

6. Explique detalhadamente, usando as tabelas 1 e 2 e o diagrama de blocos do processador na figura 1, como a instrução **beq \$s0, \$t1, loop** é executada pelo processador monociclo. O endereço desta instrução é 0x004000038 e loop é um rótulo para o endereço 0x00400014. Converta a instrução para linguagem de máquina, apresentando os campos. Apresente na figura os sinais de controle. Escreva um texto explicando como a instrução é executada.

PASSO 1: Primeiramente vamos converter a instrução **beq \$s0, \$t1** para linguagem máquina:

A instrução **beq** é do Tipo I pois seu opcode é 000 100₂.

O formato do tipo I é o seguinte:

Opcode	rs	rt	Offset
6 bits	5 bits	5 bits	16 bits
000 100			

Como nossa instrução é **beq** os registradores rs e rt serão comparados.

O número do registrador rs que é \$s0 é: 10 000₂.

O número do registrador rt que é \$t1 é: 01 001₂.

Assim ficamos com os campos preenchidos da seguinte maneira:

Opcode	rs	rt	Offset
6 bits	5 bits	5 bits	16 bits
000 100	10 000	01 001	

Para encontrar o imediato que irá para o label loop, precisamos fazer Loop - (PC + 4), para que a próxima instrução seja do endereço loop, se a comparação for verdadeira.

Imediato = Loop - (PC + 4)

0x0040 0014 - (0x0040 0038 + 4)

0x0040 0038 + 4 = 0x0040 003C

0x0040 0014 - 0x0040 003C = 0xFFFF FFFF FFF6

Imediato = FFFF FFFF FFF6 = -10

Em 16 bits fica 0x FFF6 ou 1111 1111 1111 0110

Assim ficamos com os campos preenchidos da seguinte maneira:

Opcode	rs	rt	Offset
6 bits	5 bits	5 bits	16 bits
000 100	10 000	01 001	1111 1111 1111 0110

A instrução **beq \$s0,\$t1, loop** convertida para linguagem máquina fica:

000100 10000 01001 1111 1111 1111 0110₂ ou 0x1209FFF6

PASSO 2: Explicar a execução no processador monociclo.

No processador monociclo, a instrução beq \$s0,\$t1, loop é executada da seguinte forma:

1. Obtenção da Instrução:

- O endereço armazenado no Program Counter (PC) 0x00400038 é enviado para a memória de instrução.
- A instrução 0x1209FFF6 é obtida.

2. Decodificação da Instrução:

- Os campos da instrução são divididos:
- 000100₂ vai para o controle.
- 10000₂ é o registrador a ser lido.
- 01001₂ é o segundo registrador a ser lido.
- O campo do offset vai para a extensão do sinal: 1111 1111 1111 0110₂.

3. Controle:

- A instrução beq modifica os campos DvC e UALOp0 para nível lógico 1.

4. Leitura de Registradores:

- São efetuadas leituras nos registradores \$s0 e \$t1 para realizar a comparação entre os valores armazenados neles.

5. Extensão de Sinal:

- 0xFFFF (imediato) é estendido para 32 bits para ser somado a endereço de PC+4.

6. Deslocamento do Imediato:

- O valor imediato é deslocado duas posições à esquerda, resultando em uma multiplicação por 4. Como resultado desse processo, o valor imediato torna-se 0xFFFF FFD8.

7. Operação da UAL (Unidade Aritmética e Lógica):

- Como UALop é 01₂, a operação é uma subtração(indicada por 0110₂).
- UALFonte em nível lógico 0, conectado ao sinal do multiplexador.

6.1. Seleção do Mux:

- I. UALFonte em nível lógico 0 está conectado ao sinal do multiplexador.
- II. A entrada 0 do Mux é selecionada como saída e vai para uma das entradas da UAL.

6.2. Soma na UAL:

- III. A UAL recebe o sinal 0110₂, indicando que ocorrerá uma subtração.
- IV. O valor de \$t1 é subtraído de \$s0.

8. Condição de Comparação:

- Se a subtração realizada na UAL for igual a 0, o nível lógico de FontePC é alterado para 1, pois a porta lógica AND terá ambas as entradas iguais a 1.
- Assim, a entrada do mux é 1, fazendo com que o endereço de PC + 4 + 4*imediato (estendido) saia na saída.
- Se a subtração realizada na UAL for diferente de 0, a instrução vai para o endereço, mas não realiza nenhuma ação.

9. Próxima Instrução:

- Dependendo do resultado da comparação, a próxima instrução pode ser:
 - I. PC + 4: Se os valores armazenados nos registradores \$s0 e \$t1 não forem iguais.
 - II. PC + 4 - 40: Se os valores armazenados nos registradores \$s0 e \$t1 forem iguais.

