

REV - Aplikace embedded systémů v mechatronice

úvodní proslov

Zdeněk Matěj

v2.0 (February 6, 2022)

O čem to bude:

- 1 O předmětu
- 2 "Embedded" systémy a základní koncepty
- 3 Mikrokontrolery a periferie

Základní koncepty embedded



Embedded systém

Jednouúčelový systém, ve kterém je řídicí počítač zcela zabudován do zařízení, které ovládá. Na rozdíl od univerzálních počítačů, jako jsou osobní počítače, jsou zabudované počítače většinou jednouúčelové, určené pro předem definované činnosti. Vzhledem k tomu, že systém je určen pro konkrétní účel, mohou tvůrci systém při návrhu optimalizovat pro konkrétní aplikaci.

Příklady:

bankomat, kalkulačka, pračka, meteostanice, navigační systém rakety, televize ...

Pozn: lepší, avšak krkolomnější české označení je "vestavěný" či "zabudovaný" systém.

Základní konstrukční prvky digitálních systémů

- procesor obsahuje množství tranzistorů
- tranzistory tvoří tzv. logické členy, které realizují logické funkce
- vhodnou kombinací (zpětná vazba) lze vytvořit paměťový člen
- z primitivních prvků se skládají složitější systémy

Logické členy – hradla

Logic Symbol

Truth Table

Elementary Elements

input

A

output

C

A

C

0

1

1

0

A

B

C

A

B

C

0

0

0

0

1

0

1

0

0

1

1

1

A

B

C

A

B

C

0

0

0

0

1

1

1

0

1

1

1

1

A

B

C

A

B

C

0

0

1

0

1

1

1

0

1

1

1

0

A

B

C

A

B

C

0

0

1

0

1

0

1

0

0

1

1

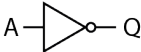
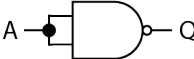

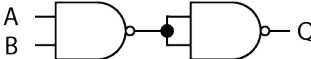

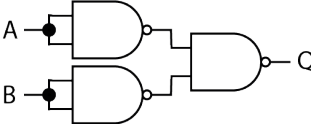
0

- NOT - negace:
 $C = \neg A = \bar{A}$
- AND - logický součin:
 $C = A \cdot B = AB$
- OR – logický součet:
 $C = A + B$

Funkčně kompletní:

- **NAND – NOT AND** $C = \overline{AB}$
- **NOR – NOT OR** $C = \overline{A + B}$

Funkčně kompletní NAND logika

| Logická funkce | Symbol funkce | Realizace funkce pomocí hradel NAND |
|----------------|---|--|
| NOT |  |  |
| AND |  |  |
| OR |  |  |

Z hradel NAND lze realizovat libovolnou logickou funkci.
Libovolný obvod lze popsat sadou booleovských funkcí.

Booleovská algebra

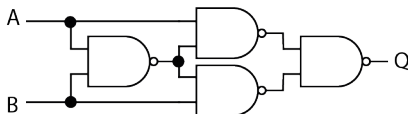
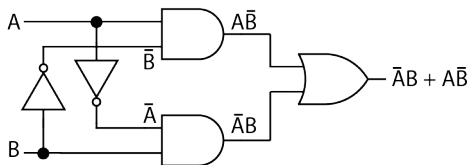
- Komutativnost: $a + b = b + a$; $ab = ba$
- Asociativita: $(a + b) + c = a + (b + c)$
- Absorpce: $a + (ab) = a$; $a(a + b) = a$
- Idempotence: $aa = a$; $a + a = a$
- Komplementarita: $a\bar{a} = 0$; $a + \bar{a} = 1$
- Distributivita: $a(b + c) = ab + ac$; $a + bc = (a + b)(a + c)$
- Ohraničenost: $0a = 0$; $1a = 1$; $0 + a = a$; $1 + a = 1$
- Dvojitá negace: $\overline{\overline{A}} = A$
- De Morganovy zákony: $\overline{A \cdot B} = \bar{A} + \bar{B}$; $\overline{A + B} = \bar{A} \cdot \bar{B}$
V kombinaci pak: $A \cdot B = \overline{\bar{A} + \bar{B}}$; $A + B = \overline{\bar{A} \cdot \bar{B}}$

Příklad: Exkluzivní součet - XOR

Realizace funkce: pravdivostní tabulka, booleovská rovnice, obvod z obecných hradel, obvod z hradel NAND

| A | B | Q |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

$$Q = A\bar{B} + \bar{A}B$$



Převod mezi intuitivní rovnicí a NAND formou: opakovaně využity De Morganovy zákony.

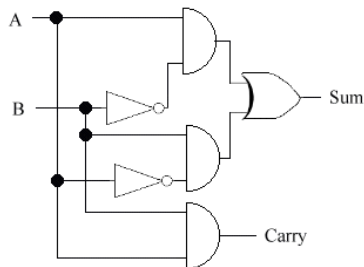
kombinační – výstup je závislý na aktuálním vstupu :

- logické funkce - booleovská logika
- aritmetické funkce - pulsčítačka, sčítačka
- řídicí funkce - multiplexor, dekodér

sekvenční – výstup je závislý na aktuálním vstupu a na celé historii vstupu od iniciálního resetu:

- klopný obvod
- čítač
- stavový automat

Kombinační obvody - pulsčítačka



Vstup:

- jednobitové hodnoty A, B

Výstup:

- součet (Sum): $S = \overline{A}B + A\overline{B}$
- Přenos do vyššího řádu:
 $C = AB$

Kombinační obvody - úplná sčítačka

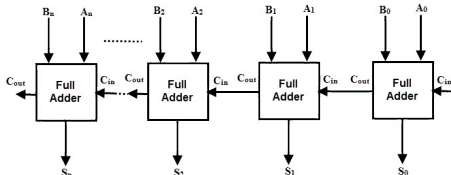
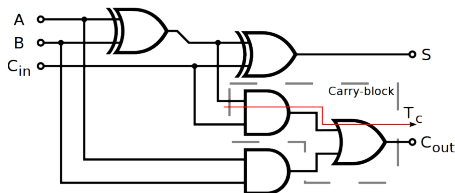
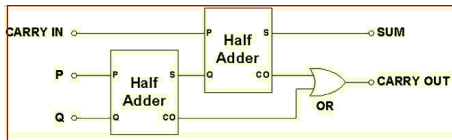
Vstup:

- hodnoty A, B
- přenos z nižšího řádu C_{in}

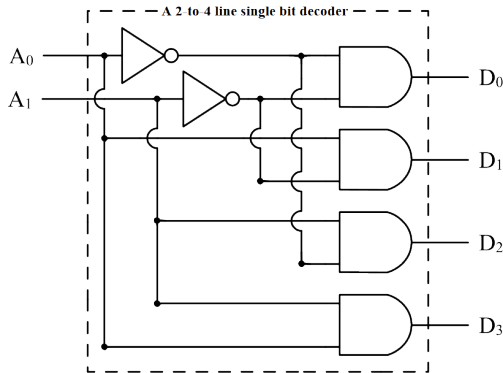
Výstup:

- součet (Sum):
$$S_h = \overline{A}B + A\overline{B}$$
$$S = \overline{S_h}C_{in} + S_h\overline{C_{in}}$$
- Přenos do vyššího řádu:
$$C_{out} = AB + S_hC_{in}$$

Bloky zřetězeny pro součet vícebitových čísel:



Řídicí obvody – dekodér



Truth Table

| A_1 | A_0 | D_3 | D_2 | D_1 | D_0 |
|-------|-------|-------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

Minterm Equations

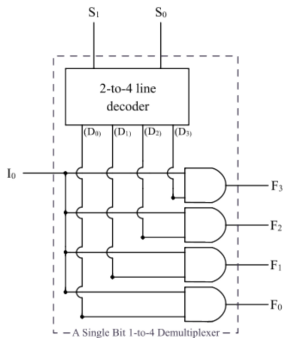
$$D_0 = \overline{A_1} \cdot \overline{A_0}$$

$$D_1 = \overline{A_1} \cdot A_0$$

$$D_2 = A_1 \cdot \overline{A_0}$$

$$D_3 = A_1 \cdot A_0$$

Řídicí obvody – multiplexing



Truth Table

| $S_1 S_0$ | I_0 | F_3 | F_2 | F_1 | F_0 |
|-----------|-------|-------|-------|-------|-------|
| 0 0 | 0 | 0 | 0 | 0 | 0 |
| 0 0 | 1 | 0 | 0 | 0 | 1 |

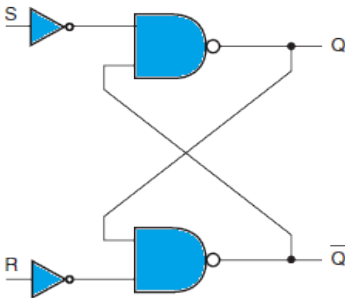
| $S_1 S_0$ | I_0 | F_3 | F_2 | F_1 | F_0 |
|-----------|-------|-------|-------|-------|-------|
| 0 1 | 0 | 0 | 0 | 0 | 0 |
| 0 1 | 1 | 0 | 0 | 1 | 0 |

| $S_1 S_0$ | I_0 | F_3 | F_2 | F_1 | F_0 |
|-----------|-------|-------|-------|-------|-------|
| 1 0 | 0 | 0 | 0 | 0 | 0 |
| 1 0 | 1 | 0 | 1 | 0 | 0 |

| $S_1 S_0$ | I_0 | F_3 | F_2 | F_1 | F_0 |
|-----------|-------|-------|-------|-------|-------|
| 1 1 | 0 | 0 | 0 | 0 | 0 |
| 1 1 | 1 | 1 | 0 | 0 | 0 |

Klopné obvody - SR Latch

Díky zpětné vazbě lze vytvořit paměťový obvod:

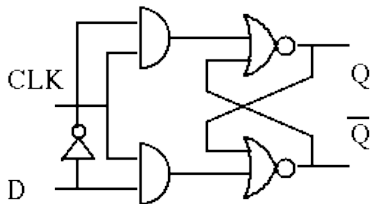
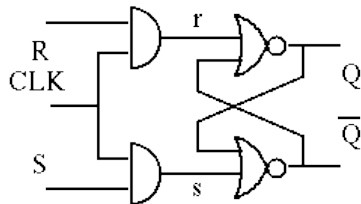


| Operation Mode | S | R | Q_{n+1} |
|----------------|---|---|-----------|
| No change | 0 | 0 | Q_n |
| SET | 1 | 0 | 1 |
| RESET | 0 | 1 | 0 |
| Forbidden | 1 | 1 | — |

Vnitřní stav obvodu se změní přivedením 1 na vstup R/S a tento stav "přetrvá" do dalšího podnětu.

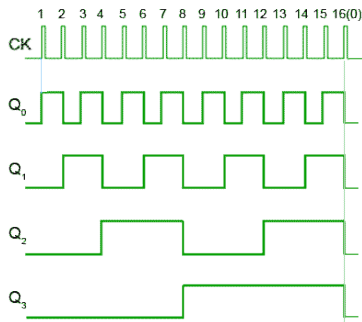
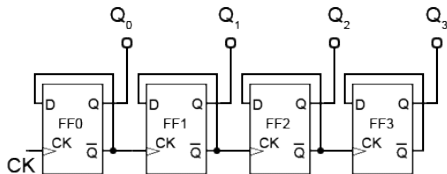
Klopné obvody - odvozené typy

U klopného obvodu typu RS (zde z hradel NOR) lze jednoduchou úpravou dosáhnout synchronizace na hodinový signál:



Klopný obvod typu D navíc odvozuje svůj stav od jednoho vstupu a eliminuje nedefinovaný stav vstupu.

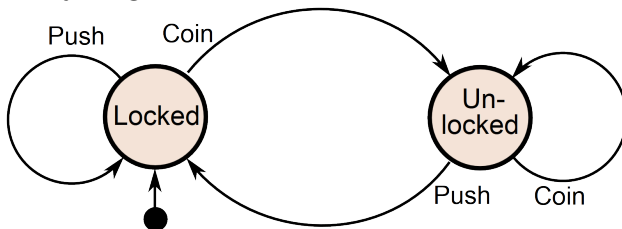
Zřetěžením několika klopných obvodů typu D vznikne čítač:



Stavový automat – Finite State Machine

- FSM klíčový koncept pro embedded, nezávisle na platformě
- konečná množina stavů, vstup a výstup, iniciální stav
- přechodová funkce: změna stavu na základě vstupu
- vystupní funkce: změna výstupu na základě stavu, případně vstupu

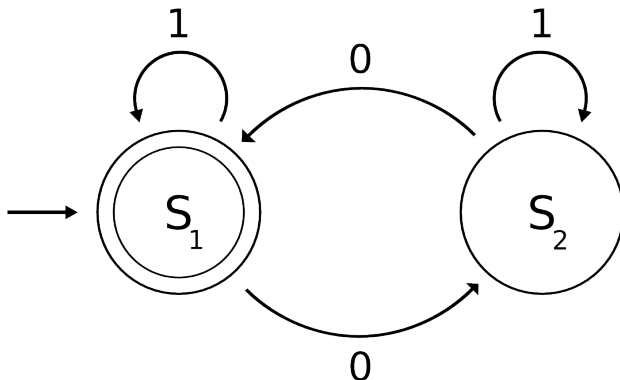
Příklad: stavový diagram "turniket na mince":



Stavový automat typu "Acceptor"

- rozpoznává vstupní posloupnost
- neprázdná množina koncových stavů
- výstupní funkce: akceptace/zamítnutí

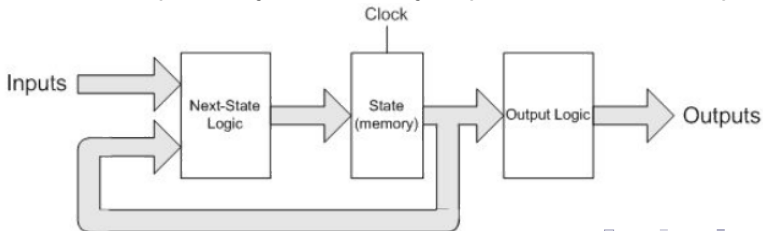
Příklad: stavový diagram "parita":



Stavový automat typu "Transducer"

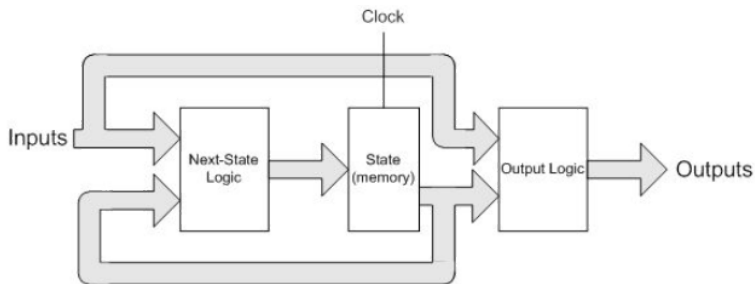
- generuje výstup na základě změn stavu
- dva základní koncepty: Moore, Mealy
- oba využívají tři základní části:
 - ▶ *current state*: realizuje aktuální stav (sekvenční)
 - ▶ *next state*: vyhodnocuje příští stav (kombinační) – realizuje přechodovou funkci
 - ▶ *output logic*: vyhodnocuje výstup (kombinační) – realizuje výstupní funkci

Moore automat: přísně synchronní, výstup asociován se vstupem



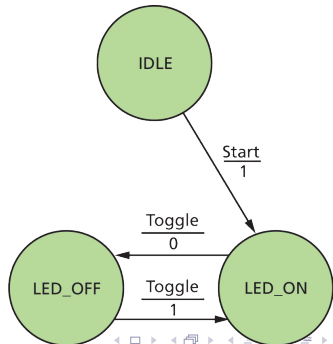
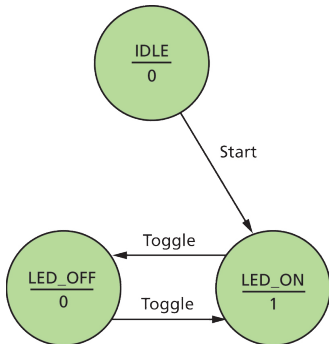
Mealy automat

- výstup na základě aktuálního stavu a vstupu vstupů, tj. výstup asociován s přechodem
- rychlejší díky asynchronní reakci, riziko hazardních stavů
- menší množství stavů než Moore

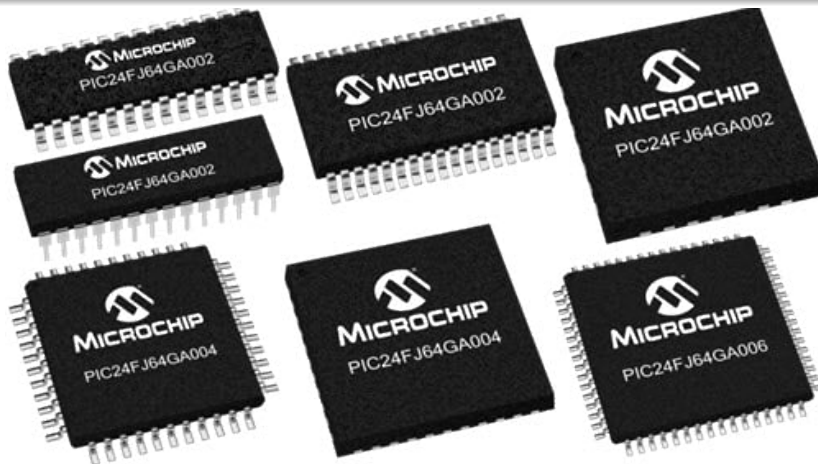


Příklad FSM: LED lampička

- vstup: tlačítko *toggle*
- výstup: LED
- množina stavů: ON, OFF
- iniciální stav: ON
- přechodová/výstupní funkce viz diagram:

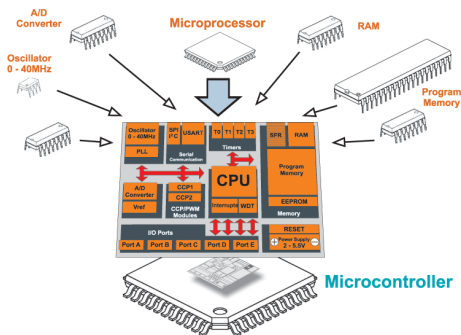


Mikrokontrolery a periferie



Jednočipový počítač alias mikrokontroler, MCU, uC

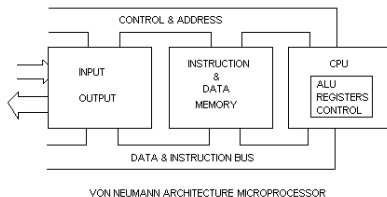
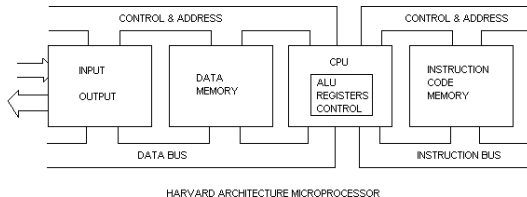
- jádrem většiny embedded systémů, zpravidla monolitický integrovaný obvod
- jádro procesoru + paměti pro data a program + obsahuje podpůrné obvody, umožňující samostatnou funkčnost + fixní množství fyzických vstupů a výstupů + periférie



Typické užití: řízení a regulace, komunikační rozhraní, uživatelské rozhraní – analogie zmenšeného stolního PC



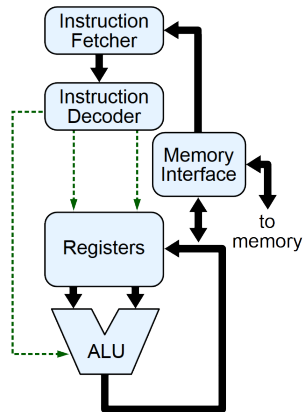
Architektura procesoru



Von Neumann vs. Harvard – společná či oddělená paměť/sběrnice
pro data a program

Jádro procesoru

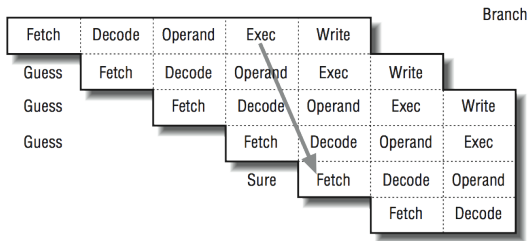
- řadič instrukcí - sekvenční automat (IF + ID)
- registry - pracovní paměť
- aritmeticko-logická jednotka - vykonává operace
- řadič sběrnic(e) - umožňuje přístup k datové a programové paměti (memory interface)



Instrukční cyklus

- Fetch - zpracovávána instrukce načtena z paměti programu
- Decode - rozpoznán typ instrukce a operandy
- Operand - načteny operandy z datové paměti do registrů
- Execute - vykonána instrukce + zápis výsledku (Write)

Příklad: řetězení zpracování (pipelining)

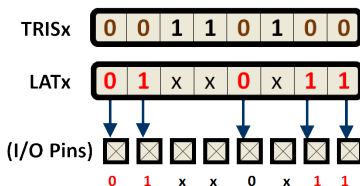


Podpůrné obvody

- datová paměť – energeticky závislá
- paměť programu (Flash)
- zásobník – HW nebo SW
- hodinový signál – reprezentuje diskrétní čas
- resetovací obvody
- **speciální funkční registry** – v adresovém prostoru datové paměti, slouží pro řízení periférií
- konfigurační pojistky – konfigurace obvodů, nutných pro start MCU
- sběrnice

Paralelní I/O (General-Purpose I/O)

- řízení fyzického výstupu (pinu): jednotlivě či sdružené n-bitové slovo
- výstup: GND / Vcc – logická 0 / 1
- třístavová logika – přepínání vstupu a výstupu
- volitelně: pull-up, open-drain
- SFR TRISx pro směr
- SFR PORTx (ev. LATx) pro čtení/zápis hodnoty

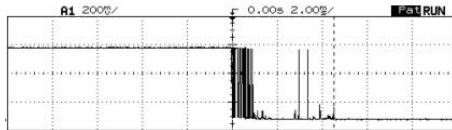
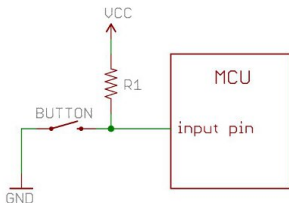


The pin level is driven to the value of
LATx for each bit set to 0 in TRISx

Typické využití: LED, displeje, tlačítka, řízení I/O

Paralelní I/O – tlačítka

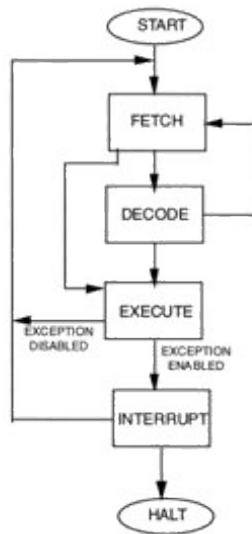
- v rozepnutém stavu náhodná hodnota na vstupu
- pull-up odpor zavede definovanou hodnotu (log. 1)
- invertovaná sémantika



- přechodové jevy při stisknutí tlačítka
- nutno číst vstup opakovaně
- rozestup typ. ≈ 10 ms

Přerušení (interrupts)

- mechanismus pro zpracování asynchronních událostí
- řadič přerušení – vyhodnotí zdroj, nastaví příznak
- tabulka vektorů přerušení (IVT)
- procesor přeruší běh programu, uloží stav procesoru a provede obsluhu přerušení
- vymazání příznaku, návrat z přerušení
- atomické operace a maskování přerušení
- priority přerušení
- souběh přerušení – odložení, vynechání
- latence přerušení



Konfigurace a obsluha přerušení

Definice obsluhy přerušení – různé metody, platformově závislé:

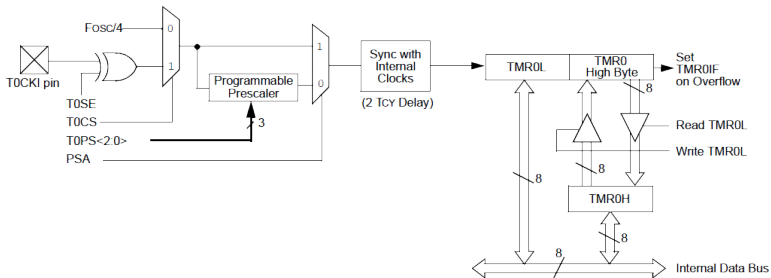
- jedna funkce + multiplexing
- předdefinované názvy procedur
- plnění tabulky vektorů přerušení

SFR:

- `intIE` (interrupt enable) – povolí konkrétní přerušení
- `intIF` (interrupt flag) – indikuje příchod přerušení, příznak nutno vymazat v rámci obsluhy
- `GIE` (global interrupt enable) – povolení funkce řadiče přerušení
- `PEIE` (peripheral interrupt enable) – povolení funkce třídy přerušení

Časovač/Čítač

- binární čítač, definovaná šířka (8, 16, 32bit)
- zdroj: systémové hodiny, externí hodiny, externí signál
- (a-)synchronní čítání
- pre/post-scaler: čítání každé n -té události
- automatický reset čítače, generování přerušení



Obsluha Časovače/Čítače

Obvyklé SFR:

- TMR_x / TMR_xH + TMR_xL: pracovní registr čítače
- TMR_xIE / TMR_xIF: povolení / příznak přerušení
- PR_x (Period Register): hodnota se porovnává s pracovním registrem, v případě shody se pracovní registr nuluje a vyvolá se přerušení
- TMR_xPS, TMR_xCKPS, TMR_xOUTPS: nastavení pre/post-scaleru, zpravidla bitový kód
- TMR_xCS (clock source): nastavení pre/post-scaleru, zpravidla bitový kód
- TMR_xON: povolení časovače

MCU obvykle obsahuje několik časovačů s různou funkcí; zároveň mohou tyto realizovat časovou základnu pro další periferie (např. PWM).

- bezpečnostní obvod, hlídá korektní běh procesoru
- čítač se zpravidla dedikovaným oscilátorem
- program MCU musí čítač pravidelně programově nulovat
- pokud čítač dosáhne stanovené hodnoty, resetuje MCU
- předpokládá se, že se MCU někde zapoměl
- konfigurace typicky pomocí pojistek

EEPROM paměť slouží k uložení konfiguračních dat.

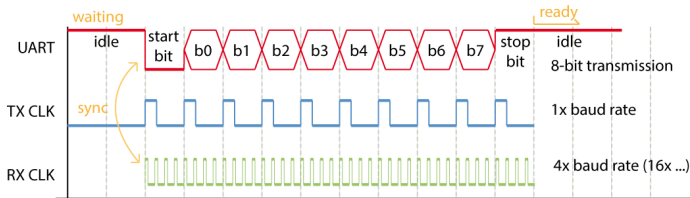
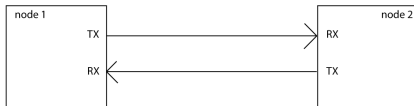
Obsluha pomocí SFR:

- **čtení:** zápisem log. 1 do položky RD registru EECON1 se hodnota na adrese EEADR uloží do EEDATA
- **zápis:** zápisem log. 1 do položky WR registru EECON1 se hodnota EEDATA uloží do paměti na adresu EEADR
- zabezpečení zápisu – před zápisem je třeba:
 - ▶ explicitně zápis povolit zápisem log. 1 do položky WREN registru EECON1
 - ▶ zapsat sekvenci 0x55, 0xAA do registru EECON2.

Podobně lze u některých MCU zapisovat do Flash paměti program nebo konfigurační pojistky.

Komunikační protokol UART

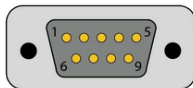
- Universal Asynchronous Receiver-Transmitter
- volitelná délka datového slova (7,8,9 bitů)
- volitelně detekční kód – parita
- volitelná délka stop-bitu
- uzly domluvené na taktování a parametrech přenosu



Rozhraní RS-232

- dříve běžná součást PC jako "sériový port"
- využívá UART
- definuje fyzické rozhraní a napěťové úrovně
- standardní datové rychlosti
- definuje metodu synchronizace a dekódování na RX straně

DB9M Connector

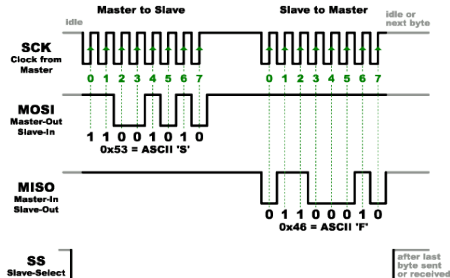
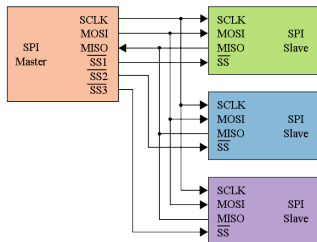


RS232 Pin Out

| Pin # | Signal |
|-------|--------|
| 1 | DCD |
| 2 | RX |
| 3 | TX |
| 4 | DTR |
| 5 | GND |
| 6 | DSR |
| 7 | RTS |
| 8 | CTS |
| 9 | RI |

Sběrnice SPI

- master-slave sériová sběrnice
- hodinový signál
- dva datové signály:
master in - slave out (MISO)
master out - slave in (MOSI)
- slave-select signály

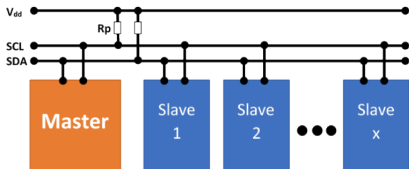


- duplexní režim možný
- korektní nastavení fáze a polarity
- libovolné napěťové úrovně i délka slova
- taktování $> 10 \text{ MHz}$

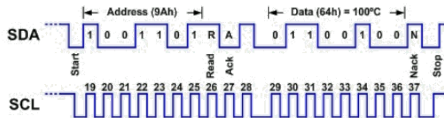
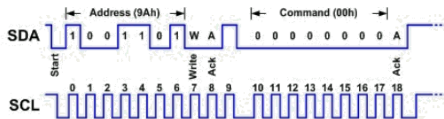
Typické užití: komunikace mezi IC nebo mezi blízkými komponenty

Sběrnice I2C

- master-slave sériová sběrnice
- hodinový signál SCL
- datový signál SDA
- externí pull-up pro střídání zdroje signálu
- uzly se mohou v roli Master střídát



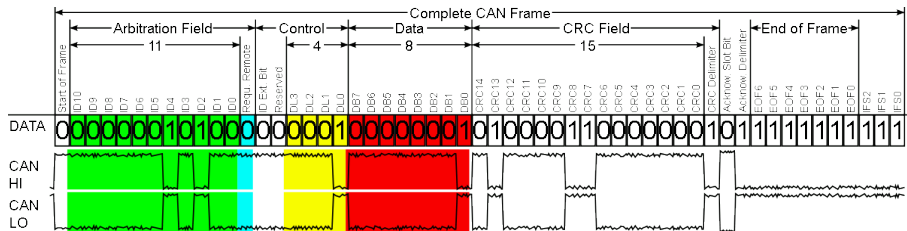
Typické užití: komunikace mezi IC



- součástí datového paketu adresa Slave uzlu
- potvrzování / timeouty
- taktování 400 KHz

CAN-bus (Controller Area Network)

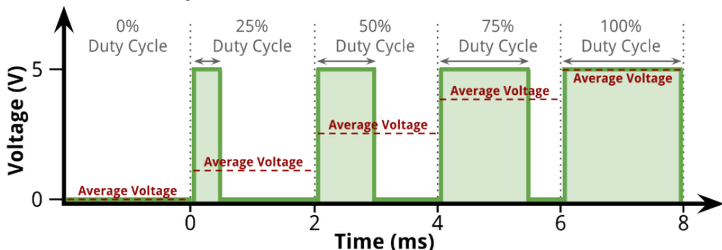
- peer-to-peer sériová sběrnice, 1 diferenciální pár
- spolehlivé doručení, prioritizace provozu
- 11-bit (28-bit) CAN-id, 8 bajtů data
- rychlost až 1 Mbit/s, robustní provoz, optické oddělení uzlů



Typické užití: automotive aplikace

Pulsně-šířková modulace (PWM)

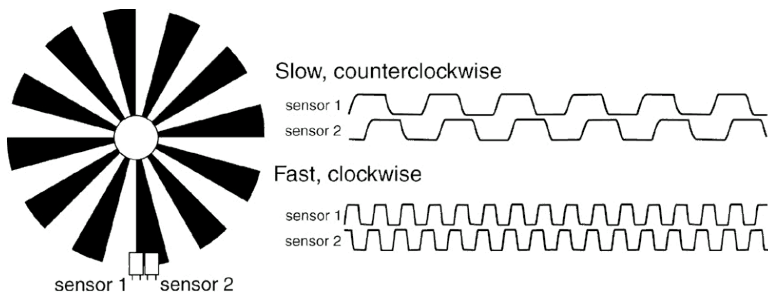
Modulační technika, která umožňuje binárně generovat spojitou veličinu změnou délky aktivní doby pulzu v rámci periody
Zpravidla řízeno čítačem, který je porovnáván s registrem periody a registrem aktivní doby.



Typické užití: motory, světelné zdroje, výkonové zátěže obecně

Kvadrurní (rotační) enkodér

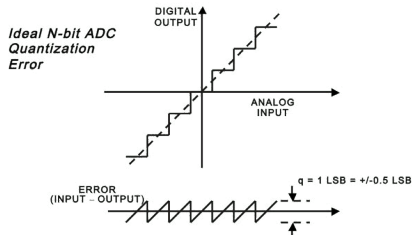
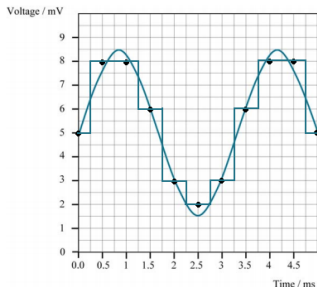
- elektromechanický/ elektrooptický systém
- převádí změnu úhlu na kvadrurní signál
- rozlišení: pulzy na otáčku
- vzájemná fáze dvojice signálů udává směr pohybu
- ev. doplněn 3. signálem pro indikaci absolutní polohy
- řízeno přerušením ($A \text{ xor } B$)



Typické užití: mechanické ovládací prvky, úhlové a lineární délkové senzory

Převodník analog-číslo (ADC)

- převod spojité veličiny na digitální reprezentace
- diskretní vzorkování
- diskretní kvantizace
- přenosová funkce



- kvantizační chyba
- vzorkovací rychlost
- vzorkovací teorém
- filtr pro anti-aliasing

Převodník číslo-analog(DAC)

- převádí digitálně reprezentovanou veličinu na analogovou
- vzorkování, kvantizace, přenos analogické opačným směru
- rekonstrukční filtr

