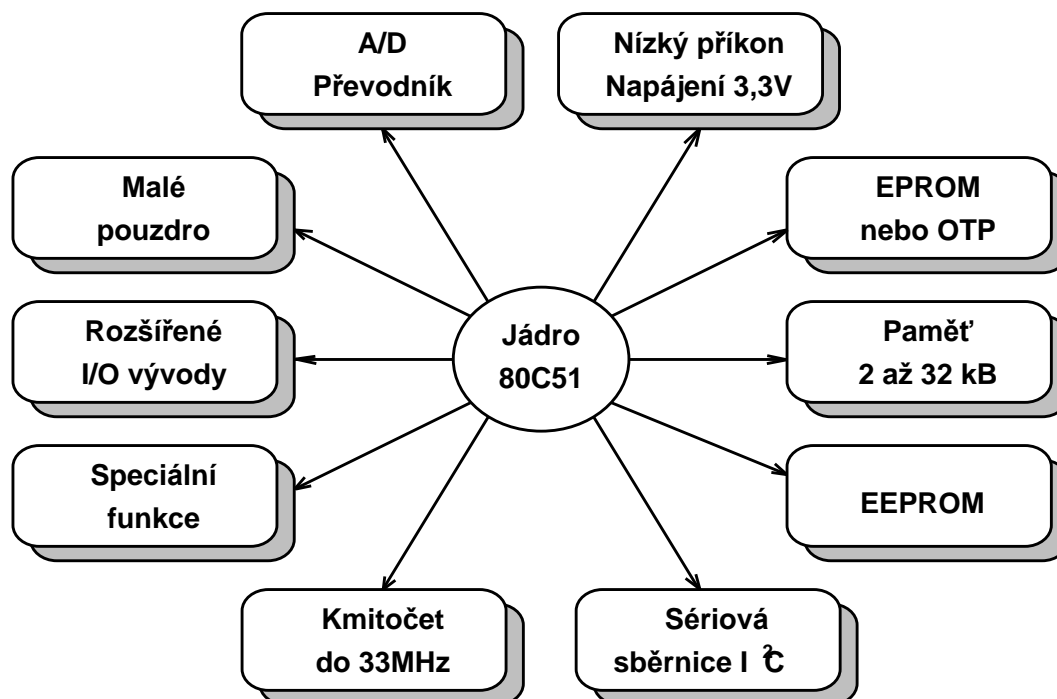


Mikroprocesory z řady 8051

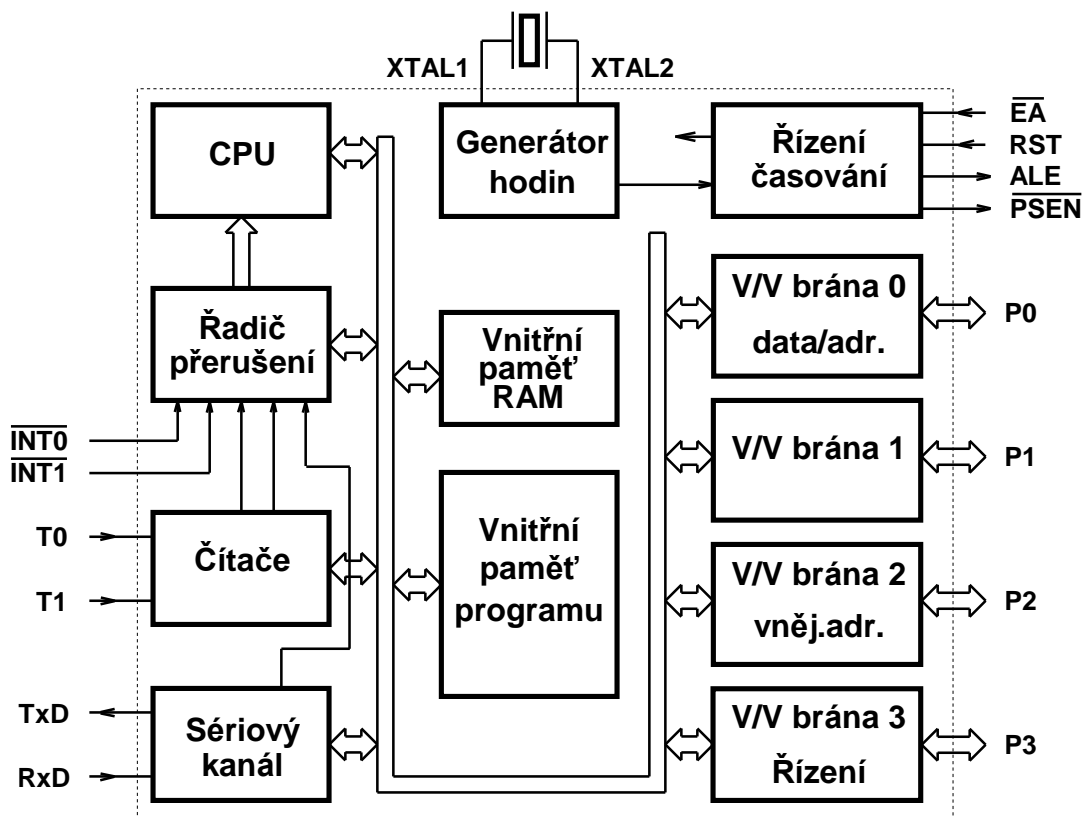
Mikroprocesor 8051 pochází z roku 1980 a je vývojově procesorem relativně starým. U návrhářů však dosáhl takové obliby, že i v současné době se řada výrobců orientuje na výrobu procesorů založených na jádře procesoru 8051, které je rozšířeno o více či méně dalších periférií. Například firma Philips vyrábí 24 různých typů těchto procesorů mající společné jádro obr.1, ke kterému jsou připojeny některé z uvedených periférií jako je: paměť programu o velikosti 2kB až 32kB, kterou lze programovat jenom jednou (provedení OTP) nebo několikrát (provedení EPROM), paměť EEPROM pro uchování konstant, rozšířená vnitřní paměť RAM na 256b, 8 nebo 10-bitový A/D převodník obvykle s osmikanálovým analogovým multiplexerem, rozšířené vstupně/výstupní vývody, komparační a záchytný systém nebo dvoudrátová přístrojová sběrnice I²C. Výrobci nabízejí procesory od základního hodinového kmitočtu 12MHz až po 33MHz ve standardním pouzdře DIL, přes provedení PLCC až po malá pouzdra PQFP. Jiné firmy se orientují na procesory s rozšířenými aritmetickými operacemi (Siemens 80C537) nebo s odlišným vnitřním časováním (Dallas 80C320), který při standardním hodinovém kmitočtu má trojnásobný výkon. Sjednocujícím základem všech těchto rozmanitých typů procesorů je vlastní jádro procesoru 8051 (80C51), které si nyní stručně popíšeme.



Obr.1 Prvky rozšiřující vlastnosti základního jádra procesoru 8051

Jádro procesoru 8051

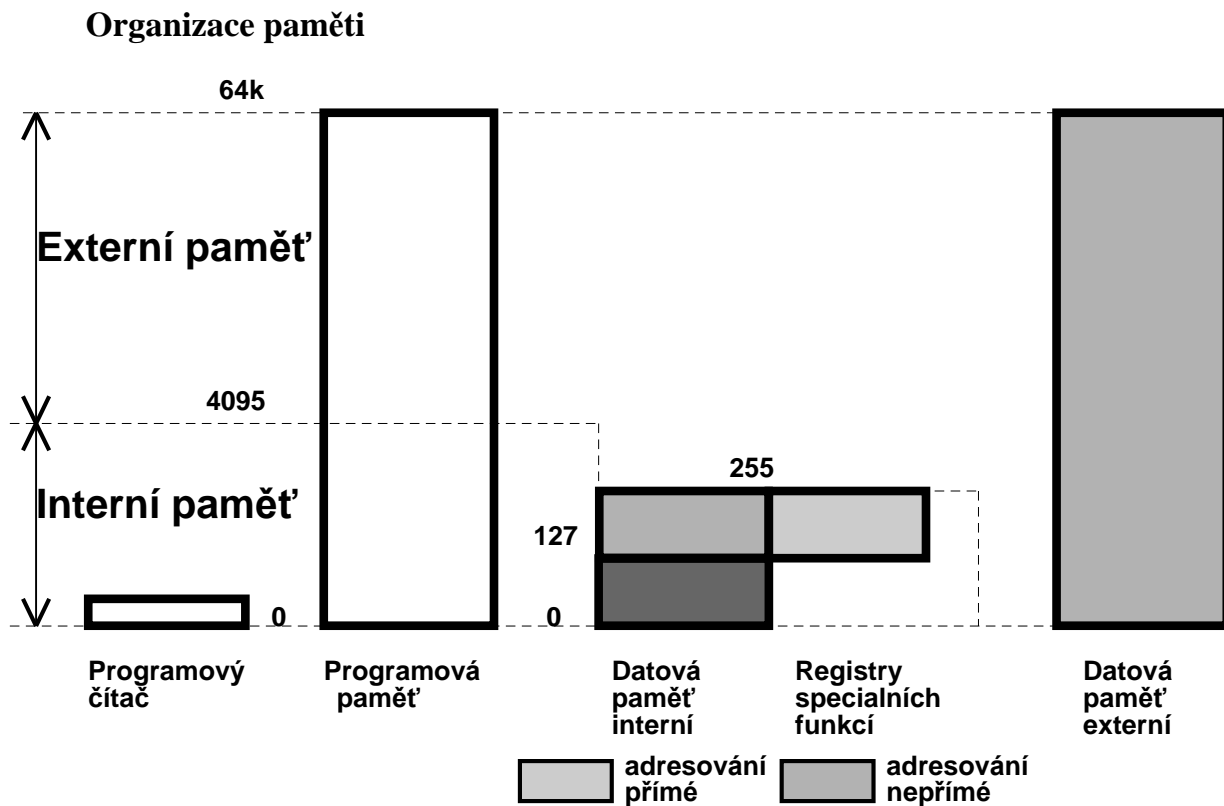
Mikroprocesor 8051 je osmibitový jednočipový mikroprocesor se smíšenou harwardskou a Von Neumanovskou architekturou, u které je oddělena programová a datová paměť, ale formát instrukcí a dat je totožný a přenáší se po stejné sběrnici. Procesor, jehož vnitřní struktura je blokově zobrazena na obr.2, je schopen samostatné činnosti po připojení vnějšího piezokeramického rezonátoru ("krystalu") na vývody XTAL1 a XTAL2 a jednoho napájecího napětí 5V. Na čipu procesoru je umístěna vlastní procesorová jednotka CPU, která je vnitřní společnou sběrnicí propojena s pamětí programu ROM nebo EPROM o kapacitě 4kB (existují i typy bez této paměti), s pamětí RAM o kapacitě 128 bytů a se čtyřmi vstupně/výstupními branami P0 až P3, které zajišťují styk procesoru s vnějšími periferiemi. Nechceme-li využívat procesor v jednočipové konfiguraci, případně nemůžeme (procesor bez vnitřní ROM/ EPROM), jsou z procesoru vyvedeny řídicí signály pro správu vnější programové ($\overline{\text{PSEN}}$) nebo datové ($\overline{\text{WR}}$, $\overline{\text{RD}}$) paměti, z nichž každá může mít až 64kB. Pro snadnější styk s periferiemi je procesor vybaven systémem přerušeni, který zpracovává 5 zdrojů přerušeni (2 externí a interní, od každého z dvou časovačů a od sériového kanálu). Jednotlivá přerušeni mají definovanou prioritu na každé ze dvou volitelných úrovní priority. Čítače, které usnadňují realizaci časování, jsou 16-bitové s hodinovým signálem odvozeným z interního generátoru hodin nebo z vnějších vstupů T0 nebo T1. Pro snazší sériový styk s nadřazenými počítači nebo jinými spolupracujícími procesory je vybaven duplexním



Obr.2 Vnitřní bloková struktura procesoru 8051

(obousměrným) sériovým kanálem. Procesor je dále vybaven Boolovským procesorem, který

umožňuje pracovat s jednotlivými bity vnitřní paměti RAM i interních periférií. Pro základní hodinový kmitočet 12MHz trvají instrukce 1 μ s nebo 2 μ s s tím, že instrukce násobení a dělení trvají 4 μ s.

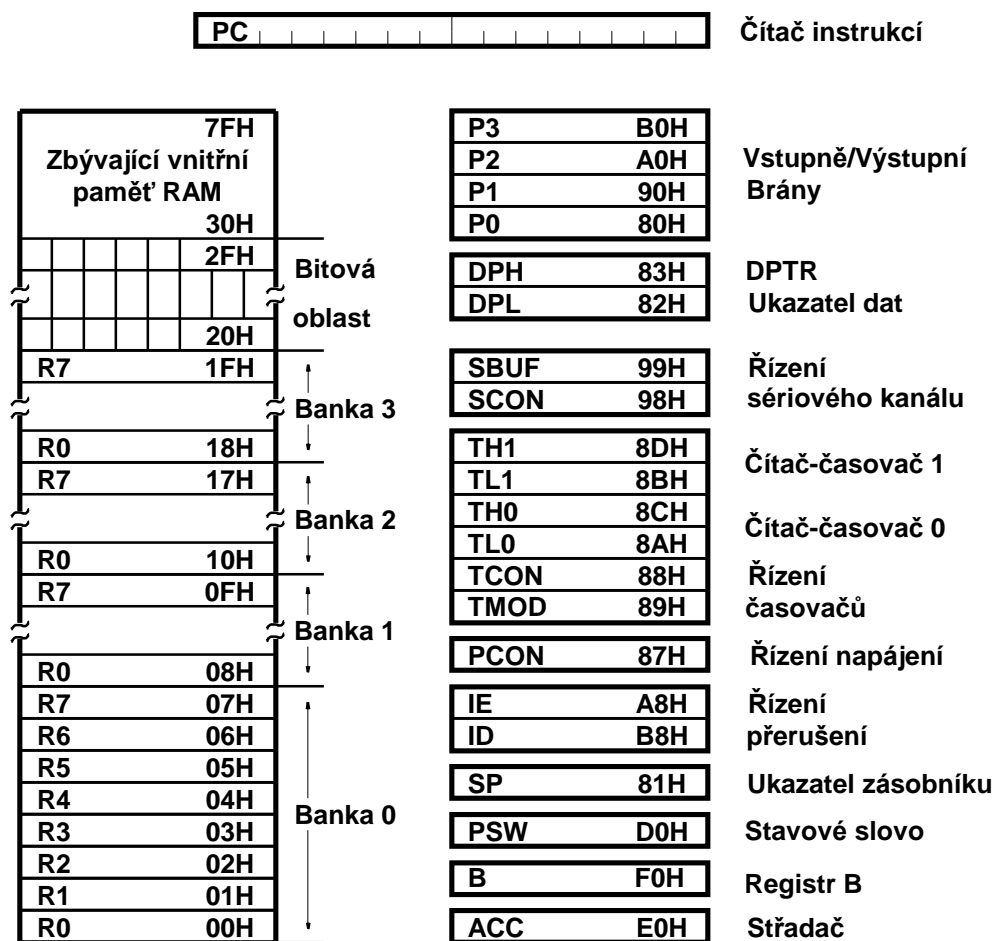


Obr.3 Struktura paměťového prostoru 8051 a jeho následovníků

Mikroprocesor 8051 má oddělené adresové prostory programu a dat, které jsou přístupné různými instrukcemi. Paměťový prostor můžeme dále dělit na vnitřní, umístěný na čipu, a vnější, který lze v případě potřeby s pomocí dalších součástek vytvořit. Proti překrývání vnitřního (4kB) a vnějšího (64kB) programového adresového prostoru je procesor vybaven vstupem \overline{EA} obr.2. Je-li vstup $\overline{EA}=0$, potom programová paměť je tvořena celou vnější pamětí, je-li vstup $\overline{EA}=1$, potom instrukce v adresovém prostoru 000H - FFFH se čtou z vnitřní paměti ROM nebo EPROM a mimo tento prostor (1000H - FFFFH) ze zbývajících 60kB vnější paměti programu obr.3. Překrývání vnitřního a vnějšího datového adresového prostoru je odstraněno tím, že přístup do každého prostoru je realizován pomocí rozdílných instrukcí. Na obr.4 je základní rozdělení vnitřní datové paměti včetně umístění speciálních registrů. Vnitřní datová paměť RAM 128 bytů je tvořena čtyřmi bankami (0,1,2,3) po osmi registrech R0, R1, až R7 (adresy 00H - 1FH), za kterými je vyhrazeno 16 bytů (adresy 20H-2FH) pro tzv. **bitoblast**. Jednotlivé bity paměťových míst počínaje nejnižší adresou (20H) a nejnižším bitem (b0 - adresa bitu 00H) jsou vzestupně přímo adresovatelné. Přímě adresovatelných bitů v této oblasti je 128 (posledním bitem s adresou 127 (7FH) je bit b7 na adrese 2FH). Zbývajících 128 adres bitů se využívá k adresování některých

významných bitů příslušejících speciálním registrům. Zbývající datová paměť RAM začíná adresou 30H a končí adresou 7FH a je uživateli volně přístupná pro přímé i nepřímé adresování.

Je-li procesor vybaven rozšířenou vnitřní datovou pamětí v adresovém prostoru 80H až FFH, potom je tato paměť s ohledem na speciální registry přístupná pouze pomocí nepřímého adresování. Vnější datová paměť RAM s kapacitou až 64kB je přístupná přes 16-bitový pomocný ukazatel datové paměti DPTR. Určitou nevýhodou je to, že DPTR může být pouze naplněn konkrétní adresou paměťového místa nebo inkrementován (DPTR nemůže být snadno zmenšován). Tato nevýhoda je částečně odstraněna u dalších typů procesoru, které mají více



Obr.4 Rozdělení vnitřní paměti RAM procesoru 8051

registrů DPTR. Vnější datová paměť může být přístupná i s pomocí registrů R0 a R1 příslušné aktivní banky, které se využívají k 8-bitovému nepřímému adresování. Zapsáním 8-bitové hodnoty do výstupní brány P2, která představuje horní část adresy při adresování vnější paměti, rozdělíme paměť RAM na 256 bloků po 256 bytech. Nepřímým adresováním pomocí registrů R0 a R1 potom můžeme zapsat nebo přečíst vybraný byte ve zvoleném bloku. Oblast speciálních funkcí (SFR) je tvořena 21 registry, které leží v adresovém prostoru 128 (80H) až 255 (FFH). To znamená za vnitřní paměti RAM nebo ve stejném prostoru jako leží rozšířená datová paměť RAM u

následovníků procesoru 8051. Z tohoto důvodu jsou speciální registry přístupné pouze pomocí přímého adresování bytů nebo případně i bitů (v registrech SFR, jejichž adresa je dělitelná osmi).

Registry speciálních funkcí

Všechny informace důležité pro činnost mikroprocesoru a jeho periferních obvodů integrovaných na čipu procesoru, jako jsou čítače, sériový kanál, přerušovací systém a u následovníků 8051 záchytný a komparační systém, jsou soustředěny do souboru registrů tzv. registrů speciálních funkcí. Jednotlivé registry si nyní postupně popíšeme a v případě registrů příslušejících a ovlivňujících činnost periférií popíšeme i s popisem vlastností těchto obvodů.

A - Střadač je základní registr aritmetickologické jednotky, který vždy obsahuje jeden operand aritmetické nebo logické operace a do něhož se ukládá výsledek této operace. Protože registr dovoluje ve spolupráci s aritmetickologickou jednotkou postupné přičítání zpracovávaných čísel, nazývá se střadač nebo **akumulátor**. Zvláštností střadače u procesoru 8051 je to, že leží v paměťovém prostoru SFR vnitřní paměti RAM procesoru a je přístupný nejen běžnými instrukcemi, ale i pomocí přímé adresy označované v mnemonice procesoru symbolickým názvem ACC.

B - Registr obsahuje jeden operand (druhý je umístěn ve střadači) pro instrukci násobení nebo dělení. Spolu se střadačem obsahují výsledek operací násobení a dělení. Nejsou-li tyto instrukce využívány, lze jej použít jako univerzální registr.

PSW - Stavové slovo mikroprocesoru se skládá ze 7 bitů, jejichž umístění je zobrazeno na obr.5 a jejich význam je následující:

C - Přenos (Carry) je nastaven při aritmetické operaci, při které dochází k přenosu z osmého (b_7) do devátého (b_8) bitu a při některých instrukcích porovnání.

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Bit
C	AC	F0	RS1	RS0	OV	---	P	Adresa = D0H
D7	D6	D5	D4	D3	D2	D1	D0	Bitová adresa (hex)

Obr.5 Rozložení příznaků ve stavovém slově

AC - Částečný přenos (Auxiliary carry) je nastaven dojde-li při sčítání k přenosu mezi čtvrtým (b_3) a pátým (b_4) bitem střadače. Příznak se využívá pouze instrukci dekadické korekce DAA, kterou aplikujeme na výsledek součtu dvou dekadických čísel vyjádřených v BCD kódu.

F0 - Uživatelský příznak **F0** může být libovolně využíván programátorem k indikaci nějaké události (např. přetečení při výpočtu v aritmetice, identifikace vnější události, atd.). Příznak je pozůstatkem po vývojově předcházejícím procesoru (8048), který nebyl vybaven takovou škálou příznaků.

RS1	RS0	Banka	Adresy reg. R0,...,R7
0	0	0	00H až 07H
0	1	1	08H až 0FH
1	0	2	10H až 17H
1	1	3	18H až 1FH

Tabulka 1.

RS1, RS0 - Určují banku, jejíž registry R0 až R7 budou používány obr.4 (například k nepřímému adresování atd.). Jednotlivé bity RS1 a RS0 lze ovládat programově pomocí logických, přesunových nebo bitových operací.

Procesor není vybaven instrukcí pro přepínání bank a po jeho vynulování je aktivní bankou banka 0 (RS1= RS0=0).

OV - Příznak přetečení (Overflow) indikuje přetečení při aritmetické operaci sčítání nebo odečítání, jestliže zpracovávaná čísla považujeme za čísla se znaménkem. Jedná se o případ, kdy součet dvou záporných čísel je kladný (došlo k přenosu mezi bity b_8 a b_7 a nedošlo k přenosu mezi bity b_7 a b_6) nebo součet dvou kladných čísel je záporný (nedošlo k přenosu mezi bity b_8 a b_7 a došlo k přenosu mezi bity b_7 a b_6), kde bit b_7 představuje znaménko. Příznak je též využíván při operaci dělení k identifikaci dělení nulou a při instrukci násobení.

P - Příznak parity (Parity flag) indikuje lichou paritu střadače. Je-li ve střadači lichý počet jedniček, potom příznak parity je nastaven P=1. Příznak je aktualizován po každé instrukci.

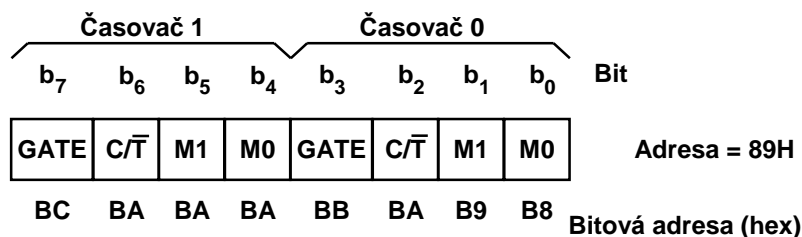
SP - Ukazatel zásobníku (Stack pointer) je osmibitový a na rozdíl od většiny procesorů je při plnění zásobníku inkrementován (hodnota ukazatele je zvětšována o jedničku). Vlastní zásobník je umístěn ve vnitřní datové paměti RAM a může být umístěn kdekoliv v této paměti tj. i v rozšířené části vnitřní datové paměti (adresy 128H až 255H u následovníků 8051), která je přístupná pouze pomocí nepřímého adresování. Po vynulování je nastaven na hodnotu SP=07H a ve většině aplikací musí být pomocí odpovídající instrukce přestaven na jinou hodnotu. **DPL, DPH** - Registry DPL a DPH tvoří nižší a vyšší 8 bitovou slabiku 16-bitového ukazatele DPTR, který slouží k 16-bitovému nepřímému adresování vnější datové paměti RAM nebo paměti programu.

PC - Čítač instrukcí (Program counter) je 16-bitový čítač instrukcí, který není přímo programově přístupný.

Čítače/časovače

Mikroprocesor 8051 obsahuje dva 16-bitové čítače, jejichž obsah je přístupný pomocí paměťově mapovaných registrů **TH0, TL0** (čítač 0) a **TH1, TL1** (čítač 1), které určují vyšší a nižší slabiku (8 bitů) příslušného čítače.

Hodinový synchronizační signál čítačů může být odvozen z oscilátoru procesoru nebo z vnějšího zdroje pomocí vstupů procesoru T0 a T1. Je-li zdrojem signálu vnitřní oscilátor procesoru, potom čítač je ve funkci **časovače** a přičítá jedničku za



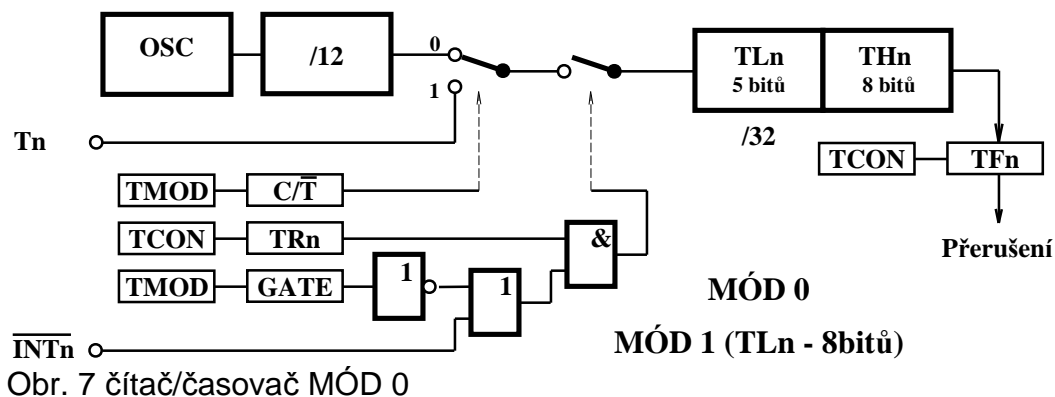
Obr.6 Rozložení bitů v registru TMOD

každý strojový cyklus, který je tvořen 12 periodami oscilátoru. Ve funkci **čítače vnějších událostí** se obsah příslušného registru zvyšuje o jedničku vždy při přechodu signálu T_n z 1→.0. Vstupy T_0 a T_1 se testují během stavu $S5P2$ každého strojového cyklu. Zjistí-li se v jednom cyklu úroveň log.1 a v příštím log.0 přičte se k obsahu čítače jednička. Nová hodnota je v čítači nastavena v době $S3P1$ následujícího cyklu za cyklem, ve kterém byla zjištěna změna. Protože zjištění změny na vstupech T_n trvá 2 strojové cykly (24 period oscilátoru), je maximální čítaný kmitočet vnějšího signálu $1/24$ kmitočtu oscilátoru mikropočítače. Logická úroveň čítaného signálu musí zůstat nezměněna vždy alespoň 1 celý strojový cyklus. Konfiguraci čítače/časovače 0 a 1 zajišťujeme naprogramováním registru $TMOD$. Vlastní čítače se programově spouští nebo zastavují nastavením nebo vynulováním bitu TR_n v registru $TCON$.

TMOD - Registr módu časovačů / čítačů (Timer/Counter mode control) se skládá ze dvou čtveřic bitů obr.6 příslušejících každému ze dvou čítačů/časovačů. Význam jednotlivých bitů je následující:

GATE - Řízení hradlování. Je-li $GATE=1$, potom čítač/časovač n je aktivován (čítá) při vstupu $\overline{INT}_n = 1$ a $TR_n=1$, kde TR_n je bit z registru $TCON$. V tomto režimu je činnost čítače ovlivňována nejenom programově pomocí bitu TR_n , ale zároveň i pomocí vnějšího signálu přivedeného na vstup \overline{INT}_n obr.7. Je-li $GATE=0$, potom čítač/ časovač n je aktivní pro $TR_n=1$ (čítač je řízen pouze programově).

C/T - Volba čítač/časovač. Bit rozhoduje o zdroji hodinových impulzů, které bude čítač počítat. Je-li $C/\overline{T} = 0$, potom se jedná o režim časovače a hodinový signál je vytvořen z hodinového synchronizačního signálu procesoru vydělením hodnotou 12. Je-li $C/\overline{T} = 1$, potom se jedná o režim čítače a hodinový signál přichází ze vstupu T_n .



M1,M0 - Volba jednoho ze čtyř módů čítače/časovače.

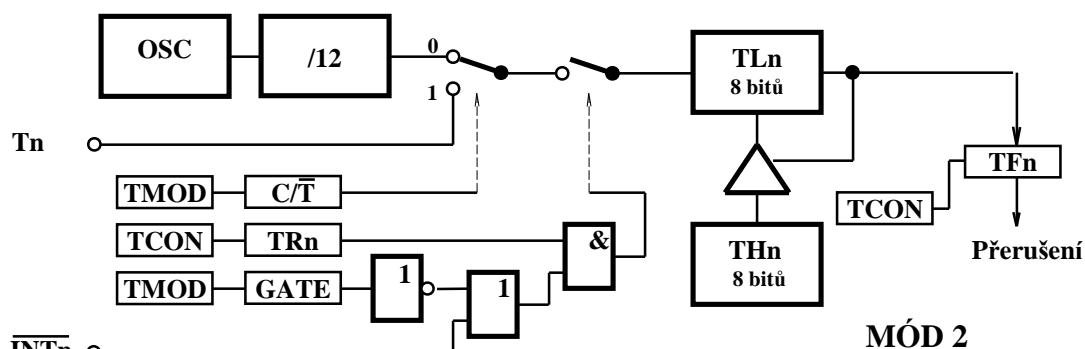
Mód 0 - $M_0=M_1=0$. Režim čítače/časovače je shodný s režimem čítače na procesoru 8048. Osmibitový čítač TH_n čítá hodinové impulzy vydělené 5 bitovým předděličem tvořeným spodními bity čítače TL_n obr.7. Oba čítače čítají vzestupně a vytváří 13-bitový čítač, který při přetečení (přechodu ze samých jedniček na samé nuly) nastaví příslušný příznakový bit TF_n v registru $TCON$, které mohou být využívány jako zdroje přerušeni procesoru. Vstup synchronizačního

signálu do časovače je povolen tehdy, je-li $TR_n=1$ (čítač je spuštěn) a současně s tím je $GATE=0$ nebo $\overline{INT1}=1$. Časovač 1 může být též využíván ke generování přenosové rychlosti sériového kanálu v módu 1 a 3.

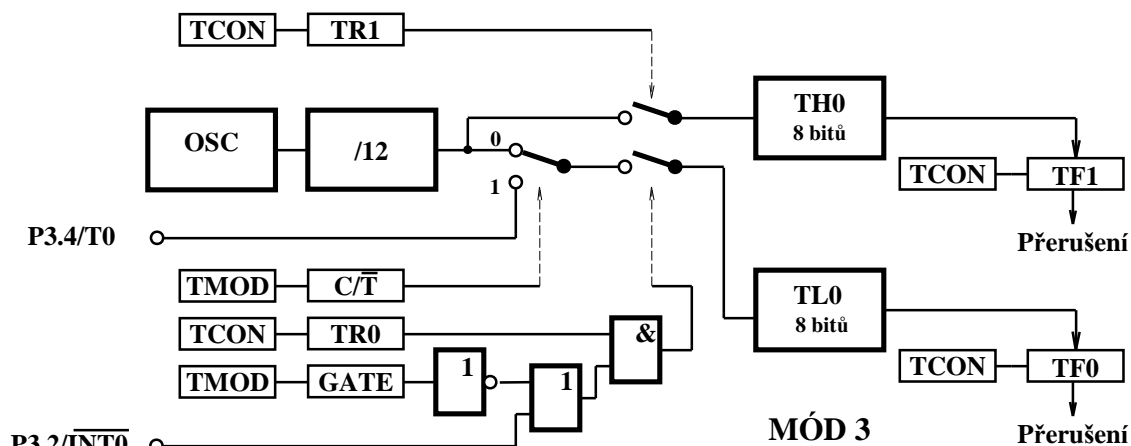
Mód 1 - $M0=1, M1=0$. Mód 1 je shodný jako mód 0 s tím rozdílem, že čítače TH_n a TL_n jsou 8-bitové a vytváří tak 16-bitový čítač. Dojde-li k přechodu ze samých jedniček na samé nuly, nastaví se příznak TF_n .

Mód 2 - $M0=0, M1=1$. V módu 2 pracuje registr TL_n jako čítač s obvodovým přednastavením na hodnotu uloženou v registru TH_n při přetečení čítače TL_n obr.8. Programové nastavení nové hodnoty v registru TH_n neovlivňuje současný stav čítače TL_n .

Mód 3 - $M0=M1=1$. V předcházejících módech byla funkce časovače 0 a 1 shodná. V módu 3 dochází u čítače 0 k jeho rozštěpení na dva samostatné osmibitové čítače $TL0$ a $TH0$. Čítač $TL0$ využívá standardní signály C/\overline{T} , $GATE$, $TR0$, $\overline{INT0}$ a $TF0$. tečení nastavuje příznak $TF1$ obr.9. Pracuje-li čítač 0 v režimu 3, potom čítač 1 může pouze generovat přenosovou rychlost pro sériový kanál nebo být využit v aplikaci, která nevyužívá přerušeni. Protože bit $TR1$ je využit pro řízení čítače 0 je zastavení nebo spuštění čítače 1 ovládáno jeho nastavením do módu 3 nebo zrušením módu 3.



Obr.8 Čítač/časovač n v módu 2 - 8-bitový čítač s přednastavením



Obr.9 Čítač/časovač 0 v módu 3 - dva 8-bitové čítače

b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀	Bit	TCON - Registr řízení
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	Adresa RAM = 88H	čítače/časovače se skládá ze čtyřech bitů příslušejících oběma časovačům a čtyřech bitů patřících vstupům vnějšího přerušeni. Na obr.10 je
8F	8E	8D	8C	8B	8A	89	88	Bitová adresa (hex)	

Obr.10 Přiřazení bitů v registru řízení čítačů

zobrazeno jejich umístění v registru TCON a mají následující význam:

TF0, TF1 - Přetečení čítače/časovače. Bit je nastaven při přechodu časovače z maximální hodnoty do nuly a je automaticky vynulován při přechodu procesoru do odpovídajícího obslužného podprogramu přerušeni.

TR0, TR1 - Spuštění čítače/časovače. Bit, který je ovládán programově, zajišťuje spuštění nebo zastavení příslušného čítače. Je-li GATE=1 v registru TMOD příslušného čítače, potom o přivedení nebo přerušeni synchronizačního hodinového signálu do čítače rozhoduje vstupní signál \overline{INRn} .

IE0, IE1 - Přijetí vnějšího přerušeni. Příslušný bit je nastaven při sestupné hraně nebo úrovni log.0 na vstupu vnějšího přerušeni \overline{INRn} v závislosti na stavu konfiguračního bitu ITn. Po přechodu procesoru do obslužného podprogramu příslušného přerušeni je bit automaticky vynulován.

IT0, IT1 - Konfigurace aktivace vnějšího přerušeni. Je-li ITn=1, potom žádost o vnější přerušeni je aktivována sestupnou hranou signálu na vstupu \overline{INTn} . Je-li ITn=0, potom žádost je aktivována úrovní log.0 na vstupu \overline{INTn} . Je-li signál na vstupu \overline{INTn} po dlouhou dobu v log.0, může být přerušeni vyvoláno i několikrát za sebou, pokud doba přerušeni je kratší než délka impulsu na vstupu \overline{INTn} . Bity se nastavují a nulují programově.

Čítač/časovač 2

Čítač/časovač 2 je implementován v klonech procesoru 8051 počínaje verzí 8052. Může pracovat jako čítač, nebo časovač, funkce je vybírána bitem $C/\overline{T}2$ speciálního funkčního registru T2CON. Čítač/časovač T2 má tři provozní režimy (viz obr. A1 a A2)

- záchytný (capture)
- s automatickým přednastavením (autoreload)
- generátor přenosové rychlosti pro seriový kanál

V obvodu 8052 jsou pro realizaci a řízení čítače/časovače T2 implementovány další speciální funkční registry. Tyto registry jsou na obr. A3

Provozní režimy T2 jsou voleny řídicími bity **RCLK, TCLK, CP/RL2** a **TR2** podle následující tabulky.

RCLK+TCLK	CP/RL2	TR2	REŽIM
0	0	1	šestnáctibitový a automatickým přednastavením
0	1	1	šestnáctibitový záchytný režim
1	x	1	generátor přenosové rychlosti

adresa	značení	funkce
0C8H	T2CON	řízení T2
0CDH	TH2	T2 horní Byte
0CCH	TL2	T2 dolní Byte
0CBH	RCAP2H	záchytný registr (horní B)
0CAH	RCAP2L	záchytný registr (dolní B)

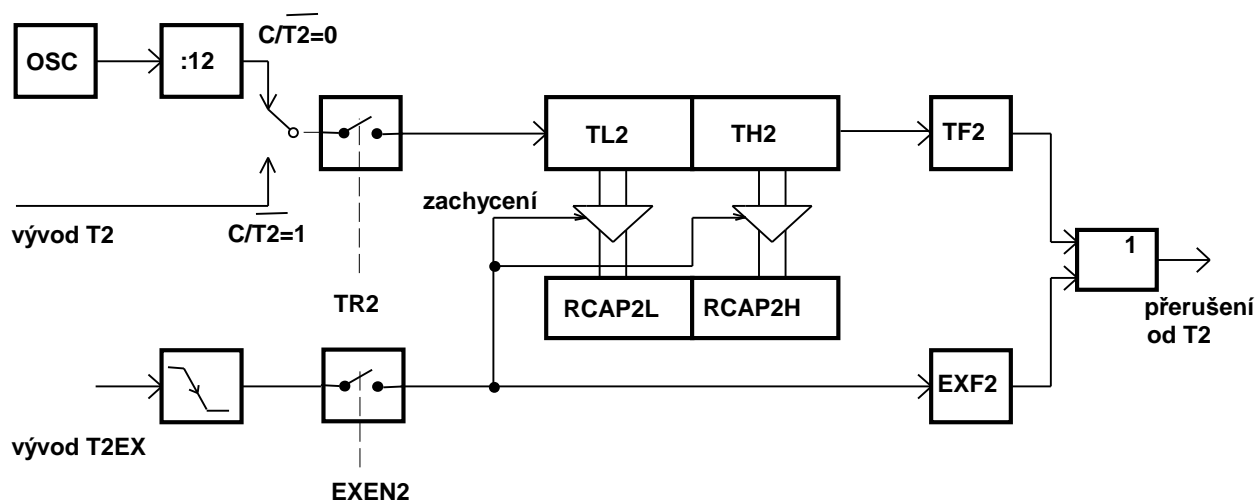
Obr. A3 Rozšíření SRF 8052

x	x	0	vypnuto
---	---	---	---------

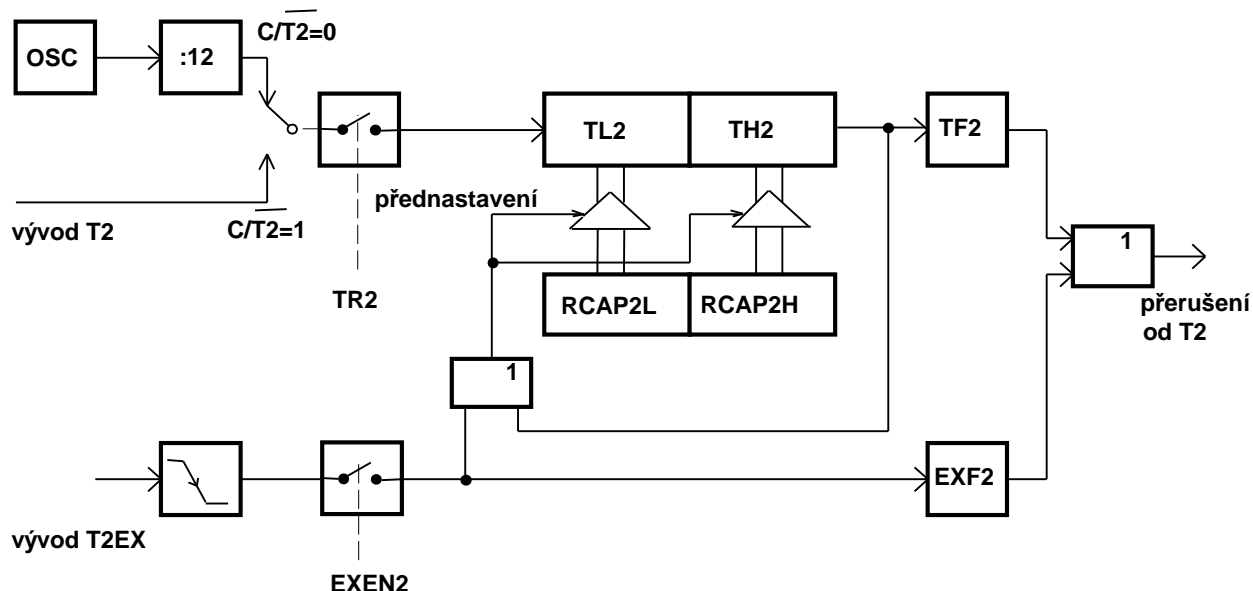
V záchytném režimu jsou možné dva způsoby provozu, které řídí bit **EXEN2** registru **T2CON**.

Pro **EXEN2 =0** pracuje T2 jako 16-ti bitový čítač/časovač, který po přetečení nastaví indikátor **TF2** a ten, pokud je to povoleno, generuje požadavek obsluhy přerušení...

Je-li bit **EXEN2=1**, pracuje T2 stejně jako v předchozím případě, pouze s jednou funkcí navíc.. Sestupná hrana vnějšího signálu, připojeného na externí vstup **T2EX** řídí zápis okamžitého obsahu registrů **TL2** a **TH2** (**T2**) do záchytných registrů **RAP2L** a **RCAP2H** a navíc příznak **EXF2**, který může generovat přerušení .



Obr. A1 Záchytný režim T2



Obr. A2 Automatické přednastavení T2

Obsah řídicího registru časovače T2 T2CON je v následující tabulce, význam řídicích bitů registru T2CON je uveden v legendě následující tabulky.

MSB					LSB		
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2

-TF2 je příznak přetečení T2, je nastavován při řetečení, nulovat je ho nutné programově nemůže být nastaven pokud RCLK, nebo TCLK jsou rovny jedné

EXF2 je příznak T2 registrující vnější událost (sestupnou hranu na vývodu T2EX). Nastavuje se pouze při EXEN2=1, nulován je programově

RCLK je příznak přijímacích hodin seriového kanálu. Je-li nastaven používá seriový kanál v režimech 1 a 3 k časování přenosu jako časovou základnu impulsy od přetečení T2, pokud

je tento příznak nulový časuje příjem seriového kanálu časovač T1

TCLK je příznak vysílacích hodin seriového kanálu. Je-li nastaven používá seriový kanál v režimech 1 a 3 k časování přenosu jako časovou základnu impulsy od přetečení T2, pokud je tento příznak nulový časuje vysílání seriového kanálu časovač T1

EXEN2 je příznak , který uvolňuje externí řízení zápisu do záchytného registru a to pouze pokud T2 generuje hodiny pro seriový kanál TR2

řídící bit spuštění($TR2=1$), nebo zastavení T2 ($TR2=0$)

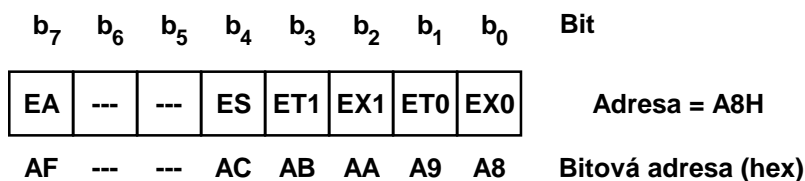
CT/\bar{T} volba časovač $CT/\bar{T}=0$ (vnitřní hodiny = $f/12$), nebo čítač vnějších událostí $CT/\bar{T}=1$

T2 čítá na sestupnou hranu

$CP/\bar{RL2}$ příznak zachycení (přednastavení). Pokud $CP/\bar{RL2}=1$ a současně $EXEN2=1$, potom při sestupné hraně signálu na vývodu T2EX(P1.1) dochází k přepisu registrů T2 (T2L a T2H) do záchytných registrů RCAP2L a RCAP2H. Pokud je tento příznak nulový a současně $EXEN2=1$ dochází se sestupnou hranou signálu na vývodu T2EX, nebo při přetečení T2 k přednastavení registrů T2 hodnotou, kterou obsahují záchytné registry RCAP2L a RCAP2H. Pokud je ale nastaven některý z bitů RCLK, nebo TCLK je externí signál ignorován stejně jako samotný příznak $CP/\bar{RL2}$ a T2 je přednastaven při každém přetečení.

Přerušení

Pro snažší komunikaci s vnějšími periferiemi je mikroprocesor 8051 vybaven přerušovacím systémem s pěti zdroji přerušení. U následovníků 8051 se počet zdrojů přerušení v závislosti na počtu vnitřních periferií zvětšuje a dosahuje až hodnoty 15. Vnější přerušení $\overline{INT0}$ a $\overline{INT1}$ mohou být vyvolána buď logickou úrovní (log.0) nebo změnou logické úrovně (sestupnou hranou 1→0). Vznikne-li vnější přerušení, je nastaven příslušný příznak IE0, IE1, který je obvodově automaticky vynulován při vyvolání obslužného podprogramu. Přerušení od časovače 0 a 1 se vyvolávají nastavením příznaků TF0 a TF1, které indikují přetečení příslušného čítače. Vyvolá-li se přerušení od časovače, potom odpovídající příznak TF_n je vynulován při vyvolání příslušného obslužného podprogramu. Přerušení od sériového kanálu se generuje logickým součtem příznaků RI a TI. Aby uživatel mohl zjistit, zda přerušení bylo generováno příznakem RI (příjem) nebo TI (vysílání), nejsou příznaky automaticky obvodově nulovány při přechodu do obslužného podprogramu. V obslužném programu je programátor nucen nejprve stanovit příčinu přerušení (od RI nebo TI) a potom příslušný příznak programově vynulovat. V obslužném podprogramu pro seriový kanál tak zároveň programově rozhoduje o tom, která žádost (RI nebo TI) bude zpracována dříve a bude tak mít vyšší prioritu. Všechny příznaky, které generují přerušení, mohou být programově vynulovány nebo nastaveny. To znamená, že přerušení mohou být generována i programově nebo vyžaduje-li to situace mohou být nevyřízené žádosti o přerušení programově zrušeny. Každý ze zdrojů přerušení je možné individuálně povolit nebo zakázat nastavením nebo vynulováním příslušného bitu v registru speciálních funkcí IE.



Obr.11 Rozložení bitů povolení přerušení

IE - Registr povolení přerušení (Interrupt enable) se skládá z 5 bitů příslušejících každému z pěti zdrojů přerušení a jednoho bitu zajišťujícího globální zakázání všech přerušení nebo povolení všech povolených přerušení označeného EA. Na obr.11 je zobrazeno umístění jednotlivých bitů v registru povolení přerušení včetně jejich přímých bitových adres s jejichž pomocí mohou být individuálně ovlivněny bez nutné znalosti stavu ostatních bitů v registru. Význam jednotlivých bitů je následující:

EA - Globální povolení přerušení (Enable all). Je-li EA=0, je celý přerušovací systém zablokován a nemůže být přijata jakákoliv žádost o přerušení. Je-li EA=1, potom mohou být přijaty žádosti těch přerušení, která mají nastavený (log.1) svůj povolovací bit tj. ES, ET1, ...atd.

ES - Povolení přerušení od sériového kanálu. Je-li ES=1 a zároveň EA=1, je přerušení od příjmu a vysílání sériového kanálu povoleno.

ET0, ET1 - Povolení přerušení od čítače/časovače 0,1. Je-li ETn=1 a zároveň EA=1, potom je povoleno přerušení způsobené přetečením čítače/časovače n (n=0 nebo 1).

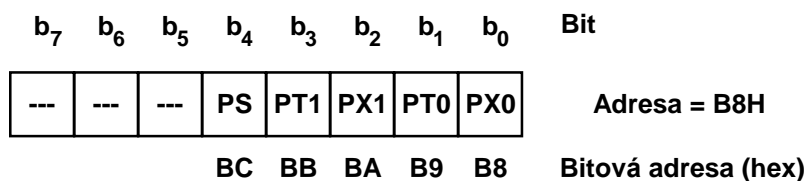
EX0, EX1 - Povolení vnějšího přerušení $\overline{INT0}$, $\overline{INT1}$. Je-li EXn=1 a zároveň EA=1, potom je povoleno přijetí vnější přerušení ze vstupu \overline{INTn} . O tom, zda přerušení bude vyvoláno sestupnou hranou signálu na vstupu \overline{INTn} nebo úrovní log.0, rozhoduje bit ITn v registru TCON.

Pro využití přerušovacího systému mikroprocesoru je nezbytné znát přiřazení priorit jednotlivým přerušením. U procesoru 8051 se výrobce rozhodl pro pevně přidělené priority jednotlivým přerušením v každé ze dvou úrovní priority. V rámci každé úrovně priority (nižší nebo vyšší) má nejvyšší prioritu vnější přerušení $\overline{INT0}$ (IE0) následované časovačem 0 (TF0), vnějším přerušením $\overline{INT1}$ (IE1), časovačem 1 (TF1) a sériovým kanálem (RI a TI), které má nejnižší prioritu. Potřebuje-li uživatel z nějakého důvodu přiřadit jednomu z přerušení nejvyšší prioritu, může mu přiřadit vyšší úroveň priority nastavením odpovídajícího bitu v registru **IP - registru priority přerušení** do log.1. Na obr.12 je zobrazeno umístění bitů úrovně priority jednotlivých zdrojů přerušení včetně jejich bitových adres, na které se můžeme odkazovat uvedenými symbolickými názvy, kde PS přísluší sériovému kanálu, PT0 a PT1 časovačům 0 a 1 a PX0, PX1 vnějším vstupům přerušení $\overline{INT0}$, $\overline{INT1}$. Přiřazením vyšší úrovně priority dosáhneme toho, že i přerušení s nižší úrovní priority bude přerušeno v důsledku žádosti přerušení s vyšší úrovní. Je-li již jedno přerušení vyvoláno (probíhá obslužný podprogram) nemůže být přerušeno přerušením třeba s vyšší prioritou v dané úrovni priority a musí počkat až na jeho dokončení. Potom budou podle vzájemné priority v dané úrovni postupně zpracovány další žádosti o přerušení. Z toho vyplývá, že přerušení s vyšší úrovní priority již nemůže být přerušeno. Otázka priority přerušení se stává důležitou v

případech současného přijetí dvou a více žádostí o přerušeni a u systémů využívajících velkého počtu přerušeni. I když volba jednotlivých priorit přerušeni v dané úrovni vychází ze zkušeností z mnoha realizací mikroprocesorových systémů, nemusí v případě využití mnoha přerušeni vyhovovat pouze dvouúrovňové členění. Proto u některých výkonných následovníků (Siemens 80C537) je přerušovací systém členěn do čtyř úrovní priority, který uživatelí poskytuje přeci jenom větší možnosti změn v prioritách jednotlivých přerušeni.

	Zdroj přerušeni	Adresa
IE0	Vnější přerušeni 0	0003H
TF0	Čítač/časovač 0	000BH
IE1	Vnější přerušeni 1	0013H
TF1	Čítač/časovač 1	001BH
RI +TI	Sériový kanál	0023H

Tabulka 2.



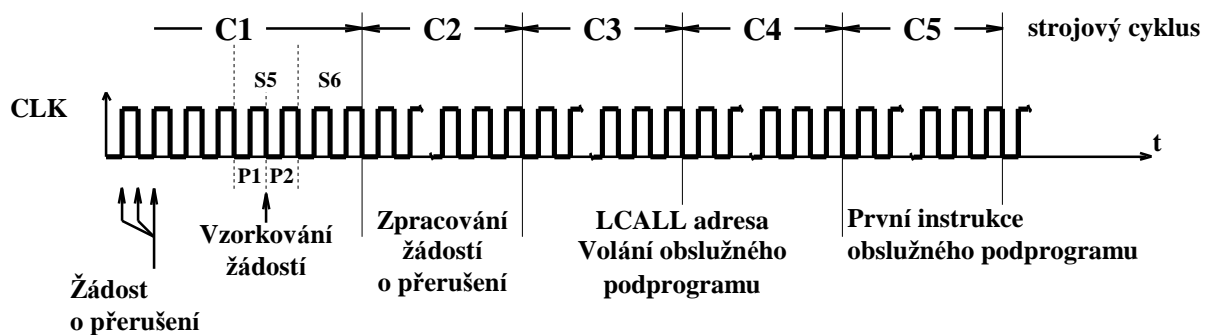
Obr.12 Přiřazení bitů úrovně priority přerušeni v IP

K využití přerušovacího systému musíme ještě znát jakým způsobem je realizováno volání příslušného obslužného podprogramu přerušeni.

U procesorů z řady 8051 má každý zdroj přerušeni definovanou pevnou adresu, na kterou po přijetí žádosti o přerušeni, dokončení probíhající instrukce a uložení návratové adresy (adresy následující instrukce za právě dokončenou instrukcí) do zásobníku, předá procesor řízení. Adresy přerušeni jsou od sebe vzdáleny pouze o 8 adresových míst viz. tab.2 a proto na nich bývá obvykle uložena pouze instrukce nepodmíněného skoku na vlastní obslužný podprogram zakončený instrukcí RETI (Return from interrupt).

Žádosti o přerušeni (příznaky IE0,IE1,TF0,TF1,RI+TI) se vzorkují v době S5P2 každého strojového cyklu procesoru a vyhodnocují se v následujícím cyklu. Je-li některý z příznaků nastaven a není splněna žádná z podmínek zabraňujících vyvolání obslužného podprogramu, přerušovací systém provede instrukci LCALL (dlouhé volání podprogramu) na příslušnou adresu určenou tab.2. Kromě individuálního nebo globálního zakázání přerušeni v registru IE, může jeho vyvolání oddálit právě probíhající přerušeni se stejnou nebo vyšší úrovní priority, doposud nedokončená instrukce nebo právě probíhající instrukce RETI (návrat z přerušeni) nebo instrukce zasahující do registrů IE a IP. Druhá podmínka zajišťuje, že před přechodem do obslužného podprogramu se rozdělaná instrukce nejprve dokončí. Poslední podmínka zajišťuje, že po instrukci RETI nebo instrukci zasahující do registrů IE a IP je vykonána ještě jedna instrukce a teprve potom je provedeno směřování na příslušný vektor přerušeni. Cyklus vyhodnocení se opakuje v každém

strojovém cyklu a vyhodnocují se v něm platné (navzorkované) hodnoty v periodě S5P2 předchozího strojového cyklu. Není-li žádost o přerušeni obsloužena, protože byla platná některá z blokovacích podmínek, pak v době, kdy už není aktivní se neobslouží, i když podmínky blokování zanikly. Žádost neobslouženého příznaku přerušeni se nikde neuchovává (vyjma příznaku), a v každém cyklu vyhodnocení se pracuje s novými hodnotami získanými v předcházejícím strojovém cyklu. Aby vzorkování vnějších přerušeni $\overline{INT0}$, $\overline{INT1}$ bylo správné, musí při aktivaci přerušeni sestupnou hranou ($0 \rightarrow 1$) trvat hodnota log.1 i log.0 na vstupech $\overline{INT0}$, $\overline{INT1}$ alespoň 12 period oscilátoru (jeden strojový cyklus). V případě aktivace nízkou úrovní signálu, musí žádost trvat tak dlouho, dokud nedojde k přechodu do obslužného podprogramu. Nebude-li žádost v průběhu obslužného podprogramu zrušena, potom dojde k opětovnému vyvolání přerušeni.



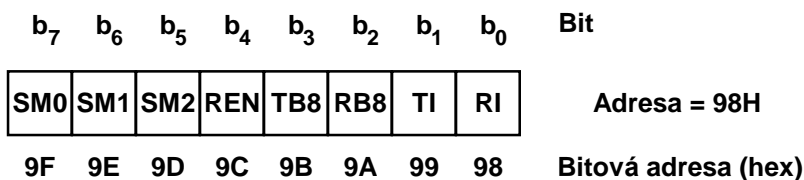
Obr.13 Nejrychlejší odezva na přerušeni

Na obr.13 je zobrazena situace při přijetí žádosti o přerušeni v případě, kdy procesor zpracovává jednocyklové instrukce nepracující s registry IE a IP. V cyklu C1 před periodou S5P2 přichází žádost o přerušeni, která se vyhodnocuje v cyklu C2. Jsou-li splněny všechny podmínky pro vyvolání přerušeni, potom v cyklech C3 a C4 je generována instrukce LCALL na příslušnou adresu obslužného podprogramu. V cyklu C5 potom může být zpracována první instrukce obslužného podprogramu. Vznikne-li žádost o přerušeni s vyšší prioritou před periodou S5P2 strojového cyklu C3, potom bude přerušeni obslouženo podle výše uvedených pravidel během strojových cyklů C5 a C6, aniž se vykoná jediná instrukce obslužného podprogramu s nižší prioritou. Mezi vznikem požadavku na přerušeni a první instrukcí obslužného programu proběhnou nejméně tři strojové cykly. Odezva na přerušeni se prodlouží při platné podmínce blokující přechod do obslužného podprogramu. V případě probíhajícího přerušeni se stejnou nebo vyšší úrovní priority, je doba čekání závislá na délce a vlastnostech obslužného podprogramu. Není-li vyhodnocení žádosti prováděno v posledním cyklu instrukce, potom prodloužení odezvy nebude větší než 3 strojové cykly (instrukce MUL a DIV trvají 4 cykly). Probíhá-li právě instrukce RETI nebo instrukce operující s IE nebo IP, prodlouží se doba odezvy nejvíce o 5 strojových cyklů (1 cyklus pro dobíhající instrukci a 4 cykly za dokončení nejdelší následující instrukce MUL nebo DIV). Využíváme-li v systému pouze jedno přerušeni, potom časová odezva bude vždy delší než 3 a kratší než 9 strojových cyklů.

Sériový kanál

Jednou z velkých výhod procesoru 8051, zvláště v době jeho vzniku, byl a je plně duplexní sériový kanál integrovaný na čipu procesoru umožňující komunikaci ve standardním 8 a 9-bitovém asynchronním režimu nebo 8-bitovém synchronním režimu s pevnou přenosovou rychlostí. Tím byla výrazně usnadněna komunikace s nadřazeným počítačem např. typu PC, k jejíž realizaci je v současné době zapotřebí jeden integrovaný obvod (MAX232, MAX233) zajišťující převod sériového kanálu v úrovních TTL na úroveň RS232C (V28). Plně duplexní sériový kanál umožňuje současně vysílat i přijímat hodnoty po tomto kanálu tvořeného minimálně 3 vodiči (RxD, TxD a Zem). Přijímací kanál je vybaven vyrovnávacím registrem, do kterého jsou uložena právě přijatá data, čímž je umožněn okamžitý příjem další dat. Přijatá data však musí být převzata dříve než je dokončen příjem následujících dat, který by způsobil přepsání původních dat. Procesor není vybaven příznaky indikující ztrátu přijatých dat (chybu přeplnění), chybu rámce a parity nebo indikaci přerušeni, které jsou obvyklé u specializovaných obvodů.

Přijímací i vysílací registr je přístupný na adrese registru speciálních funkcí se symbolickým označením **SBUF - registr sériového kanálu**. Zápisem se naplňuje vysílací registr, čtením SBUF je přečtena hodnota přijatá fyzicky odlišným posuvným přijímacím registrem přeepsaná do vyrovnávacího registru. Sériový kanál může pracovat ve čtyřech módech v závislosti na naprogramování registru SCON a nejvyššího bitu v registru PCON.



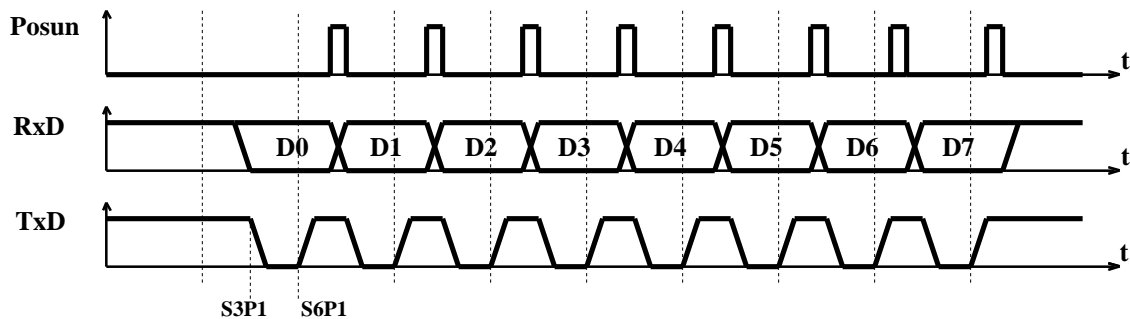
Obr.14 Rozložení bitů v registru SCON

SCON - Registr módu a řízení sériového kanálu se skládá z 8 bitů, jejichž umístění je zobrazeno na obr.14 a jejich význam je následující:

SM0, SM1 - Konfigurační bity určují jeden ze čtyř módů sériového kanálu, které jsou popsány tabulkou 3.

SM1, SM0	Mód	Typ přenosu	Bitová rychlost
0 0	0	Synchronní 8 bitový bez rám. synchronizace	$f_{osc} / 12$
0 1	1	8 bitový UART	časovač 1
1 0	2	9 bitový UART	$f_{osc} / 32, f_{osc} / 64$
1 1	3	9 bitový UART	časovač 1

Tabulka 3.



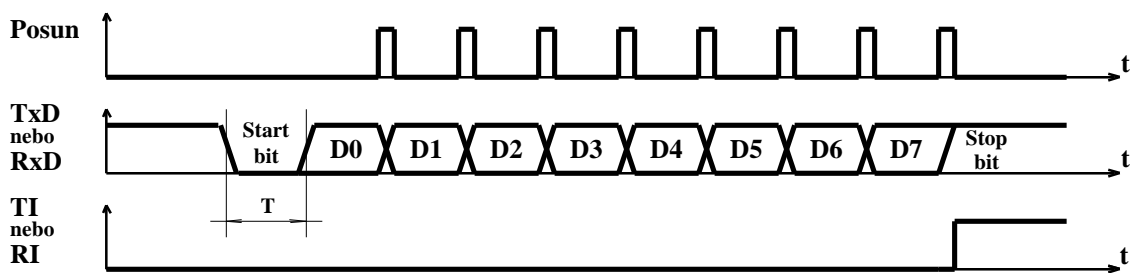
Obr.15 Časování sériového kanálu v módu 0

Mód 0 - Sériová data se vysílají nebo přijímají vstupem P3.0 označeném RxD synchronně s hodinovým posouvacím signálem vysílaným na výstupu TxD=P3.1. Vysílá se 8 datových bitů, počínaje bitem s nejmenší vahou obr.15. Přenosová rychlost je pevná a rovna 1/12 kmitočtu oscilátoru.

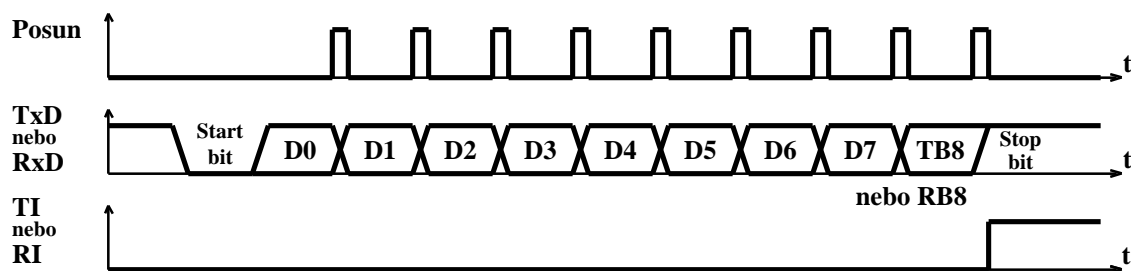
Mód 1 - 8-bitový UART obr.16. Hodnoty se vysílají výstupem TxD a přijímají vstupem RxD a skládají se z deseti intervalů určených převrácenou hodnotou přenosové rychlosti v baudech pro přenos jednotlivých bitů. První bit je vždy nulový (log.0) a představuje tzv. **start bit**, po němž následuje 8 přenášených bitů počínaje bitem s nejmenší vahou a posledním, který je vždy v log.1 a představuje tzv. **stop bit** (přesně jeden stop bit). Při příjmu se stop bit ukládá do bitu RB8 v registru SCON. Přenosová rychlost je volitelná a je určena periodou přetečení čítače/časovače 1 a hodnotou nejvyššího bitu v registru PCON. Pro přenosovou rychlost můžeme za předpokladu, že čítač 1 pracuje v módu 2, snadno odvodit tento vztah

$$\text{Přenosová rychlost} = \frac{1}{T} = \frac{2^{\text{SMOD}}}{32} * \frac{f_{\text{osc}}}{12 * [256 - (\text{TH1})]}$$

kde (TH1) je obsah registru TH1 a f_{osc} je kmitočet oscilátoru. Pro dosažení nízkých přenosových rychlostí se využívá čítač v módu 1, kde po vzniku přerušení provedeme programové přednastavení potřebnou 16-bitovou hodnotou.



Obr.16 Časování sériového kanálu v módu 1



Obr.17 Časování sériového kanálu v módu 2 a 3

Mód 2 - 9-bitový UART. Při vysílání je na vývodu TxD generováno 11 bitů v odpovídajících 11 intervalech daných přenosovou rychlostí uvedených start bitem (log.0) a ukončených jedním stop bitem (log.1) jako v předcházejícím módu. Přijímaný znak přichází přes vstup RxD. Devátým vysílaným bitem je hodnota bitu TB8 z registru SCON a přijatý 9-tý bit se ukládá do bitu RB8 v registru SCON a stop bit se ignoruje. Devátý bit může být využit k přenosu hodnoty (9 bitů) nebo k přenosu zabezpečovacího bitu například parity P (ochrany proti chybě v jednom bitu při přenosu), kterou získáme z registru PSW, jestliže ve střadači je uložena přenášená hodnota. Přenosová rychlost je dána $1/32$ nebo $1/64$ kmitočtu oscilátoru v závislosti na hodnotě bitu SMOD v registru PCON.

Mód 3 - 9-bitový UART s programovatelnou přenosovou rychlostí. Příjem i vysílání hodnot probíhá stejně jako v módu 2 s tím, že přenosová rychlost je určena periodou přetečení čítače/časovače 1 a nastavení bitu SMOD.

Ve všech čtyřech režimech se vysílání spouští instrukcí, která využívá SBUF jako cílový registr (zápisem do SBUF). Příjem v módu 0 se spouští podmínkami RI=0 a REN=1. V ostatních režimech se příjem spouští příchodem start bitu při REN=1.

SM2 - Bit povolující vytvoření víceprocesorové sériové sběrnice v módu 2 a 3. Je-li v módu 2 a 3 nastaven příznak SM2=1, pak se RI nenastaví (nemůže být vyvoláno přerušení), jestliže přijatý devátý bit (RB8) má hodnotu log.0. V módu 1 může být SM2 využit ke kontrole platnosti stop bitu a příjmu dat s jenom platným stop bitem. V módu 0 se bit SM2 nevyužívá.

REN - Povolení příjmu (REN=1). Bit se nastavuje i nuluje programově

TB8 - Devátý datový bit při vysílání. Vysílá se v módech 2 nebo 3. Nastavuje a nuluje se programově.

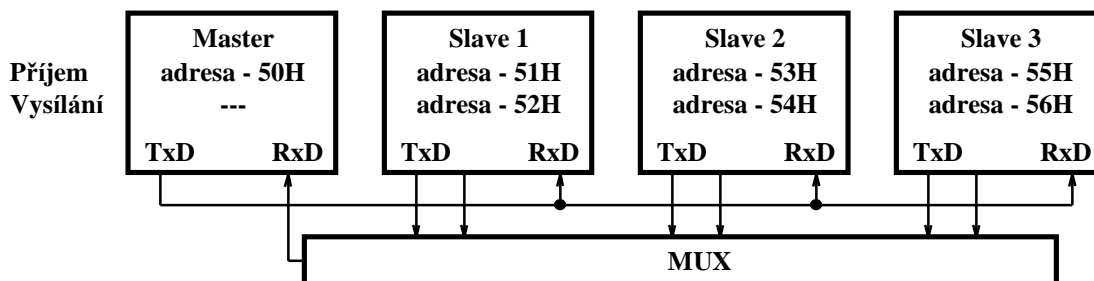
RB8 - Devátý datový bit při příjmu. Přijímá se v módech 2 nebo 3. V módu 1 při SM2=0, obsahuje RB8 přijatý stop bit. V módu 0 se RB8 nevyužívá.

TI - Příznak prázdného vysílacího posuvného registru se nastavuje obvodově v okamžiku vysílání 8 přenášeného bitu v módu 0 nebo na začátku vysílání stop bitu v ostatních módech. Příznak TI je společně s příznakem RI zdrojem žádosti o přerušení sériového kanálu a proto nemůže být z důvodu identifikace zdroje (od TI nebo RI) přerušení nulován obvodově. Proto uživatel sám musí po přijetí žádosti rozlišit, zda se jedná o žádost od příjmu (RI) nebo vysílání (TI), a teprve potom příslušný příznak programově vynulovat.

RI - Příznak přijatých platných dat se nastaví na konci příjmu 8 bitu v módu 0 nebo uprostřed přijímaného stop bitu v ostatních módech. Příznak se stejně jako TI nuluje programově, aby bylo možné rozlišit příčinu přerušení.

Multiprocessorová komunikace

Režim 2 a režim 3 je vybaven prostředky, které umožňují realizovat multiprocessorovou komunikaci. V těchto režimech se přijímá devět datových bitů, z nichž devátý bit se ukládá do



Obr.18 Jedna z možných multiprocessorových konfigurací s procesory 8051

bitu RB8 v řídicím registru sériového kanálu SCON. Sériový kanál lze naprogramovat pomocí bitu SM2 z registru SCOM tak, aby se při příjmu stop bitu aktivovalo přerušení od sériového kanálu jedině tehdy, je-li logická hodnota RB8=1. Této možnosti lze využít v klasické multiprocessorové konfiguraci (Master - slave) následujícím způsobem obr.18. Chce-li nadřízený procesor (master) přenášet do jednoho z podřízených procesorů (slave) blok dat, vyšle nejdříve adresový byte, který indentifikuje cílový procesor. Adresový byte se liší od datového bitu logickou hodnotou právě devátého bitu. U adresového bytu je v devátém bitu log.1, zatímco datový byte obsahuje v devátém bitu log.0. Je-li logická hodnota SM2=1, pak nebude žádný procesor přerušován datovým bytem, avšak adresový byte přeruší všechny podřízené procesory. Testováním přijatého bytu může každý procesor zajistit, zda je adresován a následující datové údaje jsou určeny pro jeho činnost. Adresovaný podřízený procesor vynuluje bit SM2 a připraví se k příjmu datových bytů. Ten podřízený procesor, který nebyl adresován, ponechá svůj bit SM2 nastaven a pokračuje v původní činnosti před přerušením a ignoruje přecházející datové byty. Formát takové zprávy, která se přenáší mezi procesory je tvořen adresou následovanou vlastní informací s délkou n bytů takto

$\langle \text{adresa} \rangle, \langle 1 \text{ byte dat} \rangle, \langle 2 \text{ byte dat} \rangle, \mathbf{K}, \langle n \text{ byte dat} \rangle$

Například pro přenos hodnot 55H a AAH z procesory master do slave 1 bude vypadat následovně

101010001	001010101	010101010
adresa	55H	AAH

Komunikace z podřízeného do řídicího mikroprocesoru bude probíhat obdobným způsobem s tím, že například jinou adresou vyzve nadřízený procesor jeden z podřízených procesorů k přenosu dat. Ten zajistí připojení výstupu TxD na vstup RxD nadřízeného procesoru a realizuje přenos, který je v takovéto konfiguraci nemusel být uveden adresou. Vzhledem k tomu, že výstupy brány 3, kde je

vývod TxD, jsou realizovány s otevřeným kolektorem, mohou být jednotlivé výtupy TxD paralelně spojeny a připojeny na vstup RxD nadřazeného procesoru. Podmínkou takového propojení je, že vysílání bude realizováno pouze jedním podřazeným procesorem.

Režimy se sníženou spotřebou

U aplikací, kde není vyžadována soustavná činnost a jsou kladeny přísné požadavky na spotřebu energie je obvykle nutné u procesorů využívat režimů se sníženou spotřebou. Procesory z řady 8051 umožňují v závislosti na tom, v jaké technologii jsou vyrobeny režimy se sníženým příkonem a se sníženým napájením. U starých procesorů vyrobených v technologii HMOS je snížení příkonu možné pouze v režimu s vypnutým napájením, při kterém chceme uchovat obsah paměti RAM na čipu. Pro tento účel byl procesor vybaven vstupem pro pomocný (záložní) zdroj napětí U_{PD} sdružený se vstupem nulování procesoru RST. Zjistí-li uživatelský systém, že dochází k poklesu napájecího napětí, musí přerušit činnost procesoru a přesunout důležité údaje do paměti RAM na čipu procesoru. Potom připojí zdroj pomocného napětí na vstup RST a vygeneruje tak nulování procesoru (RESET). Tato činnost musí být ukončena dříve než napětí U_{CC} poklesne pod spodní úroveň provozního napětí určeného výrobcem. Po opětovném naběhnutí napájení musí zůstat záložní napětí tak dlouho, dokud nedojde k spolehlivé činnosti oscilátoru procesoru (10ms), U_{CC} nedosáhne alespoň spodní úroveň provozního napětí. Po splnění všech podmínek může být provozní napětí odpojeno a procesor může zahájit normální činnost. Moderní procesory vyrobené v CHMOS technologii jsou vybaveny dva režimy se sníženým příkonem, do kterých lze procesor uvést nastavením příslušných bitů v registru PCOM. Vstupem, na který se přivádí záložní napětí, je vývod pro vlastní napájení U_{CC} .

b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀	Bit
SMOD	---	---	---	---	---	PD	IDL	Adresa RAM 87H

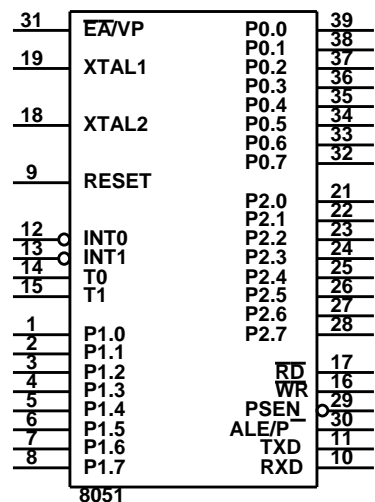
Obr.18 Rozložení bitů v registru PCON

PCON - Registr řízení napájení (Power control) obsahuje u jádra procesoru 80(C)51 jeden nebo tři bity obr.19. První odlišností tohoto registru od všech předcházejících je to, že jeho bity nejsou bitově adresovatelné a musí být jeho obsah měněn zápisem celého nového bytu. Nejvyšší bit označený SMOD nemá s řízením napájení nic společného a je pouze doplňujícím bitem pro řízení sériového kanálu SMOD sloužícím k zdvojnásobení jeho přenosové rychlosti. Jedná-li se o procesor, který je vyroben v HMOS technologii, potom je tento bit jediným bitem využívaným z registru PCOM. U procesorů vyrobených v technologii CMOS jsou využívány další dva bity označené PD a IDL, které slouží k uvedení procesoru do módu se sníženou spotřebou. Instrukce, která nastaví bitu **IDL** do log.1 je poslední instrukcí provedenou před přechodem do tzv. **Idle** módu, u kterého jsou všechny aktivity jádra procesoru zastaveny. Oscilátor i periferní obvody jako je sériový kanál, časovače, atd. pokračují v dál. Zůstává zpracován stav procesoru jako je ukazatel zásobníku, programový čítač, stavové slovo, střadač i zbývající speciální registry a vnitřní stav

datové paměti RAM. Stav vstupně/výstupních bran drží svůj logický stav, který měly v době přechodu do Idle módu. Signály ALE a $\overline{\text{PSEN}}$ se stávají neaktivní. Uvedený mód, při kterém klesne spotřeba obvodu na cca 15mA, lze opustit přijetím libovolného nezamaskovaného přerušení některé z periférií nebo vynulováním procesoru. Přijetí přerušení obvodově vynuluje bit IDL a procesor obslouží příslušné přerušení a přejde do normální činnosti. Po provedení instrukce RETI a ještě jediné instrukce může být znovu uveden do módu Idle. Nastavením bitu **PD** do log.1 uvedeme procesor do tzv. **Power down** módu, u kterého jsou zastavena činnost celého procesoru včetně periférií i oscilátoru a díky tomu se sníží odběr ze zdroje až na desítky μA . Stejně jako v předcházejícím módu je instrukce, která nastaví bit PD do log.1 poslední instrukcí provedenou před přechodem do power down módu. Zůstává zachován stav registrů speciálních funkcí a vnitřní paměti RAM. Oproti předcházejícímu módu nemohou být generována přerušení od vnitřních periférií a procesor může uvedený mód opustit pouze vynulováním. Tato funkce však znovu definuje obsah všech registrů speciálních funkcí, obsah paměti RAM se nezmění. V tomto režimu může být dále snížen odběr snížením záložního napájení až na hodnotu 3,3V. S výhodou tak lze využít obvodů MAX690 až MAX696 sdružujících watchdog, komparátor napětí a přepínač záložního napájení.

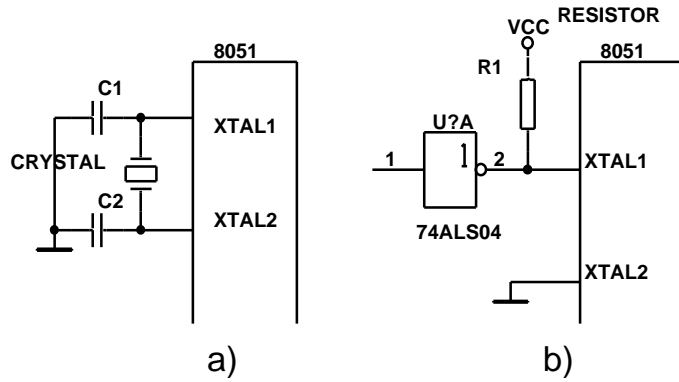
Zapojení vývodů mikroprocesoru 8051

Klasický mikroprocesor 8051 se vyráběl v plastickém nebo keramickém pouzdře DIL se čtyřiceti vývody obr.19, nyní se většinou využívá plastického pouzdra PLCC se čtyřiceti čtyřmi vývody. Ke své činnosti vyžaduje jedno napájecí napětí $U_{cc} = 5\text{V}$ a $U_{ss} = 0\text{V}$ a připojení piezokeramického rezonátoru ("krystalu") k vývodům XTAL1 a XTAL2 obvodu vnitřního oscilátoru. Procesor je vybaven 4 vstupně/výstupními branami P0 až P3 z nichž P0, P2 a P3 obsahují další funkce, které jsou závislé na tom, zda využíváme vnější paměť programu nebo dat, vnější přerušení, vnější vstupy časovačů nebo sériový kanál. Pro řízení vnější paměti je procesor



Obr.19

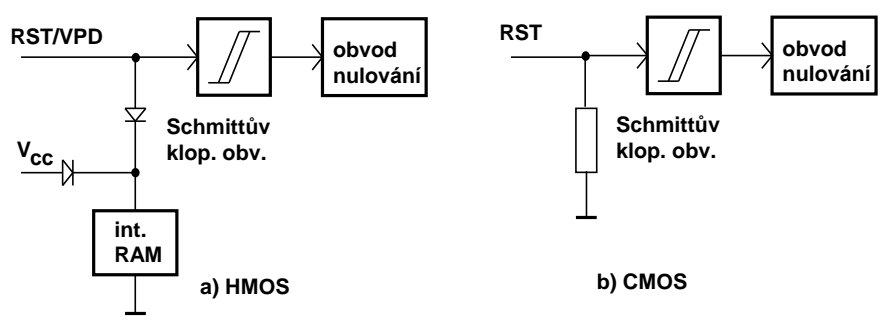
vybaven řídicími signály ALE (pro zápis spodní poloviny platné adresy A0 až A7) $\overline{\text{PSEN}}$ (pro čtení z vnější paměti programu) a $\overline{\text{RD}}$ a $\overline{\text{WR}}$ (pro čtení nebo zápis do vnější datové paměti). Kromě uvedených signálů je procesor vybaven již zmíněným vstupem EA, který určuje přístup k vnější programové paměti, a nulovacím vstupem RESET s aktivní úrovní v log.1. Na tomto místě je třeba upozornit na to, že řada následovníků má aktivní úroveň nulovacího impulsu opačnou (log.0), jak bývá u většiny procesorů obvyklé. Na obr.20 je zobrazen nejjednodušší nulovací obvod, který musí zajistit, aby $\text{RESET}=1$ nejméně po dva strojové cykly v době plné činnosti oscilátoru. Ačkoliv se úloha zdá být jednoduchá, je třeba si uvědomit, že napájecí zdroj může mít relativně pomalý náběh a vnitřní oscilátor je obvykle nastartován mezi 5ms až 10ms po dosažení napájecího napětí. Je třeba připomenout, že tento jednoduchý obvod není vhodný pro systémy se zálohovanou pamětí dat RAM nebo EEPROM. V okamžiku odpojení nebo poklesu napájecího



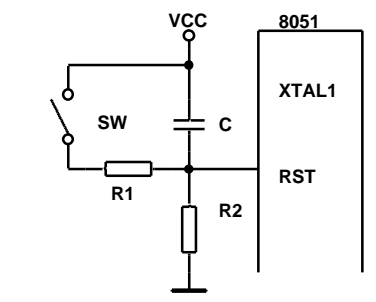
Obr.22 Synchronizace mikroprocesoru 8051

napětí nedojde k vytvoření nulovacího impulsu, který by vyřadil procesor z činnosti po dobu napájecího napětí mimo tolerance stanovené výrobcem. Necháme-li procesor v tomto rozsahu napájecího napětí pracovat, potom má "dostatek času" k přepsání zálohovaných dat. V takových případech musíme použít nulovací obvod (TL7705) nebo nulovací generátor s obvodem watchdog (MAX690 až MAX699). Obvod nulování u verze HMOS je spojen s vývodem nulování RST/VDP a

současně se jím přivádí záložní napájecí napětí při úsporném režimu pro paměť RAM. U verze CMOS se záložní napětí přivádí vývodem U_{cc}. Provedení RST vstupu pro obě verze je na obr. 21a), b). Obsahy registrů speciálních funkcí po resetu jsou



Obr.21



Obr. 20 Reset obvod

v následující tabulce .

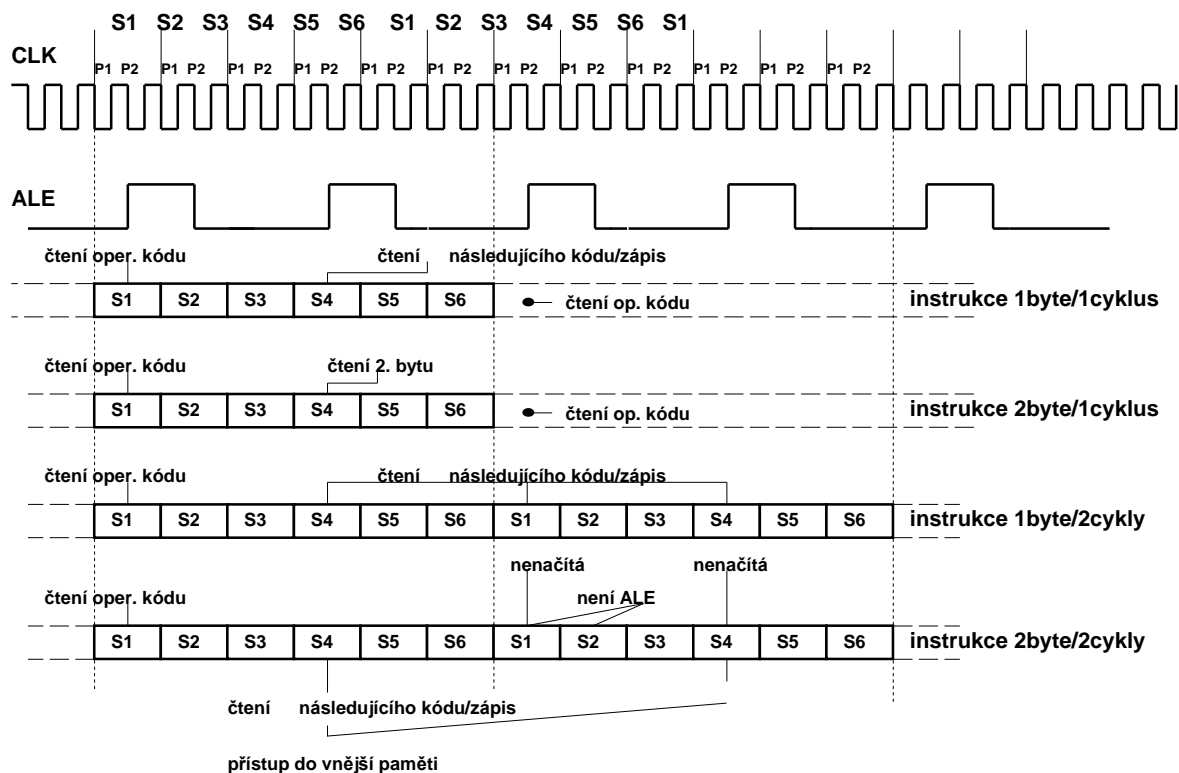
registr	obsah	registr	obsah	registr	obsah	registr	obsah
PC	000h	DPTR	0000h	TL0, 1	00h	SBUF	náhodný
ACC	00h	P0-P3	00h	SCON	00H	PCON/H	0xxxxxxxB
B	00h	TMOD	00h	T2CON	00H	PCON/C	0xxx0000B
PSW	00h	TCON	00h	RCAP2H	00H	IP	xx000000B
SP	07h	TH0, 1	00h	RCAP2L	00H	IE	0x000000B

Synchronizační signál je obvykle tvořen vnitřním oscilátorem, ke kterému připojujeme krystal dle obr.22a. Kmitočet oscilátoru u standardního mikroprocesoru 8051 se může pohybovat v rozsahu 1,2MHz až 12MHz. U následovníků zůstává spodní kmitočet shodný, ale nejvyšší kmitočet může dosáhnout až 33MHz. V takovém případě budeme-li využívat synchronizační kmitočty nad 20MHz, pak budeme většinou využívat vnější oscilátory, které připojíme k mikroprocesoru dle obr. 22 b.

Časování centrální procesorové jednotky

Strojový cyklus procesoru se skládá ze šesti stavů označených S1, S2, ... , S6, z nichž každý je dále rozdělen na dvě fáze P1 a P2. Každý strojový cyklus je tak tvořen 12 fázemi označovanými S1P1, S1P2, S2P1, ... , S6P2 shodnými s periodami synchronizačního oscilátoru. Na obr.23 je zobrazeno časování čtení a vykonání čtyř možných typů instrukcí procesoru 8051 vzhledem k jeho vnitřnímu časování, které je uživateli nedostupné. Pro lepší orientaci jsou na obr.23 zobrazeny i průběhy signálů ALE a XTAL2. Signál ALE se normálně aktivuje dvakrát během jednoho strojového cyklu. Poprvé ve fázi S1P2 až S2P1 a podruhé během S4P2 až S5P1 s výjimkou přístupu do vnější paměti dat (MOVX), kdy v druhém cyklu je vynechán první cyklus ALE. Z tohoto důvodu není vhodné využívat signál ALE k časování v navrhovaném systému.

Realizace jednocyklové instrukce začíná ve fázi S1P2 uložením přečteného operačního znaku do registru instrukcí. Ve stavu S4 ještě téhož strojového cyklu se provádí čtení ještě jednoho bytu z následujícího paměťového místa. Je-li přečtený byte využit v instrukci (dvoubytová a jednocyklová nebo dvoubytová a dvoucyklová instrukce), potom čítač instrukcí je inkrementován. Je-li zpracovávaná instrukce jednobytová a jednocyklová, potom přečtený byte ve stavu S4 je ignorován a čítač instrukcí se nemění. Instrukce je vykonána ve fázi S6P2. V případě jednobytové dvoucyklové instrukce je situace stejná jako u předcházejícího případu s tím, že čtení dalšího bytu je ignorováno třikrát. Jedinou výjimkou je instrukce přístupu do vnější paměti MOVX, která jako 1 bytová dvoucyklová instrukce negeneruje ve fázi S1P2 až S2P1 druhého cyklu signál ALE a nerealizuje oba výběry dalšího bytu (čtecí cyklus externí datové paměti obr.24.). Na obr. 24 je souhrně naznačeno časování externích periférií a interních rozhraní (bran a seriového kanálu). Nižší byte adresy externí datové paměti vystupuje bránou P0 v dobách S5P1 až S6P1 a musí být zapsána do záchytného registru signálem ALE, signál RD# trvá od S1P1 do S3P2, uvolní výstup datové paměti a ten je vzorkován v době S2P2 a přenesen na interní sběrnici procesoru. Horní byte adresy (obsah DPH)

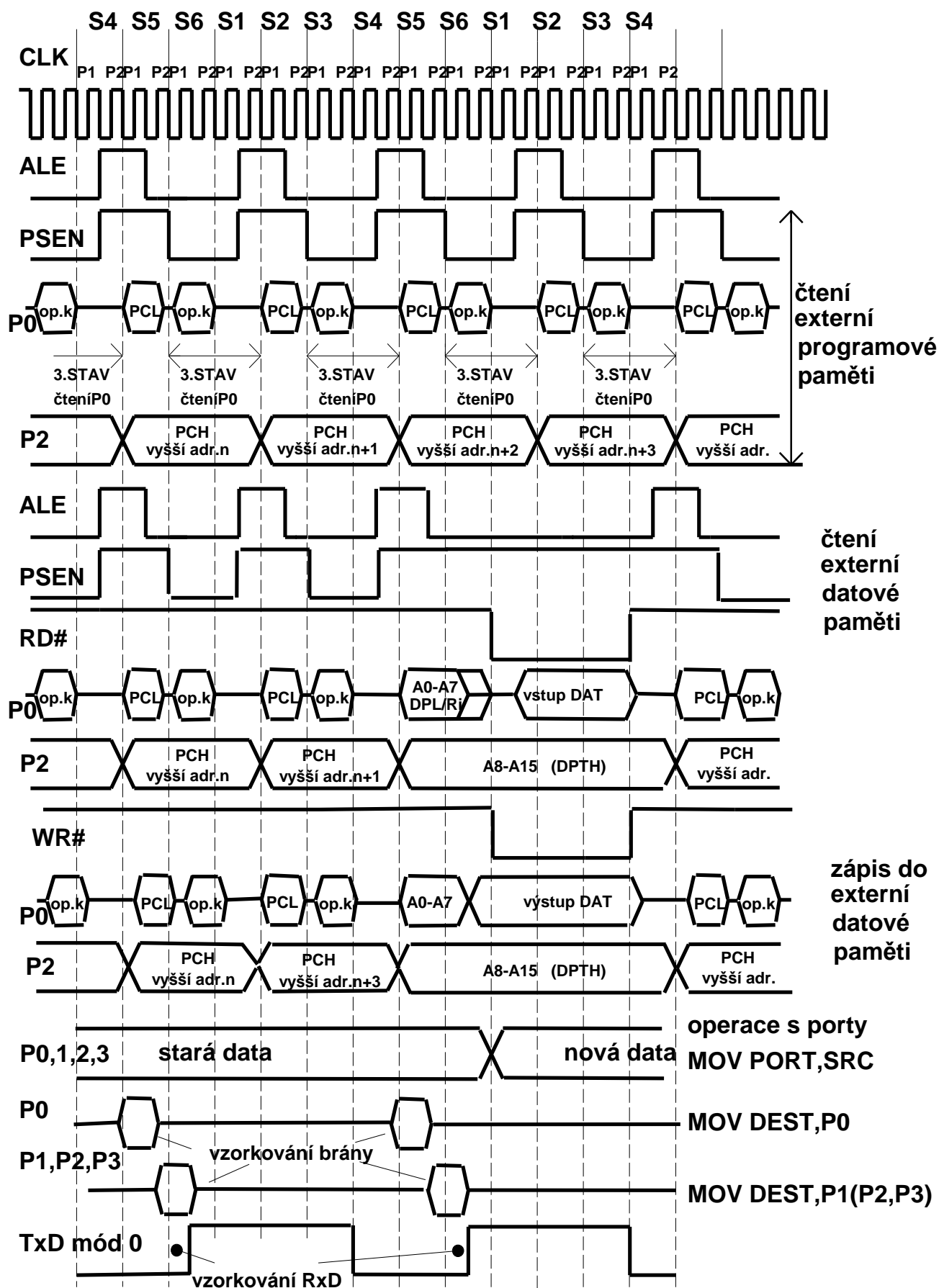


Obr. 23 Časování procesoru 8051

vystupuje bránou P2 v dobách S5P1 až S4P2. Zápis dat do externí datové paměti probíhá obdobně, pouze je ve stejných dobách jako \overline{RD} generován signál \overline{WR} a data vystupují bránou P0 v dobách S6P2 až S4P1. Při instrukci MOV PORT, SRC (SRC je název obsahu zdrojového registru) se mění stav brány na konci instrukčního cyklu ve fázi S1P1, Při čtení brány (nikoliv obsahu registru brány) je nutné nejprve zapsat do registru brány slabiku 0FFh a pak instrukcí MOV DEST,Pk (k=0,1,2,3) vzorkovat signály na vývodech brány Pk ve fázi S5P1(brána P0), nebo S5P2 (brány P1,2,3). Ve stejné fázi je vzorkován signál RxD a v návaznosti na fázi S5P2 tj. od fáze S6P1 do fáze S2P2 včetně je roven jedné hodinový signál seriového kanálu. Signál TxD je mění s náběžnou hranou hodin seriového kanálu. Vstupy externích přerušení INT0# a INT1# jsou vzorkovány v každém strojovém cyklu ve fázi S5P2 a pokud jsou aktivní a potvrzené začne se hardwarové volání obsluhy provádět jako příští instrukce. Volání samo trvá dva cykly.

Bitový (booleovský) procesor

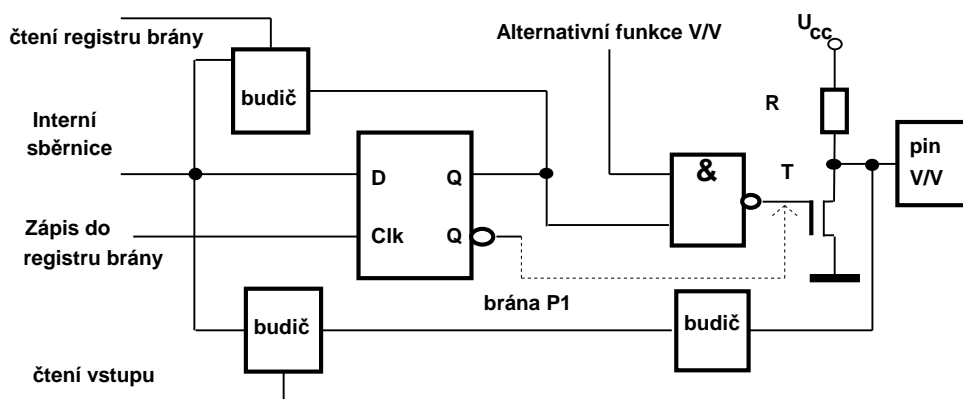
Stařadačem bitového procesoru je bit přenosu C. Pracuje nad bitově adresovatelnou částí RAM a bity bitově adresovatelných registrů speciálních funkcí (všechny registry v interní paměti výhradně přímo adresované, jejichž adresa je dělitelná osmi beze zbytku). Umožňuje nastavení, nulování, negaci a přesun do jiného bitu. Větvit program podle bitu a provádět logický součin a součet dvou bitů.



Obr. 24. Časování externích pamětí, čtení/zápis bran

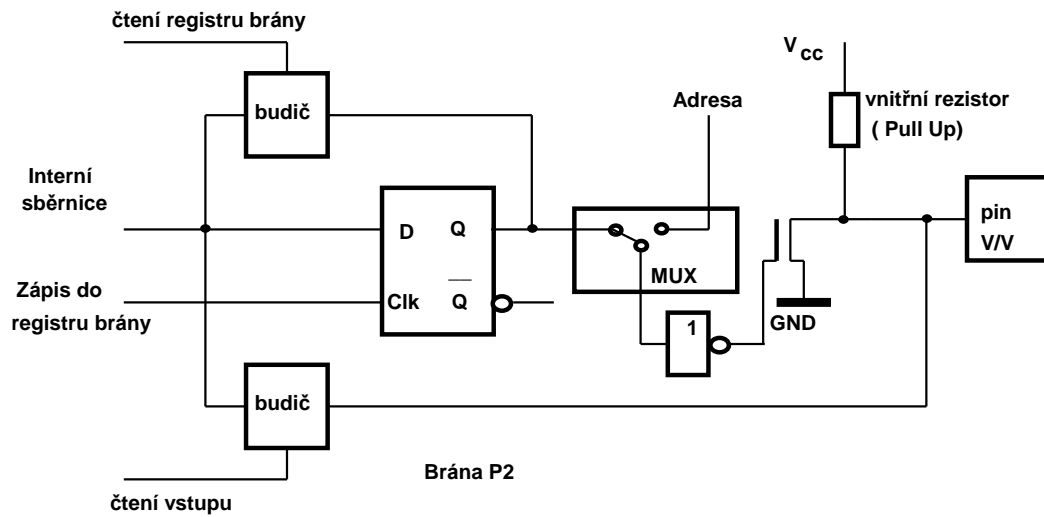
Paralelní vstupy/výstupy

Paralelní číslicové vstupy/výstupy (paralelní číslicové rozhraní procesoru) jsou samozřejmou výbavou každého jednočipového mikropočítače. Jejich obvodové řešení se vyvíjelo od nejjednoduššího uspořádání výstupu s jediným výstupním registrem brány a vlastním výstupem tvořeným spínacím tranzistorem který má v kolektoru upínací rezistor, realizovaný MOS tranzistorem v saturačním režimu, až k čistě obousměrnému provedení brány s třístavovým výstupem, podporovaným třemi registry, **výstupním, vstupním a registrem řízení směru** (npř.u procesoru 80C166). Zapojení výstupu brány P1 (bez alternativní funkce - čárkovaný spoj) a současně bran P3 až P6 u klonu npř. 80552, je na obr. 25. Hodnota odporu rezistoru R je 40 až 60 k Ω , což je hodnota příliš velká pro dostatečně rychlý zápis "log. jedničky" na vývod, brány, proto se krátkodobě, na dvě periody oscilátoru, na výstup připíná další upínací rezistor (MOS tranzistor). Obvodové provedení krátkodobého připínání dalšího upínacího rezistoru pro verzi HMOS je na obr. 28 a pro verzi HCMOS na obr. 29. Při čtení hodnot signálů na vývodech bran P1, P3 a dalších je nutné, aby do registrů bran byly u čtených vývodů zapsány jedničky, jinak je čtena hodnota signálu zapsaného do registru brány. Brána P2 je u mikropočítačů s externí pamětí používána jako adresová sběrnice a proto výstupní obvod brány P2 (obr.26) obsahuje multiplexor. V literatuře se brány a P1, P2, P3 označují jako kvasibidirekcionální, jedná se ovšem o výstupy s otevřeným kolektorem a vnitřním upínacím rezistorem. Brána P0 se v rozšířeném módu mikroprocesoru využívá jako multiplexovaná datová/adresová sběrnice. Na rozdíl od ostatních bran nemá vnitřní upínací rezistor.

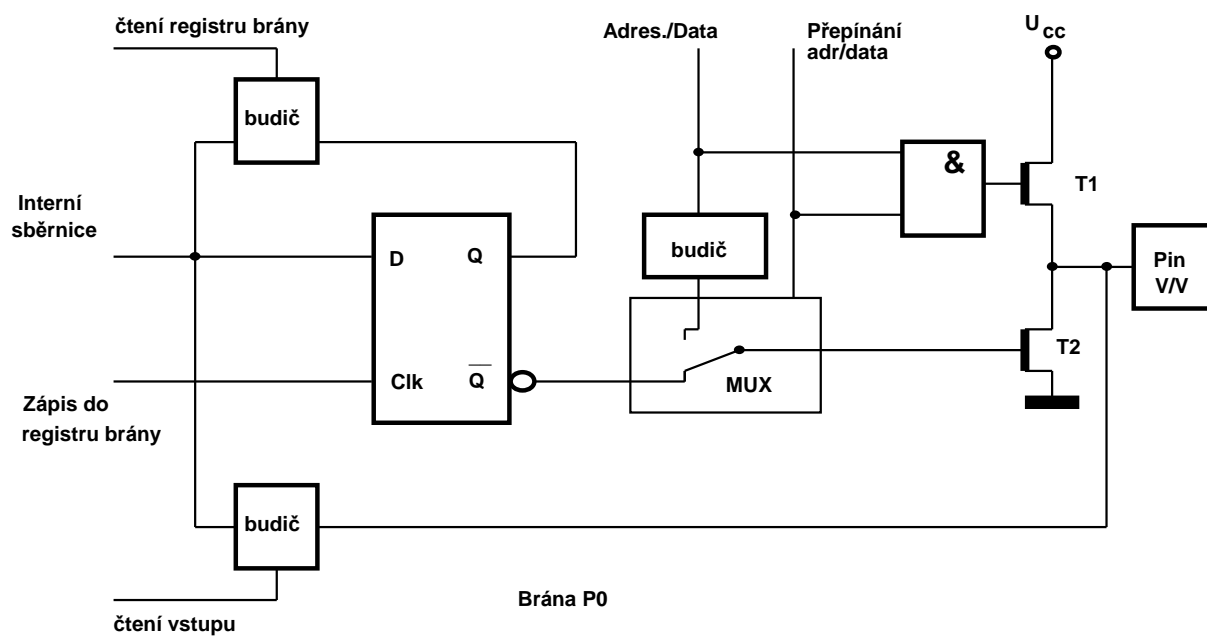


Obr.25 Základní struktura V/V obvodu brány 1, 3, 4, 5, 6.0 až 6.2

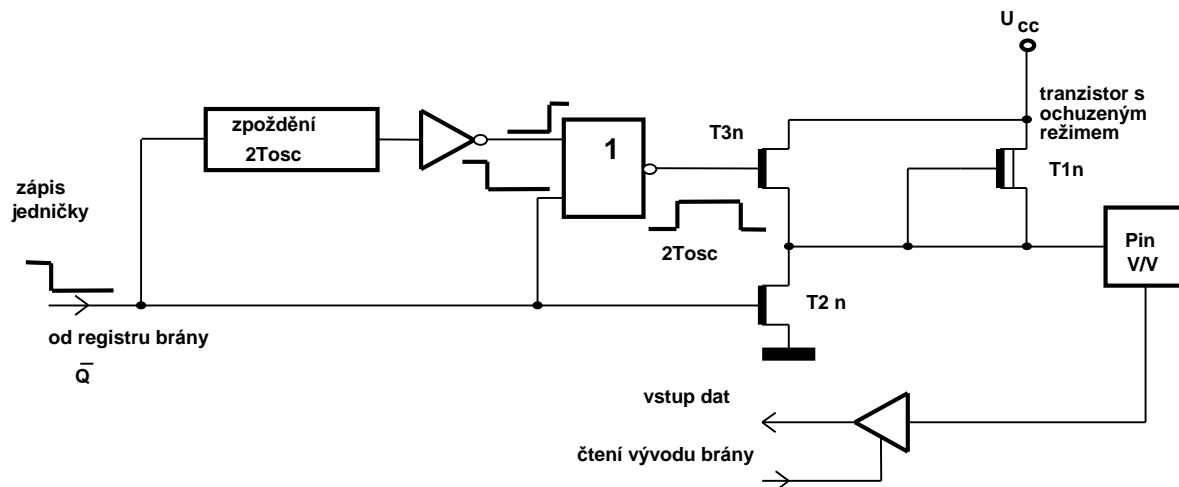
U verze HMOS je horní tranzistor rozepnutý (výstup je stavu vysoké impedance) s výjimkou, kdy je ve funkci multiplexované sběrnice. Zapojení výstupu brány P0 je na obr. 27. Přípustné zatížení bran P1, P2, P3 představují 4 vstupy LSTTL a brány P0 8 vstupů LSTTL. Pokud brána P0 je používána jako normální paralelní číslicová brána a jsou k ní připojeny MOS obvody s předepsanými vyššími úrovněmi "log. 1" je nutné výstupní úroveň brány P0 zvýšit externími upínacími rezistory (5 až 15k Ω), pokud je brána použita jako externí sběrnice není upínání nutné..



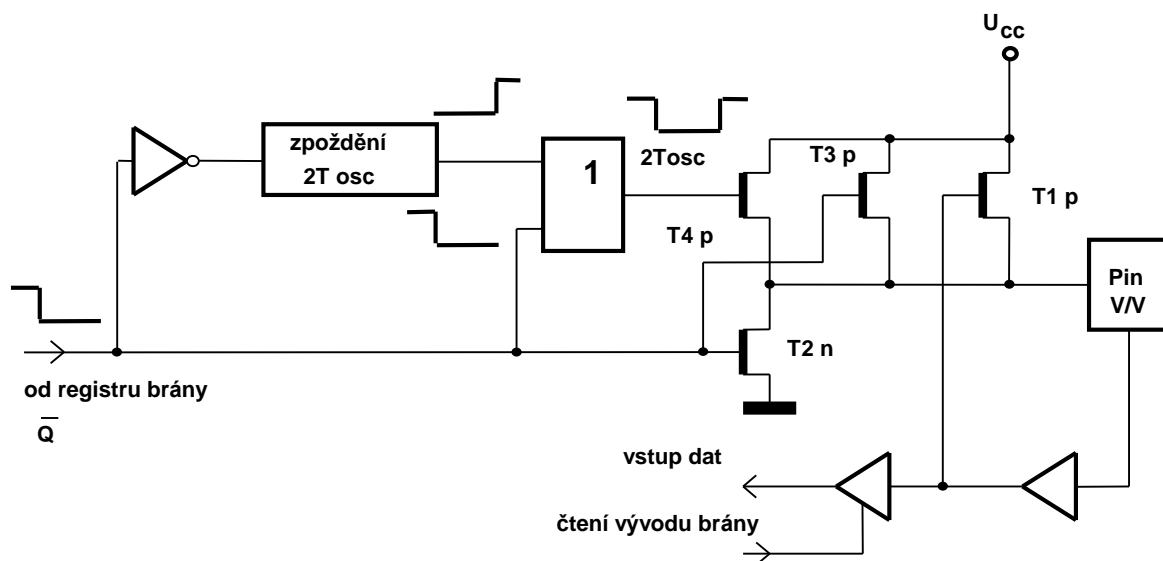
Obr.26 V/V obvod s vnitřním upínacím rezistorem brána P2



Obr.27 Struktura V/V obvodu brána P0



Obr. 28 Obvodové provedení výstupu brány - verze HMOS



Obr. 29 Obvodové řešení brány s otevřeným kolektorem - verze HCMOS

Na obrázcích 28. a 29. je naznačeno obvodové řešení výstupu procesoru, které zabezpečí rychlé nabíjení kapacity vývodu brány při změně výstupního signálu 0[®] 1. Výstup brány je v obou technologiích HMOS i HCMOS doplněn tranzistorem, který při zápisu "jedničky" spíná na vývod procesoru napětí U_{dd} po dobu dvou period kmitočtu oscilátoru f_{osc}. Doplnkový tranzistor (T3n-verze HMOS, T4p-verze HCMOS) má v sepnutém stavu odpor kanálu asi 5k Ω a jeho spínací signál generuje tvarovač impulsů tvořený zpožďovací linkou, invertorem a logickým členem (NOR - HMOS a OR - HCMOS).

Krokování programu 8051

Krokování programu u procesoru 8051 dovoluje struktura a funkce přerušovacího systému uskutečnit programově. Vychází se z toho, že žádost o přerušování nebude akceptována pokud je obsluhován požadavek přerušování se stejnou nebo vyšší prioritou, nebo pokud nebyla provedena alespoň jedna další instrukce po návratové instrukci **RETI**. Naprogramujeme-li jedno z externích přerušování např. INT0# tak, aby bylo aktivováno úrovní "log.0", na vývod ITO připojíme generátor jednoho impulsu, jehož šířka je větší než perioda strojového cyklu a zakončíme obslužný program tohoto požadavku na přerušování sekvencí instrukcí :

```
JNB P3.2,$           ; stop pro INT0=0, podmíněný skok na adresu=(PC)=$
JB  P3.2,$           ; stop pro INT0=1
(JB P3.2,VOLMON) ;pomíněný skok na adresu volání monitoru místo JB P3.2,$
RETI                 ; návrat do krokovaného programu a provedení jedné instrukce
```

V uvedeném těle programu je v závorkách vloženo volání monitoru, který umožní prohlížet a modifikovat obsah vnitřní i vnější paměti procesoru. Signál na vstupu INT0 je vhodné generovat generátorem jednoho pulsu bez zákmitů.

Programování CPU 51

Programy pro CPU 51 je možno vytvářet ve strojovém kódu (prakticky se nepoužívá), v jazyce symbolických instrukcí JSI, který se také někdy označuje jako jazyk symbolických adres - JSA, nebo ve vyšších programovacích jazycích na př. jazyce C. s tím, že zdrojové programy vytvořené v JSI nebo vyšších jazycích jsou přeloženy kompilátory do cílového (strojového) kódu.

V následující kapitole je představen instrukční soubor CPU 51 a v další pak základy syntaxe jedné verze assembleru ASM51.

Program CPU 51/52 pracuje s programovým čítačem PC (nepřímo) a s registrovými bankami, zásobníkovou pamětí (STACK), registry speciálních funkcí SFR (special function register) a bitově adresovatelnou zápisníkovou pamětí. Přístup do vnějších pamětí procesoru zprostředkovávají registry speciálních funkcí (SFR).

Způsoby adresování CPU 51

Architektura CPU51 rozlišuje datovou a programovou paměť a liší se i způsoby adresování obou pamětí.

Adresování registrů

Adresování registrů je inherentní, tzn. že adresa registru vybrané registrové banky je obsažena přímo, nebo kódovaná v operačním kódu (první slabice instrukce). U CPU51 bity PSW RS1, RS0 vybírají registrovou banku a operační kódy instrukcí, které pracují s registry banky nejnižšími třemi bity přímo adresují registr banky. Mnemonika používá označení registru Rn, kde n (0 až 7) je adresa registru.

Přímé adresování

operandy - slabiky (Byty)

Přímé adresování bytů se používá v dolním adresovém prostoru vnitřní paměti (adresy 0-127H) a v adresovém prostoru registru speciálních funkcí SFR. Jejich adresy jsou v následující tabulce. Přímá adresa bytu (8 bitů) tvoří druhou slabiku instrukce. V popisu instrukcí je vyjádřena jako "direct" (npř. MOV Rn,direct) a v JSI za "direct" se dosazuje buď hodnota adresy (npř. 66H), nebo symbolický název přímé adresy definovaný pseudoinstrukcí ASM51

(npř. **data1 equ 66H** ;[pseudoinstrukce]

a **MOV Rn,data1** ; [instrukce]

;instrukce přesouvá do registru Rn obsah paměťového místa s adresou 66H).

operandy - bity

Přímo lze adresovat jednotlivé bity osmi slabik v paměťovém prostoru s adresami 20H - 2FH, celkem 128 bitů (tyto bity je možné uživatelsky využívat jako indikátory -(flags), nebo pro bitové manipulace). Další 128 bitů je přímo adresovatelných v paměťovém prostoru SFR v registrech jejichž adresa je dělitelná 8. Jedná se tudíž o adresy 80H, 88H, 90H, 98H atd. až F0H, F8H. V ASM51 se obecně přímá adresa bitu označuje zkratkou **BIT**, v JSI pak má podobu buď

- a) přímé adresy - 0H až FFH; 0H je adresa LSB(least significant bit) ve slabice interní datové paměti na adrese 20H a FFH je adresa MSB (most significant bit) ve slabice paměti SFR na adrese F8H
- b) přímou adresou bitu v přímo adresované slabice; npř. 33.7 je adresa MSB ve slabice na adrese 21H. 33 je dekadická adresa slabiky a 7 (oddělená \cdot) je adresa bitu
 - c) jako b), přímá adresa slabiky je symbolická ; npř. P0.0 je adresa LSB registru P0, který má adresu 80H tj. 128 dekadicky. (další možné vyjádření této adresy je podle b, 128.0)

Nepřímé adresování

bázové nebo indexované

- při adresování programové paměti jako bázový registr vystupuje programový čítač PC, jako index registr slouží registr DPTR (data pointer registr -16bitů) a registr A (8 bitů) obsahuje relativní posun adresy (obsah registru A se interpretuje jako číslo v komplementárním zápisu se znaménkem, které určuje posun v intervalu -128 až +127). Adresování je vhodné pro práci s programovými tabulkami (LUT - Look Up Table). Jako ilustrující příklad poslouží instrukce

MOVC A,@A+PC - bázové adresování s relativním posunutím,

kde @ je symbol nepřímé adresy **MOVC A,@A+DPTR** - indexované adresování

které přesouvají byte z programové paměti na fyzické adrese vytvořené součtem základu ((PC), nebo (DPTR)) a posunutí (A).

- při adresování datové paměti se používá pouze indexované adresování bez relativního posunu, kdy jako index registr vystupuje registr DPTR, npř. instrukce **MOVX A,@DPTR**

registrové

registrové nepřímé adresování (@Ri) znamená, že fyzickou adresu operandu představuje obsah registru R0, nebo R1 pracovní registrové banky. Registry R0 nebo R1 určuje hodnota LSB operačního kódu. Tímto adresováním je přístup k interní i externí datové paměti. V obou případech lze adresovat pouze 256 slabik. Pro interní datovou paměť v adresovém prostoru 80H až FFH, která není využívána jako zásobník je to jediná možnost přístupu., pro vnější datovou paměť je umístění 256-ti slabičné stránky adresované obsahem registru Ri do paměťového prostoru 64kBytů dáno obsahem záchytného registru brány P2.

Bezprostřední adresování

bezprostřední adresování (Immediate addressing) znamená, že byt/byty (slabiky) v instrukci připojené za slabiku operačního kódu jsou data (konstanty/adresy). V mnemonickém popisu instrukce se označují data v instrukci symbolem #, npř v instrukce, která zapisuje do registru DPTR 16-ti bitová data má podobu **MOV DPTR,#data16**, data16 určují hodnoty druhého a třetího bytů instrukce.

Módy adresování při instrukcích větvení programů

Relativní adresování

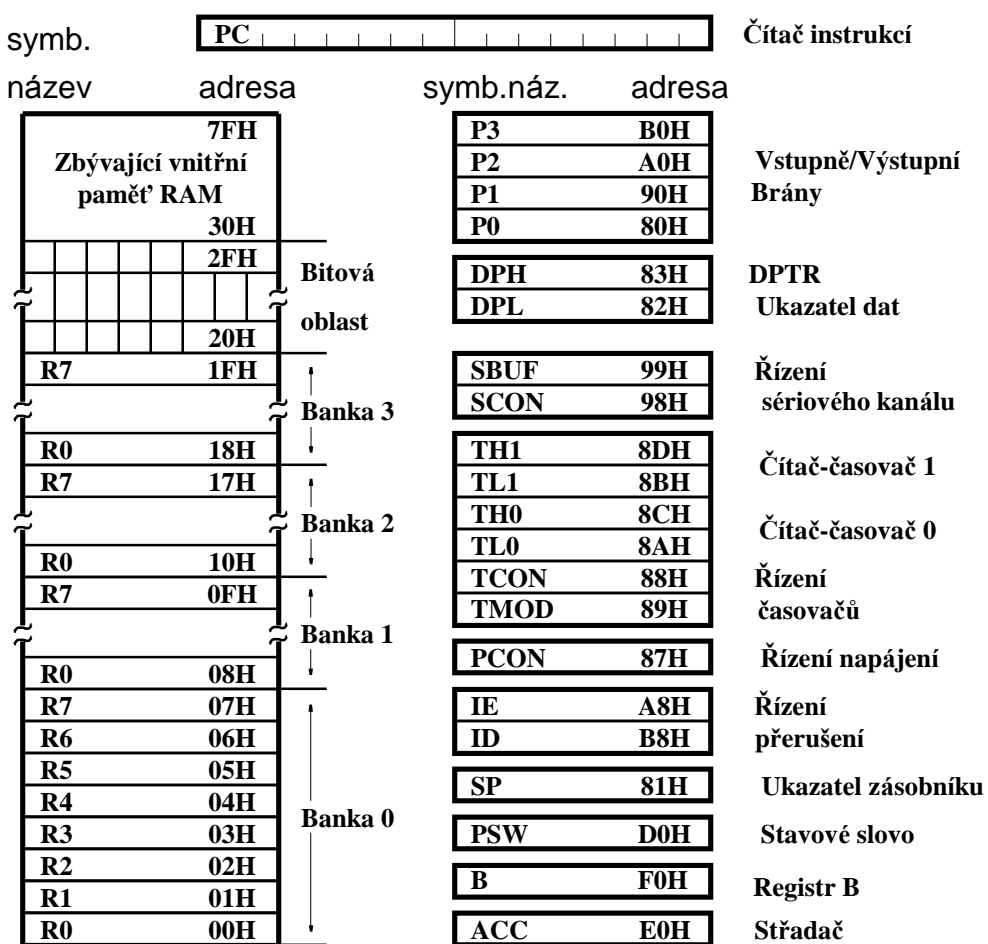
cílovou adresu skoku/volání určuje posunutí, poslední byte instrukce, jehož obsah (sedmibitové číslo se znaménkem) změní aktuální obsah programového čítače PC o -128 až +127. Aktuální obsah PC je stav PC po načtení instrukce větvení.

Absolutní adresování (addr11)

cílová adresa skoku/volání je dána spojením pěti horních bitů PC a jedenácti bitů adresy obsažených v instrukci tak, že tři vyšší bity adresy jsou ve slabice operačního kódu a osm nižších bitů adresy je v druhé slabice instrukce. Adresování je možné pouze vpřed. Prakticky to tedy znamená, že pokud npř. adresa instrukce **AJMP** nebo **ACALL** je 07FEH může program pokračovat, podle adresy v instrukci, na adresách 0800H až 0FFFH.

Dlouhé adresování (addr16)

cílovou adresu skoku/volání určují druhá a třetí slabika instrukce, skok lze provádět do libovolného místa adresového prostoru CPU.



Přehled adres vnitřní paměti procesoru 8051

Současné verze (klony) procesoru 8051 mají sice stejně organizovanou vnitřní datovou paměť a paměť SFR (speciálních funkčních registrů), ale protože jsou vybaveny větším množstvím interních periferních obvodů, které je nutné řídit, je i počet SFR v jejich interní paměti nutně větší.

V tabulce Tab. SFR552 je přehled mnemonického značení , adres a počátečního stavu registrů speciálních funkcí CPU 80C552 fy Philips.

TAB.552

symbol	popis/funkce	přímá ADR	bity adresy, obsahu registru nebo alternativní funkce brány								stav reset
			MSB				LSB				
ACC+	akumulátor	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
ADCH*	a/d přev. high	C6H									xxH
ADCON	a/d přev. řízení*	C5H	adc1	adc0	adex	adci	adcs	aadr2	aadr1	aadr0	x0H
B+	B registr	F0H	F7	F6	F5	F4	F3	F2	f1	F0	00H
CTCON	capture řízení*	EBH	ctn3	ctp3	ctn2	ctp2	ctn1	ctp1	ctn0	ctp0	00H
CTH3*	capture high 3	CFH									xxH
CTH2*	capture high 2	CEH									xxH
CTH1*	capture high 1	CDH									xxH
CTH0*	capture high0	CCH									xxH
CMH2*	compare high 2	CBH									00H
CMH1*	compare high 1	CAH									00H
CMH0*	compare high 0	C9H									00H
CTL3*	capture low 3	AFH									xxH
CTL2*	capture low 2	AEH									xxH
CTL1*	capture low 1	ADH									xxH
CTL0*	capture low 0	ACH									xxH
CML2*	compare low 2	ABH									00H
CML1*	compare low1	AAH									00H
CML0*	compare low0	A9H									00H
DPTR: data pointer (dva byty - adresování externí datové paměti)											
DPH	data point. high	83H									00H
DPL	data point. low	82H									00H
IEN0*+	interupt enbl. 0	A8H	EA	EAD	ES1	ES0	ET1	EX1	ET0	EX0	00H
IEN0*+	adresy bitů reg.		AF	AE	AD	AC	AB	AA	A9	A8	
IEN1*+	interupt enbl. 1	E8H	ET2	ECM2	ECM1	ECM0	ECT3	ECT2	ECT1	ECT0	00H
IEN1*+	adresy bitů reg.		EF	EE	ED	EC	EB	EA	E9	E8	
IP0*+	interupt priorita	B8H	-----	PAD	PS1	PS0	PT1	PX1	PT0	PX0	x00H
IP0*+	adresy bitů reg.		BF	BE	BD	BC	BB	BA	B9	B8	
IP1*+	interupt priorita	F8H	PT2	PCM2	PCM1	PCM0	PCT3	PCT2	PCT1	PCT0	00H
IP1*+	adresy bitů reg.		FF	FE	FD	FC	FB	FA	F9	F8	
P5*	brána 5	C4H	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	xxH
P4*+	brána 4	C0H	CMT1	CMT0	CMSR5	CMSR4	CMSR3	CMSR2	CMSR1	CMSR0	FFH
P4*+	adresy bitů reg.		C7	C6	C5	C4	C3	C2	C1	C0	
P3*+	brána 3	B0H	RDn	WRn	T1	T0	INT1n	INT0n	TxD	RxD	FFH
P3*+	adresy bitů reg.		B7	B6	B5	B4	B3	B2	B1	B0	
P2*+	brána 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8	FFH
P2*+	adresy bitů reg.		A7	A6	A5	A4	A3	A2	A1	A0	
P1*+	brána 1	90H	SDA	SCL	RT2	T2	CT3I	CT2I	CT1I	CT0I	FFH
P1*+	adresy bitů reg.		97	96	95	94	93	92	91	90	
P0*+	brána 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
P0*+	adresy bitů reg.		87	86	85	84	83	82	81	80	
PCON*	řízení příkonu	87H	SMOD	----	----	WLE	GF1	GF0	PD	IDL	00H
PSW+	stavové slovo	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00H
PWMP*	PWM dělič	FEH									00H
PWM1*	PWM registr 1	FDH									00H
PWM0*	PWM registr 0	FCH									00H
RTE*	res/togg. enable	EFH	TP47	TP46	TP45	TP44	TP43	TP42	TP41	TP40	00H

TAB 552. Registry speciálních funkcí 80C552 pokračování

symbol	popis/funkce	dir. ADR	bity adresy, obsahu registru nebo alternativní funkce brány								stav reset
			MSB				LSB				
SP	stack pointer	81H									C7H
S0BUF	ser. port 0 buff.	99H									xxH
S0CON+	ser. port 0 řízení	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
S0CON+	adresa bitů reg.		9F	9E	9D	9C	9B	9A	99	98	
S1ADR*	ser. port1 adresa	DBH	----- SLAVE ADDRESS -----							GC	00H
S1DAT*	ser. port 1 data	DAH									00H
S1STA*	ser. port 1 stav	D9H	SC4	SC3	SC2	SC1	SC0				F8H
S1CON+	ser. port1 řízení	D8H	CR2	ENS1	STA	ST0	SI	AA	CR1	CR0	00H
S1CON	adresa bitů reg.		DF	DE	DD	DC	DB	DA	D9	D8	
STE*	set enable	EEH	TG47	TG46	SP45	SP44	SP43	SP42	SP41	SP40	00H
TH1	časovač1 high	8DH									00H
TH0	časovač0 high	8CH									00H
TL1	časovač1 low	8BH									00H
TL0	časovač0 low	8AH									00H
TMH2	časovač2 high	EDH									00H
TML2	časovač2 low	ECH									00H
TMOD	mód časovačů0,1	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
TCON+	řízení čas. 0,1	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
TCON+	adresy bitů reg.		8F	8E	8D	88C	8B	8A	89	88	
TM2CON	řízení čas.2	EAH	T2IS1	T2IS0	T2ER	T2B0	T2P1	T2P0	T2MS1	T2MS0	00H
TM2IR+	čas.2 ukaz. int.	C8H	T20V	CMI2	CMI1	CMI0	CTI3	CTI2	CTI1	CTI0	00H
TM2IR+	adresa bitů reg.		CF	CE	CD	CC	CB	CA	C9	C8	
T3*	časovač 3	FFH									00H

*- registr modifikován, nebo přidán k registrům procesoru 8051

V tabulce TAB 552 je uvedeno 56 speciálních funkčních registrů, které jsou použity v procesoru firmy Philips 80C552.

Rozdíly mezi procesory 8051 a 80C552 z hlediska programování

interní programová paměť - procesor může obsahovat vnitřní programovou paměť 8kB, která je programovaná buď maskou při výrobě (typ 83C552), nebo elektricky u uživatele (typ 87C552). Typ 80C552 nemá vnitřní programovou paměť. Přepínání vnitřní a vnější paměti řídí signál \overline{AE} stejně jako u 8051. Při $\overline{AE}=0$ je program čten z vnější programové paměti, při $\overline{AE}=1$ jsou instrukce čteny z interní programové paměti jejíž adresy jsou 0000H až 1FFFH, od adresy 2000H čte procesor instrukce z paměti vnější.

datová paměť - datová paměť je stejná jako u procesoru 8052, to znamená, že paměťový prostor vnitřní datové paměti je tvořen dvěma oblastmi 128 Bytů. První z nich je adresovatelná přímo i neřímo (adresy 00h - 7FH) a druhá pouze nepřímou (adresy 80H - FFH). Zásobníkovou paměť lze umístit do celého prostoru vnitřní datové paměti.

interní periferie - procesor je vybaven osmi paralelními branami z nichž jedna je výhradně vstupní a lze jí alternativně používat pro vstup až osmi analogových signálů do A/D převodníku, dále navíc proti procesoru 8051 obsahuje časovač T2, který je spojen se záchytným a komparačním systémem, hlídací časovač T3, přístrojový seriový kanál IIC (kanál Inter Integrated Circuits) a

dva kanály PWM (mohou být použity jako převodníky D/A) Vše je řízeno 56-ti registry SFR (viz Tab 552)

Instrukční soubor CPU 51

Instrukční soubor procesoru 8051 lze rozdělit do těchto sedmi skupin:

- **Přesunové instrukce** umožňují přesun 8 bitové hodnoty z registru Rr a přímo nebo nepřímou adresovanou vnitřní datové paměti do střadače a obráceně **MOV**, přesun hodnoty z nebo do střadače z nepřímou adresované vnější datové paměti **MOVX**, přesun hodnoty do střadače z nepřímou adresované programové paměti **MOVC**, záměnu obsahu střadače s obsahem registru Rr a přímo nebo nepřímou adresované vnitřní datové paměti **XCH** a **XCHD**. Posledními instrukcemi, které lze zařadit do této skupiny jsou instrukce pro práci se zásobníkem **PUSH** pro uložení a **POP** pro vyjmutí hodnoty ze zásobníku z přímo adresovaného místa vnitřní datové paměti.
- **Aritmetické instrukce** umožňují zvětšení (inkrementace) **INC** a zmenšení (dekrementace) **DEC** střadače, registru a přímo nebo nepřímou adresovaného místa ve vnitřní datové paměti, 8-bitový aritmetický součet střadače s registrem, přímo nebo nepřímou adresovaným místem ve vnitřní datové paměti nebo přímo uvedenou hodnotou bez příznaku přenosu **ADD** nebo s příznakem přenosu **ADDC**, 8-bitový rozdíl střadače s příznakem přenosu s registrem, přímo nebo nepřímou adresovaným místem ve vnitřní datové paměti nebo přímo uvedenou hodnotou a dekadické korekce po binárním součtu dvou čísel v BCD kódu **DA**. Aritmetické instrukce obsahují i instrukce pro 8-bitové násobení **MUL** a dělení **DIV** dvou hodnot bez znaménka uložených ve střadači a registru B a instrukci pro realizaci jednotkového doplňku střadače **CPL**.
- **Logické operace** umožňují logický součin **ANL**, součet **ORL** a operaci výhradně-nebo (EX-OR = neekvivalence) **XRL** střadače s registrem, přímo nebo nepřímou adresovaným místem ve vnitřní datové paměti nebo přímo uvedenou hodnotou nebo přímo adresovaným paměťovým místem a střadačem nebo přímo uvedenou hodnotou. Protože vstupně/výstupní brány jsou paměťově adresované, jsou v logických operacích obsaženy i operace s bránami.
- **Posunové instrukce** umožňují 8 a 9-bitové logické rotace střadače doleva **RL** a **RLC** nebo doprava **RR** a **RRC** případně s příznakem přenosu (9-bitů). Mezi posunové instrukce můžeme zařadit i instrukci výměny horního a dolního půlbytu střadače **SWAP**.
- **Bitové operace** umožňují nastavit **SETB** nebo nulovat **CLR** kterýkoliv z 256-ti přímo adresovatelných bitů procesoru 8051. Bitové instrukce umožňují přesun **MOV**, logický součin **ANL** a logický součet **ORL** mezi přímo adresovaným bitem a příznakem přenosu C, který přejímá pro bitové operace funkci střadače.

- **Skokové instrukce** umožňují nepodmíněný dlouhý **LJMP** (16-bitový) a krátký **AJMP** (11-bitový) skok, podmíněné relativní skoky (7-bitové) závislé na nulovosti a nenulovosti střadače, příznaku přenosu C nebo přímo adresovaném bitu ve vnitřní datové paměti. Umožňují pouze nepodmíněná dlouhá **LCALL** (16-bitové) a krátké **ACALL** (11-bitové) volání podprogramů a nepodmíněné návraty z podprogramů **RET** a obslužných podprogramů přerušení **RETI**.
- **Sdružené instrukce**, které v sobě obsahují dvě samostatné operace a urychlují tak realizaci programových smyček, umožňují dekrementovat registr nebo přímo adresované paměťové místo a realizovat relativní skok (7-bitů) při jeho nenulovosti **DJNZ** nebo porovnat střadač s přímo adresovaným paměťovým místem, střadač, registr nebo nepřímo adresované paměťové místo s přímo uvedenými daty a realizovat relativní skok (7-bitů) při jejich neshodě **CJNE**.

SYNTAXE INSTRUKCE	POPIS INSTRUKCE		
ACALL adr11	$PC(10 \div 0) \leftarrow \text{adr } 11$		
	Ovlivňuje: ---	Slov: 2	Cyklů: 2
Volání podprogramu uvnitř 2kB adresového prostoru (adresa je 11-bitová). Instrukce uloží návratovou adresu do zásobníku s tím, že nejprve uloží nižší a potom vyšší slabiku. Volaný podprogram musí ležet uvnitř 2kB stránky, v které leží instrukce následující po instrukci ACALL.			
ADD A,<zdrojová slabika> Sčítání střadače se zdrojovou slabikou			
ADD A, Rr	$(A) \leftarrow (A) + (Rr)$		
ADD A, @Rr	$(A) \leftarrow (A) + ((Rr))$, kde $r = 0,1$		
	Ovlivňuje: C,AC,OV,P	Slov: 1	Cyklů: 1
ADD A, adresa	$(A) \leftarrow (A) + (\text{adresa})$		
ADD A, #data	$(A) \leftarrow (A) + \text{data}$		
	Ovlivňuje: C,AC,OV,P	Slov: 2	Cyklů: 1
Instrukce přičte obsah adresované slabiky ke střadači a výsledek v něm ponechá. Adresovanou slabikou může být registr aktivní banky Rr , přímo uvedená adresa (8 bitů), nepřímo adresované paměťové místo registrem R0 nebo R1 (obsah registru určuje adresu místa jehož obsah se bude přičítat) nebo přímo uvedená hodnota data .			
ADDC A,<zdrojová slabika> Sčítání střadače se slabikou a příznakem přenosu			
ADDC A, Rr	$(A) \leftarrow (A) + (Rr) + (C)$		
ADDC A, @Rr	$(A) \leftarrow (A) + ((Rr)) + (C)$, kde $r = 0,1$		
	Ovlivňuje: C,AC,OV,P	Slov: 1	Cyklů: 1
ADDC A, adresa	$(A) \leftarrow (A) + (\text{adresa}) + (C)$		

ADDC A, #data	$(A) \leftarrow (A) + \text{data} + (C)$		
	Ovlivňuje: C,AC,OV,P	Slov: 2	Cyklů: 1
Instrukce přičte obsah adresované slabiky a příznakový bit přenosu C ke střadači a výsledek v něm ponechá.			
AJMP adr11	$PC(10 \div 0) \leftarrow \text{adr } 11$		
	Ovlivňuje: ---	Slov: 2	Cyklů: 2
Krátký nepodmíněný skok na adresu uvnitř 2kB stránky.			
ANL <cílová slabika>, <zdrojová slabika>			
Operace logického součtu mezi cílovou a zdrojovou slabikou.			
ANL A, Rr	$(A) \leftarrow (A) \text{ AND } (Rr)$, kde $r = 0,1,K,7$		
ANL A, @Rr	$(A) \leftarrow (A) \text{ AND } ((Rr))$, kde $r = 0,1$		
	Ovlivňuje: P	Slov: 1	Cyklů: 1
ANL A, adresa	$(A) \leftarrow (A) \text{ AND } (\text{adresa})$		
ANL A, #data	$(A) \leftarrow (A) \text{ AND } \text{data}$		
ANL adresa, A	$(\text{adresa}) \leftarrow (\text{adresa}) \text{ AND } (A)$		
	Ovlivňuje: P - pro zápis do střadače	Slov: 2	Cyklů: 1
ANL adresa, #data	$(\text{adresa}) \leftarrow (\text{adresa}) \text{ AND } \text{data}$		
	Ovlivňuje: ---	Slov: 3	Cyklů: 2
Instrukce ANL provede logický součin mezi odpovídajícími bity cílového a zdrojového bytu a výsledek uloží do cílové slabiky. Pro operaci s výstupní bránou (přímá adresa) se operace provede mezi výstupním registrem a zdrojovým bytem a nikoliv vstupními signály.			
ANL C, <zdrojový bit>	Logický součin bitů		
ANL C, bit	$(C) \leftarrow (C) \text{ AND } (\text{bit})$		
ANL C, /bit	$(C) \leftarrow (C) \text{ AND } (\overline{\text{bit}})$		
	Ovlivňuje: C	Slov: 2	Cyklů: 2
Logický součin příznaku přenosu C s přímo adresovaným bitem. Výsledek operace se uloží do příznaku C. Je-li před adresou bitu lomítko, potom hodnota bitu bude před operací negována. Adresovaný bit lze adresovat jenom přímou adresou .			
CJNE <cílová slabika>, <zdrojová slabika>,relativní adresa			
Porovnej a skoč, když se operandy nerovnají.			
CJNE A, adresa, relativní adresa	$(PC) \leftarrow (PC) + 3$ Je-li $(A) \neq (\text{adresa})$ pak $(PC) \leftarrow (PC) + \text{relativní adresa}$ jinak $(PC) \leftarrow (PC) + 2$		

CJNE A, #data, relativní adresa	$(PC) \leftarrow (PC) + 3$ Je-li $(A) \neq \text{data}$ pak $(PC) \leftarrow (PC) + \text{relativní adresa}$ jinak $(PC) \leftarrow (PC) + 2$		
CJNE Rr, #data, relativní adresa	$(PC) \leftarrow (PC) + 3$ Je-li $(Rr) \neq \text{data}$, kde $r = 0,1,K,7$ pak $(PC) \leftarrow (PC) + \text{relativní adresa}$ jinak $(PC) \leftarrow (PC) + 2$		
CJNE @Rr, #data, relativní adresa	$(PC) \leftarrow (PC) + 3$ Je-li $((Rr)) \neq \text{data}$, kde $r = 0,1$ pak $(PC) \leftarrow (PC) + \text{relativní adresa}$ jinak $(PC) \leftarrow (PC) + 2$		
	Ovlivňuje: C	Slov: 3	Cyklů: 2
Instrukce porovná střadač nebo registr Rr nebo slabiku adresovanou registry R0,R1 se zdrojovou slabikou (adresa,data). V případě jejich nerovnosti provede relativní skok (maximálně o +127 nebo -128 bytů) na požadovanou adresu. Adresa skoku se vypočte přičtením relativního posunu k čítači instrukcí, který byl nejprve třikrát inkrementován v důsledku čtení prováděné instrukce. Je-li cílová slabika menší než zdrojová je zároveň nastaven příznak přenosu C=1, jinak ho vynuluje C=0.			
CLR A	$(A) \leftarrow 0$		
	Ovlivňuje: P	Slov: 1	Cyklů: 1
Instrukce vynuluje obsah střadače.			
CLR C	$(C) \leftarrow 0$		
	Ovlivňuje: C	Slov: 1	Cyklů: 1
CLR bit	$(\text{bit}) \leftarrow 0$		
	Ovlivňuje: AC, F0, RS1, RS0, OV, P jen je-li adresován	Slov: 2	Cyklů: 1
Instrukce vynuluje adresovaný bit.			
CPL A	$(A) \leftarrow \overline{(A)} = 1(A)$		
	Ovlivňuje: ---	Slov: 1	Cyklů: 1
Instrukce neguje každý bit střadače a vytváří tak jeho jednotkový doplněk.			
CPL C	$(C) \leftarrow \text{NOT}(C) = \overline{(C)}$		
	Ovlivňuje: C	Slov: 1	Cyklů: 1
CPL bit	$(\text{bit}) \leftarrow \text{NOT}(\text{bit}) = \overline{(\text{bit})}$		
	Ovlivňuje: C, AC, F0, RS1, RS0, OV, P jen je-li adresován	Slov: 2	Cyklů: 1
Instrukce neguje (invertuje) adresovaný bit.			

DA A	Dekadická korekce střadače		
	Ovlivňuje: C, AC, P	Slov: 1	Cyklů: 1
Instrukce koriguje obsah střadače po binárním sčítání dvou dekadických čísel vyjádřených v BCD kódu tak, aby výsledek opět tvořil dvě čtyřbitová BDC čísla. Je-li hodnota na nižších čtyřech bitech >9 nebo AC=1, potom se ke střadači přičte hodnota 6. Je-li hodnota na vyšších čtyřech bitech >9 nebo C=1, potom se ke střadači přičte hodnota 60H.			
DEC slabika	Dekrementace registru nebo paměťového místa		
		Slov: 1	Cyklů: 1
DEC A	$(A) \leftarrow (A) - 1$ ovlivňuje P		
DEC Rr	$(Rr) \leftarrow (Rr) - 1$ kde $Rr = 0,1,K,7$		
DEC adresa	$(adresa) \leftarrow (adresa) - 1$		Slov: 2
DEC @Rr	$(Rr) \leftarrow (Rr - 1)$ kde $Rr = 0,1$		
Instrukce odečte od obsahu adresovaného paměťového místa hodnotu jedna. Po zmenšení hodnoty 00H dojde k podtečení na hodnotu FFH. Dekrementace obsahu výstupní brány zmenšuje obsah přečtený z registru brány a nikoliv ze vstupně/výstupních vodičů.			
DIV AB	A ← quocient A/B B ← zbytek A/B		
	Ovlivňuje: OV,P a C=0	Slov: 1	Cyklů: 4
Instrukce provádí celočíselné dělení obsahu střadače s obsahem registru B. Celá část podílu zůstává ve střadači, zbytek (nikoliv desetinná část) zůstává v registru B. Při dělení nulou se nastaví příznak přetečení OV=1.			
DJNZ Rr, relativní adresa	$(Rr) \leftarrow (Rr) - 1$, kde $r = 0,1,K,7$ Je-li $(Rr) \neq 0$ pak $(PC) \leftarrow (PC) + \text{relativní adresa}$ jinak $(PC) \leftarrow (PC) + 2$		
	Ovlivňuje: OV,P a C=0	Slov: 1	Cyklů: 4
Instrukce odečte od adresového registru jedničku a zjistí zda výsledek je nulový. Je-li výsledek nenulový provede skok na definovanou adresu.			
DJNZ adresa, relativní adresa			
$(adresa) \leftarrow (adresa) - 1$, Je-li $(adresa) \neq 0$ pak $(PC) \leftarrow (PC) + \text{relativní adresa}$ jinak $(PC) \leftarrow (PC) + 2$			
	Ovlivňuje: OV,P a C=0	Slov: 1	Cyklů: 4
Instrukce odečte od adresového paměťového místa jedničku a zjistí zda výsledek je nulový. Je-li výsledek nenulový provede skok na definovanou adresu.			
INC slabika	Inkrementace registru nebo paměťového místa		
	Ovlivňuje: ---	Slov: 1	Cyklů: 1
INC A	$(A) \leftarrow (A) + 1$		
INC Rr	$(Rr) \leftarrow (Rr) + 1$ kde $r = 0,1,K,7$		
INC adresa	$(adresa) \leftarrow (adresa) + 1$		Slov: 2

INC @Rr	$((Rr)) \leftarrow ((Rr)) + 1$ kde $r = 0,1$		
Instrukce přičte od obsahu adresovaného paměťového místa hodnotu jedna. Po zvětšení hodnoty FFH dojde k přetečení na hodnotu 00H. Inkrementace obsahu výstupní brány zvětšuje obsah přečtený z registru brány a nikoliv ze vstupně/výstupních vodičů.			
INC DPTR	$(DPTR) \leftarrow (DPTR) + 1$		
	Ovlivňuje: ---	Slov: 1	Cyklů: 2
Instrukce přičte jedničku k registrovému páru DPH a DPL, které vytváří 16-bitový ukazatel datové paměti DPTR. Dojde-li při přičítání k přetečení u registru DPL (FFH→00H) potom je přičtena jednička k registru DPH. Přičítání jedničku k DPTR probíhá modulo 2^{16} (FFFFH +1 →0000H). Instrukce INC DPTR je jedinou 16-bitovou instrukcí v instrukčním souboru procesoru 8051.			
JB bit, relativní adresa	Je-li bit=1 potom skoč		
	Ovlivňuje: ---	Slov: 3	Cyklů: 2
Instrukce testuje adresovaný bit a v případě jeho nastavení (log.1) provede skok na adresu kterou vypočte jako součet čítače instrukcí a relativní adresy.			
JBC bit, relativní adresa	Je-li bit=1 potom skoč a nuluj bit (bit←0)		
	Ovlivňuje: ---	Slov: 3	Cyklů: 2
Instrukce testuje adresovaný bit a v případě jeho nastavení (log.1) provede skok na adresu kterou vypočte jako součet čítače instrukcí a relativní adresy a vynuluje testovaný bit.			
JC relativní adresa	Je-li C=1 potom skoč		
	Ovlivňuje: ---	Slov: 2	Cyklů: 2
Instrukce testuje příznak přenosu a v případě jeho nastavení (log.1) provede skok na adresu kterou vypočte jako součet čítače instrukcí a relativní adresy.			
JMP @A+DPTR	$(PC) \leftarrow (A) + (DPTR)$		
	Ovlivňuje: ---	Slov: 1	Cyklů: 2
Instrukce nepřímého nepodmíněného skoku na adresu určenou 16-bitovým součtem obsahu střadače (8 bitů bez znaménka) s obsahem ukazatele datové paměti DPTR.			
JNB bit, relativní adresa	Je-li bit=0 potom $(PC) \leftarrow (PC) +$ relativní adresa		
	Ovlivňuje: ---	Slov: 3	Cyklů: 2
Instrukce testuje adresovaný bit a v případě jeho nulovosti (log.0) provede skok na adresu, kterou vypočte jako součet čítače instrukcí a relativní adresy.			
JNC relativní adresa	Je-li C=0 potom $(PC) \leftarrow (PC) +$ relativní adresa		
	Ovlivňuje: ---	Slov: 2	Cyklů: 2
Instrukce testuje příznak přenosu a v případě jeho nulovosti (log.0) provede skok na adresu, kterou vypočte jako součet čítače instrukcí a relativní adresy.			
JNZ relativní adresa	Je-li $(A) \neq 0$ potom $(PC) \leftarrow (PC) +$ relativní adresa		
	Ovlivňuje: ---	Slov: 2	Cyklů: 2

Instrukce testuje obsah střadače a v případě jeho nenulovosti provede skok na adresu, kterou vypočte jako součet čítače instrukcí a relativní adresy.			
JZ relativní adresa	Je-li $(A) = 0$ potom $(PC) \leftarrow (PC) + \text{relativní adresa}$		
	Ovlivňuje: ---	Slov: 2	Cyklů: 2
Instrukce testuje obsah střadače a v případě jeho nulovosti provede skok na adresu, kterou vypočte jako součet čítače instrukcí a relativní adresy.			
LCALL adr16	$PC(15 \div 0) \leftarrow \text{adr } 16$		
	Ovlivňuje: ---	Slov: 3	Cyklů: 2
Instrukce vykoná nepodmíněné volání podprogramu z přímo uvedené adresy ve svém druhém (vyšší byte) a třetím (nižší byte) bytu. Před uložením přečtené adresy do čítače instrukcí (PC), uloží návratovou adresu (současný stav (PC) = adresa následující instrukce) do zásobníku.			
LJMP adr16	$PC(15 \div 0) \leftarrow \text{adr } 16$		
	Ovlivňuje: ---	Slov: 3	Cyklů: 2
Instrukce vykoná nepodmíněný skok na adresu přímo uvedenou ve svém druhém (vyšší byte) a třetím (nižší byte) bytu. Adresa může ležet kdekoli v 64kB adresovém prostoru.			
MOV <cílová slabika>, <zdrojová slabika>			
Přesun bytu z paměťového místa na jiné paměťové místo ve vnitřní datové paměti.			
MOV A, Rr	$(A) \leftarrow (Rr)$, kde $r = 0, 1, K, 7$		
MOV A, @Rr	$(A) \leftarrow ((Rr))$, kde $r = 0, 1$		
MOV Rr, A	$(Rr) \leftarrow (A)$, kde $r = 0, 1, K, 7$		
MOV @Rr, A	$((Rr)) \leftarrow (A)$, kde $r = 0, 1$		
	Ovlivňuje: ---	Slov: 1	Cyklů: 1
MOV A, adresa	$(A) \leftarrow (\text{adresa})$		
MOV A, #data	$(A) \leftarrow \text{data}$		
MOV Rr, #data	$(Rr) \leftarrow (A)$, kde $r = 0, 1, K, 7$		
MOV @Rr, #data	$((Rr)) \leftarrow \text{data}$, kde $r = 0, 1$		
MOV adresa, A	$(\text{adresa}) \leftarrow (A)$		
	Ovlivňuje: ---	Slov: 2	Cyklů: 1
MOV Rr, adresa	$(Rr) \leftarrow (\text{adresa})$, kde $r = 0, 1, K, 7$		
MOV adresa, Rr	$(\text{adresa}) \leftarrow (Rr)$, kde $r = 0, 1, K, 7$		
MOV @Rr, adresa	$((Rr)) \leftarrow (\text{adresa})$, kde $r = 0, 1$		
MOV adresa, @Rr	$(\text{adresa}) \leftarrow ((Rr))$, kde $r = 0, 1$		
	Ovlivňuje: ---	Slov: 2	Cyklů: 2
MOV adresa1, adresa2	$(\text{adresa1}) \leftarrow (\text{adresa2})$		
MOV adresa, #data	$(\text{adresa}) \leftarrow \text{data}$		

	Ovlivňuje: ---	Slov: 3	Cyklů: 2
Instrukce MOV přesune obsah zdrojové slabiky do cílové slabiky bez ovlivnění jakýchkoliv příznaků.			
MOV <cílový bit>, <zdrojový bit>			
Přesun hodnoty mezi přenosem a daným bitem.		Ovlivňuje: ---	
MOV C, bit	$(C) \leftarrow (\text{bit})$	Slov: 2	Cyklů: 1
MOV bit, C	$(\text{bit}) \leftarrow (C)$	Slov: 2	Cyklů: 2
Instrukce přesune obsah druhého a třetího bytu instrukce do ukazatele dat (DPTR). Druhý byte do DPH a třetí byte do DPL.			
MOV DPTR, #data16	$(DPTR) \leftarrow \text{data16}$ tj. $(DPH) = \text{data}(15 \div 8)$ $(DPL) = \text{data}(7 \div 0)$		
	Ovlivňuje: ---	Slov: 3	Cyklů: 2
Instrukce přesune obsah druhého a třetího bytu instrukce do ukazatele dat (DPTR). Druhý byte do DPH a třetí byte do DPL.			
MOVC A, @A+<bázový registr>		Přesuň byte z paměti programu	
MOVC A, @A+DPTR	$(A) \leftarrow ((A) + (DPTR))$		
MOVC A, @A+PC	$(A) \leftarrow ((A) + (PC))$		
	Ovlivňuje: ---	Slov: 1	Cyklů: 2
Instrukce přesune byte z programové paměti (operační kód nebo konstantu) do střadače. Adresa místa, z kterého jehož obsah se přesouvá, získáme jako 16-bitový součet obsahu stradače (8 bitů) a ukazatelem dat DPTR nebo čítačem instrukcí PC. Pro případ čítače instrukcí je jeho obsah před provedením instrukce inkrementován (ukazuje na následující instrukci).			
MOVX <cílová slabika>, <zdrojová slabika>			
Přesun byte z/do vnější datové paměti			
MOVX A, @DPTR	$(A) \leftarrow ((DPTR))$		
MOVX @DPTR, A	$((DPTR)) \leftarrow (A)$		
MOVX A, @Rr	$(A) \leftarrow ((Rr))$, kde $r = 0,1$		
MOVX @Rr, A	$((Rr)) \leftarrow (A)$, kde $r = 0,1$		
	Ovlivňuje: P - pro zápis do střadače	Slov: 1	Cyklů: 2
Instrukce přesune byte ze/do střadače do/z vnější paměti dat. Instrukce mohou využívat 16-bitovou nebo 8-bitovou nepřímou adresu. V prvním případě se vysílá adresa uložená v DPTR na bránu P2 (DPH) a bránu P0 (DPL). V případě druhém se vysílá na bránu P0 adresa uložená v registru R0 nebo R1 a na bráně P2 zůstává hodnota, která do ni byla naposledy zapsána.			
MUL AB	$(A) \leftarrow \text{nižší byte } (A) * (B)$, $(B) \leftarrow \text{vyšší byte } (A) * (B)$		
	Ovlivňuje: OV,P a C=0	Slov: 1	Cyklů: 4

Instrukce vynásobí dvě osmibitová čísla bez znaménka uložená ve střadači a registru B. Je-li součin větší než hodnota 255 (FFH), nastaví se příznakový bit přetečení OV=1. V opačném případě jej vynuluje.			
NOP	Prázdná operace		
	Ovlivňuje: ---	Slov: 1	Cyklů: 1
Kromě čítače instrukcí neovlivňuje instrukce žádné registry a příznaky.			
ORL <cílová slabika>, <zdrojová slabika>			
Operace logického součtu mezi cílovou a zdrojovou slabikou.			
ORL A, Rr	$(A) \leftarrow (A) \text{ OR } (Rr)$, kde $r = 0,1,K,7$		
ORL A, @Rr	$(A) \leftarrow (A) \text{ OR } ((Rr))$, kde $r = 0,1$		
	Ovlivňuje: P	Slov: 1	Cyklů: 1
ORL A, adresa	$(A) \leftarrow (A) \text{ OR } (\text{adresa})$		
ORL A, #data	$(A) \leftarrow (A) \text{ OR } \text{data}$		
ORL adresa, A	$(\text{adresa}) \leftarrow (\text{adresa}) \text{ OR } (A)$		
	Ovlivňuje: P - pro zápis do střadače	Slov: 2	Cyklů: 1
ORL adresa, #data	$(\text{adresa}) \leftarrow (\text{adresa}) \text{ OR } \text{data}$		
	Ovlivňuje: ---	Slov: 3	Cyklů: 2
Instrukce ORL provede logický součin mezi odpovídajícími bity cílového a zdrojového bytu a výsledek uloží do cílové slabiky. Pro operaci s výstupní bránou (přímá adresa) se operace provede mezi výstupním registrem a zdrojovým bytem a nikoliv vstupními signály.			
ORL C, <zdrojový bit>	Logický součet bitů		
ORL C, bit	$(C) \leftarrow (C) \text{ OR } (\text{bit})$		
ORL C, /bit	$(C) \leftarrow (C) \text{ OR } \overline{(\text{bit})}$		
	Ovlivňuje: C	Slov: 2	Cyklů: 2
Logický součet příznaku přenosu C s přímo adresovaným bitem. Výsledek operace se uloží do příznaku C. Je-li před adresou bitu lomítko, potom hodnota bitu bude před operací negována. Adresovaný bit lze adresovat jenom přímou adresou .			
POP adresa	$(\text{adresa}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$		
	Ovlivňuje: ---	Slov: 2	Cyklů: 2
Instrukce vyzvedne obsah vrcholu zásobníku a uloží jej na adresované paměťové místo. Pak odečte od ukazatele zásobníku jedničku.			
PUSH adresa	$(SP) \leftarrow (SP) + 1, ((SP)) \leftarrow (\text{adresa})$		
	Ovlivňuje: ---	Slov: 2	Cyklů: 2
Instrukce přičte k ukazateli zásobníku jedničku a potom uloží obsah adresovaného místa do vrcholu zásobníku (zásobník je vždy ve vnitřní datové paměti).			

RET	$(PC(15 \div 8)) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$ $(PC(7 \div 0)) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$		
	Ovlivňuje: ---	Slov: 1	Cyklů: 2
Instrukce návratu z podprogramu vyzvedne ze zásobníku dva byty a uloží je do čítače instrukcí (PC). Nejprve vyjme vyšší byte a potom nižší byte a odečte od ukazatele zásobníku hodnotu dvě.			
RETI	$(PC(15 \div 8)) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$ $(PC(7 \div 0)) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$		
	Ovlivňuje: ---	Slov: 1	Cyklů: 2
Instrukce návratu z obslužného podprogramu přerušení vyzvedne ze zásobníku dva byty a uloží je do čítače instrukcí (PC). Nejprve vyjme vyšší byte a potom nižší byte a odečte do ukazatele zásobníku hodnotu dvě. Nakonec se povolí přijetí žádostí o přerušení se stejnou nebo nižší úrovní priority. Stavové slovo (jako 8048) se automaticky neobnovuje.			
RL A	$(A(n+1)) \leftarrow (A(n)), \text{ pro } n = 0,1,K,6$ $(A(0)) \leftarrow (A(7))$		
	Ovlivňuje: ---	Slov: 1	Cyklů: 1
Instrukce osmibitové logické rotace střadače o jednu pozici vlevo.			
RLC A	$(A(n+1)) \leftarrow (A(n)), \text{ pro } n = 0,1,K,6$ $(A(0)) \leftarrow (C), (C) \leftarrow (A(7))$		
	Ovlivňuje: C, P	Slov: 1	Cyklů: 1
Instrukce devítibitové logické rotace střadače a příznaku přenosu o jednu pozici vlevo.			
RR A	$(A(n)) \leftarrow (A(n+1)), \text{ pro } n = 0,1,K,6$ $(A(7)) \leftarrow (A(0))$		
	Ovlivňuje: ---	Slov: 1	Cyklů: 1
Instrukce osmibitové logické rotace střadače o jednu pozici vpravo.			
RLC A	$(A(n)) \leftarrow (A(n+1)), \text{ pro } n = 0,1,K,6$ $(A(7)) \leftarrow (C), (C) \leftarrow (A(0))$		
	Ovlivňuje: C, P	Slov: 1	Cyklů: 1
Instrukce devítibitové logické rotace střadače a příznaku přenosu o jednu pozici vpravo.			
SETB <bit>	Nastav bit		
SETB C	$(C) \leftarrow 1$		
	Ovlivňuje: C	Slov: 1	Cyklů: 1
SETB bit	$(\text{bit}) \leftarrow 1$		
	Ovlivňuje: ---	Slov: 2	Cyklů: 1
Instrukce nastaví přímo adresovaný bit na log.1.			
SJMP relativní adresa	$(PC) \leftarrow (PC) + 2, (PC) \leftarrow (PC) + \text{relativní adresa}$		
	Ovlivňuje: ---	Slov: 2	Cyklů: 2

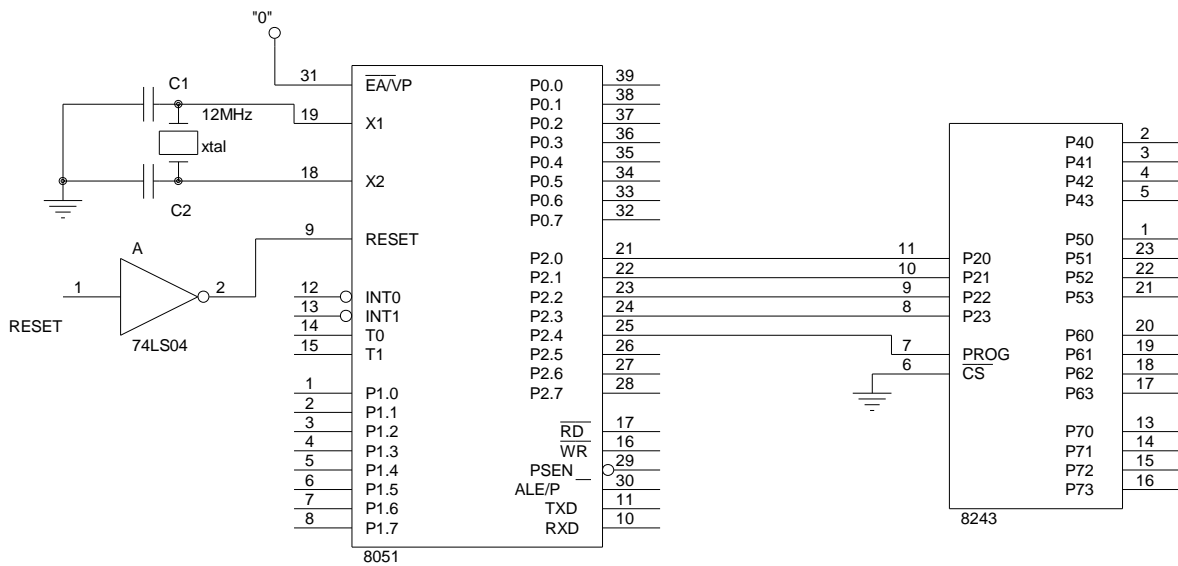
Instrukce realizuje krátký nepodmíněný skok na definovanou adresu, která se vypočítá jako součet čítače instrukcí ukazujícího na adresu následující instrukce a posunu (-128;127) určeného relativní adresou.			
SUBB A,<zdrojová slabika> Odečítání slabiky a příznaku přenosu od střadače.			
SUBB A, Rr	$(A) \leftarrow (A) - (Rr) - (C)$, kde $r = 0,1,K,7$		
SUBB A, @Rr	$(A) \leftarrow (A) - ((Rr)) - (C)$, kde $r = 0,1$		
	Ovlivňuje: C,AC,OV,P	Slov: 1	Cyklů: 1
SUBB A, adresa	$(A) \leftarrow (A) - (adresa) - (C)$		
SUBB A, #data	$(A) \leftarrow (A) - data - (C)$		
	Ovlivňuje: C,AC,OV,P	Slov: 2	Cyklů: 1
Instrukce odečte obsah adresované slabiky od střadače včetně příznaku přenosu (výpůjčky) a výsledek v něm ponechá. Adresovanou slabikou může být registr aktivní banky Rr , přímo uvedená adresa , nepřímo adresované paměťové místo registrem R0 nebo R1 (obsah registru určuje adresu místa jehož obsah budu odečítat) nebo přímo uvedená hodnota data . Je-li při výpočtu vyžadována v bitu b_7 výpůjčka, je příznakový bit C nastaven, v opačném případě je vynulován.			
SWAP A	$(A) \leftarrow (A(3 \div 0) * 16 - A(7 \div 4))$		
	Ovlivňuje: ---	Slov: 1	Cyklů: 1
Instrukce prohodí navzájem obsah nižšího a vyššího "půlbytu" střadače. Instrukce se shodná s osmibitovou rotací o čtyři bity vpravo nebo vlevo.			
XCH A,<slabika>	Zaměň obsah střadače a adresované slabiky		
XCH A, Rr	$(A) \leftrightarrow (Rr)$		
XCH A, @Rr	$(A) \leftrightarrow ((Rr))$, $r = 0,1$		
	Ovlivňuje: P	Slov: 1	Cyklů: 1
XCH A, adresa	$(A) \leftrightarrow (adresa)$		
	Ovlivňuje: P	Slov: 2	Cyklů: 1
Instrukce vymění obsah střadače a určeného registru nebo adresovaného paměťového místa.			
XCHD A,@Rr	$A(3 \div 0) \leftrightarrow (Rr(7 \div 4))$, kde $r = 0,1$		
	Ovlivňuje: P	Slov: 1	Cyklů: 1
Instrukce prohodí navzájem obsah nižší "půlbyte" střadače s nepřímo adresovaným paměťovým místem.			
XRL <cílová slabika>, <zdrojová slabika> Operace neekvivalence (EX-OR) mezi cílovou a zdrojovou slabikou.			
XRL A, Rr	$(A) \leftarrow (A) \oplus (Rr)$, kde $r = 0,1,K,7$		
XRL A, @Rr	$(A) \leftarrow (A) \oplus ((Rr))$, kde $r = 0,1$		
	Ovlivňuje: P	Slov: 1	Cyklů: 1

XRL A, adresa	$(A) \leftarrow (A) \oplus (\text{adresa})$		
XRL A, #data	$(A) \leftarrow (A) \oplus \text{data}$		
XRL adresa, A	$(\text{adresa}) \leftarrow (\text{adresa}) \oplus (A)$		
	Ovlivňuje: P - pro zápis do střadače	Slov: 2	Cyklů: 1
XRL adresa, #data	$(\text{adresa}) \leftarrow (\text{adresa}) \oplus \text{data}$		
	Ovlivňuje: ---	Slov: 3	Cyklů: 2
Instrukce ORL provede logický součin mezi odpovídajícími bity cílového a zdrojového bytu a výsledek uloží do cílové slabiky. Pro operaci s výstupní bránou (přímá adresa) se operace provede mezi výstupním registrem a zdrojovým bytem a nikoliv vstupními signály.			

Připojení externích periférií k procesoru 8051

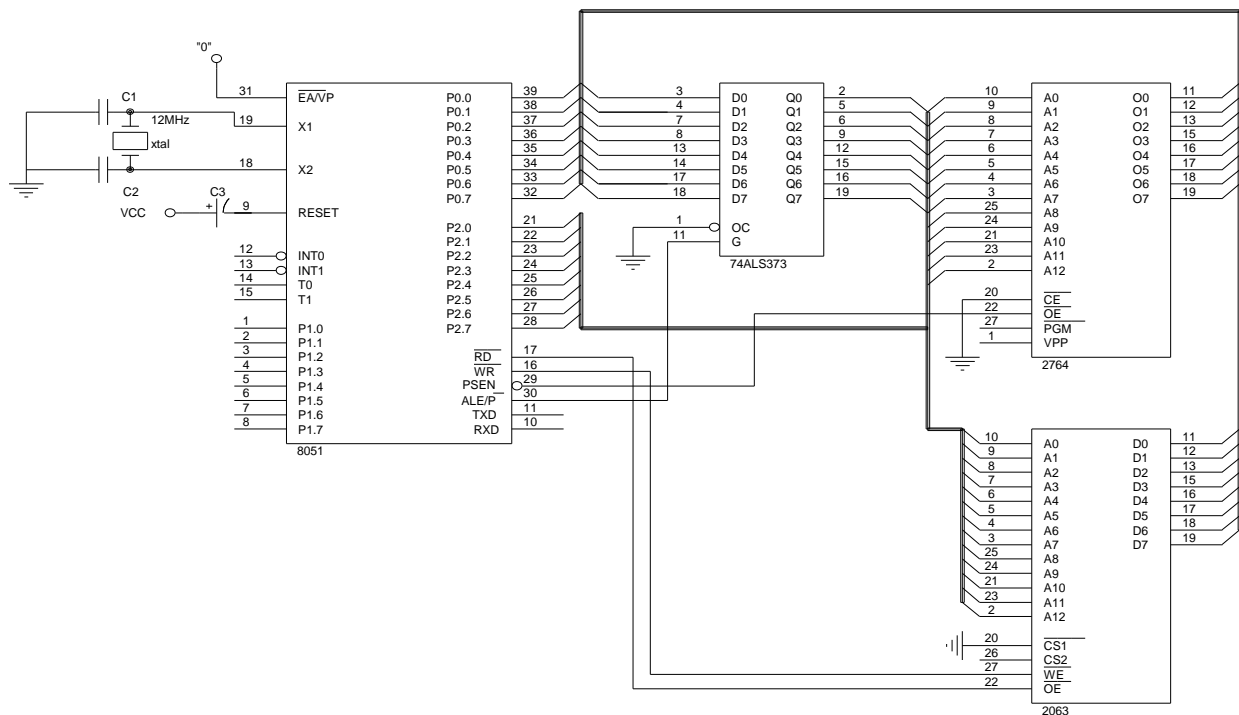
Pokud aplikace procesoru vyžaduje větší paměť, počet vstupů/výstupů, případně další pomocné obvody jako jsou hlídač časovač, obvod reálného času, hlídač napájecího napětí a pod., je možné k procesoru připojit externí obvody a komunikovat s nimi pomocí bran procesoru jednak standardně, jednak je tzv. paměťově mapovat a ke komunikaci použít externí datovou a adresovou sběrnici procesoru. Ve druhém případě je třeba k zachycení nižšího bytu adresy použít vnější záchytný adresový registr. Některé rozšiřující obvody ve své struktuře obsahují tento registr a jsou pro komunikaci s procesorem přizpůsobeny. Pokud navíc je nutné použít při rozšíření systému další logiku např. adresové dekodéry, přepínače, jednoduché obvodové řadiče a pod. je vhodné tyto obvody implementovat pomocí programovatelných obvodů PAL, GAL, FPGA. Pro ilustraci je uvedeno :

- a) rozšíření vstupů/výstupů obvodem 8243, který sice není speciálně určen pro procesor 8051 (byl navržen pro Intelský procesor 8048), ale s programovou obsluhou je možné poměrně efektivně rozšířit počet V/V. Programová obsluha 8243 je uvedena mezi příklady programů 8051. Obsluha je nutno vyhradit 5 vývodů V/V procesoru k přenosu dat a řízení obvodu 8243 a pokud těchto obvodů použijeme více, dalšími vývody V/V obvodu 8243 adresujeme a to buď přímo, nebo prostřednictvím adresového dekodéru. Připojení jednoho obvodu 8243 k procesoru 8051 je na obr.30, jeho výběr je trvalý
- b) na obr.31 je připojení externí datové a programové paměti, které lze označit jako standardní. U mikropočítačů používaných jako vývojové kity (evaluation boards) se



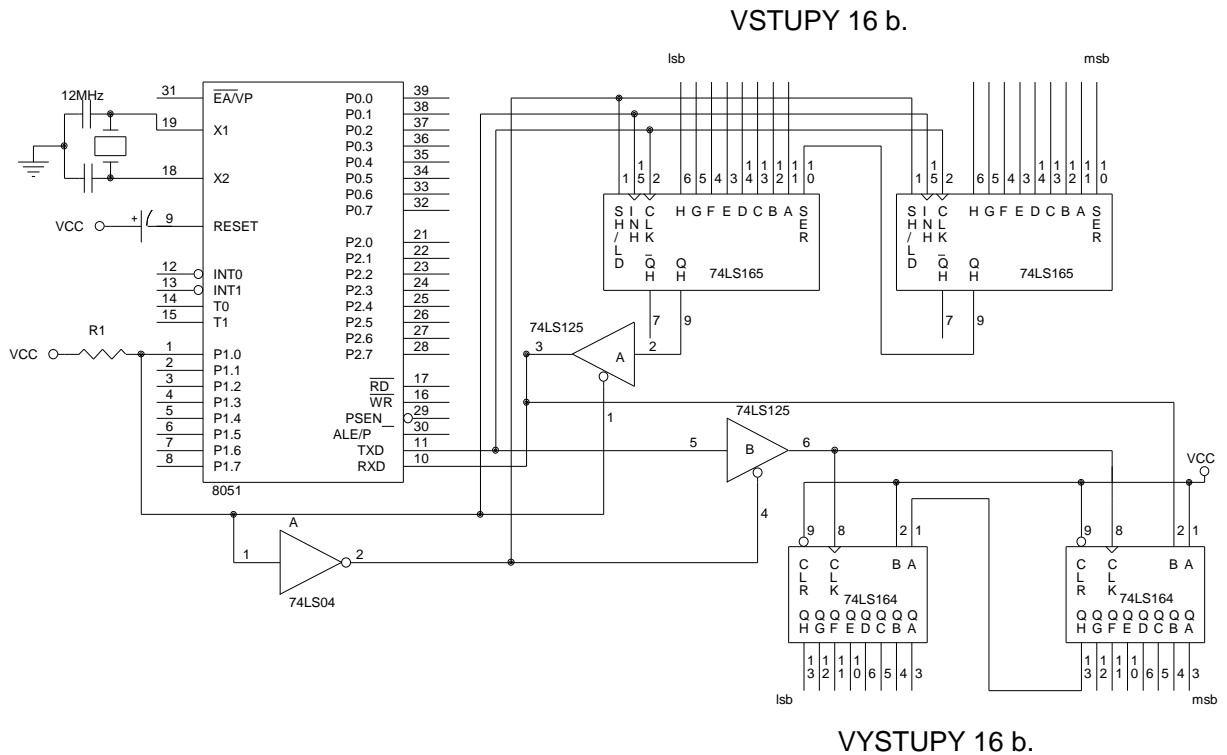
Obr. 30 Rozšíření vstupů/výstupů obvodem 8243

žadá možnost zavádět vyvíjené programy do datové paměťové oblasti a v ní je pak spouštět a ladit. V tom případě je nutné, aby paměťové prostory programový a datový se překrývaly. To se jednoduše docílí tím, že na řídicí vstup paměti OE# (output enable) se přivede logický součin řídicích signálu procesoru PSEN# a RD# c) na obr. 32 je zobrazeno rozšíření vstupů/výstupů pomocí posuvných registrů obsluhovaných seriovým kanálem procesoru pracujícím v módu 0 (synchronní seriový přenos).

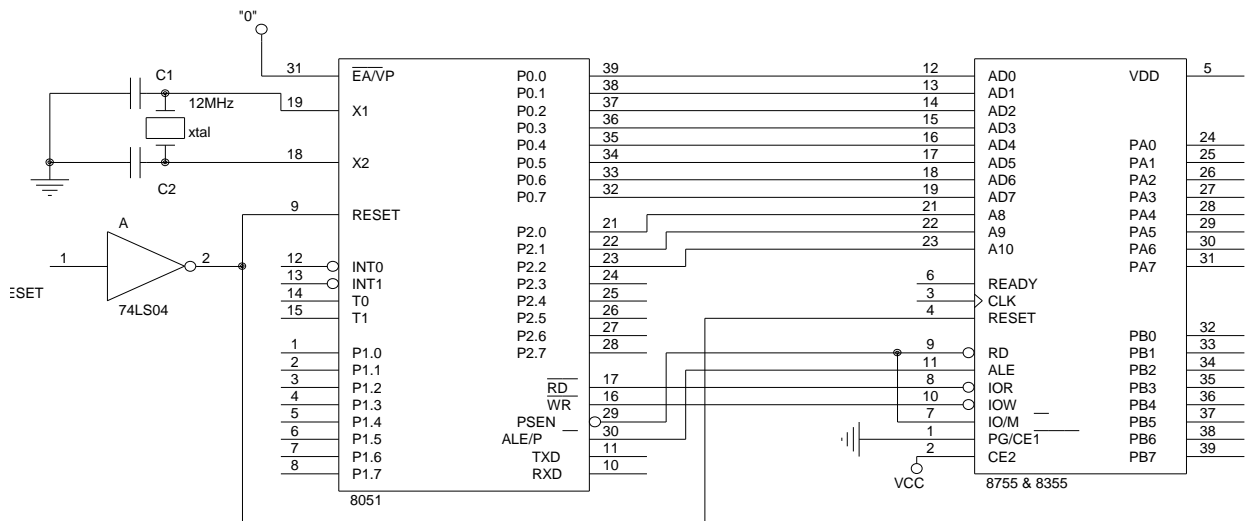


Obr. 31 Připojení externí programové a datové paměti

Rozšíření V/V posuvnými registry



Obr. 32 Rozšíření vstupů/výstupů pomocí posuvných registrů

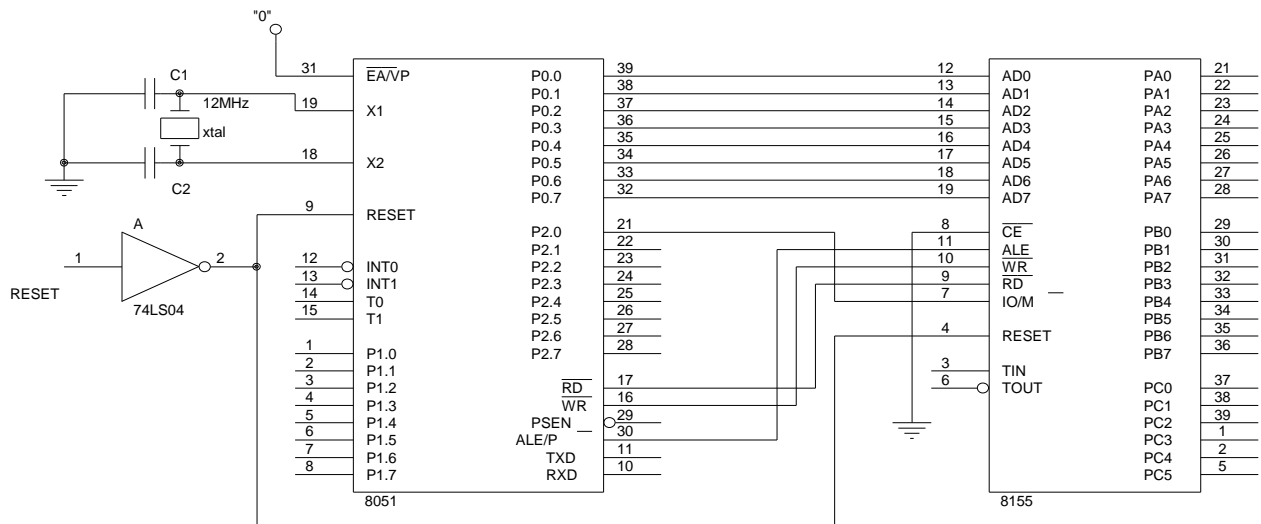


Obr. 33 Připojení externí programové paměti a rozšíření V/V

V zapojení jsou použity starší posuvné registry 74164 (8bitů s paralelními výstupy) a 74165 (8bitů s paralelními vstupy), které nemají třístavové výstupy a je proto nutné, pokud žádáme obousměrný provoz na jediném datovém datovém vstupu procesoru, oddělit seriový výstup vstupního posuvného registru 74165 od seriového vstupu výstupního posuvného registru 74164. Z novějších prvků se doporučují registry 74HC595 (Serial-in/Paralel-out) a 74HC589 (Paralel-in Serial-out), které při propojení seriových vstupů - výstupů nevyžadují prepínací logiku. d)

na obr. 33 je rozšíření vstupů/výstupů a připojení externí programové paměti pomocí kombinovaného obvodu 8355/8755 (2kB ROM/EPROM a dvě paralelní číslicové brány) z rodiny MCS85

- e) na obr 34 je rozšíření vstupů/výstupů a připojení externí datové paměti pomocí kombinovaného obvodu 8155 (256bitů RAM a dvě paralelní číslicové brány 8bitů a jedna paralelní brána 6bitů) z rodiny MCS85.



Obr. 34 Připojení externí datové paměti , časovače a rozšíření V/V