

# Arquitetura de Sistemas Digitais (FTL066)

## O Processador

### Quarta Lista de Exercícios

- 1) Considere um pequeno multiplexador com 4 entradas de dados de 1 bit.
  - a) Este bloco contém somente lógica, somente *flip-flops*, ou ambos?
  - b) Mostre como este bloco pode ser implementado. Use somente portas AND, OR, NOT.
  - c) Estenda a letra b) com buffers *tri-state*.
  - d) Implemente a letra b) usando um FPGA.
  
- 2) Diferentes instruções utilizam diferentes blocos de *hardware* na implementação básica do ciclo único.

	Instrução	Interpretação
a.	AND Rd, Rs, Rt	$\text{Reg}[\text{Rd}] = \text{Reg}[\text{Rs}] \text{ AND } \text{Reg}[\text{Rt}]$
b.	ADD Rd, Rs, Rt	$\text{Reg}[\text{Rd}] = \text{Reg}[\text{Rs}] + \text{Reg}[\text{Rt}]$
c.	SW Rt, Offs(Rs)	$\text{Mem}[\text{Reg}[\text{Rs}] + \text{Offs}] = \text{Reg}[\text{Rt}]$
d.	LW Rt, Offs(Rs)	$\text{Reg}[\text{Rt}] = \text{Mem}[\text{Reg}[\text{Rs}] + \text{Offs}]$

- a) Quais são os valores dos sinais de controle gerados pelo controle do slide 43 ([http://home.ufam.edu.br/lucascordeiro/arquitetura/slides/4\\_o\\_processador.pdf](http://home.ufam.edu.br/lucascordeiro/arquitetura/slides/4_o_processador.pdf))?
  - b) Quais recursos (blocos) executam uma função útil?
- 3) Quais das seguintes frases estão corretas para uma instrução *load*?
- a) *MemtoReg* deve ser setado para fazer com que o dado da memória seja enviado para o arquivo de registrador.
  - b) *MemtoReg* deve ser setado para fazer com que o registrador de destino correto seja enviado para o arquivo de registrador.
  - c) Nós não nos importamos sobre a definição de *MemtoReg* para *loads*.
- 4) O caminho de dados de ciclo único, descrito conceitualmente em sala de aula, deve ter memória de instruções e dados separada, porque:
- a) O formato das instruções e dos dados são diferentes no MIPS, e deste modo diferentes memórias são necessárias.
  - b) Tendo memórias separadas é menos dispendioso.

c) O processador opera em um ciclo e não pode usar uma memória de única porta para dois acessos diferentes dentro desse ciclo.

5) Diferentes unidades de execução e blocos de lógica digital têm diferentes latências (tempo necessário para realizar o trabalho).

	<b>I-Mem</b>	<b>Add</b>	<b>Mux</b>	<b>ULA</b>	<b>Regs</b>	<b>D-Mem</b>	<b>Control</b>
a.	200ps	70ps	20ps	90ps	90ps	250ps	40ps
b.	750ps	200ps	50ps	250ps	300ps	500ps	300ps

Qual é o tempo necessário para executar as instruções AND, ADD, LOAD, SW e BEQ? Responda esta questão com base no circuito do slide 43 (mencionado na questão 2).

6) Projete um circuito lógico com três entradas A, B e C, cuja saída será nível alto apenas quando a maioria das entradas for nível alto. Implemente este circuito usando o FPGA.

7) A latência de componentes individuais do caminho de dados afetam o tempo de ciclo de relógio de todo caminho de dados.

	<b>I-Mem</b>	<b>Add</b>	<b>Mux</b>	<b>ULA</b>	<b>Regs</b>	<b>D-Mem</b>	<b>Sign-extend</b>	<b>Shift-left-2</b>
a.	200ps	70ps	20ps	90ps	90ps	250ps	15ps	10ps
b.	750ps	200ps	50ps	250ps	300ps	500ps	100ps	5ps

- Qual é o tempo de ciclo de relógio se somente as instruções do tipo ULA (add, AND, etc.) são executadas?
- Qual é o tempo de ciclo de relógio se somente instruções LW são suportadas?
- Qual é o tempo de ciclo de relógio se somente instruções ADD, BEQ, LW e SW são suportadas?

8) Examinaremos como *pipeline* afeta o tempo de ciclo do processador. Assuma as seguintes latências de cada estágio individual:

	<b>IF</b>	<b>ID</b>	<b>EX</b>	<b>MEM</b>	<b>WB</b>

a.	250ps	350ps	150ps	300ps	200ps
b.	200ps	170ps	220ps	210ps	150ps

- a) Qual é o tempo de ciclo de relógio em um processador com e sem *pipeline*?
- b) Qual é a latência total das instruções LW e SW em um processador com e sem *pipeline*?

9) Para cada sequência de código abaixo, declare se o mesmo deve parar, pode evitar paradas usando encaminhamento ou pode executar sem parar ou encaminhar.

#### Sequência 1

```
lw $t0, 0($t0)
add $t1, $t0, $t0
```

#### Sequência 2

```
add $t1, $t0, $t0
addi $t2, $t0, #5
addi $t4, $t1, #5
```

#### Sequência 3

```
addi $t1, $t0, #1
addi $t2, $t0, #2
addi $t3, $t0, #2
addi $t3, $t0, #4
addi $t5, $t0, #5
```

10) Para as seguintes instruções, assuma que elas são executadas em um caminho de dados com 5 estágios de *pipeline*.

```
ADD R5, R2, R1
LW R3, 4(R5)
LW R2, 0(R2)
OR R3, R5, R3
SW R3, 0(R5)
```

```
LW R2, 0(R1)
AND R1, R2, R1
LW R3, 0(R2)
LW R1, 0(R1)
SW R1, 0(R2)
```

- a) Quais são os perigos que podemos encontrar?
- b) Como eles podem ser resolvidos?

11) Implementa uma ULA (Unidade Lógica Aritmética) usando VHDL que realize as seguintes funções: AND, OR, XOR, ADD, MUL, DIV, SUB, *set-on-less-than*, NOR, NOT.

**Data de entrega: 19 de agosto de 2014 (terça-feira).**

**Após esta data será descontado 2 pontos por dia de atraso.**

**A lista de exercícios deve ser resolvida e entregue individualmente.**

**05/08/2014**