

Práce s koprocесorem FPU

Ing. Dominika Regéciová
Výzkumná skupina formálních modelů

Vysoké učení technické v Brně, Fakulta informačních technologií
Božetěchova 1/2, 612 66 Brno - Královo Pole
iregeciova@fit.vutbr.cz



- **Jaký bude obsah registru AL po provedení následujících instrukcí?**

```
mov al,5  
shl al,3
```

- **Co bude výsledkem následujících instrukcí?**

```
mov ax,1501  
mov bx,3  
idiv bx
```

- Jaký bude obsah registru AL po provedení následujících instrukcí?

```
mov al,5  
shl al,3  
40
```

- Co bude výsledkem následujících instrukcí?

```
mov ax,1501  
mov bx,3  
idiv bx
```

Chyba, nutno přidat mov dx, 0 (pokud dělíme BX, dělenec je v DX:AX)

- ALU - Arithmetic Logic Unit
 - Obecné registry EAX, EBX, ECX, EDX
 - Příznakový registr EFLAGS
 - Zásobník s ukazateli ESP a EBP
 - Používáno na všech předchozích cvičení
- FPU - Floating-point Unit
 - Vlastní registry
 - Vlastní příznakový registr
 - Vlastní instrukční sada (instrukce začínající písmenem F)
 - Bude zkoušeno v testu další týden!
- ALU a FPU jsou dvě hardwarově nezávislá zařízení!

- Pro obecné použití má FPU vyhrazeno 8 registrů `st0` – `st7`
 - Registry FPU fungují jako zásobník
 - Každý registr má 80 bitů
 - Nová data jsou vkládána vždy do `st0`
 - Předchozí obsah `st0` je odsunut do `st1`, z `st1` do `st2`, atd.
 - Naopak, při odstranění hodnoty z `st0` se do něj přesune hodnota z `st1`
- Pozor, neplést si zásobník FPU a zásobník ALU
 - Zásobník FPU může snad přetéct i podtéct
 - Hodnoty jsou v FPU vždy uloženy v plovoucí řádové čárce
 - Zobrazení obsahu registrů FPU v NASM
Settings → Common → Show all registers in debug

- Pro načtení FP hodnoty se používá `FLD src`, načítat lze z:
 - Paměti 32b (float)
 - Paměti 64b (double)
 - Jiného registru
- Pro načtení integeru se používá `FILD src`, načítat lze z:
 - Paměti 16b (word integer)
 - Paměti 32b (short integer)
 - Paměti 64b (long integer)
- Načtení speciálních konstant
 - `FLDZ` - načte `0.0`
 - `FLD1` - načte `1.0`
 - `FLDPI` - načte π

- Pro ukládání hodnot se používá `FST dest`, jsou 4 varianty:
 - `FST` - uložení FP hodnoty bez odstranění z `st0`
 - `FIST` - uložení integeru bez odstranění z `st0`
 - `FSTP` - uložení FP hodnoty s odstraněním z `st0`
 - `FISTP` - uložení integeru s odstraněním z `st0`
- Hodnota může být uložena do:
 - Paměti 16b (word integer)
 - Paměti 32b (short integer, float)
 - Paměti 64b (long integer, double)
 - Jiného FPU registru
- Příponu `P` pro odstranění vrcholu zásobníku lze přidat i k mnoha jiným instrukcím

- Pro základní aritmetické instrukce `FADD`, `FSUB`, `FMUL`, `FDIV`
Existují 3 možnosti použití:
 - `FADD [var] ; st0 = st0 + [var]`
 - `FADD stn ; st0 = st0 + stn`
 - `FADD stn, st0 ; stn = stn + st0`
- Knihovní funkce pro vstup a výstup
 - `CALL ReadFloat ; EAX = stdin (32b)`
 - `CALL ReadDouble ; st0 = stdin (64b)`
 - `CALL WriteFloat ; stdout = EAX (32b)`
 - `CALL WriteDouble ; stdout = st0 (64b)`

- Porovnávací instrukce
 - `FTST ;porovna st0 a 0.0`
 - `FCOM src ;porovna st0 a src`
- Porovnání nastaví příznaky C3, C2 a C0 ve stavovém registru FPU
 - `st0 > src` \rightarrow C3 = 0, C2 = 0, C0 = 0
 - `st0 < src` \rightarrow C3 = 0, C2 = 0, C0 = 1
 - `st0 = src` \rightarrow C3 = 1, C2 = 0, C0 = 0
 - neporovnatelné \rightarrow C3 = 1, C2 = 1, C0 = 1
- Řídící instrukce:
 - `FSTSW AX ;ulozi stavovy registr FPU do AX`
 - `SAHF ;ulozi AH do (casti) registru EFLAGS`
- Po převodu platí:
 - `ZF == C3`
 - `PF == C2`
 - `CF == C0`
- Můžeme tak použít standardní podmíněné skoky

- Pokročilé aritmetické instrukce

- FSQRT ;st0 = odmocnina z (st0)
- FSIN ;st0 = sin(st0) v radiánech
- FCOS ;st0 = cos(st0) v radiánech
- FABS ;st0 = abs(st0)
- FCHS ;st0 = -(st0)
- FXCH dest ;prohodí st0 a dest (implicitně st1)

- Příště: třetí test, za který můžete získat až 6 bodů
- Programování koprocesoru
- Můžete používat vlastní poznámky, přehled instrukcí a materiály k přednáškám
- Cvičení v 13. týdnu **nebude** (svátek práce)
 - Lze si nahradit u jiného cvičícího v úterý
 - Doporučuji se podívat na obsah probírané látky v rámci wiki stránek a stránek ostatních cvičících
- Příští týden bude zveřejněno zadání bonusové úlohy

