Teoria da Computação

UNESP - Univerisade Estadual Paulista - Presidente Prudente

SEM1 2020

Celso Olivete Júnior

Trab. Aula 08: Computador Digital (Von Neumann)

Gilmar Francisco de Oliveira Santos

1 Máquina de Turing Multifita para a Arquitetura

A arquitetura construída consta no arquivo em anexo "MT_aula_08.jff"

Ou no link: MT_aula_08.jff.

2 Casos de Teste

Caso 1: Programa que carrega o valor “111”, soma a “1” e salva no endereço “1000”.

Entrada:

```
#1000*1000#1000*111#1001*1#0*1011000#1*100#10*1011001#11*000#100*1101000#
```

```
1  Carrega a entrada na fita 1
2  Inicia a fita 2 com 0
3  Busca na memória(Fita 1) o endereço 0
4  Lê a operação de LOAD do endereço 1000 no reg X(fita 3)
5  Busca na memória o endereço 1000
6  Copia o conteúdo do endereço 1000 no reg X(fita 3).
7  Incrementa PC para 1
8  Lê o conteúdo de PC, busca na memória a instrução no endereço 1
9  Lê a operação SWAP(100)
10 Realiza a troca do conteúdo de X com Y, usando a fita 5 como mediadora
11 Incrementa PC para 10
12 Lê o conteúdo de PC, busca na memória a instrução no endereço 10
13 Lê a operação de LOAD do endereço 1001 no reg X(fita 3)
14 Busca na memória o endereço 1001
15 Copia o conteúdo do endereço 1001 no reg X(fita 3).
16 Incrementa PC para 11
17 Lê o conteúdo de PC, busca na memória a instrução no endereço 11
18 Lê a operação de ADD regs X e Y
19 Copia X e Y para a fita 5
20 Realiza a soma e copia o resultado para X
21 Incrementa PC para 100
22 Lê o conteúdo de PC, busca na memória a instrução no endereço 100
23 Lê a operação de STORE X no endereço 1000
24 Escreve no começo da fita 1 a posição de memória 1000 com o valor 1000
25 Incrementa PC para 101
26 Lê o conteúdo de PC, busca na memória a instrução no endereço 101
```

```
27      Não encontra -> Halt
28      Finaliza execução
```

Resultado:

```
#1000*1000#1000*111#1001*1#0*1011000#1*100#10*1011001#11*000#100*1101000#
```

Caso 2: Programa que faz o And lógico entre os mesmos valores do caso 1 e salva no endereço "1000".

Entrada:

```
#1000*111#1001*1#0*1011000#1*100#10*1011001#11*010#100*1101000#
```

```
1   Carrega a entrada na fita 1
2   Inicia a fita 2 com 0
3   Busca na memória(Fita 1) o endereço 0
4   Lê a operação de LOAD do endereço 1000 no reg X(fita 3)
5   Busca na memória o endereço 1000
6   Copia o conteúdo do endereço 1000 no reg X(fita 3)
7   Incrementa PC para 1
8   Lê o conteúdo de PC, busca na memória a instrução no endereço 1
9   Lê a operação SWAP(100)
10  Realiza a troca do conteúdo de X com Y, usando a fita 5 como mediadora
11  Incrementa PC para 10
12  Lê o conteúdo de PC, busca na memória a instrução no endereço 10
13  Lê a operação LOAD(101) do endereço 1001 no reg X(fita 3)
14  Busca na memória o endereço 1001
15  Copia o conteúdo do endereço 1001 no reg X(fita 3)
16  Incrementa PC para 11
17  Lê o conteúdo de PC, busca na memória a instrução no endereço 11
18  Lê a operação AND(010)
19  Realiza o AND lógico entre X e Y, aplicando a tabela verdade
20  Armazena o resultado em X
21  Incremente PC para 100
22  Lê o conteúdo de PC, busca na memória a instrução no endereço 100
23  Lê a operação STORE(110) do conteúdo do reg X para o endereço 1000
24  Copia o conteúdo de X para o início da memória, salvo com o endereço 1000
25  Incrementa PC para 101
26  Busca a instrução no endereço 101, não encontra -> Halt
27  Finaliza execução
```

Resultado:

#1000*001#1000*111#1001*1#0*1011000#1*100#10*1011001#11*010#100*1101000#

Caso 3: Programa que faz o Not no valor “1101” e o salva no endereço “1000”.

Entrada:

```
#1000*1101#0*1011000#1*011#10*1101000#
```

```
1  Carrega a entrada na fita 1
2  Inicia a fita 2 com 0
3  Busca na memória(Fita 1) o endereço 0
4  Lê a operação de LOAD do endereço 1000 no reg X(fita 3)
5  Busca na memória o endereço 1000
6  Copia o conteúdo do endereço 1000 no reg X(fita 3).
7  Incrementa PC para 1
8  Lê o conteúdo de PC, busca na memória a instrução no endereço 1
9  Lê a operação NOT
10 Inverte os bits do reg X
11 Incremente PC para 10
12 Lê o conteúdo de PC, busca na memória a instrução no endereço 10
13 Lê a operação 110(STORE) do conteúdo do reg X para o endereço 1000
14 Copia o conteúdo de X para o início da memória, salvo com o endereço 1000
15 Incrementa PC para 11
16 Busca a instrução no endereço 11, não encontra -> Halt
17 Finaliza execução
```

Resultado:

```
#1000*0010#1000*1101#0*1011000#1*011#10*1101000#
```

3 Solução Adotada

Ciclo de Funcionamento

A execução é inicializada pelo carregamento da entrada na Fita 1, que será a memória de trabalho, então o valor 0 é pré-carregado na Fita 2, a qual será representado o PC, inicia-se então o ciclo de busca e execução de instruções, cada instrução encontrada, representada pelos 3 primeiros bits do conteúdo do endereço alvo, é executada de acordo com o que se segue abaixo, para então ocorrer uma operação de incremento do PC, usando da própria unidade de Adição, com exceção da operação de JUMP, que carrega diretamente o valor no PC, e o ciclo de busca é reiniciado.

Gerenciamento de Memória

A busca é realizada na memória, Fita 1, de maneira linear, estando os endereços presentes na memória disposto de maneira aleatória, para a busca de endereços de intruções o valor utilizado é o do PC, enquanto que para endereços de variáveis, o endereço é alocado temporariamente na fita 5, para então a busca linear ser iniciada.

As escritas na memória são feitas no início, dadas as complexidades envolvidas ao sobrescrever posições de memória no meio, com isso a busca de endereços sempre irá utilizar o primeiro endereço encontrado, começando sempre da direita, de modo a garantir que novos valores para as posições de memória sejam possíveis.

Endereços não Encontrados fazem a MT travar, porém caso seja um endereço fornecido pelo PC e não for encontrado, a máquina finaliza a execução, vai para o estado de Halt.

3.1 Operações

STORE

Rebobina a memória, ou seja, volta para o início, e escreve da esquerda para direita, primeiro o conteúdo do registrador X, adiciona o carácter de separação "*", escreve então o endereço alvo e adiciona o carácter "#", finalizando a adição ou sobrescrita da nova posição de memória.

Então chama a operação de Incrementar o PC e busca a próxima instrução.

LOAD

Busca na memória o endereço alvo, ao encontrar, copia seu conteúdo diretamente para o registrador X.

Então chama a operação de Incrementar o PC e busca a próxima instrução.

JUMP

A operação de JUMP copia o endereço presente na memória diretamente para a Fita 2(PC) e retorna para o estado de Busca de Instrução.

SWAP

A operação de SWAP troca os conteúdos dos registradores X e Y, usando a fita 5 como intermediária, primeiramente copia o conteúdo de X para a fita 5, depois copia o valor de Y para X, para então copiar o valor da fita 5 para Y.

Então chama a operação de Incrementar o PC e busca a próxima instrução.

NOT

A operação lógica NOT é realizada diretamente no registrador X, a MT itera sobre todos os bits e inverte seus valores diretamente na própria fita.

Então chama a operação de Incrementar o PC e busca a próxima instrução.

AND

A operação de AND lógico é realizada da esquerda para direita, aplicando a tabela verdade do operador lógico AND simultâneamente aos registradores X e Y, quando um **branco** é encontrado em um dos registradores, zero é escrito na posição correspondente.

Então chama a operação de Incrementar o PC e busca a próxima instrução.

ADD

Para a operação de AND, a Fita 5 é majoritariamente utilizada, para isso é feito a cópia dos valores contidos no registrador X e Y para a Fita 5, onde a operação a ser realizada é no seguinte formato:

```
#b_x#b_y#d
```

Onde b_x e b_y representam respectivamente os valores copiados dos registradores X e Y em binário, e d representa um código para a escolha de qual será o destino do resultado, sendo os valores possíveis: P(usado para somas com o valor de PC, ou seja, usado no incremento das instruções), F(resultado final de uma operação de soma normal ou subtração, vai ser copiado no Reg. x), S(resultado parcial de uma subtração, o carry é excluído e o valor retroalimentado somando 1).

Então chama a operação de Incrementar o PC e busca a próxima instrução.

SUB

A operação de subtração é feita reutilizando-se o a estrutura para Adição, porém a estratégia é utilizar a **soma binária com complemento de 1**, sendo assim o valor do Reg. Y é copiado para a Fita 5 com seus valores de Bits invertidos e então somado com o valor do Reg. X, o resultado é intermediário, então descarta-se o primeiro bit(carry) do resultado, dado que só trabalhamos com números positivos, esse resultado é então somado com 1, para só então seu resultado ser salvo no Reg. X. Caso os valores de X e Y sejam iguais, a MT escreve diretamente zero no Reg X.

Então chama a operação de Incrementar o PC e busca a próxima instrução.