budou ve stavu vysoké impedance?



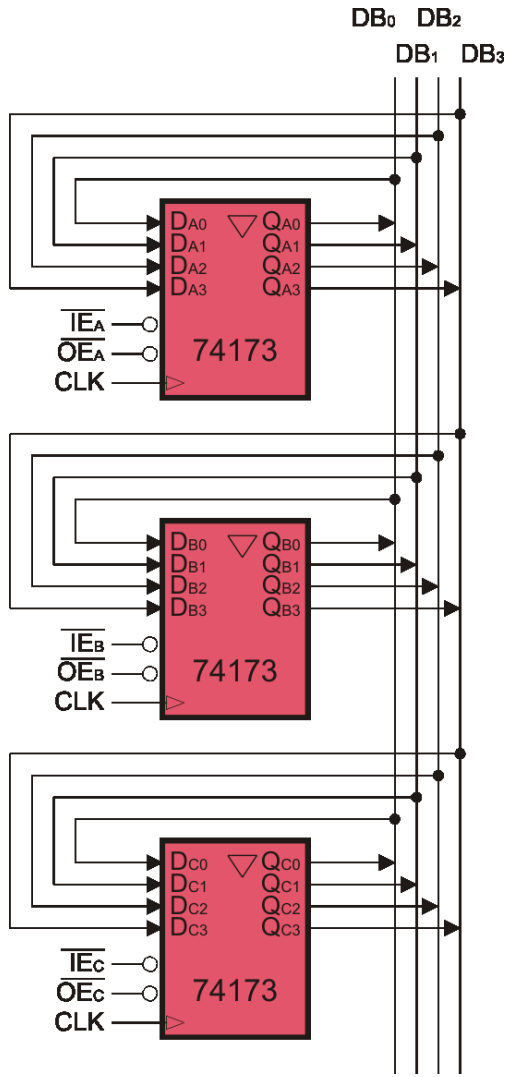
Obrázek: Komunikace po sběrnici.

Příklad přenosu dat po sběrnici

- Příklad zapojení tří 4bitových registrů 74173 (4bitový 3stavový klopný obvod typu D) na 4bitovou sběrnici, tvořenou vodiči DB_3–DB_0.
- Registry obsahují tři řídicí signály:
  - $\overline{IE}$ (Input Enable) – nízká úroveň aktivuje vstupní obvody,
  - $\overline{OE}$ (Output Enable) – nízká úroveň aktivuje výstupní obvody,
  - CLK – hodinový signál.

Příklad

Zajistěte přenos dat z registru B do registru A, [B]→[A]. Nechť [B] = 1100.



Obrázek: 4bitové registry 74173.

Příklad přenosu dat po sběrnici, řešení

Řešení

- *Výstupy:*
  - *pouze registr B musí mít aktivní výstupní obvody, tj.* $\overline{OE}_A = 1$, $\overline{OE}_C = 1$, $\overline{OE}_B = 0$,
  - *obsah registru B je tím zapsán na datovou sběrnici.*
- *Vstupy:*
  - *pouze registr A musí mít aktivní vstupní obvody, tj.* $\overline{IE}_A = 0$, $\overline{IE}_B = 1$, $\overline{IE}_C = 1$.
- *Následující aktivní hrana hodinového signálu zajistí přenos (přečtení) dat ze sběrnice do klopných obvodů registru A.*
- *Přenos dat je tedy zajištěn generováním korektní posloupností řídicích signálů – zajišťuje řídicí jednotka.*

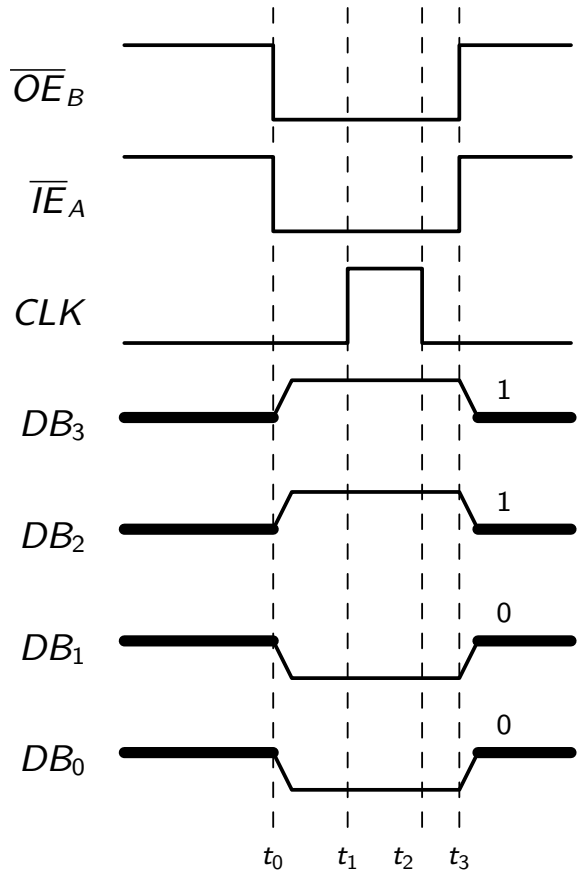
Příklad přenosu dat po sběrnici, řešení

Řešení

- *Platná data jsou v registru A přeneseny v časovém intervalu $t_1 - t_2$.*

Příklad

Proč je aktivní úroveň řídicích signálů v nízké úrovni?



Obrázek: Časové průběhy na řídicí a datové sběrnici.

Rozšíření sběrnice

- Běžně se používá větší počet registrů než 3. Obecně lze na sběrnici připojit libovolný počet registrů. Zvyšuje se tím ale počet řídicích signálů $\overline{OE}$, $\overline{IE}$ a vstupně/výstupních vodičů.
- Obdobně jako u 4bitové sběrnice fungují sběrnice 8bitové, 16bitové, nebo 32bitové.
- Zobecněný princip:
  - jeden registr (paměťová buňka, zařízení) má aktivované výstupní obvody a zapisuje data na sběrnici,
  - jiné zařízení má aktivované vstupní obvody, čímž může data ze sběrnice číst,
  - při aktivní hraně hodinového signálu dojde k zápisu dat do registru – synchronní řízení.
- Šířka sběrnice (počet vodičů) je dána šířkou datového slova, přenášeného v systému. 8bitové slovo $\Rightarrow$ 8 vodičů, 16bitové slovo $\Rightarrow$ 16 vodičů, ... Některé systémy obsahují kombinací několika různě-širokých sběrnic.
- Počet zařízení, připojených na sběrnici je rozdílný případ od případu. Závisí na velikosti paměti systému a na počtu vstupních a výstupních zařízeních, které musí s CPU komunikovat po sběrnici.
- Všechna zařízení musí být připojena přes 3stavové buffery:
  - některá zařízení tyto buffery již obsahují přímo na čipu, např. registry 74173 (viz výše),
  - ostatní zařízení musí být ke sběrnici připojeny přes tzv. řadič sběrnice (Bus Driver).

Řadič sběrnice/Bus Driver

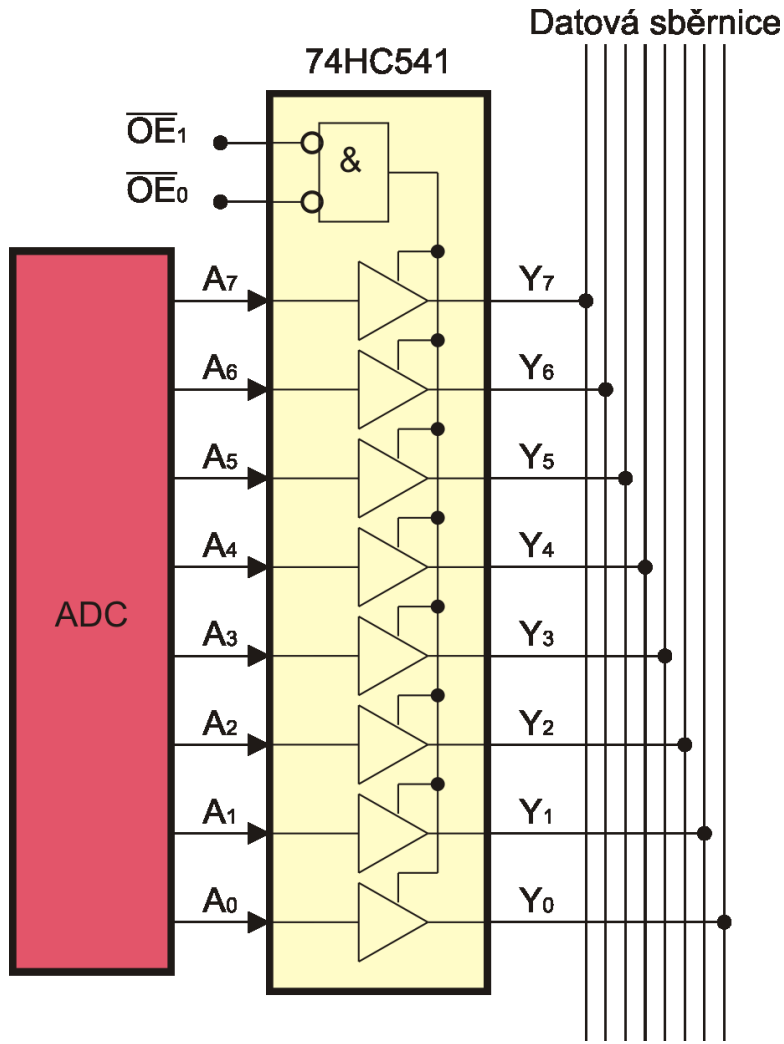
- Řadič sběrnice obsahuje 3stavový výstup s velmi nízkou výstupní impedancí, což způsobuje rychlé nabíjení a vybíjení celkové kapacitní reaktance (odpor) sběrnice $\Rightarrow$ krátké doby přechodu. (Řadič sběrnice tedy nezhoršuje doby přechodu.)
- Kapacitní odpor je tvořen kumulací parazitních kapacitancí všech vstupů a výstupů, připojených ke sběrnici a může zhoršit doby přechodu.
- Příklad řadiče sběrnice – 8bitový 74HC541.

Využití řadiče sběrnice

- Připojení výstupu 8bitového A/D převodníku k datové sběrnici pomocí 74HC541.

Tabulka: Funkční tabulka.

$\overline{OE}_1$	$\overline{OE}_0$	A_n	Y_n
0	0	0	0
0	0	1	1
1	X	X	Z
X	1	X	Z

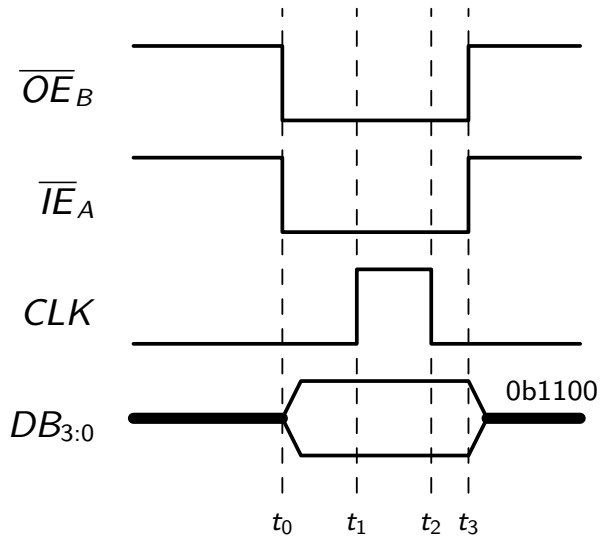


Obrázek: Řadič sběrnice

Zjednodušená reprezentace časových průběhů

- Časové průběhy jednotlivých signálů na sběrnici lze z důvodu přehlednosti zápisu sdružit do jediného průběhu – výhodné u systémů s 8, 16, nebo s 32 vodiči.

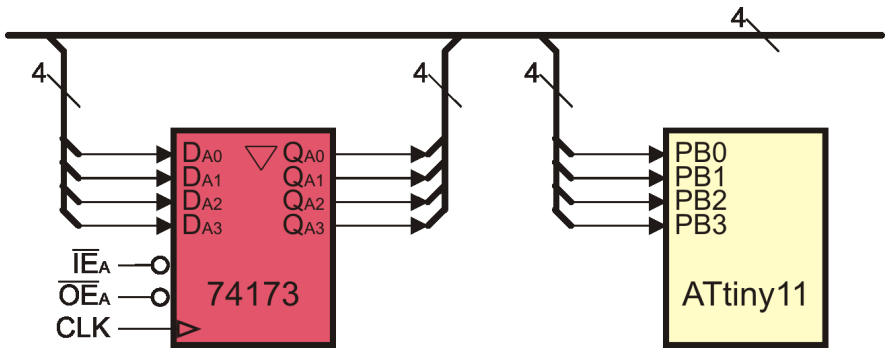
- Ukázka: Přenos dat ze zařízení B do A, viz předešlý případ [B]=1100:
  - $\overline{OE}_B = 0$; $\overline{IE}_A = 0$,
  - ostatní řídicí vstupy/výstupy nejsou zakresleny.
- Platná data na sběrnici v době trvání aktivní úrovně hodinového impulsu (interval $t_2 - t_1$).
- Neplatná data v intervalech $t_1 - t_0$ a $t_3 - t_2$.



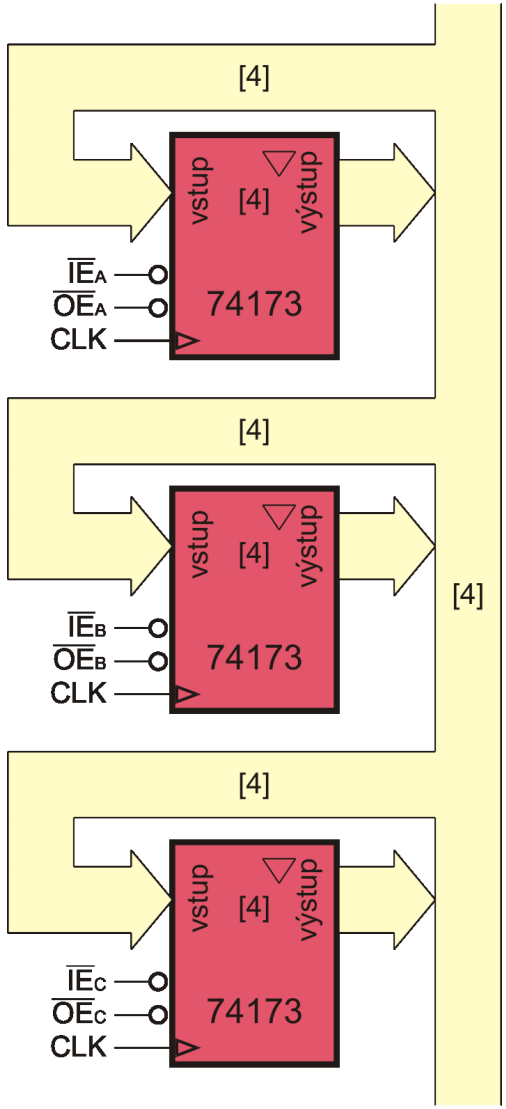
Obrázek: Zjednodušené časové průběhy na datové sběrnici.

Zjednodušená reprezentace sběrnic

- Z důvodu připojení velkého množství zařízení ke sběrnici se používá zjednodušené znázornění vodičů ve schématech:
  - pomocí širokých šipek; číslo v [·] znázorňuje šířku registrů/počet připojených vodičů,
  - pomocí jediného vodiče (fyzicky nespojeno) s indikovanou šířkou (počtem bitů) \4.



Obrázek: 4bitový registr 74173 a MCU ATtiny11.

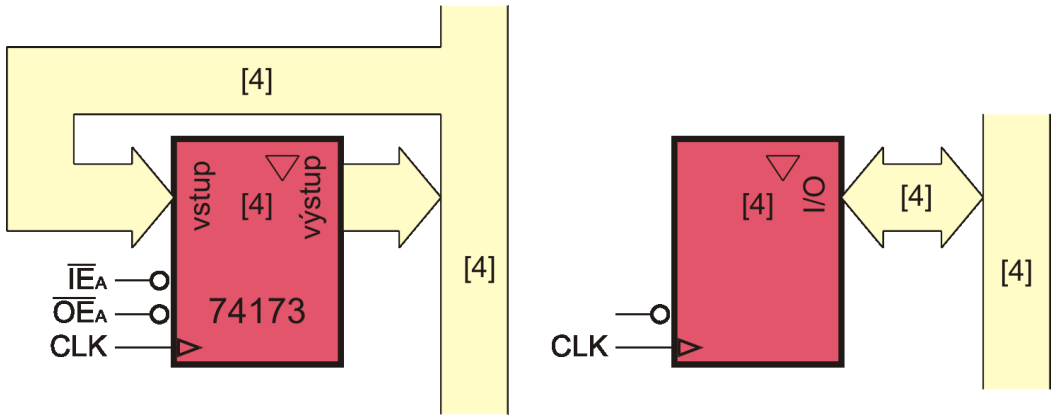


Obrázek: Zjednodušené znázornění sběrnice.

Obousměrná sběrnice

- Uvažovaný obvod (registr 74173) má vstupy i výstupy napojené na shodný vodič datové sběrnice, např. vstupní signál D_{C0} je propojen s výstupem O_{C0} přes vodič DB_0 (neplatí při použití řadiče sběrnice):
  - D_{C0} – vstup č. nula registru C,
  - O_{C0} – výstup č. nula registru C.
- Z důvodu snížení počtu pinů, bylo vyvinuto uspořádání s propojením vstupů a výstupů uvnitř integrovaných obvodů.
- Každý pin (I_0/O_0 až I_n/O_n) může pracovat jako vstupní nebo výstupní v závislosti na řídicích signálech. Proto jsou piny I/O nazývány jako obousměrné datové linky.
- Mnoho paměťových obvodů a mikroprocesorů samozřejmě umožňuje obousměrný přenos.

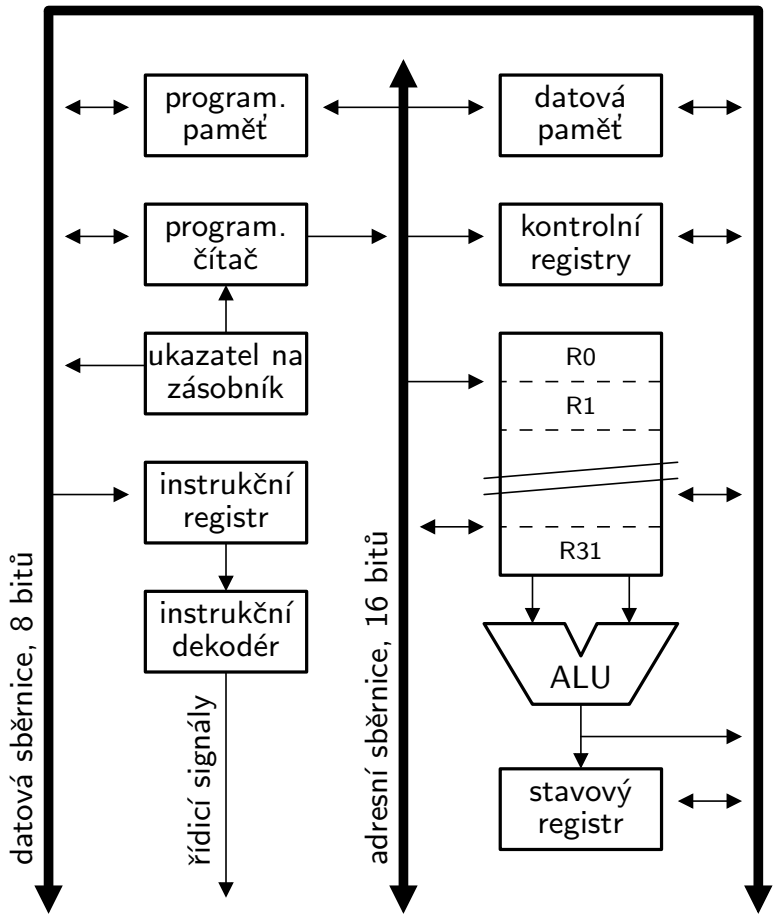
Obousměrná sběrnice



Obrázek: Propojení I/O pinů uvnitř obvodu.

Adresní, datová, řídicí sběrnice

- V mikroprocesorových systémech se obecně vyskytují tři typy sběrnic.
- Adresní sběrnice: přenáší n-bitovou adresu z mikroprocesoru do paměti:
  - mikrokontroléry AVR obsahují 16bitovou adresní sběrnici, tj. mohou adresovat $2^{16} = 65\,536$ pozic v paměti nebo ve vstupně/výstupním zařízení,
  - čtení/zápis z/do paměti musí být doprovázen příslušným řídicím signálem.
- Datová sběrnice: obousměrná 8bitová sběrnice (u 8bitových systémů) pro přenos dat do/z paměti.
  - Datová sběrnice nemusí obsahovat jen "data", ale také instrukce, které mají být vykonány.
- Řídicí sběrnice je skupina všech časovacích a řídicích signálů, nutných pro korektní synchronizaci operací mikroprocesoru s ostatními součástmi systému.
  - Některé řídicí signály jsou výstupem mikroprocesoru a některé jsou vstupy od I/O zařízení. Počet a typ signálů se liší,
  - Mezi základní signály patří READ/WRITE, generovaný mikroprocesorem a obsahují ho všechny mikroprocesory (příp. jeho ekvivalent),
  - Signál Enable.

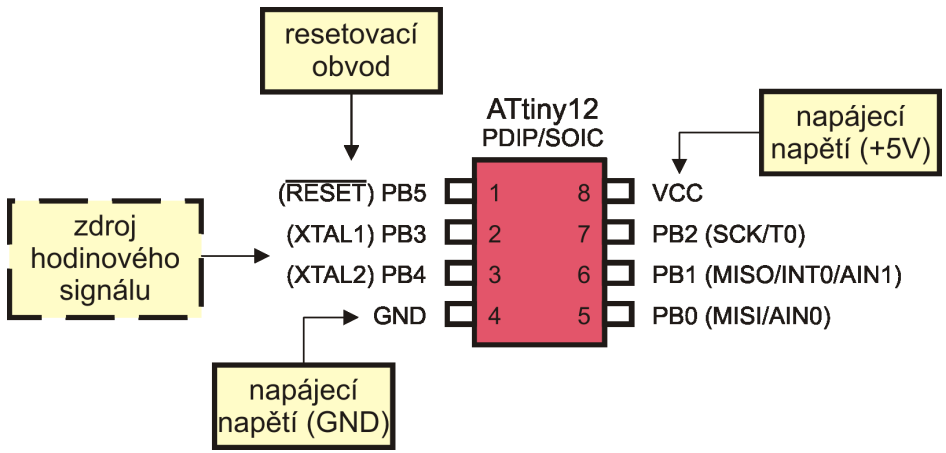


Obrázek: Zjednodušené blokové schéma MCU typu AVR.

Obsah přednášky

Pomocné obvody řídicí aplikace

- Není vždy nutné využívat výkonné mikrokontroléry. U jednoduchých aplikací postačí např. ATtiny12 (pouzdro PDIP8).
- Jednoduché řídicí aplikace s mikrokontroléry vyžadují (po hardwarové stránce):
  - zdroj napájení,
  - zdroj hodinového signálu,
  - resetovací obvod.



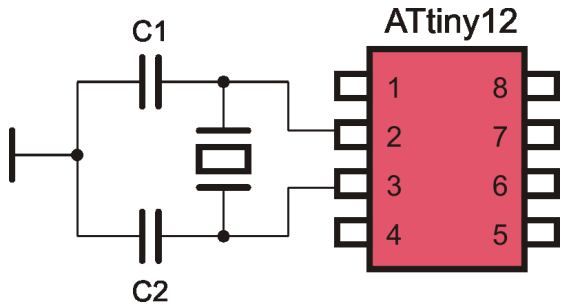
Obrázek: Blokové zapojení řídicí aplikace.

Pomocné obvody řídicí aplikace, pokračování

- Typ napájení závisí na mobilní/desktopové aplikaci:
  - napájení z baterií, z rozvodné sítě s transformací, z datových portů PC (RS-232, USB), pomocí solárních panelů, . . .,
- hodinový signál procesoru řídí časování AVR jádra. Od něho jsou odvozeny časovací signály pro všechny periférie (Flash, A/D převodník, I/O modul, čítač, . . .),
- obecné možnosti generování hodinového signálu:
  - externí zdroj hodinového signálu na vstupním pinu XTAL1,
  - vnější RC oscilátor: RC oscilátor způsobuje nízkou časovou přesnost,
  - interní RC oscilátor: nízká časová přesnost; běžné hodnoty jsou 1, 2, 4 a 8 MHz, závislost přesnosti oscilací na napětí, nejsou potřeba žádné externí součástky.

Pomocné obvody řídicí aplikace, dokončení

- Možnosti generování hodinového signálu, pokračování:
  - interní hodinový generátor s externím krystalem: nejrozšířenější varianta; vstupy XTAL1 a XTAL2 představují vstup a výstup invertujícího zesilovače oscilátoru; vyžaduje jen dva kondenzátory (typické hodnoty od 12 pF do 22 pF).



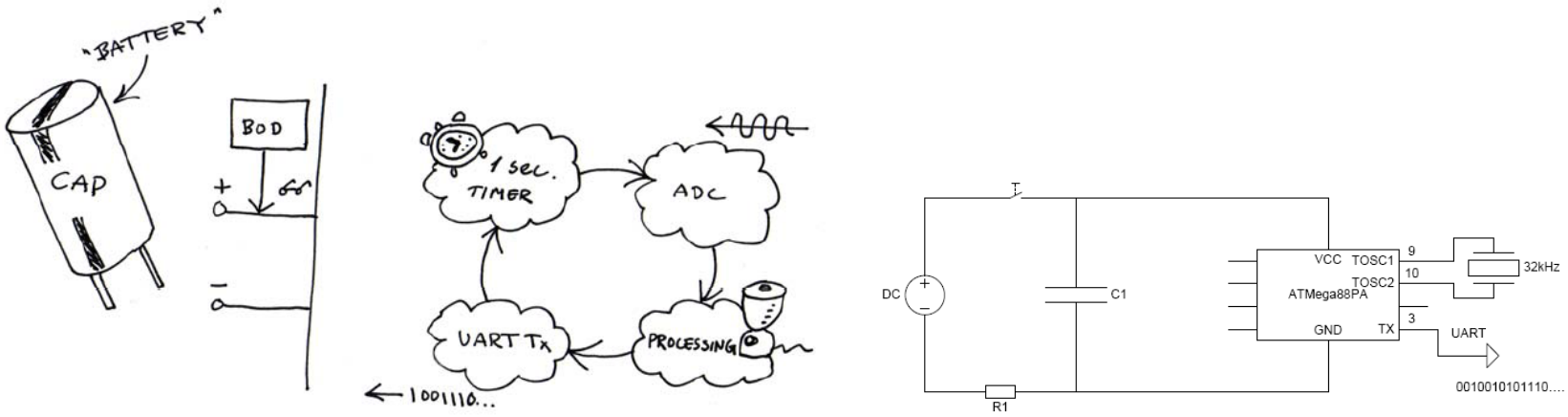
Obrázek: Externí krystal.

Pozn.: Zdroj hodinového signálu je vybrán na základě tzv. programovatelné propojky (viz dále). Propojky je možné přeprogramovat.

ATmega16 je dodáván s přednastaveným interním RC oscilátorem o frekvenci 1 MHz.

Snižování spotřeby

- Vhodnou změnou softwaru lze snížit proudový odběr; vyšší životnost baterií.



Obrázek: Testovací aplikace s ATmega88PA [AVR4013].

- Viz např. Application Note AVR4013: picoPower Basics:
  - jeden A/D převod,
  - simulace 1000 fází "zpracování signálu",
  - převod hodnoty na ASCII řetězec,
  - vyslání dat přes UART,
  - a znovu . . .

Snižování spotřeby

Tabulka: Doba trvání testovací aplikace.

Metody redukce spotřeby	Životnost baterie
Žádná	6 s
Ošetření nepoužitých pinů a interních periférií	9 s
Předdělička hodinového signálu	40 s
Režimy snížené spotřeby	198 s
Redukce doby v aktivním módu	217 s

(1) $f_{CPU} = 8$ MHz, UART 19 200 Bd, polling (opakované dotazování) časovače 2, zda uplynula 1 s.

(2) Použití pull-up rez. u nevyužitých pinů – nepřeklápí se; odpojení napájení od nevyužívaných periférií (SPI, TWI, Timer0, Timer1).

(3) Snížení frekvence $f_{CPU} = 2$ MHz, UART 19 200 Bd.

(4) Během čekání na další zpracování $\rightarrow$ sleep mód (idle, sleep, power save).

(5) Softwarové kalibrování oscilátoru pro zvýšení přenosové rychlosti 115,5 kbps; předdělička CPU zůstává.

Snižování spotřeby – ukázka některých funkcí avr-libc

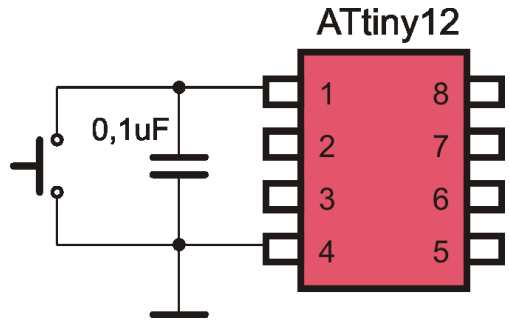
```
1  #include <avr/io.h>              // header file for ATtiny44A microcontroller
2  #include <avr/sleep.h>           // sleep_enable(), sleep_disable(), sleep_cpu()
3  #include <avr/power.h>           // see google: ``avr libc power''
4  ...
5
6      power_timer0_disable() ;     // turn off timer/counter0
7      power_timer1_disable() ;     // turn off timer/counter1
8      power_adc_disable() ;        // turn off ADC
9      power_usi_enable() ;         // turn on usi (Universal Serial Interface)
10
11     // prescale system clock from 8 MHz to 500 kHz
12     clock_prescale_set( clock_div_16 ) ;
13
14     // set sleep mode to Power-save
15     set_sleep_mode( SLEEP_MODE_PWR_SAVE ) ;
16
17     // enable sleep (possible to put the device into sleep mode when executing↩
           the sleep instruction)
18     sleep_enable() ;
19
20     sleep_bod_disable() ;        // disable BOD while in sleep mode
21     sleep_cpu() ;                // execute sleep instruction
```

Reset systému

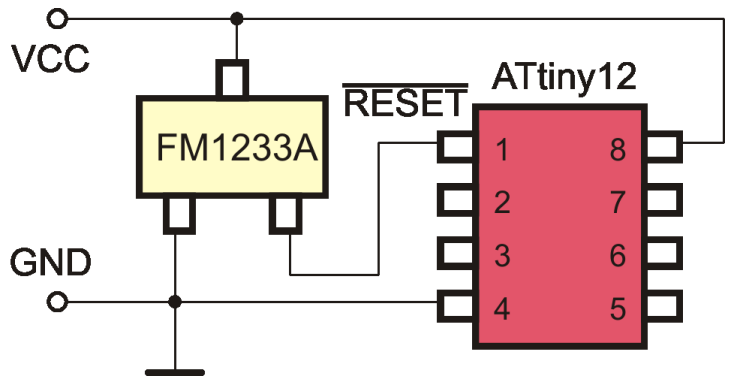
- Přerušení reset má nejvyšší prioritu může přerušit již běžící obsluhy všech přerušení, nebo vykonávající program.
- Události vyvolávající reset systému:
  - úroveň na resetovacím pinu mikrokontroléru je nižší než práh V_{POT} (Power-on reset threshold),
  - externí reset: přítomnost nízké úrovně na vstupu $\overline{Reset}$
  - přetečení ochranného časovače watchdog (musí být povoleno),
  - napájecí napětí má nižší úroveň než práh V_{BOT} (Brown-out reset threshold) (musí být povoleno),
  - reset pomocí JTAG rozhraní.
- Reset způsobí opětovný výkon programu od adresy `0x0000` (reset vektor) a slouží k nastavení mikrokontroléru do definovaného (známého) stavu, tj.:
  - všechny periférie (včetně watchdogu) jsou odpojeny,
  - všechny paralelní I/O porty jsou nastaveny jako vstupní,
  - všechna přerušení jsou vypnuta.

Generování externího resetu

- Procesory mají kromě vnitřního resetovacího obvodu, který je závislý na napájecím napětí, také externí resetovací obvod:
  - pomocí tlačítka,
  - pomocí generátorů resetovacího signálu, kdy je napájecí napětí monitorováno a při jeho poklesu je generován reset impulz. Např.: FM1233A (Fairchild Semiconductor), ADM709 (Analog Devices), . . .,
  - ošetření stisku je hardwarově realizováno uvnitř mikrokontroléru, nebo unvitř externího generátoru.



Obrázek: Tlačítko.



Obrázek: Externí generátor reset.

Obsah přednášky

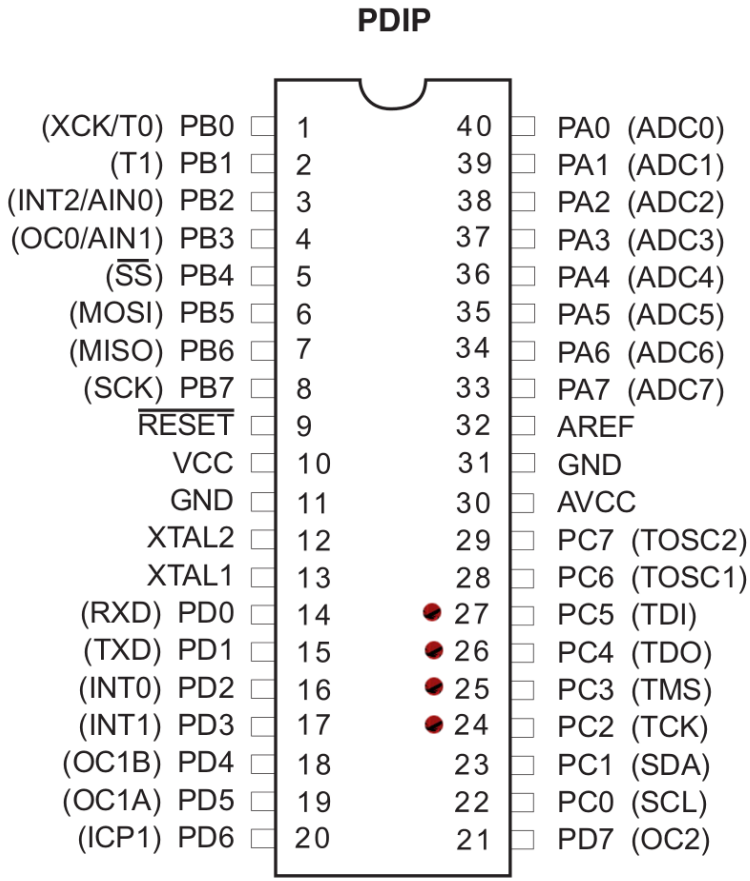
Programovatelné propojky

- Programovatelné propojky (anglicky: Fuse bits) upřesňují použití některých periferií. ATmega16 obsahuje dva byty těchto propojek:
  - povolení/zakázání ladění na čipu, použití rozhraní JTAG (programování + ladění aplikací), použití sériového programování, určení velikosti bootovací sekce ve Flash, volba start-up prodlevy, volba hodinového zdroje, ...

Fuse	**Popis funkce**
OCDEN	Povolení ladění přímo na čipu.
JTAGEN	Povolení JTAG rozhraní.
SPIEN	Povolení sériového programování ISP.
CKOPT	Clock options.
EESAVE	Možnost uchování EEPROM dat i po smazání čipu.
BOOTSZ1:0	Velikost bootovací sekce ve Flash (128 až 1 024 slov).
BOOTRST	Vektory přerušení do bootovací sekce.
BODLEVEL	Výběr úrovně detekce výpadku napájení (brown-out).
BODEN	Povolení detekce výpadku napájení.
SUT1:0	Výběr start-up prodlevy (0; 4,1 ms, 65 ms).
CKSEL3:0	Výběr zdroje hodinového signálu.

- Pozn.: AVR Fuse Calculator (`http://www.engbedded.com/fusecalc/`).

Umístění JTAG (Joint Test Action Group)



Obrázek: Pouzdro mikrokontroléru ATmega16 – umístění periférie JTAG.

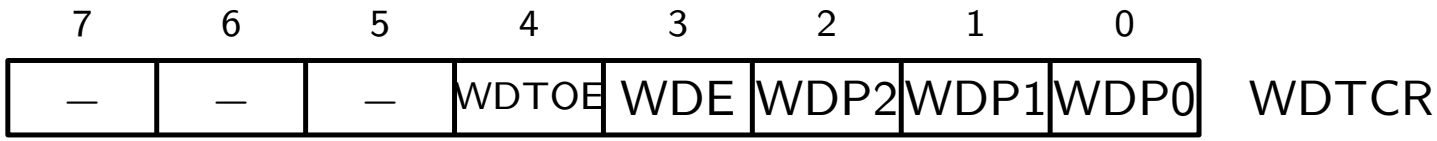
Bootovací část paměti Flash

- Programová paměť Flash je rozdělena na dvě části:
  - aplikační část (anglicky: Application section): k uložení aplikace, která se má vykonávat,
  - bootovací část (anglicky: Boot loader section): zde může být uložen program, který umožňuje přeprogramování aplikační (i bootovací) části Flash.
- Bootloader umožňuje komunikovat s "okolním světem" pomocí dostupných protokolů (UART, SPI, I2C, ...) a také umožňuje měnit obsah aplikační části programové paměti.
- Využití: programátor (propojení PC softwaru a prog. paměti MCU) – aktualizace firmwaru elektronických zařízení.
- Velikost bootovací části je nastavitelná pomocí propojky BOOTSZ1:0 na hodnoty 128, 256, 512 nebo 1 024 paměťových slov.
- Je možnost přesunout vektory přerušení do bootovací části.

Obsah přednášky

Časovač watchdog v JSA

- Interní (”hlídací”) časovač watchdog je nutné opakovaně nulovat – typicky v nekonečné smyčce. Při přetečení způsobí reset systému.
  - Nulování pomocí instrukce `WDR` – Watchdog Reset.
- Přetečení může být způsobeno pomalou obsluhou přerušení/podprogramů, nebo zacyklením vně nekonečné smyčky.
- Ovládání prostřednictvím WDTCR (Watchdog Timer Control Register):
  - WDE – Watchdog Enable,
  - WDP – Watchdog Timer Prescaler.



Obrázek: Struktura kontrolního registru WDTCR (Watchdog Timer Control Register).

Časovač watchdog v JSA

```
1  .include <m16def.inc>      ; popisný soubor ATmega16
2  .cseg                      ; paměťový segment Flash
3  .org 0x0000                ; ulož od adresy 0x0000
4      RJMP reset             ; skoč na návěští reset
5  .org 0x0002                ; vektor přerušení INT0
6      RJMP tlacitko          ; skoč na návěští tlacitko
7
8  .org 0x002A                ; ulož od adresy 0x002A
9  reset:
10     ...                    ; definice zásobníku
11     ...                    ; naplnění použitých I/O registrů
12     LDI R16, (1<<WDE) | (1<<WDP2) | (1<<WDP1) | (1<<WDP0)
13     OUT WDTCR, R16         ; povolení watchdogu s předděličkou @ 2,1 s
14     SEI                    ; globální povolení přerušení
15
16 loop:                      ; nekonečná smyčka
17     WDR                    ; nulování watchdogu
18     RJMP loop              ; skok na návěští loop
19
20 tlacitko:
21     ...                    ; obsluha přerušení INT0
22     RJMP tlacitko          ; CHYBNÉ zacyklení
23     RETI                   ; ukončení obsluhy přerušení
```

Časovač watchdog v jazyce C

- Vložení instrukce AVR do zdrojového kódu v jazyce C bez nutnosti předávání parametrů:
  - funkce `asm()` ;
  - parametrem je řetězec obsahující posloupnost instrukcí, které se mají přímo vložit do překompilované aplikace,
  - jednotlivé instrukce lze oddělovat `\n`:
    Např.: `asm( "WDR\nDEC R16") ;`

Časovač watchdog v jazyce C

```
1  #include <avr\io.h>          // hlavičkový soubor pro MCU
2  #include <avr\interrupt.h>   // hlavičkový soubor pro přerušení
3
4  ISR( INT0_vect ){            // obsluha externího přerušeni INT0
5      ...                      // kód obsluhy přerušení
6      while( 1 ) ;             // CHYBNÉ zacyklení
7  }
8
9  int main( void ){
10     ...                      // kód hlavní funkce
11
12     // povolení watchdogu s předděličkou @ 2,1 s
13     WDTCR |= (1<<WDE) | (1<<WDP2) | (1<<WDP1) | (1<<WDP0) ;
14
15     sei() ;                  // povolení globálního přerušení
16     while( 1 ){              // nekonečná smyčka
17         asm( "WDR" ) ;       // nulování watchdogu
18     }
19     return( 1 ) ;            // výstupní hodnota funkce = 1
20 }
```