

Otázka: Mikroprocesorová technika (MIT)

Předmět: ELP, Informatika

Přidal(a): jasal

Mikrokontroler řady 8051

Základní rozdělení:

- Je to **8bitový** (jednočipový) kontrolér s **Harvardskou strukturou**, u které je rozdělena programovací a datová paměť
- Napájení má hodnotu **5V**

Základní informace k tvorbě programu:

- **Registr** - část paměti, kterou využíváme pro ukládání dat a tvorbu programů
- Všechny registry u 8051 jsou 8 bitové
- Značení jednotlivých bitů:

b.7	b.6	b.5	b.4	b.3	b.2	b.1	b.0
-----	-----	-----	-----	-----	-----	-----	-----

Číselné soustavy:

Binární soustava - používáme **0 a 1**

- Zápis: 0111 B

Hexadecimální soustava - používáme: **0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F**

- Zápis: 0FF H

Desítková soustava - používáme: **0,1,2,3,4,5,6,7,8,9**

- Zápis: 23 D

128	64	32	16	8	4	2	1

Důležité registry paměti RAM u mikrokontroleru řady 8051:

ACC - střadač (akumulátor)

- Patří mezi nejdůležitější registry (8bitový registr)
- Slouží pro matematické a aritmetické operace
- Značení: **ACC**

B - pomocný registr, který využíváme pro aritmetické operace

- 8 bitový registr

Vstupně/výstupní porty (brány) - 8 bitové registry

- Značení: **P0,P1,P2,P3**

Pomocné registry R0 až R7 - nacházení se v bance (**banka 0, banka 1, banka 2, banka 3**)

- V každé bance je 8 registrů (**R0, R1, R2, R3, R4, R5, R6, R7**)

Instrukce:

- Základní instrukční soubor obsahuje **102** instrukcí (čím více instrukcí, tím je provádění programu pomalejší - složitost instrukcí)
- Instrukční soubor se rozděluje do několika skupin

Základní instrukce:

1) Přesunové - MOV - z 1 paměťového místa přesuneme data do jiného paměťového místa

|tab|MOV|tab|ACC, #45D

- Existují různé modifikace MOV

2) Aritmetické - zde patří:

ADD - aritmetický součet

MUL - aritmetické násobení

DIV - aritmetické dělení

INC - inkrementace - přičte do daného registru hodnotu 1

DEC - dekrementace - odečte od daného registru hodnotu 1

!!! Aritmetické operace se provádějí pomocí registru ACC!!!

PSW - stavové slovo (je to registr z bitů, které nám určují stavy jiných registrů)

- 8 bitový registr

C	.	F0	RS1	RS0	.	.	.
---	---	----	-----	-----	---	---	---

- **C... Carry bit à příznakový bit** - při přetečení registru ACC dochází ke změně hodnoty bitu z na **1**
- **F0 - uživatelský bit (programátorský bit)**
- **RS0, RS1 - určují banku**

SP - ukazatel zásobníku

IE - interrupt enable = povolení přerušení

- 8bitový registr

EA	.	.	ES	ET1	EX1	ET0	EX0
----	---	---	----	-----	-----	-----	-----

- **EA** - slouží ke globálnímu povolení přerušení
- **ES** - povolení sériové komunikace
- **EX0** - povolení vnějšího přerušení (**INT0**)
- **ET0** - povolení čítače, časovače (**T0**)
- **EX1** - povolení vnějšího přerušení (**INT1**)
- **ET1** - povolení čítače, časovače (**T1**)
- Tento registr slouží k povolení přerušení
- Každý bit tohoto registru povoluje dané přerušení, jestliže má hodnotu **1**

SETB _____ **EA**; **SETB** _____ **EX0** à povolení vnější **INT0**

Registr IP - nastavení priorit přerušení

- uK 8051 - má 2 úrovně priorit přerušení

PCON - řízení napájení

Další registry pro řízení a fci. Čítačů a časovačů:

TMOD; TCON; TL0; TH0 à čítače/časovače **T0**

TL12; TH15 à čítače/časovače **T1**

Skokové instrukce

1) nepodmíněné skokové instrukce:

JMP _____ **NÁVĚŠTÍ**

SJMP _____ **NÁVĚŠTÍ** à krátká skoková instrukce

LJMP _____ **NÁVĚŠTÍ** à dlouhá skoková instrukce

Př) **SJMP** _____ \$ à programový čítač skáče na stejný řádek

2) podmíněné skokové instrukce:

- **Bitové:**

JB__bit, START

- Je-li **daný bit roven 1**, skáče na návěští
- Je-li **daný bit roven 0**, jde dál

JNB__bit, SKOK

- Je-li **daný bit roven 0**, skáče na návěští
- Je-li **daný bit roven 1**, jde dál

JC - je-li **C=1**, skočí na NÁVĚŠTÍ

- Je-li **C=0**, jde dál

JNC - je-li **C=0**, skočí na NÁVĚŠTÍ

- Je-li **C=1**, jde dál

JZ - jestliže střadač **ACC=0**, skočí na NÁVĚŠTÍ

- jestliže střadač **ACC=1**, jde dál

Bitové instrukce - jsou to instrukce, které pracují s jednotlivými bity (případně s registrem ACC)

CLR - instrukce, která nuluje daný bit

- Instrukce nuluje střadač ACC

CLR__bit

CPL - instrukce, která neguje daný bit (případně ACC)

CPL__bit

SETB - do daného bitu uloží log 1 (ACC - nepoužívá)

SETB_bit

Přesunové instrukce - instrukce slouží k přesunu jednotlivých bitů ve střadači ACC

- Mohou být 8 bitové nebo 9 bitové

a) 8bitová rotace

RR - rotace doprava - jednotlivé bity se přesunou doprava

RR__A

RL - rotace doleva

RL__A

b) 9bitová rotace - využíváme přesunu bitu o 1 místo v registru ACC včetně příznakového bitu C

RRC - přesun bitu doprava včetně příznakového bitu C

RRL - přesun bitu doleva včetně příznakového bitu C

Sdružené instrukce - jsou to instrukce, které provádějí více kroků a trvají nejdelší čas

DJNZ - odečte od daného registru 1 a porovná tuto hodnotu s 0

- Je-li různá od 0, skáče na NÁVĚŠTÍ, je-li rovna 0, pokračuje dál

DJNZ_A,SKOK (pokud A=0 po odečtení, jde dál a neskáče na NÁVĚŠTÍ)

CJNE - porovnává obsahy 2 registrů a jestliže se nerovnaj, skočí na NÁVĚŠTÍ a jestliže se rovnají, jde dál

CJNE_A1, A2, A3 (A1...argument 1, A2...argument 2, A3...argument 3)

Časová prodleva - pauza

Pauza DL:

MOV_R1,#225

Pauz: MOV R2, #11111111B

DJNZ_R2, \$

DJNZ_R1, Pauz

RET

Pauza KR:

MOV_R1, #225

DJNZ_R1, \$

RET

(RET...instrukce pro návrat do programu a vrací se na řádek pod instrukci ACALL [CALL])

Přerušení = při přerušení dochází k okamžitému zastavení hlavního programu skoku do podprogramu, jeho vykonání a návrat do hlavního programu (místa přerušení)

- 5 zdrojů přerušení:
- **INT 0** - vnější přerušení
- **INT 1** - vnější přerušení
- Čítač/časovač - **T0**
- Čítač/časovač - **T1**
- Sériová komunikace : **TxD, RxD**

Vnější přerušení: (INT 0, INT 1)

- Posláním **log 0** od pinu **2**, nebo **P3.3** dojde k zastavení hlavního programu a skoku do podprogramu, jeho vykonání a návratu do místa, kde došlo k přerušení a pokračování v programu
- Každý program přerušení končí instrukcí **RETI**

Jednotlivé kroky pro definování a použití vnějšího přerušení:

```
ORG 0H
```

```
SJMP START
```

```
ORG 03H
```

```
SJMP PODPROGRAM_0
```

```
ORG 30H
```

```
;
```

```
START:          MOV IE,#10000101B (poslední 1 pro INT0, 3. Od zádu pro INT1)
```

```
MOV TCON,# 0000101B
```

```
PODPR:.....RETI....
```

Nastavení informace pro programový čítač:

```
ORG 03H - INT0
```

```
ORG 13H - INT1
```

ORG 0BH - T0

ORG 01BH - T1

ORG 023H - SERIOVÁ KOMUNIKACE

Registr TCON - patří do skupiny registrů řízení časovačů

- Tento registr se skládá ze 4 bitů příslušejících oběma čítačům a časovačům a 4 bitů, příslušejících vnějším přerušením
- **8** bitový registr

pro T0 a T1				pro INT0 a INT1			
TF1	TR1	TF0	TR0				

TR0 -> log0 à spustí časovač **T0**

TR1 -> log0 à spustí časovač **T1**

Čítače/časovače - daný mikrokontrolér může pracovat buď v režimu časovače, nebo v režimu čítače

- Tato fce se nastavuje v registru **TMOD** (v případě našich příkladů používáme časovače)

TMOD

T1				T0			
Gate	C/T	M1	M0	Gate	C/T	M1	M0

C/T Log0 à časovač Log1 à čítač

GATE - řízení hradlování - je-li hodnota log0 je čítač řízen pouze v programu

M0/M1 - jsou bity, které určují MOD čítače, časovače = nastavení daného čítače, časovače jak bude pracovat

- Máme **MOD0** až **MOD3**

MOD0 - M0 = 0; M1 = 0

- Čítač/časovač je v MODu0 **13 bitový**

$TL_n = 5$ $TH_n = 8$

MOD1 - M0 = 1; M1 = 0

- Čítač/časovač je **16 bitový**

$TL_n = 8$ $TH_n = 8$

MOD2 - M0 = 0; M1 = 1

- Čítač/časovač je **maximálně 8 bitový**

Registr **TH** = přednastavení hodnoty pro registr **TL**

MOD3 - M0 = 1; M1 = 1

- v MODu 3 využíváme oba čítače/časovače

Definování PODPROGRAMU - návratové instrukce:

RET - (podprogram volaný instrukcí **ACALL**) à návrat na řádek pod **ACALL**

RETI - (přerušeni) à návrat do místa, kde došlo k přerušeni

Registry čítačů/časovačů: Registr **TL0**; Registr **TH0**; Registr **TL1**; Registr **TH1**

DPTR = ukazatel dat - je to **16 bitový** registr, ale skládá se ze 2 registrů: **DPL**, **DPH**

Přerušovací systém:

- umožňuje jakousi komunikaci mezi probíhajícím hlavním programem a určitými hardwarovými částmi mikroprocesoru
- je-li přerušeni od dané hardwarové části povoleno, procesor přeruší vykonání hl. programu a začne se vykonávat program zajišťující obsluhu vzniklého přerušeni
- je-li program obsluhy přerušeni dokončen, procesor se vrací k vykonávání hl. programu od místa, ve kterém skončil před přechodem do programu obsluhy přerušeni

Instrukční soubor CPU 8051

- IS - skupina instrukcí

Lze rozdělit do 7 skupin:

- Přesunové instrukce - **MOV**
- Aritmetické instrukce - **INC, ADD, DEC, MUL, DIV, SUBB, CPL**
- Logické operace - **ANL** (log. součin), **ORL** (log. součet)
- Posunové instrukce - umožňují 8 a 9 bitové log. rotace střadače (**RR, RL, RRC, RLC**)
- Bitové operace - umožňují nastavit (**SETB**), nebo nulovat (**CLR**), kterýkoliv R 2256 přímo adresovatelných bitů mikrokontroléru 8051
- Skokové instrukce - umožňují skok - **JMP, LJMP (16 BITOVÝ), AJMP (11 BITOVÝ), LCALL, ACALL, JB, JNB, JC, JNC, JZ, JNZ**
- Sdružené instrukce - **DJNZ, CJNE**