



Instituto de Computação  
Engenharia de Computação

Mateus Fernando Felismino da Silva Patriota

**Universidade Federal de Alagoas**

Lista 2 - Processador MIPS, Desempenho, Mono-ciclo e  
Multi-ciclo

Maceió

2019

1)

- a) Os dados e instruções em MIPS são armazenados em 4 bytes, ou seja, 32bits.
- b) O endereço de memória é ocupado por uma palavra de 4 bytes, ou seja, 4 posições de memória.
- c) A arquitetura MIPS possui 32 registrados para armazenamento geral, como endereço e dados, possui formato \$n, sendo  $0 \leq n \leq 31$ . O registrador 0, é descrito como \$zero e possui valor 0.
- d) As instruções possuem 3 formatos:

**Formato R**

Opcode – 6bits	Reg s. - 5bits	Reg t. - 5bits	Reg d. - 5bits	Função - 6bits
----------------	----------------	----------------	----------------	----------------

**Formato I**

Opcode – 6bits	Reg s. - 5bits	Reg t. - 5bits	Imediato - 16bits
----------------	----------------	----------------	-------------------

**Formato J**

Opcode – 6bits	Endereço - 26bits
----------------	-------------------

Ou seja, todo opcode possui 6bits de espaço e cada registrador 5. É possível verificar, que toda instrução possui 32bits, 4 bytes.

2)

Monociclo

- Simplicidade de implementação, em datapath e controle.
- Todas as instruções utilizam o mesmo período do clock.
- O ciclo do clock é mais lento, pois é definido por aquela instrução que possui maior complexidade.
- Cada unidade lógica só pode ser utilizada uma vez por ciclo.
- Maior custo de HW, necessita de duplicação de componentes.

Multiciclo

- Cada unidade funcional pode ser utilizada mais de uma vez por instrução.
- Cada etapa leva 1 ciclo de clock.
- Instruções possuem diferente CPI.
- Compartilha unidades funcionais durante a execução de uma única execução.

3)

Vantagens

- Facilidade em executar operações aritméticas simples ao HW.
- Formato genérico.

Desvantagens

- Necessidade de implementação de diversas linhas de códigos complexos quando a realizar operações aninhadas.
- Necessita de mais instruções caso tenha mais entradas.

- 4) Mips é considerada como arquitetura LOAD/STORE porque suas operações lógicas e aritméticas são executadas exclusivamente entre registradores da arquitetura ou entre constantes imediatas e registradores. As operações de acesso à memória executam, apenas, ou uma leitura ou escrita, daí o LOAD/STORE.

#### Vantagem

- Velocidade em operações com registradores.
- Facilidade na implementação de operações paralelas, PipeLine.

#### Desvantagem

- Possui apenas 32 registradores, logo, compensa na leitura e escrita na memória, causando operações relativamente mais lentas.

5)

- a) Considerando a fórmula de tempo de execução

$$T = \frac{N \times CPI}{F}$$

Onde:

N = número de instruções a serem executadas.

CPI = Ciclos por instrução.

F = Frequência do processador.

Considerando N como um número qualquer de instruções, temos:

Para P1 :

$$T = \frac{N \times 1.5}{2 \times 10^9}, T = N * 0.75 * 10^{-9}$$

Para P2 :

$$T = \frac{N \times 1}{1.5 \times 10^9}, T = N * 0.66 * 10^{-9}$$

Para P2 :

$$T = \frac{N \times 2.5}{3 \times 10^9}, T = N * 0.83 * 10^{-9}$$

Dessa maneira, o processador que possui menos tempo de execução para N instruções é P2, logo P2 detém o melhor desempenho.

- b) Para P1:

$$10 = \frac{N \times 1.5}{2 \times 10^9}, N = 2 \times 10^9 / 1.5 = 13.33 * 10^9 \text{ instruções.}$$

$$C = N \times CPI, C = 13.333 * 10^9 * 1.5 = 20 * 10^9 \text{ Ciclos.}$$

**Para P2:**

$$10 = \frac{N \times 1}{1.5 \times 10^9}, N = 15 \times 10^9 / 1 = 15 * 10^9 \text{ instruções.}$$

$$C = N \times CPI, C = 15 \times 10^9 * 1 = 15 * 10^9 \text{ Ciclos.}$$

**Para P3:**

$$10 = \frac{N \times 2.5}{3 \times 10^9}, N = 30 \times 10^9 / 2.5 = 12 * 10^9 \text{ instruções.}$$

$$C = N \times CPI, C = 12 \times 10^9 * 2.5 = 30 * 10^9 \text{ Ciclos.}$$

**c) Para P1:**

$$7 = \frac{13 \times 10^9 \times 1.5 \times 1.2}{f}, f = 23.4 \times 10^9 / 7 = 3.34 * 10^9.$$

**Para P2:**

$$7 = \frac{15 \times 10^9 \times 1 \times 1.2}{f}, f = 18 \times 10^9 / 7 = 2.57 * 10^9.$$

**Para P3:**

$$7 = \frac{12 \times 10^9 \times 2.5 \times 1.2}{f}, f = 36 \times 10^9 / 7 = 5.14 * 10^9.$$

**6)**

**a)** Considerando que a quantidade de ciclos é dada por:

$$C = \sum_{i=0}^n N_i \times CPI_i$$

Temos que,

**Para P1:**

$$T = \frac{(10^6 * 1) * 0.1 + (10^6 * 2) * 0.2 + (10^6 * 3) * 0.5 + (10^6 * 4) * 0.20}{1.5 * 10^9}$$

$$T = 1.86 * 10^{-3} \text{ segundos.}$$

Para P2:

$$T = \frac{(10^6 * 2) * 0.1 + (10^6 * 2) * 0.2 + (10^6 * 2) * 0.5 + (10^6 * 2) * 0.20}{2 * 10^9}$$

$$T = 1 * 10^{-3} \text{ segundos.}$$

**Logo P2, tem melhor desempenho.**

**b)** Para P1:

$$CPI = C / N$$

$$CPI = 2.7999 * 10^6 / 10^6 = \mathbf{2.8 \text{ Ciclos por instrução.}}$$

Para P2:

$$CPI = C / N$$

$$CPI = 2 * 10^6 / 10^6 = \mathbf{2 \text{ ciclos Ciclos por instrução.}}$$

**c)** Para P1, temos **2.8\*10<sup>6</sup> Ciclos de clock.**

Para P2, temos **2\*10<sup>6</sup> Ciclos de clock.**

- 7)** Considerando que a nova instrução funcionará no formato R, ou seja, possui 6bits para opcode, 5bits para cada registrador (rs e rt). Podemos, dessa maneira, reutilizar uma instrução já pré implementada, no caso a instrução add. Assim, consideramos que a instrução [25 - 21] contém rs, e [20 - 16] contém rt.

Em seguida, adiciona-se um MUX ao sistema, na entrada do Read R.2. Esse mux é definido como: Se 0, o valor recebido é rt, caso seja 1, recebe a constante 0 (\$zero). O Opcode da nova instrução fica responsável pelo sinal da unidade de controle para definir o valor do MUX.

Dessa maneira a nova instrução utilizaria de add, com o valor constante zero. O resultado será salvo no Write Data, utilizando o registrador rt como Write Register.