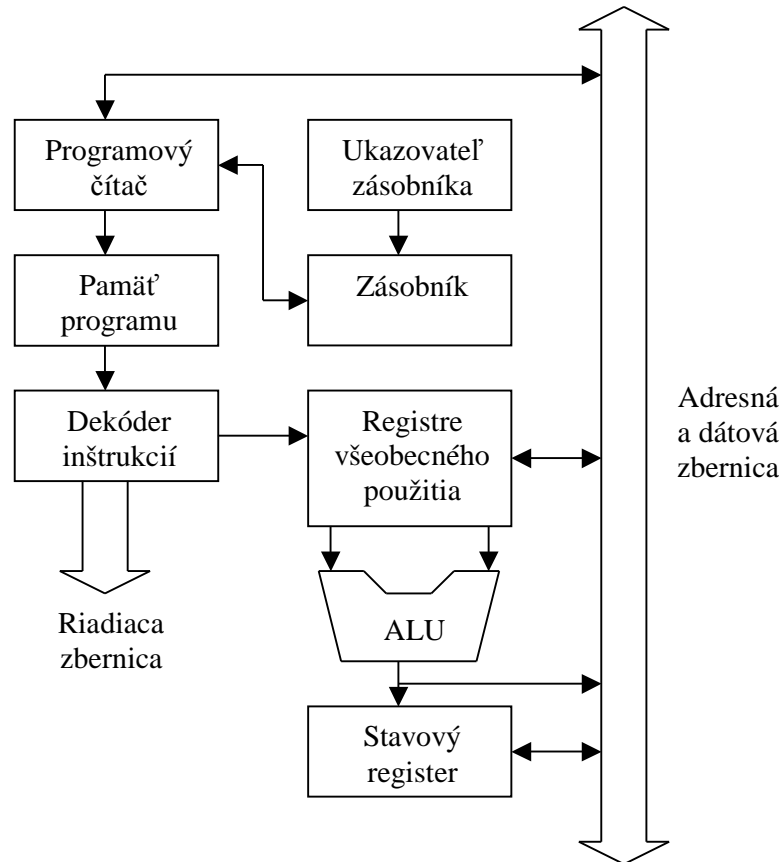


ÚVOD

CPU emulátor slúži na emuláciu jednoduchého 8-bitového procesora. Emulácia je softvérovo-hardvérová. V praxi to znamená, že väčšina inštrukcií procesora je vykonávaná výlučne pomocou emulačného programu (CPUemul) a len inštrukcie pre komunikáciu s externou pamäťou a vstupno-výstupnými (V/V) zariadeniami sú vykonávané s pomocou hardvérového emulátora.

ŠTRUKTÚRA PROCESORA



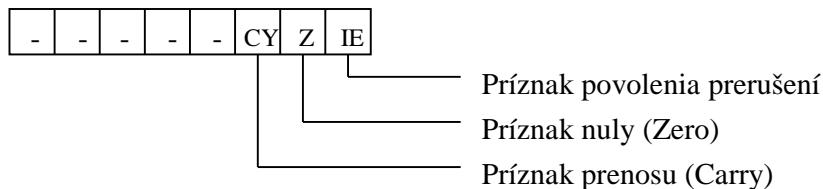
Obr.1 Architektúra emulovaného procesora

Aritmeticko-logická jednotka (ALU) podporuje aritmetické a logické funkcie medzi dvoma registrami alebo medzi registrom a konštantou. Jednoregistrové operácie sú taktiež vykonávané v ALU. Procesor používa Harwardskú architektúru – s oddelenými pamäťami a zbernicami dát a programu.

REGISTRE PROCESORA

Procesor obsahuje štyri **8-bitové registre na všeobecné použitie**. Označujú sa **A, B, C, D**. Tieto registre je možné použiť vo všetkých inštrukciách pre prácu s 8-bitovými registrami. Napríklad je možné načítať konštantu do registra, vykonať rôzne aritmetické operácie (či už s dvomi registrami, s jedným registrom, alebo s jedným registrom a konštantou) a tiež logické operácie a posuny.

Podľa výsledku matematických operácií sa nastavujú príznakové bity **stavového registra F** (Flag register), ktorý je tiež 8-bitový, ale využité sú len spodné 3 bity:



Procesor obsahuje tiež tri **16-bitové registre**:

- **PC** (Program Counter) – ukazovateľ nasledujúcej inštrukcie
- **SP** (Stack Pointer) – ukazovateľ vrcholu zásobníka
- **MP** (Memory Pointer) – ukazovateľ pre nepriame adresovanie pamäte

Do 16-bitových registrov SP a MP je možné buď priamo zapísať konštantu, alebo dvojicu registrov AB. Na čítanie z týchto registrov je potrebné použiť dvojicu registrov CD.

Zásobník má veľkosť 64 kB (65536 bajtov). Veľkosť programu je obmedzená na maximálne 65536 inštrukcií.

ZBERNICA

Emulovaný procesor je 8-bitový, t.j. **dátová zbernica má šírku 8 bitov. Šírka adresnej zbernice je 16 bitov**. Tomu zodpovedá adresný pamäťový a vstupno-výstupný priestor s rozsahom 0-65535.

Riadiaca zbernica obsahuje 9 signálov, z toho je 5 výstupných a 4 sú vstupné.

Výstupné riadiace signály:

- \overline{MW} (Memory Write) – zápis do pamäte
- \overline{MR} (Memory Read) – čítanie z pamäte
- \overline{IW} (Input/Output Write) – zápis do V/V zariadenia
- \overline{IR} (Input/Output Read) – čítanie z V/V zariadenia
- \overline{IA} (Interrupt Acknowledge) – potvrdenie prijatia požiadavky o prerušenie

Vstupné riadiace signály:

- IT (Interrupt) – požiadavka o prerušenie
- RY (Ready) – potvrdenie pripravenosti zariadenia
- BQ (Bus Request)
- BA (Bus Acknowledge)

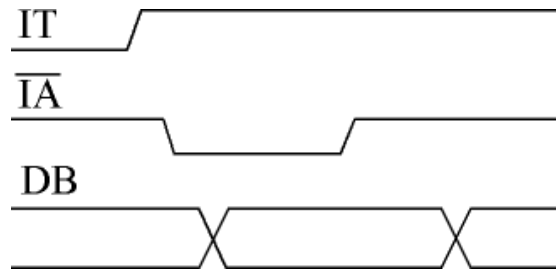
PRERUŠENIA

Procesor umožňuje použiť až 16 prerušení. Pri požiadavke o vyvolanie prerušení prostredníctvom signálu IT, ak je povolené prerušenie, potvrdí procesor prijatie požiadavky signálom \overline{IA} . Následne prečíta procesor z dátovej zbernice číslo obslužného programu (0–15) a zavolá príslušný obslužný program. Názov obslužných programov je int00 – int0f. Pri vyvolaní prerušení je automaticky zakázané ďalšie prerušenie.

Ak je v obslužnom programe príkaz na povolenie prerušení (zvyčajne býva tento príkaz až na konci obslužného programu, ale môže byť aj inde), je možné aj vnorenie prerušení. Počet vnorených prerušení je obmedzený len veľkosťou zásobníka.

Prerušenie je možné vyvolať buď podľa úrovne signálu IT (keď je v logickej úrovni „1“), alebo pri zmene signálu IT (kontroluje sa zmena z „0“ na „1“). Konkrétny spôsob vyvolávania prerušení je možné nastaviť v emulačnom programe.

Na nasledovnom obrázku je znázornený časový priebeh signálov pri zisťovaní čísla prerušení procesorom.



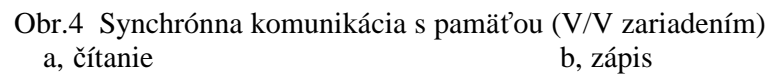
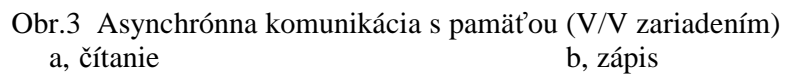
Obr.2 Načítanie čísla prerušení

Postupnosť krokov pri zisťovaní čísla prerušení:

1. Zariadenie vyšle požiadavku na prerušenie ($IT = 1$)
2. Procesor oznámi prijatie požiadavky ($\overline{IA} = 0$)
3. Zariadenie zapíše na dátovú zbernicu číslo prerušení
4. Po uplynutí daného času procesor prečíta číslo prerušení a zruší signál \overline{IA}

KOMUNIKÁCIA PO ZBERNICI

Komunikácia procesora s pamäťou, alebo s vstupno-výstupnými zariadeniami môže byť synchronná alebo asynchronná. Konkrétny spôsob komunikácie sa nastavuje v emulačnom programe. Na nasledovných obrázkoch sú znázornené časové priebehy signálov pre všetky možnosti komunikácie.



PROGRAM CPUemul

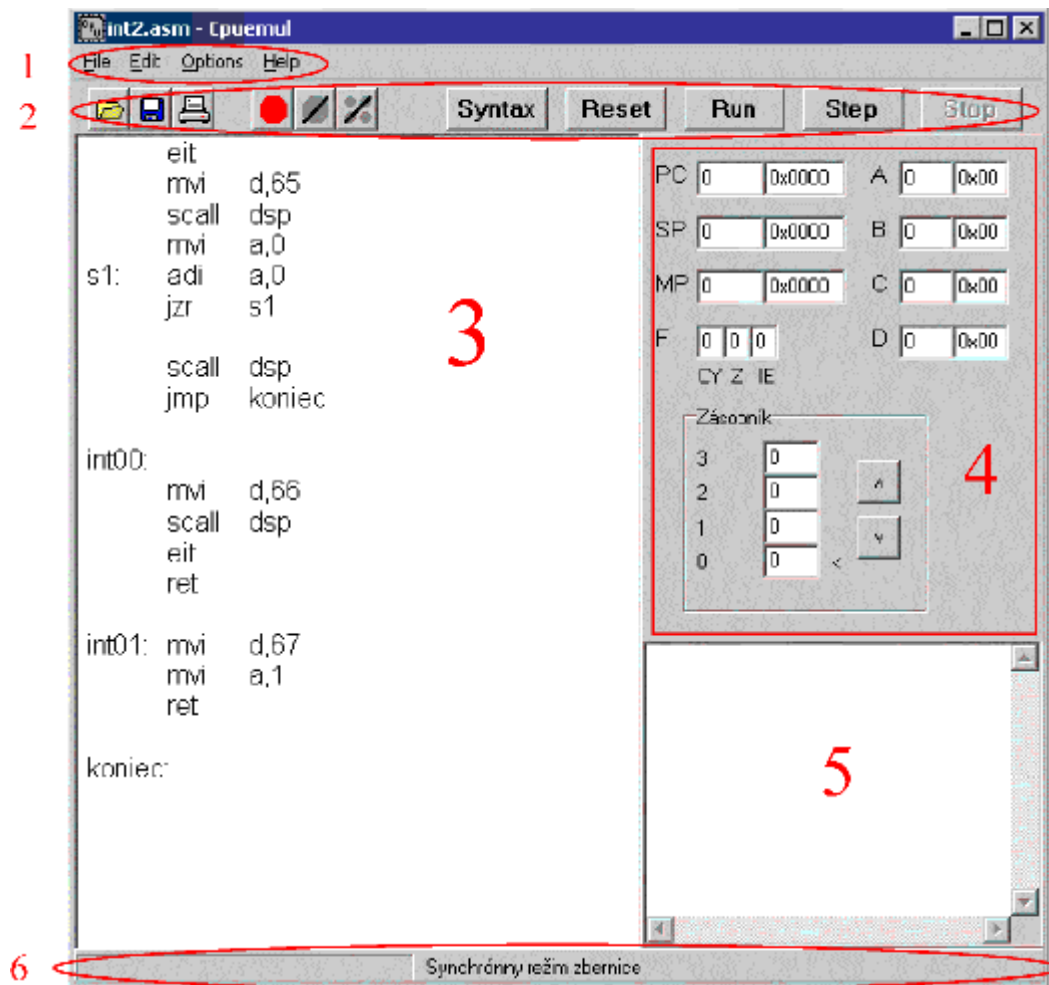
Umožňuje vytváranie a spúšťanie programov písaných v špeciálnom jazyku symbolických adries (asembleri). Program je možné používať aj samostatne (bez hardvérového emulátora), ale v tom prípade nie je možné využiť tie inštrukcie procesora, ktoré prístupujú do pamäte alebo vstupno-výstupného zariadenia.

Po spustení programu musí užívateľ zadať identifikačné číslo, ktoré je v tvare Sxxx (xxx = 000...999). Podľa zadaného čísla sa vytvorí domovský adresár užívateľa D:\Student\Sxxx (pokiaľ taký adresár ešte neexistuje). Programy vytvorené užívateľom je možné uložiť len do domovského adresára.

Hlavné okno programu sa skladá zo šiestich častí. Sú to:

1. Menu
2. Lišta s ovládacími tlačidlami
3. Okno pre editáciu programu
4. Stav registrov procesora a zásobník
5. Displej
6. Riadok s helpom

Jednotlivé časti okna sú znázornené na obrázku:



Menu obsahuje položky:

- File – práca so súbormi
 - New – vytvorenie prázdneho súboru na editáciu
 - Open – otvorenie existujúceho súboru
 - Save – uloženie súboru
 - Save As – uloženie súboru pod iným menom
 - Print – tlač súboru
 - Exit – ukončenie práce s emulátorom
- Edit – editácia súboru
 - Undo – vrátenie poslednej zmeny
 - Cut – vystrihnutie označeného textu
 - Copy – kopírovanie označeného textu
 - Paste – vloženie textu na pozíciu kurzora
- Options – nastavenia emulátora
 - Synchr. – synchrónny režim zbernice
 - Asynchr. – asynchrónny režim zbernice
 - INT - level – vyvolanie prerušenia od úrovne signálu IT
 - INT - change – vyvolanie prerušenia od zmeny signálu IT
 - Micro Step – povolenie/zakázanie mikrokrokovania
- Help
 - About – informácia o verzii softvéru

FUNKCIA TLAČIDIEL

- Syntax:** Umožňuje skontrolovanie vytvoreného programu. Ak program obsahuje chyby, vypíše sa hlásenie o prvej chybe a označí sa miesto v programe, ktoré obsahuje chybu. V prípade, že je program bez syntaktických chýb, vypíše sa o tom hlásenie.
- Reset:** Spôsobí vynulovanie všetkých registrov procesora a zásobníka. Ak bol program práve v režime krokovania, krokovanie sa ukončí.
- Run:** Umožňuje spúšťanie programu plnou rýchlosťou. Emulačný program načítava jednotlivé inštrukcie a vykonáva ich. Ak emulátor nájde breakpoint, zastaví sa vykonávanie programu na danej inštrukcii a aktuálny riadok sa vyznačí. Počas behu programu sa aktualizuje výpis obsahu registrov procesora a obsah zásobníka. Keď je spustená emulácia, nie je možné editovať program.
- Step:** Umožňuje krokovanie vytvoreného programu po jednotlivých inštrukciách, alebo po mikroinštrukciách (režim mikrokrokovania). Počas krokovania je nasledovná inštrukcia zvýraznená. Kedykoľvek je možné prerušiť krokovanie a spustiť program plnou rýchlosťou (Run). V režime mikrokrokovania je správanie emulátora obdobné, s tým rozdielom, že pri inštrukciách pristupujúcich na zbernicu (LMI, SMI, LMR, SMR, INN, OUT) je umožnené krokovanie inštrukcie podľa jednotlivých zásahov na zbernicu. Počas mikrokrokovania uvedených inštrukcií sa v pravom dolnom rohu hlavného okna vypisuje test popisujúci nasledovnú akciu na zbernici. V režime mikrokrokovania je možné spustiť program plnou rýchlosťou až keď je ukončená celá inštrukcia.
- Stop:** Slúži na zastavenie behu programu v prípade, že došlo ku zacykleniu (nesprávne napísaný program, zlyhanie asynchrónnej komunikácie ...)