



Nome: GABARITO

Matrícula: 19/0012345

## Prova 2

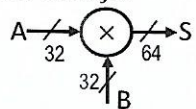
1) (2.4) Mantendo a compatibilidade binária com a ISA RISC-V RV32IMF implemente as instruções abaixo nos processadores Uniciclo, Multiciclo e Pipeline vistos em aula, indicando nas folhas em anexo as modificações necessárias nos Caminhos de Dados e nos Blocos de Controle. Dicas: Se necessitar alteração na ULA explique-a na folha de almaço.

a) (1.2) mul t0, t1, t2

b) (1.2) uret

# PC=UEPC

Você dispõe do seguinte módulo de multiplicação:



2) (6.0) A detecção de exceções e interrupções é uma das partes mais complexas na construção de um processador, pois envolve um desvio não previsto no programa a uma rotina de tratamento desses efeitos.

Considere que o processador RISC-V possua um pino de ligação externa que, quando ativado '1', indique que o dispositivo externo (teclado) colocou um dado novo (código ASCII da tecla pressionada) em seu buffer (endereço 0xFF200004). O pino de interrupção fica ativado até que o processador indique ao dispositivo que efetuou a leitura do dado, quando então o dispositivo o desativa '0'.

a) (1.0) Explique qual é a diferença conceitual entre exceção e interrupção.

b) (3.0) Modifique adequadamente os Caminhos de Dados e os Blocos de Controle dos processadores Uniciclo, Multiciclo e Pipeline, nas folhas em anexo, de forma que, a cada vez que uma interrupção ocorrer: o endereço PC da instrução que iria ser executada seja salvo no registrador UEPC e a rotina de tratamento, cujo endereço já está no registrador UTVECT, seja chamada.

c) (1.0) Crie uma rotina de tratamento de interrupção que coloque o valor ASCII da tecla pressionada no registrador a0, indique ao dispositivo que a leitura foi feita através do zeramento do conteúdo do endereço do buffer, e retorne ao programa principal. Dica: Considere que você dispõe da instrução uret (implementada no item 1.b).

d) (1.0) Faça um programa principal que incremente continuamente um contador no registrador s0 e a cada vez que a tecla 'r' (ASCII 114) for pressionada reinicie esse contador do 0, pause a contagem pelo pressionamento da tecla 'p' (112) e continue a contagem pelo pressionamento da tecla 'c' (99).

3) (3.0) Considerando apenas os seguintes tempos dos elementos dos caminhos de dados originais vistos em aula e apresentados nos anexos (sem considerar as suas modificações das questões anteriores):

- Tempo de acesso à memória de dados ou de instruções para leitura: 500ps
- Tempo de acesso às memórias de dados para escrita: 250ps
- $t_{pd}$  de qualquer operação da ULA: 200ps
- $t_{pd}$  dos somadores de 32 bits: 100ps
- $t_{co}$  de qualquer registrador (flip-flop): 50ps
- $t_{su}$  de qualquer registrador (flip-flop): 25ps
- $t_{pd}$  de qualquer decodificador e multiplexador: 20ps
- $t_{pd}$  da unidade de geração de imediato: 20ps
- $t_{pd}$  das unidades de controle/circuitos combinacionais: 20ps
- $t_{pd}$  das portas AND e OR: 5ps

Dica: Reveja a estrutura interna do Banco de Registradores e do Controle Multiciclo.

```
PROC: addi a0, a0, 4  → 3 Bolhas
      lw t0, 0(a0)    → 3 Bolhas
      sw t0, -4(a0)
      bne a0, a1, PROC
      ret              → 1 Bolha
```

Para a ISA reduzida de 9 instruções desenvolvida em aula, calcule: i) qual a máxima frequência de clock utilizável e ii) o tempo de execução do procedimento PROC com os argumentos a0=0x10010000 e a1=0x10010010, para os processadores:

a) (1.0) Processador Uniciclo;

b) (1.0) Processador Multiciclo;

c) (1.0) Processador Pipeline; onde todos os hazards são solucionados apenas com inserção de bolhas, o banco de registradores é escrito na borda de subida e o branch previsto como não tomado é avaliado no 2º ciclo.

Dica: Considere addi similar a add, bne similar a beq e jalr similar a jal

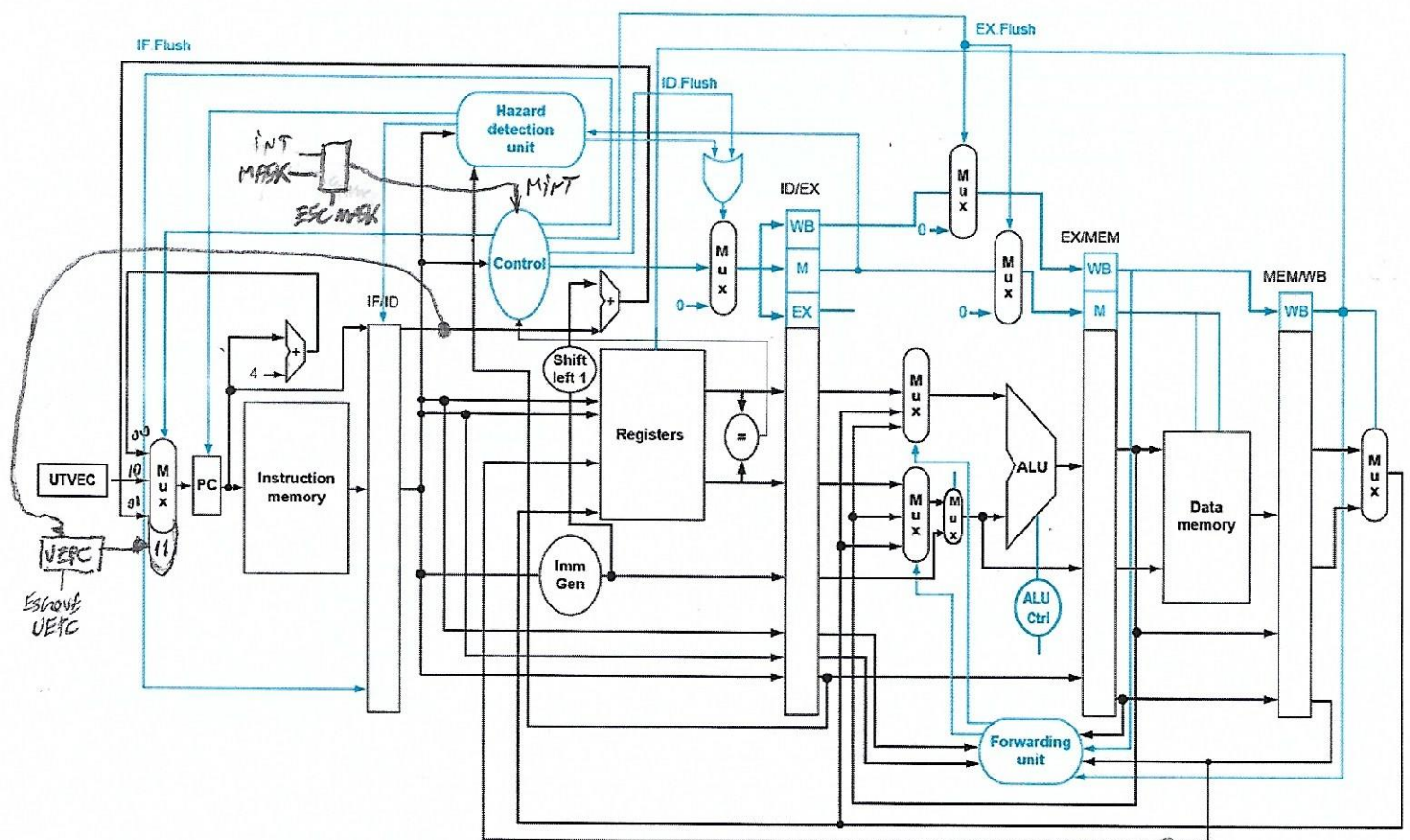
Como diria o Mestre Yoda: "Complique não o que simples é!"











INT TRATADO NO ESTAGIO ID  
L7PC DO ID E SALVO NO UERC.

MULS OS PRECISA MUDANCA NA ULA

1) a) EXP. DA ULA  
b) MUX, REG UERC, CONTROLE  
2) b) MUX, ESQUELETO UERC, CONTROLE  
MINT E MASK,

Estágio	IF					ID			EX				
	IF.Flush	PCSrc	ESQUELETO UERC	MASK	ESQUELETO MASK	ID.Flush			EX.Flush	ALUSrc	ALUOp		
Tipo R	0	00	0	X	0	0			0	0	10		
lw	0	00	0	X	0	0			0	1	00		
sw	0	00	0	X	0	0			0	1	00		
beq&=	1	01	0	X	0	0			0	x	xx		
beq&≠	0	00	0	X	0	0			0	x	xx		
URET	1	11	0	1	1	0			0	x	xx		
MINT=1	1	10	1	0	1	0			0	x	xx		
						5001							

Estágio	MEM				WB		
	MemRead	MemWrite			RegWrite	Mem2Reg	
Tipo R	0	0			1	0	
lw	1	0			1	1	
sw	0	1			0	x	
beq&=	0	0			0	x	
beq&≠	0	0			0	x	
URET	0	0			0	x	
MINT=1	0	0			0	x	

## PROVA 2

## GABARITO

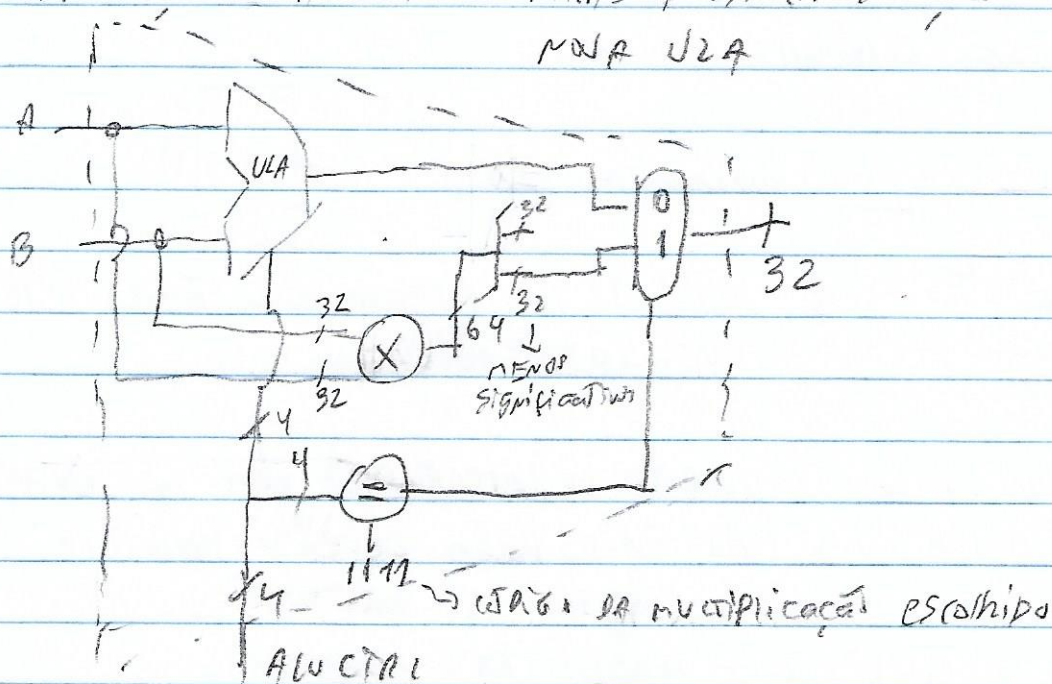
1)

d) MUL  $t_0, t_1, t_2$  $t_0 = \text{LOW} \{t_1 \times t_2\} \quad [31:0]$ 

OPCODE: TIPO-X 0XB3

FUNCT3: 000 FUNCT7: 00000001

\*MODIFICAÇÕES NA ULA: VÁRIAS POSSÍVEIS SOLUÇÕES  
PARA ULA



1,2

E

\*INCREMENTAR CONTROLE DA ULA P/ RECONHECER FUNCT7 E FUNCT3 DO MUL  
E GERAR O CÓDIGO 1111

ASSIM O MUL FICA AUTOMATICAMENTE IMPLEMENTADO NOS  
3 PROCESSADORES! NA NECESSITANDO DE NENHUMA MODIFICAÇÃO  
ADICIONAL NOS CAMINHOS DE DADOS E BLOCOS DE CONTROLE



b) URET # PC = UERC

OPCODE: 273 : 1110011

FUNC13: 000

FUNC7: 0000000

RS2: 00010

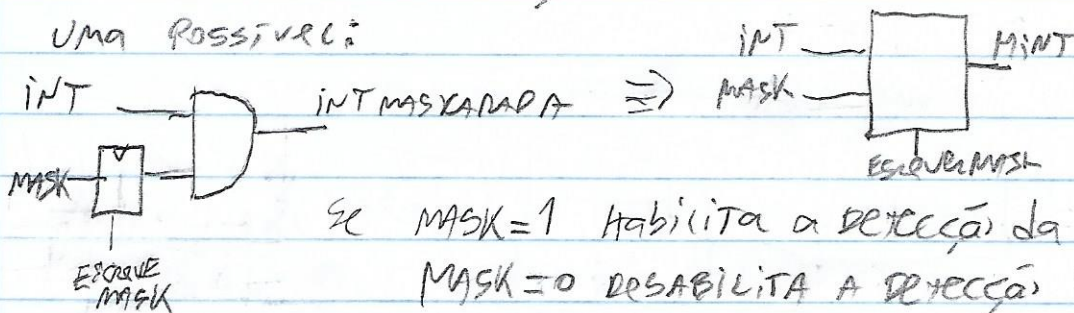
NAS FO Ihas

## 2) interrupção por HARDWARE

PROBLEMA: UMA VEZ IDENTIFICADA/RECONHECIDA A SOLICITAÇÃO DO DISPOSITIVO EXTERNO DEVEMOS INICIAR A ROTINA DE TRATAMENTO DA interrupção, MASCARANDO O PÍMO DE interrupção PARA NÃO ENTRAR EM LOOP DO.

EXISTEM DIVERSAS SOLUÇÕES.

UMA POSSÍVEL:



SE MASK=1 HABILITA A DETECÇÃO DA interrupção  
MASK=0 DESABILITA A DETECÇÃO " "

A ROTINA DE TRATAMENTO DE interrupção deve ser executada COM MASK=0. AO SAIR DELA DEVE-SE HABILITAR MASK=1  
↳ POSSÍVEL SOLUÇÃO

→ DETECÇÃO → DESABILITA MASK=0

→ URET → HABILITA NOVAMENTE MASK=1

- a) EXCEÇÃO → Evento interno ao processador  
INTERRUPÇÃO → Evento externo ao processador

b) NAS FOLHAS

c) INTERRUPÇÃO / EXCEÇÃO NÃO DEVE ALTERAR REGISTROS

```
INT: addi sp, sp, -4  
sw t0, 0(sp)  
li t0, 0xFF200004  
lw a0, 0(t0)  
sw zero, 0(t0)  
lw t0, 0(sp)  
addi sp, sp, 4  
uret
```

```
d) MAIN: li t0, 114 # 'Y'  
li t1, 112 # 'P'  
li t2, 99 # 'C'
```

```
ZERAR: li s0, 0
```

```
LOOP: addi s0, s0, 1  
beq a0, t0, ZERAR  
beq a0, t1, PAUSA  
j LOOP
```

```
PAUSA: beq a0, t2, LOOP  
j PAUSA
```



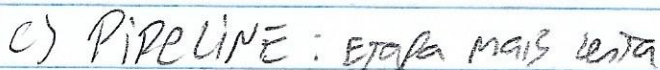
a) UNICiclo : insinuação MA.3. Letra

$$\bar{T} = 50 + 500 + 20 + 20 + 200 + 500 + 20 + 25 = 1,335 \text{ ns}$$

Programa: Executa  $4 + 4 + 4 + 4 + 1 = 17$  instruções

b) MULTICICLO : Etapa mais lenta

Q. No 4 Matriz



$f_{MAX} = 1,39 \text{ GHz}$  //

adig la g sw bre g m jala g

PROGRAMA =  $4 \times (1 + 3 + 1 + 3 + 1 + 1 + 1) - 1 + 1 + 1 = 45 \text{ ciclos}$

$$t_{\text{proc}} = 95 \times 575 \text{ p} = 25,87 \text{ ns}$$