**TAREFA DE MICRO ARQUITETURA.**

**Nome: Guilherme Augusto Henckel, Gabriel Krummenauer de Morais, Allan Kevin Scain.**

**1.** Codifique as seguintes micro-instruções para a micro-arquitetura Mic1, mostrada na figura abaixo, explicando como você realizou cada passo da codificação e determinou o valor de cada campo da micro-instrução:

0. mar:= pc; rd;

1. pc:= pc + 1; rd;

2. ir:= mbr; if n then goto 28;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Amux** | **Cond** | **ALU** | **Desl** | **mar** | **mbr** | **rd** | **wr** | **valc** | **C** | **B** | **A** | **Ender** |
| **0** | 0 | 0 | 2 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 5 | 0 | 0 |
| **2** | 1 | 1 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 28 |

0. mar:= pc; rd;

**0 {**

**AMUX = 0 >** Para que o operando A da ULA seja um dos 16 registradores;

**COND = 0 >** Não existe sequenciamento;

**ALU = 2 >** Para que o PC passe direto pela ULA;

**DESL = 0 >** Para não ocorrer deslocamento;

**MAR = 1 >** Para que o PC seja escrito no registrador MAR;

**MBR = 0 >** Não carrega o MBR;

**RD = 1 >** Para que seja feita uma leitura na RAM;

**WR = 0 >** Não haverá nenhuma escrita;

**VALC = 0 >** Para que não entre no barramento C, para ser escrito num registrador;

**C = 0 >** Não é usado, pois VALC = 0;

**B = 0 >** Para o PC entrar no barramento B;

**A = 0 >**Não é usado;

**ENDER = 0 >**Não é usado, pois COND = 0;

**}**

1. pc:= pc + 1; rd;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Amux** | **Cond** | **ALU** | **Desl** | **mar** | **mbr** | **rd** | **wr** | **valc** | **C** | **B** | **A** | **Ender** |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 5 | 0 | 0 |

**1 {**

**AMUX = 0 >** Para que o operando A da ULA seja um dos 16 registradores;

**COND = 0 >** Não existe sequenciamento;

**ALU = 0 >** Para que execute a soma PC + (+1);

**DESL = 0 >** Para não ocorrer deslocamento;

**MAR = 0 >** Para que não seja escrito no MAR;

**MBR = 0 >** Não carrega o MBR;

**RD = 1 >** Para que seja feita uma leitura na RAM;

**WR = 0 >** Para que seja uma leitura de memória.

**VALC = 1 >** Para que o PC seja escrito em PC;

**C = 0 >** Não é usado, pois VALC = 0;

**B = 5 >** Para que o +1 seja o operando B da ULA;

**A = 0 >**Para o PC seja o operando A da ULA;

**ENDER = 0 >**Não é usado, pois COND = 0;

**}**

2. ir:= mbr; if n then goto 28;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Amux** | **Cond** | **ALU** | **Desl** | **mar** | **mbr** | **rd** | **wr** | **valc** | **C** | **B** | **A** | **Ender** |
| **2** | 1 | 1 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 28 |

**2{**

**AMUX = 1 >** Para entrar valor do MBR;

**COND = 1 >** Indica sequenciamento se N for “verdade”;

**ALU = 2 >** Para que o MBR passe direto pela ULA;

**DESL = 0 >** Para não ocorrer deslocamento;

**MAR = 0 >** Para que nada seja escrito no MAR;

**MBR = 1 >** Para que carregue endereço de MBR;

**RD = 0 >** Para que não seja feita uma leitura na RAM;

**WR = 0 >** Para que não seja feita uma escrita na RAM;

**VALC = 0 >** Para que não entre no barramento C, para ser escrito num registrador;

**C = 0 >** Não é usado, pois VALC = 0;

**B = 0 >** Não é usado;

**A = 0 >**Não é usado;

**ENDER = 28 >**Que indica a próxima instrução a ser executada após COND;

**}**

B)

3: tir :=lshift(ir + ir); if n then goto 19;

4: tir :=lshift(tir); if n then goto 11;

5: alu := tir; if n then goto 9;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Amux** | **Cond** | **ALU** | **Desl** | **mar** | **mbr** | **rd** | **wr** | **valc** | **C** | **B** | **A** | **Ender** |
| **3** | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 4 | 3 | 3 | 19 |
| **4** | 0 | 1 | 2 | 2 | 0 | 0 | 0 | 0 | 1 | 4 | 0 | 4 | 11 |
| **5** | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 9 |

3: tir:=lshift(ir + ir); if n then goto 19;

**3{**

**AMUX = 0 >** Para que o operando A da ULA seja um dos 16 registradores;

**COND = 1 >** Indica sequenciamento se N for “verdade”;

**ALU = 0 >** zero para executar a soma de IR + IR;

**DESL = 2 >** 2 para deslocar para a esquerda.

**MAR = 0 >** Para não ser escrito em MAR;

**MBR = 0 >** Não carrega o MBR;

**RD = 0 >** Processo não é de leitura.

**WR = 0 >** Processo não é de escrita.

**VALC = 1 >** Para que TIR receba a soma da ULA pelo barramento C;

**C = 4 >** Para que o TIR (registrador 4) seja o registrador de destino;

**B = 3 >** Para que o IR (registrador 3) entre na ULA;

**A = 3 >**Para que o IR (registrador 3) entre na ULA;

**ENDER = 19 >**Que indica a próxima instrução a ser executada após COND;

**}**

4: tir:=lshift(tir); if n then goto 11;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Amux** | **Cond** | **ALU** | **Desl** | **mar** | **mbr** | **rd** | **wr** | **valc** | **C** | **B** | **A** | **Ender** |
| **4** | 0 | 1 | 2 | 2 | 0 | 0 | 0 | 0 | 1 | 4 | 0 | 4 | 11 |

**4 {**

**AMUX = 0 >** Para que o operando A da ULA seja um dos 16 registradores;

**COND = 1 >** Indica sequenciamento se N for “verdade”;

**ALU = 2 >** Para que o TIR passe direto pela ULA;

**DESL = 2 >** Para ocorrer um deslocamento à esquerda;

**MAR = 0 >** Para que nada seja escrito no MAR;

**MBR = 0 >** Não carrega o MBR;

**RD = 0 >** Para que não seja feita uma leitura na RAM;

**WR = 0 >** Para que não seja feita uma escrita na RAM;

**VALC = 1 >** Para que TIR seja escrito em TIR;

**C = 4 >** Para que o TIR (registrador 4) seja o registrador de destino;

**B = 0 >** Não é usado;

**A = 0 >**Para que o TIR (registrador 4) entre na ULA;

**ENDER = 11 >**Que indica a próxima instrução a ser executada após COND;

**}**

5: alu:= tir; if n then goto 9;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Amux** | **Cond** | **ALU** | **Desl** | **mar** | **mbr** | **rd** | **wr** | **valc** | **C** | **B** | **A** | **Ender** |
| **5** | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 9 |

**5{**

**AMUX = 0 >** Para que o operando A da ULA seja um dos 16 registradores;

**COND = 1 >** Indica sequenciamento se N for “verdade”;

**ALU = 2 >** Para que o TIR passe direto pela ULA;

**DESL = 0 >** Para não ocorrer deslocamento;

**MAR = 0 >** Para que nada seja escrito no MAR;

**MBR = 0 >** Não carrega o MBR;

**RD = 0 >** Para que não seja feita uma leitura na RAM;

**WR = 0 >** Para que não seja feita uma escrita na RAM;

**VALC = 0 >** Para que não entre no barramento C, para ser escrito num registrador;

**C = 0 >** Não é usado, pois VALC = 0;

**B = 0 >** Não é usado;

**A = 4 >**Para que o TIR (registrador 4) entre na ULA;

**ENDER = 9>**Que indica a próxima instrução a ser executada após COND;

**}**

C)

6: mar:= ir; rd / 0000 = LODD

7: rd;

8: ac:= mbr; goto 0;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Amux** | **Cond** | ALU | Desl | mar | mbr | rd | wr | valc | C | B | A | Ender |
| 6 | 0 | 0 | 2 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 3 | 0 |
| 7 | 0 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 1 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

6: mar:= ir; rd / 0000 = LODD

**6 {**

**AMUX = 0 >**Para que o operando A da ULA seja um dos 16 registradores;

**COND = 0>**Não existe sequenciamento;

**ALU = 2 >**Para que o IR passe direto pela ULA;

**DESL = 0 >**Para não ocorrer deslocamento;

**MAR = 1 >**Para que o IR seja escrito no registrador MAR;

**MBR = 0 >**Não carrega o MBR;

**RD = 1 >**Para que seja feita uma leitura na RAM;

**WR = 0 >**Para que não seja feita uma escrita na RAM;

**VALC = 0 >**Para que não entre no barramento C, para ser escrito num registrador;

**C = 0 >** Não é usado, pois VALC = 0;

**B = 0 >** Não é usado.

**A = 3 >**Para que o IR (registrador 3) entre na ULA;

**ENDER = 0 >** Não é usado, pois COND = 0;

**}**

7: rd;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Amux** | **Cond** | **ALU** | **Desl** | **mar** | **mbr** | **rd** | **wr** | **valc** | **C** | **B** | **A** | **Ender** |
| 7 | 0 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

**7 {**

**AMUX = 0 >**Para que o operando A da ULA seja um dos 16 registradores;

**COND = 0>**Não existe sequenciamento;

**ALU = 2 >**Para que o ENDEREÇO passe direto pela ULA;

**DESL = 0 >**Para não ocorrer deslocamento;

**MAR = 0 >**Não carrega o MBR;

**MBR = 0 >**Para que nada seja escrito no MBR;

**RD = 1 >**Para que seja feita uma leitura na RAM;

**WR = 0 >**Para que não seja feita uma escrita na RAM;

**VALC = 0 >**Para que não entre no barramento C, para ser escrito num registrador;

**C = 0 >** Não é usado, pois VALC = 0;

**B = 0 >**Não é usado.

**A = 0 >**Não é usado.

**ENDER = 0 >** Não é usado, pois COND = 0;

**}**

8: ac:= mbr; goto 0;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Amux** | **Cond** | **ALU** | **Desl** | **mar** | **mbr** | **rd** | **wr** | **valc** | **C** | **B** | **A** | **Ender** |
| 8 | 1 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

**8 {**

**AMUX = 1 >**Para entrar valor do MBR;

**COND = 0>**Não existe sequenciamento;

**ALU = 2 >**Para que o MBR passe direto pela ULA;

**DESL = 0 >**Para não ocorrer deslocamento;

**MAR = 0 >**Para que nada seja escrito no MAR;

**MBR = 1 >**Para que carregue endereço de MBR;

**RD = 0 >**Para que não seja feita uma leitura na RAM;

**WR = 0 >**Para que não seja feita uma escrita na RAM;

**VALC = 1 >**Para MBR ser escrito em AC;

**C = 1 >** Para que o AC (registrador 4) seja o registrador de destino;

**B = 0 >**Não é usado.

**A = 0 >**Não é usado.

**ENDER = 0 >** Não é usado, pois COND = 0;

**}**

D)

9: mar:= ir; mbr:= ac: wr

10: wr; goto 0;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Amux** | **Cond** | **ALU** | **Desl** | **mar** | **mbr** | **rd** | **wr** | **valc** | **C** | **B** | **A** | **Ender** |
| 9 | 1 | 0 | 2 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 3 | 0 | 0 |
| 10 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

**9 {**

**AMUX = 1 >**Para entrar valor do MBR;

**COND = 0>**Não existe sequenciamento;

**ALU = 2 >**Para que o IR e o AC passem direto pela ULA;

**DESL = 0 >**Para não ocorrer deslocamento;

**MAR = 1 >**Para que o IR seja escrito em MAR;

**MBR = 1 >**Para que carregue endereço de MBR;

**RD = 0 >**Para que não seja feita uma leitura na RAM;

**WR = 1 >**Para que seja feita uma escrita na RAM;

**VALC = 1 >**Para que seja feita uma escrita pelo barramento C de IR em MAR;

**C = 1 >** Para que o AC (registrador 4) seja o registrador de destino;

**B = 3 >**Para o IR entrar no barramento B;

**A = 0 >**Não é usado.

**ENDER = 0 >** Não é usado, pois COND = 0;

**}**

10: wr; goto 0;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Amux** | **Cond** | **ALU** | **Desl** | **mar** | **mbr** | **rd** | **wr** | **valc** | **C** | **B** | **A** | **Ender** |
| 10 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

**10 {**

**AMUX = 0 >**Para entrar latch, não MBR;

**COND = 0>**Não existe sequenciamento;

**ALU = 2 >**Para que o ENDEREÇO passe direto pela ULA;

**DESL = 0 >**Para não ocorrer deslocamento;

**MAR = 0 >**Para que nada seja escrito em MAR;

**MBR = 0 >**Não carrega o MBR;

**RD = 0 >**Para que não seja feita uma leitura na RAM;

**WR = 1 >**Para que seja feita uma escrita na RAM;

**VALC = 0 >**Para que não entre no barramento C, para ser escrito num registrador;

**C = 0 >** Não é usado, pois VALC = 0;

**B = 0 >**Não é usado;

**A = 0 >**Não é usado.

**ENDER = 0 >** Não é usado, pois COND = 0;

**}**

E)

11: alu:= tir; if n then goto 15

12: mar:= ir; rd

13: rd;

14: ac:= mbr + ac; goto 0;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Amux** | **Cond** | **ALU** | **Desl** | **mar** | **mbr** | **rd** | **wr** | **valc** | **C** | **B** | **A** | **Ender** |
| 11 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 15 |
| 12 | 0 | 0 | 2 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 3 | 0 | 0 |
| 13 | 0 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |

11: alu:= tir; if n then goto 15

**11 {**

**AMUX = 0 >**Para entrar latch, não MBR;

**COND = 1>**Quebra de sequenciamento se N for “verdade”;

**ALU = 2 >**Para que o TIR passe direto pela ULA;

**DESL = 0 >**Para não ocorrer deslocamento;

**MAR = 0 >**Para que nada seja escrito em MAR;

**MBR = 0 >**Não carrega o MBR;

**RD = 0 >**Para que não seja feita uma leitura na RAM;

**WR = 0 >**Para que não seja feita uma escrita na RAM;

**VALC = 0 >**Para que não entre no barramento C, para ser escrito num registrador;

**C = 0 >** Não é usado, pois VALC = 0;

**B = 0 >**Não é usado;

**A = 4 >**Para o TIR entrar no barramento A;

**ENDER = 15 >** Indica a próxima instrução a ser executada após COND;

**}**

12: mar:= ir; rd

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Amux** | **Cond** | **ALU** | **Desl** | **mar** | **mbr** | **rd** | **wr** | **valc** | **C** | **B** | **A** | **Ender** |
| 12 | 0 | 0 | 2 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 3 | 0 | 0 |

**12 {**

**AMUX = 0 >**Para entrar latch, não MBR;

**COND = 0>**Não existe sequenciamento;

**ALU = 2 >**Para que o IR passe direto pela ULA;

**DESL = 0 >**Para não ocorrer deslocamento;

**MAR = 1 >**Para que o IR seja escrito em MAR;

**MBR = 0 >**Não carrega o MBR;

**RD = 1 >**Para que seja feita uma leitura na RAM;

**WR = 0 >**Para que não seja feita uma escrita na RAM;

**VALC = 0 >**Para que não entre no barramento C, para ser escrito num registrador;

**C = 0 >** Não é usado, pois VALC = 0;

**B = 3 >**Para o IR entrar no barramento B;

**A = 0 >**Não é usado.

**ENDER = 0 >** Não é usado, pois COND = 0;

**}**

13: rd;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Amux** | **Cond** | **ALU** | **Desl** | **mar** | **mbr** | **rd** | **wr** | **valc** | **C** | **B** | **A** | **Ender** |
| 13 | 0 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

**13 {**

**AMUX = 0 >**Para entrar latch, não MBR;

**COND = 0>**Sem quebra de deslocamento;

**ALU = 2 >**Para que o ENDEREÇO passe direto pela ULA;

**DESL = 0 >**Para não ocorrer deslocamento;

**MAR = 0 >**Para que nada seja escrito em MAR;

**MBR = 0 >**Não carrega o MBR;

**RD = 1 >**Para que seja feita uma leitura na RAM;

**WR = 0 >**Para que não seja feita uma escrita na RAM;

**VALC = 0 >**Para que não entre no barramento C, para ser escrito num registrador;

**C = 0 >** Não é usado, pois VALC = 0;

**B = 0 >**Não é usado;

**A = 0 >**Não é usado;

**ENDER = 0 >** Não é usado, pois COND = 0;

**}**

14: ac:= mbr + ac; goto 0;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Amux** | **Cond** | **ALU** | **Desl** | **mar** | **mbr** | **rd** | **wr** | **valc** | **C** | **B** | **A** | **Ender** |
| 14 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |

**14 {**

**AMUX = 1 >**Indica que vai entrar valor do MBR.

**COND = 0>**Sem quebra de deslocamento;

**ALU = 0 >**Para a ULA fazer a soma de MBR + AC;

**DESL = 0 >**Para não ocorrer deslocamento;

**MAR = 0 >**Para que nada seja escrito em MAR;

**MBR = 1 >**Para que carregue o ENDERÇO de MBR;

**RD = 0 >**Para que não seja feita uma leitura na RAM;

**WR = 0 >**Para que não seja feita uma escrita na RAM;

**VALC = 1 >**Permite que seja feita uma escrita pelo barramento C, da soma da ULA em AC;

**C = 1 >** Leva o resultado da soma para AC;

**B = 1 >**Para o AC entrar no barramento B;

**A = 0 >**Não é usado.

**ENDER = 0 >** Não é usado, pois COND = 0;

**}**