1 Descrição

o trabalho consiste em implementar um processador emaulado em duas versoes.

O módulo de carga deve estar em C e a implementação da CPU em Assembly x86-64

Todos os grupos vao implementar a mesma maquina 😊

2 Referencia

Como referencia são postados dois arquivos maquina_piulha.asm e maquina_pilha.c

No projeto fazemos a leitura direto do teclado, recomendamos usar um arquivo txt com o código de maquina da cpu

Montar o assembly

```
nasm -f elf64  maquina_pilha.asm
```

Compilar e linkeditar

```
gcc -m64 -no-pie maquina_pilha.c maquina_pilha.o  -o   processador
```

executar

```
./processador
```

3 Escpecificacao

Máquina de 8 bits com endereçamento de 16 bits

memoria: 64 kbytes (memoria: db 65536 db 0)

maquina de segmento único de memoria

3.1 Flags

- **Zero -** setado em 1 quando o resultado é zero, setado em 0 se diferente de zero
- **Negativo** - setado em 1 se resultado é menor que zero, serado em 0 caso contrario
- **Carry -** Setado em 1 se tivermos um estouro de capacidade

3.2 Registradores

Endereco /generico- 16 bits

- A0 (codigo 0)

- A1 (código 1)

Generico 8 bits

- D0 (código 2)
- D1 (código 3)
- D2 (código 4)
- D3 (código 5)

Genérico 32 bits

- H0 (código 6)
- H1 (código 7)

Operações podem ser realizadas apenas com registradores do mesmo tamanho.

4. ISA

Onde aprece R temos um registrador, XX 1 byte

R é codificado em 4 bits, na pare alta do endereço, logo temos a possibilidade de usar dois registradores em 1 byte

PUSH R Código 00

Empilha o valor de um registrador

Exemplo push D0 Maquina 0020

Push H0 Maquina 0070

POP R Código 01

desempilha o valor de um registrador

Exemplo pop D1 Maquina 0130

Pop A0 Maquina 0100

Load R, Const Código 02

Atribui ao registrador R a constante Const

Exemplo Load A0,0FAC -> 02000FAC

Load D0,00 -> 021000

Load R, [XXXX] Código 03

Atribui ao registrador R o valor presente na posição XXXX de memoria

Exemplo

Load H0,[1010] -> 03601010

Load D0,[0000] -> 03200000

Load R, [Rx] Código 04

Atribui ao registrador R o valor presente na posição de memoria presente em Rx (Rx pode ser A0 ou A1)

Exemplo

Load H0,[A0] -> 0460

Load D0,[A1] -> 0431

Store [XXXX],R Código 05

Armazena no endereço XXXX de memoria o valor do registrador R

Exemplo

Store [0100],A0 -> 05010000

Store [Rx],R Código 06

Armazena no endereço de memoria em Rx (A0,A1 apenas) o valor de R

Exemplo

Store [A0],D0 -> 0602

HALT Codigo 07

Termina o programa e volta para o código C de carga

Exemplo Halt -> 07

ADD R1,R2 Código 08

R1=R1+R2 seta Zero, Negativo, Carry . R1 e R2 do mesmo tamanho

Exemplo : ADD A0,A1 -> 0801

SUB R1,R2 Código 09

R1=R1-R2 seta Zero, Negativo, Carry . R1 e R2 do mesmo tamanho

Exemplo : SUB A0,A1 -> 0901

AND R1,R2 Código 0A

R1=R1 and R2 seta Zero, Negativo, Carry . R1 e R2 do mesmo tamanho

Exemplo : AND A0,A1 -> 0A01

OR R1,R2 Código 0B

R1=R1 or R2 seta Zero, Negativo, Carry . R1 e R2 do mesmo tamanho

Exemplo : OR A0,A1 -> 0B01

XOR R1,R2 Código 0C

R1=R1 xor R2 seta Zero, Negativo, Carry . R1 e R2 do mesmo tamanho

Exemplo : XOR A0,A1 -> 0C01

NOT R Código 0D

R1= not R1 seta Zero, Negativo, Carry .

Exemplo : NOT A0 -> 0D00

CMP R1,R2 Código 0E

Equivale a um sub sem destruir R1, apenas seta flags

Exemplo Cmp D0,D1 -> 0E34

JMP Código 0F

Retira do topo da pilha um endereço (2 bytes) e desvia para o mesmo.

Exemplo: Jmp -> 0F

JL Código 10

Se negativo==1 realiza um jmp

Exemplo: JL -> 10

JG Código 11

Se negativo==0 e Zero=0 realiza um jmp

Exemplo: JG -> 11

JLE Código 12

Se negativo==1 ou e Zero=1 realiza um jmp

Exemplo: JLE -> 12

JGE Código 13

Se negativo==0 realiza um jmp

Exemplo: JGE -> 13

JC Código 14

Se carry==1 realiza um jmp

Exemplo: JC -> 14

JNC Código 15

Se carry==0 realiza um jmp

Exemplo: JNC -> 15

IN R Código 16

Lê um caracter em ascII para R (R deve ser D0,D1,D2,ou D3)

Exemplo: IN D0 1620

OUT R Código 17

Escreve um caracter em ascII para R (R deve ser D0,D1,D2,ou D3)

Exemplo: OUT D0 1720