

Registr SP – stack pointer, **PC** – adresa následující instrukce, **CCR** – obsahuje flagy pro větvení programu, **EXR**- řízení přerušení a trasování

Paměti bez napájení – Flash, EPROM, EEPROM

Maskované přerušení nastavit – programově – nastavením registru

Nemaskovatelné přerušení lze povolit nebo zakázat - nelze je zakázat žádným způsobem

Výhodné přerušení – přenos po znacích

Testování signálu -žádost o uvolnění sběrnice (BREQ) a při aktivitě uvolněna – po dokončení strojového cyklu

Několik IRQ – obslužen s nejvyšší prioritou

adresní vývody procesoru po odpovědi BUSACK na signál BREQ – odpojeny

periferní obvody mikrokontrolérů – paralelní IO porty, sériové rozhraní, obvody čítačů a časovačů
vnitřní paměť I8051 – 128Byte

připojení vnější paměti k I8051 – s využitím paralelních portů

statické CMOS – malá spotřeba v klidovém stavu

ochrana paměti pomocí baterie - statické CMOS

probíhající DMA přenos – procesor odpojen od sběrnice

RAM udrží data – připojeno napájení + periodické zotavení

Ve kterém okamžiku je testován stav vstupního pinu přerušení (INT) a při jeho aktivitě je vyvolán obslužný program?-po dokončení instrukce

Přenos – vhodné použití DMA – blokové přenosy

Dynamické paměti RAM mají kromě jiných vlastností i některé y následujících:

-možnost automatického zotavení dat....., vysokou rychlost – nejvyšší z dostupných typů paměti, vnitřního čítače adres zotavení, možnost rychlého stránkového čtení dat

Při návratu z obslužného programu přerušení se návratová adresa určí: -přečte se z vrcholu zásobníku

Paměti EPROM -lze po naprogramování vymazat UV zářením a znovu naprogr. - celkem cca 100x

Napište názvy alespoň 4 různých cyklu, které lze aplikovat dynamické paměť

-čtecí, zápisový, cykl read-write, zotavovací

Následující tabulka porovnává vlastnosti statických a dyn paměti.....

-Dynamická RAM –nízká cena, velká kapacita, nízká rychlost

-Statická RAM-opak Dynamické

Při provádění obslužného programu přerušení jsou další přerušení

-povolena nebo zakázána v závislosti na konkrétním obslužném programu

Statická pamet RAM udrží obsah -jen při pripojenem napájecím napeti

Jak dlouhá adresa je zapotřebí k adresování pameti o kapacite 32kbyte?

- $32 \cdot 1024 = 32\,768$, $32\,768 / 8 = 4096$, $4096 = 2^{12}$ vysledek **12**

Každá bunka dynamicke pameti RAM se musí zotavovat priblizne po -10ms

Které z nasledujicich duvodu vedou k tomu ze se pamet flash nepouziva jako hlavni operacni pamet počítače - omezeny počet povolených zápisů, (možná i menší rychlost oproti RAM)

Pro které typy prenosu je nejmene vhodne použití přerušení po kazdem znaku - velmi rychle přenosy

Převody mezi soustavami:

Přímý kód - binární kód; **Přímý se znaménkem** – první bit znaménkový

Doplňkový kód – + - stejný jako binární -- na 16bitů – $2^{16}-1$ -hledané číslo +1

Kód s posunutou nulou - na 8bitů – nula = 2^7-1

Pohyblivá řádová čárka - znaménko + 8bitů exponent (posunutá nula) + mantisa (bez první jedničky)

BCD kód - číslice – 4bity, korekce +- 0110(6)

DMA přenos:Datový registr V/V je plný;V/V generuje DRQ; DMA řadič generuje BREQ; Procesor dokončí sběrniceový cykl, odpojí se od sběrnice, aktivuje BUSACK;DMA aktivuje DACK a vysílá na sběrnici adresu a povel „zápis“;V/V řadič vysílá na datovou sběrnici obsah datového registru; V/V řadič zruší DRQ; DMA řadič BREQ a DACK, přestane vysílat adresu a povel „zápis“. Registr adresy se inkrementuje, registr délky se dekrementuje; CPU zruší BUSACK;CPU se připojí na sběrnici

(10 b.) Napište v assembleru H8S program, který převede řetězec (8 znaků) s ASCII zobrazením binárního obsahu registru R0L na jeho binární obsah (např. pro řetězec se znaky '0', '1', '1', '0', '1', '0', '0', '1' bude odpovídající obsah registru R0L = 01101001b). Řetězec leží v paměti, adresa prvního znaku s nejvyšším bitem registru R0L je uložena v ER6, další znaky jsou postupně na vyšších adresách.

```
• h8300s

• data

vstup: .asciz "01101001"
      • space 100
stack:

• text
• global _start

_start:
mov.l #stack, ER7
mov.l #vstup, ER6
xor.b R0L, R0L

etiznak:
mov.b @ER6, R1L
cmp.b #0, R1L
beq   hotovo
and.b #1, R1L
shl.b R0L
add.b R1L, R0L
inc.l #1, ER6
bra   etiznak

hotovo:
    **

koniec: bra   koniec
```

