UNIVERSIDAE ESTADUAL DO PARANÁ - *CAMPUS* APUCARANA

Carlos Henrique Sonego Do Prado

Relatório Técnico – Ciclos de Instrução

APUCARANA – PR
2024

Carlos Henrique Sonego do Prado

Relatório Técnico – Ciclos de Instrução

Trabalho apresentado a disciplina de Arquitetura e Organização de Computadores, do curso de Bacharelado em Ciência da Computação.

Professor: Guilherme Henrique de Souza Nakahata

APUCARANA – PR
2024

SUMÁRIO

INTRODUÇÃO

O trabalho tem o como conteúdo principal a criação e implementação de um código-fonte que realize siclos de instrução. O código foi desenvolvido em linguagem Java. A ideia era utilizar como base uma Máquina de von Neumann na logica realizar os ciclos que essa maquina realiza. Em síntese é isso que é para ser capaz de realizar nosso código.

CAPÍTULO 1
OBJETIVOS

O objetivo principal do código criada pode ser interpretado como recriar os ciclos com pelo menos algum tipo de fidelidade ao original, neste código o usuário irá falar o código e seus operando que de acordo com suas logicas vai manipular algo dentro da estrutura da Máquina de von Neumann. De modo geral, o código fonte vai estar lendo os dados e executando eles utilizando a lógica previamente dita. Está representado na figura 1 a tabela de códigos e na imagem 2 um exemplo de teste.

Código da Instrução	Operandos	Resultado
000001	#pos	MBR ← #pos
000010	#pos #dado	#pos ← #dado
000011	#pos	MBR ← MBR + #pos
000100	#pos	MBR ← MBR - #pos
000101	#pos	MBR ← MBR * #pos
000110	#pos	MBR ← MBR / #pos
000111	#lin	JUMP to #lin
001000	#lin	JUMP IF Z to #lin
001001	#lin	JUMP IF N to #lin
001010	-	MBR ← raiz_quadrada(MBR)
001011	-	MBR ← - MBR
001111	#pos	#pos ← MBR
001100	-	NOP

Figura1

Posição	Opcode	Operando1	Operando2	Operação
1	000010	251	5	Armazena 5 na posição 251
2	000010	252	10	Armazena 10 na posição 252
3	000010	253	15	Armazena 15 na posição 253
4	000001	251		MBR recebe o conteúdo da posição 251
5	000011	252		MBR recebe o conteúdo dele mesmo somado com o conteúdo da posição 252
6	000011	253		MBR recebe o conteúdo dele mesmo somado com o conteúdo da posição 253
7	001111	254		Posição 254 recebe o conteúdo de MBR
8	001100			Fim, no operation

Figura 2

CAPÍTULO 2
MOTIVAÇÃO E RECUROS UTILIZADOS

Baseando-se no exposto anteriormente, devemos explicitar os motivos para a realização do trabalho, ou seja, o objetivo final e os recursos utilizados para que isso seja comprido.

2.1 Motivação

Como citado no capitulo que trata acerca dos objetivos do projeto em questão, a motivação também seria a realização de um código fonte funcional em que possamos demonstrar o pleno funcionamento de uma máquina de ciclos.

2.2 Estrutura de dados

Para tal projeto, foi preciso utilizar para armazenamento dos dados int[] memoria com tamanho 250, além disso String[] instruções que guarda todas as instruções digitadas pelo usuário, também foi criado uma variável para o PC,MBR,IR e MAR, além de uma variável para quando o pulo for 0 e quando for negativo.
Foi criado métodos onde cada um deles realizava algo explicito no menu. Imagem 3 com o menu.

```
System.out.println("------------------");
System.out.println("      OPCOES:     ");
System.out.println("------------------");
System.out.println("1 - INSERIR       ");
System.out.println("2 - VER INSTRUCOES");
System.out.println("3 - VER MEMORIA   ");
System.out.println("4 - EXECUTAR      ");
System.out.println("5 - LIMPAR        ");
System.out.println("6 - SAIR          ");
```

Imagem 3

O primeiro dos métodos foi uma para inserção das instruções onde nesse método pedia o código o operando 1 e operando 2. Detalhado na imagem 4;

```java
public static void inserirInstrucao(Scanner scanner) {
    if (numInstrucoes >= tamanho_maximo_de_instrucoes) {
        System.out.println(x:"Numero maximo de instrucoes atingido.");
        return;
    }

    String codigo;
    do {
        System.out.print(s:"Digite o codigo da instrucao (001100 para terminar as instrucoes): ");
        codigo = scanner.nextLine();
        if (!codigo.equals(anObject:"001100")) {
            System.out.print(s:"Digite o operando 1: ");
            String operando1 = scanner.nextLine();
            System.out.print(s:"Digite o operando 2: ");
            String operando2 = scanner.nextLine();
            String instrucao = codigo + " " + operando1 + " " + operando2;
            instrucoes[numInstrucoes] = instrucao;
            numInstrucoes++;
        }
    } while (!codigo.equals(anObject:"001100"));
}
```

Imagem 4

Em segundo foi criado um método para poder ver a tabela com os códigos e suas instruções. Detalhado na imagem 5.

```java
public static void verInstrucoes() {
    System.out.println("|Codigo|Operando 1|Operando 2|Resultado                |");
    System.out.println("|000001|#pos      |          |MBR <- #pos              |");
    System.out.println("|000010|#pos      |#dado     |#pos <- #dado            |");
    System.out.println("|000011|#pos      |          |MBR <- MBR + #pos        |");
    System.out.println("|000100|#pos      |          |MBR <- MBR - #pos        |");
    System.out.println("|000101|#pos      |          |MBR <- MBR * #pos        |");
    System.out.println("|000110|#pos      |          |MBR <- MBR / #pos        |");
    System.out.println("|000111|#lin      |          |JUMP to #lin             |");
    System.out.println("|001000|#lin      |          |JUMP IF Z to #lin        |");
    System.out.println("|001001|#lin      |          |JUMP IF N to #lin        |");
    System.out.println("|001010|          |          |MBR <- raiz quadratica(MBR)|");
    System.out.println("|001011|          |          |MBR <- -Mbr              |");
    System.out.println("|001111|#pos      |          |#pos <- MBR              |");
    System.out.println("|001100|          |          |NOP                      |");
    System.out.println();
}
```

Imagem 5

Em terceiro foi criado um metodo para poder executar os codigos nela estava presente a logica por traz de cada codigo e mostrava a manipulação do IR ,MBR ,MAR e PC com o decorer de cada instrução sendo lida. Foi criada um metodo para mostar a memoria depois da execução dos codigo e outro metodo para limpar tudo para o caso de estiver lotado a memoria .

2.3 Linguagem de Programação e demais informações

A linguagem escolhida para a implementação do código foi Java, de maneira que se utiliza-se a linguagem que a maioria da sala aprendeu no segundo ano de formação para implementar as instruções a ser seguida pela máquina. A única biblioteca diretamente utilizada foi a scanner.

CAPÍTULO 3
RESULTADOS

Mediante os objetivos apresentados, o resultado esperado seria o pleno funcionamento de um codigo que exemplifique o funcionamento de uma Máquina de von Neumann ou ciclo de maquina. Dessarte, com a implementação completa e revisada, os resultados foram meio que atingidos,houve pontos de logica onde não estam totalmente apurada devido a eu ter pego para fazer o projeto de ultima hora resaltando que deveria ter sido começado antes. Abaixo nas imagens 6,7,8,9 , podemos ver o funcionamento de alguns dos codigos propostos.

Figura 6

```
Ciclo de Instrucao 3:
PC = 2
MAR = 2
MBR = 10
IR = 000010
Executando: memoria[3] <- 15
Estado após execução:
PC = 2
MAR = 3
MBR = 15
IR = 000010

Ciclo de Instrucao 4:
PC = 3
MAR = 3
MBR = 15
IR = 000010
Executando: MBR <- memoria[1] = 5
Estado após execução:
PC = 3
MAR = 1
MBR = 5
IR = 000001

Ciclo de Instrucao 5:
PC = 4
MAR = 1
MBR = 5
IR = 000001
Executando: MBR <- MBR + memoria[2] = 15
Estado após execução:
PC = 4
MAR = 2
MBR = 15
IR = 000011
```

Figura 7

```
Ciclo de Instrucao 6:
PC = 5
MAR = 2
MBR = 15
IR = 000011
Executando: MBR <- MBR + memoria[3] = 30
Estado após execução:
PC = 5
MAR = 3
MBR = 30
IR = 000011

Ciclo de Instrucao 7:
PC = 6
MAR = 3
MBR = 30
IR = 000011
Executando: memoria[4] <- MBR = 30
Estado após execução:
PC = 6
MAR = 4
MBR = 30
IR = 001111
```

Figura 8

```
Endereco 0: 0
Endereco 1: 5
Endereco 2: 10
Endereco 3: 15
Endereco 4: 30
Endereco 5: 0
Endereco 6: 0
Endereco 7: 0
Endereco 8: 0
Endereco 9: 0
Endereco 10: 0
```

Figura 9

CONCLUSÃO

Tendo em vista os objetivos estabelecidos para o trabalho, pode-se dizer que os objetivos foram alcançados pela metade, já que nem todos os códigos estão em seu pleno funcionamento além de que a interface esta de certa forma confusa para o entendimento pleno da manipulação do MBR,IR,MAR,PC.

REFERÊNCIAS

NAKAHATA, Guilherme. Arquitetura e Organização de Computadores. 2º bimestre. Disponível em: https://github.com/GuilhermeNakahata/UNESPAR-2024/blob/main/Arquitetura%20e%20Organizacao%20de%20Computadores/2%C2%B0%20Bimestre/Aulas/Aulas_ArquiteturaComputadores_11_06_2024.pdf. Acesso em: 27 jul. 2024.