

Unidade de Ponto Flutuante

Programação Aplicada a Ciência da Computação.
Prof. Dr. Eduardo S. Pereira.

[http:
//eduardopereira.upcursosetreinamentosonline.com/](http://eduardopereira.upcursosetreinamentosonline.com/)

11 de maio de 2018

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Unidade de
Ponto
Flutuante
Exercício

1 Unidade de Ponto Flutuante

2 Exercício

Unidade de Ponto Flutuante

- Também chamada de co-processador matemático
- A ULA não trabalha bem com números fracionário, muito pequenos ou muito grandes
- Noa processadores antigos eram auxiliados por outro processador mais especializado para realização de operações matemáticas complexas.
- A partir do processador 80486, os processadores passaram a incorporar a função de co-processamento matemático
- Os processadores passaram a receber mais um componente interno que foi chamado de Unidade de Ponto Flutuante.

Unidade de Ponto Flutuante

- Também chamada de co-processador matemático
- A ULA não trabalha bem com números fracionário, muito pequenos ou muito grandes
- Noa processadores antigos eram auxiliados por outro processador mais especializado para realização de operações matemáticas complexas.
- A partir do processador 80486, os processadores passaram a incorporar a função de co-processamento matemático
- Os processadores passaram a receber mais um componente interno que foi chamado de Unidade de Ponto Flutuante.

Unidade de Ponto Flutuante

- Também chamada de co-processador matemático
- A ULA não trabalha bem com números fracionário, muito pequenos ou muito grandes
- Noa processadores antigos eram auxiliados por outro processador mais especializado para realização de operações matemáticas complexas.
- A partir do processador 80486, os processadores passaram a incorporar a função de co-processamento matemático
- Os processadores passaram a receber mais um componente interno que foi chamado de Unidade de Ponto Flutuante.

Unidade de Ponto Flutuante

- Também chamada de co-processador matemático
- A ULA não trabalha bem com números fracionário, muito pequenos ou muito grandes
- Noa processadores antigos eram auxiliados por outro processador mais especializado para realização de operações matemáticas complexas.
- A partir do processador 80486, os processadores passaram a incorporar a função de co-processamento matemático
- Os processadores passaram a receber mais um componente interno que foi chamado de Unidade de Ponto Flutuante.

Unidade de Ponto Flutuante

- Também chamada de co-processador matemático
- A ULA não trabalha bem com números fracionário, muito pequenos ou muito grandes
- Nos processadores antigos eram auxiliados por outro processador mais especializado para realização de operações matemáticas complexas.
- A partir do processador 80486, os processadores passaram a incorporar a função de co-processamento matemático
- Os processadores passaram a receber mais um componente interno que foi chamado de Unidade de Ponto Flutuante.

Unidade de Ponto Flutuante

- Em NASM existem instruções específicas para tratar com pontos flutuantes

Unidade de Ponto Flutuante

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Unidade de
Ponto
Flutuante

Exercício

```
%include "io.inc"
section .data
    a: dd 4.0
    b: dd 3.0
    c: dq 0.0
    msg: db "soma = %f",10,0
    msg2: db "raiz = %f",10,0
    msg3: db "multiplicacao = %f",10,0
    msg4: db "Divisao = %f",10,0

section .text
extern printf
global CMAIN
CMAIN:
    mov ebp, esp; for correct debugging
    ;Inicializando a FPU
    finit
    ;Colocando os valores de a e b na pilha
    ;para realizar operacoes de ponto flutuante
    fld dword [a]
    fld dword [b]
    ;somando os dois itens na pilha
    fadd
    ;armazenando a respota em c
    fstp qword [c]

    ;Colocando a parte L do dado em c na pilha
    push dword [c+4]
    ;Colocando a parte H do dado em c na pilha
    push dword [c]
    push dword msg
    call printf
    add esp,12
    xor eax, eax
    ret
```

Unidade de Ponto Flutuante

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Unidade de
Ponto
Flutuante

Exercício

```
%include "io.inc"
section .data
    a: dd 4.0
    b: dd 3.0
    c: dq 0.0
    msg: db "soma = %f",10,0
    msg2: db "raiz = %f",10,0
    msg3: db "multiplicacao = %f",10,0
    msg4: db "Divisao = %f",10,0

section .text
extern printf
global CMAIN
CMAIN:
    finit
    fld dword [a]
    ;Calculando a raiz quadrada de a
    fsqrt
    fstp qword [c]

    push dword [c+4]
    push dword [c]
    push dword msg2
    call printf
    add esp,12
    xor eax, eax
    ret
```

```
%include "io.inc"
section .data
    a: dd 4.0
    b: dd 3.0
    c: dq 0.0
    msg: db "soma = %f",10,0
    msg2: db "raiz = %f",10,0
    msg3: db "multiplicacao = %f",10,0
    msg4: db "Divisao = %f",10,0

section .text
extern printf
global CMAIN
CMAIN:
    finit
    fld dword [a]
    fld dword [b]
    ;realizando a multiplicacao de a com b
    fmul
    fstp qword [c]

    push dword [c+4]
    push dword [c]
    push dword msg3
    call printf
    add esp,12
    xor eax, eax
    ret
```

Unidade de Ponto Flutuante

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Unidade de
Ponto
Flutuante

Exercício

```
%include "io.inc"
section .data
    a: dd 4.0
    b: dd 3.0
    c: dq 0.0
    msg: db "soma = %f",10,0
    msg2: db "raiz = %f",10,0
    msg3: db "multiplicacao = %f",10,0
    msg4: db "Divisao = %f",10,0

section .text
extern printf
global CMAIN
CMAIN:
    finit
    fld dword [a]
    fld dword [b]
    ;realizando a divisao de a por b
    fdiv
    fstp qword [c]

    push dword [c+4]
    push dword [c]
    push dword msg4
    call printf
    add esp,12
    xor eax, eax
    ret
```

Unidade de Ponto Flutuante

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Unidade de
Ponto
Flutuante

Exercício

```
%include "io.inc"
%macro somavetor 3
    mov ebx,0
    mov ecx, %2
    fldz ;st0 <- 0
    while:
        fld dword [%1+4*ebx]
        fadd
        inc ebx
        cmp ebx, ecx
        jl while
    end_while:
    fstp qword [%3]
%endmacro
```

```
section .data
    tmp: dq 0.0
    soma: dd 0.0
    msg: db "soma = %e",10,0
    vetor:      dd      7.36464646465
                dd      0.930984158273
                dd      10.6047098049
                dd      14.3058722306
                dd      15.2983812149
                dd      -17.4394255035
                dd      -17.8120975978
                dd      -12.4885670266
                dd      3.74178604342
                dd      16.3611827165
                dd      -9.1182728262
                dd      -11.4055038727
                dd      4.68148165048
                dd      -9.66095817322
                dd      5.54394454154
                dd      13.4203706426
                dd      18.2194407176
                dd      -7.878340987
                dd      -6.60045833452
                dd      -7.98961850398

    N: equ ($-vetor)/4
```

```
section .text
extern printf
global CMAIN
CMAIN:
    mov ebp, esp; for correct debugging
    somavetor vetor, N, tmp

    push dword [tmp+4]
    push dword [tmp]
    push dword msg
    call printf
    add esp, 12

    xor eax, eax
    ret
```

Unidade de Ponto Flutuante

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Unidade de
Ponto
Flutuante

Exercício

```
%include "io.inc"
%macro somavetor 3
    mov ebx,0
    mov ecx, %2
    fldr %st0 <- 0
    while:
        fld dword [%1+4*ebx]
        fadd
        inc ebx
        cmp ebx, ecx
        jl while
    end_while:
    fstp qword [%3]
%endmacro

section .data
    tmp: dq 0.0
    soma: dd 0.0
    msg: db "soma = %e",10,0
    vetor:
        dd      7.36464646465
        dd      0.930984158273
        dd      10.6047098049
        dd      14.3058722306
        dd      15.2983812149
        dd      -17.4394255035
        dd      -17.8120975978
        dd      -12.4885670266
        dd      3.74178604342
        dd      16.3611827165
        dd      -9.1182728262
        dd      -11.4055038727
        dd      4.68148165048
        dd      -9.66095817322
        dd      5.54394454154
        dd      13.4203706426
        dd      18.2194407176
        dd      -7.878340987
        dd      -6.60045833452
        dd      -7.98961850398

    N: equ (5-vetor)/4

section .text
extern printf
global MAIN
MAIN:
    mov ebp, esp; for correct debugging
    somavetor vetor, N, tmp

    push dword [tmp+4]
    push dword [tmp]
    push dword msg
    call printf
    add esp,12

    xor eax, eax
    ret
```


- Escreva uma sub-rotina para o cálculo da potência de dois números, sendo que a base deve ser do tipo ponto flutuante e a potência um inteiro.
- Use como base o seguinte código:

```
POTENCIA:
    mov ecx, [valor]
    mov eax, [valor]
    mov ebx, 1
    while:
        cmp ebx, [potc]
        je end_while
        mul ecx
        inc ebx
        jmp while
    end_while:

    mov [resultado], eax
    ret
```

Prog. Apl. CC

Dr. E. S.
Pereira

Sumário

Unidade de
Ponto
Flutuante

Exercício

Grato

MUITO OBRIGADO.

Referências

- <https://www.nasm.us/doc/nasmdoc4.html>
- https://en.wikibooks.org/wiki/X86_Assembly/Floating_Point
- http://www.science.smith.edu/dftwiki/index.php/CSC231_Floating-Point_Assembly_Examples