



Nome: GABARITO

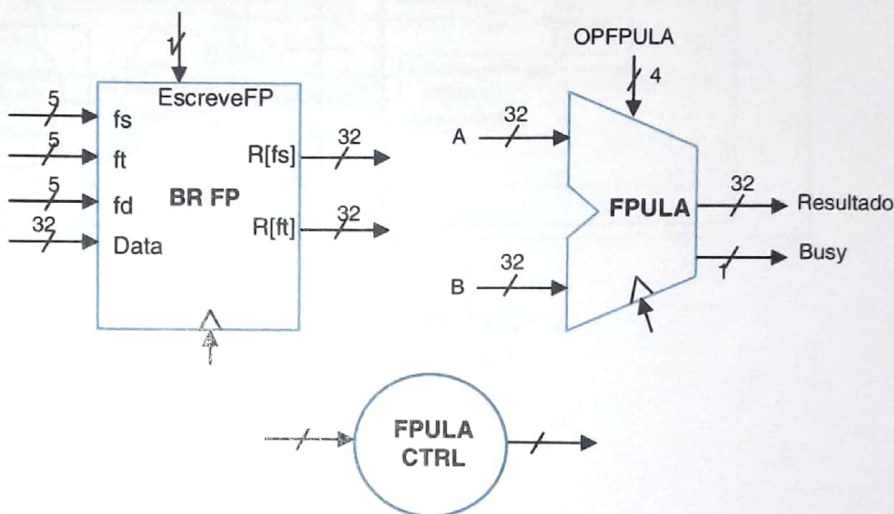
Matrícula:   /

## Prova 2

1)(4.0) Hoje em dia, a Unidade de Ponto Flutuante (FPU) é uma parte do processador relativamente cara de ser implementada, e pode não ser necessária em sistemas computacionais de baixo custo (brinquedos, etc.) ou em chips dedicados a uma aplicação específica que não requer esse tipo de operação matemática (MP3 player, etc.).

Dada uma implementação do Coprocessador 1 do MIPS, em IEEE 754 precisão simples, que possa executar as instruções tipo FR add.s, sub.s, mul.s, div.s, sqrt.s, abs.s, neg.s, floor.w.s, ceil.w.s e round.w.s. Considere que a FPULA seja capaz de executar as operações relacionadas no quadro abaixo e que indique, através de um sinal Busy, quando uma operação esteja pronta, após um certo número de ciclos de clock de processamento.

OPFPULA	Operação da FPULA
0000	Resultado = A
0001	Resultado = B
0010	Resultado = A + B
0011	Resultado = A - B
0100	Resultado = A x B
0101	Resultado = A ÷ B
0110	Resultado = $\sqrt{A}$
0111	Resultado =  A
1000	Resultado = -A
1001	Resultado = (int)floor(A)
1010	Resultado = (int)ceil(A)
1011	Resultado = (int)round(A)
1100	Resultado = (float)(A)
1101	Resultado = (A==B)?1:0
1110	Resultado = (A<B)?1:0
1111	Resultado = (A<=B)?1:0



- a) (1.0) Defina a tabela verdade do bloco FPULA CTRL, indicando quais são os sinais de entrada e de saída;  
b) (2.0) Modifique o caminho de dados do Processador Multiciclo, no verso desta folha, de forma a incluir o Banco de Registradores FP, a FPULA e o bloco FPULA CTRL;  
c) (1.0) Modifique o controle do Processador Multiciclo, no verso desta folha, de modo que a definição de um sinal externo FPUenable=1, faça com que o processador reconheça as instruções FR definidas e as processe **eficientemente**, e ignorando-as caso FPUenable=0;

2)(4.0) Implemente as instruções específicas para operações com pilha (push e pop) nos Processadores Uniciclo, Multiciclo e Pipeline, modificando nas folhas em anexo os Caminhos de Dados (0.5/cada) e os sinais dos Blocos de Controle (0.5/cada). Desenhe na folha de respostas as modificações que se fizerem necessárias no Banco de Registradores a ser usado nas 3 versões (1.0),

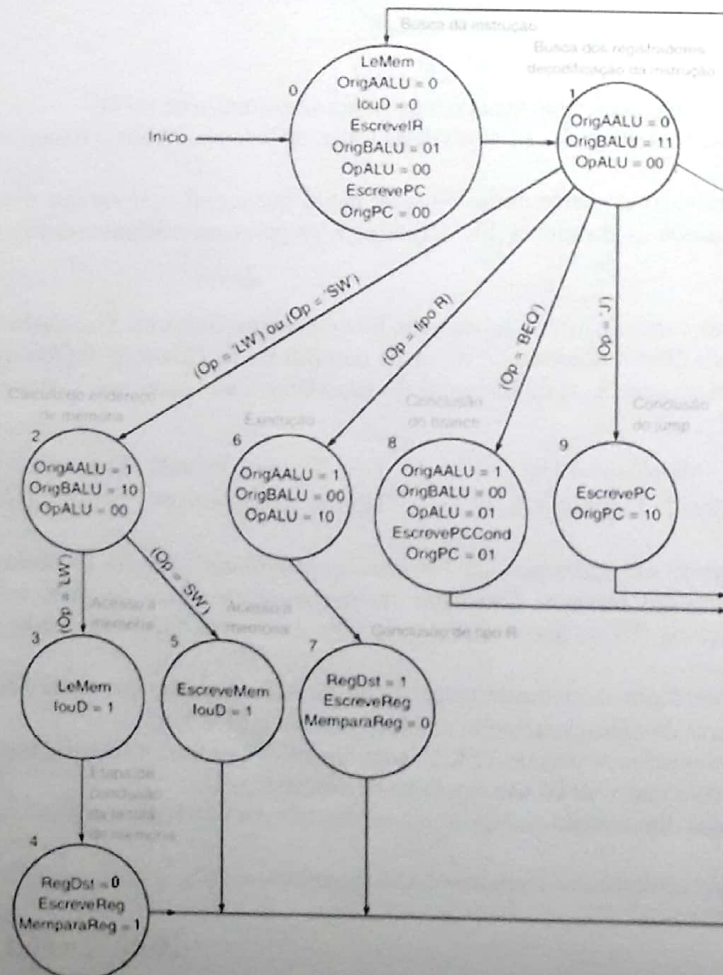
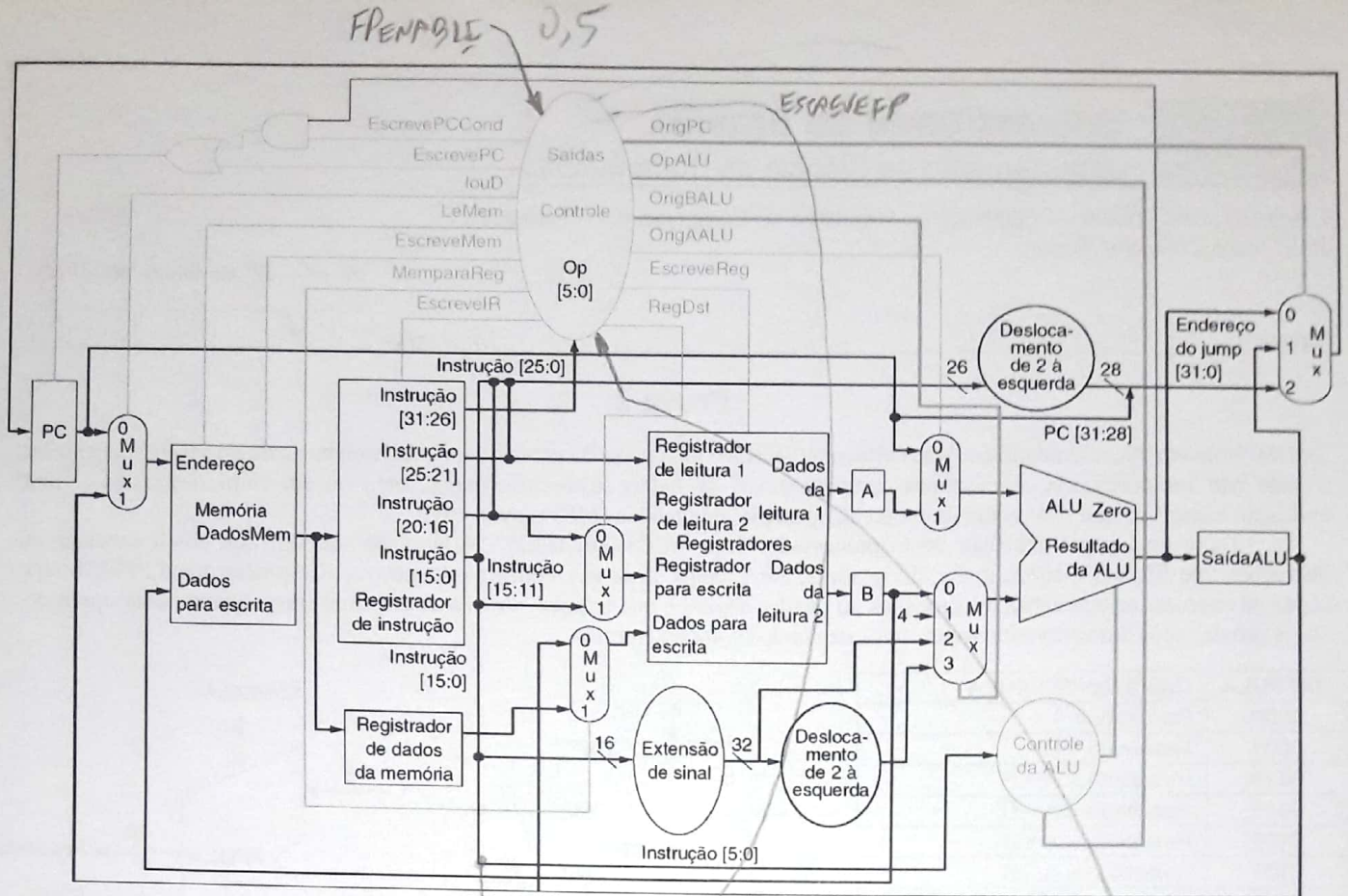
- a) (1.5) push.w rs # \$sp=\$sp - 4; Mem[\$sp]=R[rs] Tipo-R Opcode:0x00 Funct=0x14  
b) (1.5) pop.w rd # R[rd]=Mem[\$sp]; \$sp=\$sp + 4 Tipo-R Opcode:0x00 Funct=0x15

3) (2.5) Aplicações tais como DVD players e MP3 players, são chamadas de “streaming workloads” devido ao acesso inerentemente sequencial aos dados com muito pouco reuso dos mesmos. Considere um streaming workload de vídeo que acesse um conjunto de 1 GiB de dados presentes na memória RAM, que o acesso seja feito em words (32 bits), sendo o endereçamento a byte de tamanho 32 bits.

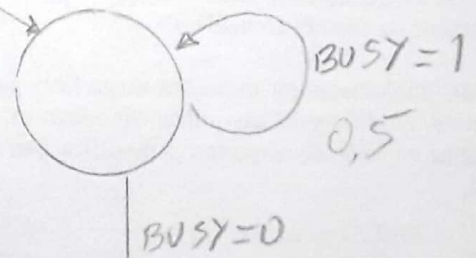
- a) (1.0) Caso seja utilizada uma memória cache de dados diretamente mapeada de 32 KiB (tamanho convencional) com blocos de W words. Faça um gráfico que apresente a taxa de falhas x tamanho do bloco para  $W \in [1, 2, 3, 4, 5]$ .  
b) (0.5) Caso seja utilizada uma memória cache associativa N-way de 32 KiB com blocos de tamanho  $4^b$  bytes. Quais os valores de N e b que minimizam a taxa de falhas? Considere que a MMU use a política de descarte LRU.  
c) (0.5) Caso o tamanho da memória cache de dados diretamente mapeada seja aumentada para 64 KiB, o que ocorre com as taxas de falhas do item a) ?  
d) (0.5) Caso seja necessário acessar 2GiB de dados contínuos na memória RAM, o que ocorre com as taxas de falhas do item a) ?

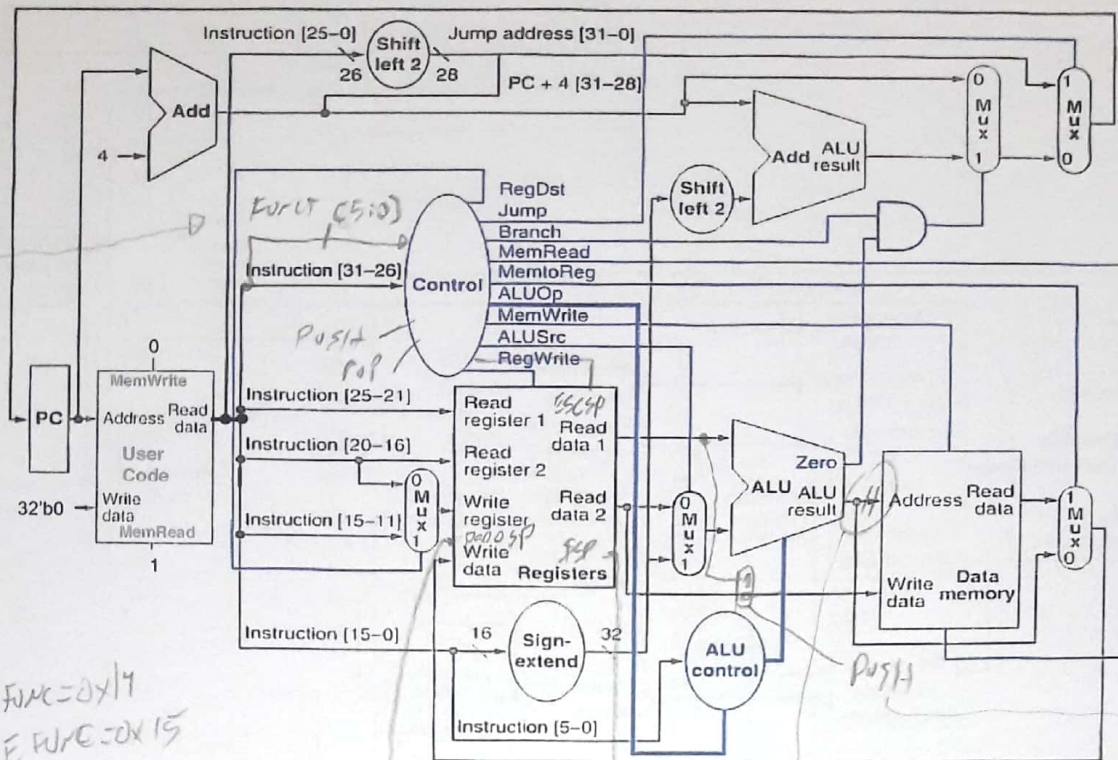
**Boa Sorte!!!**



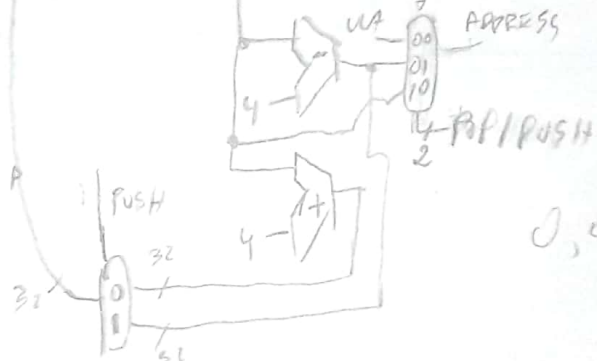


OP=0x11 E FPENABLE=1





P1  
 identificação  
 OPcode = 0 E Func = 0x14  
 OPcode = 0 E Func = 0x15



0,5

Instrução	RegDst	OrigALU	Mempara Reg	Escreve Reg	Le Mem	Escreve Mem	Branch
formato R	1	0	0	1	0	0	0
lw	0	1	1	1	1	0	0
sw	X	1	X	0	0	1	0
beq	X	0	X	0	0	0	1
PUSH.w	X	X	X	0	0	1	0
POP.w	1	X	1	1	1	0	0

Instrução	ALUOp1	ALUOp0	Escreve SP	PUSH	POP			
formato R	1	0	0	0	0			
lw	0	0	0	0	0			
sw	0	0	0	0	0			
beq	0	1	0	0	0			
PUSH.w	X	X	1	1	0			
POP.w	X	X	1	0	1			







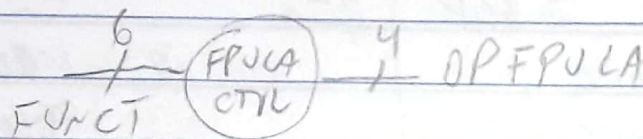


## GABARITO

1)

a) Bloco de controle da FPUA

As instruções tipo FR são definidas pelo campo FUNCT da instrução.

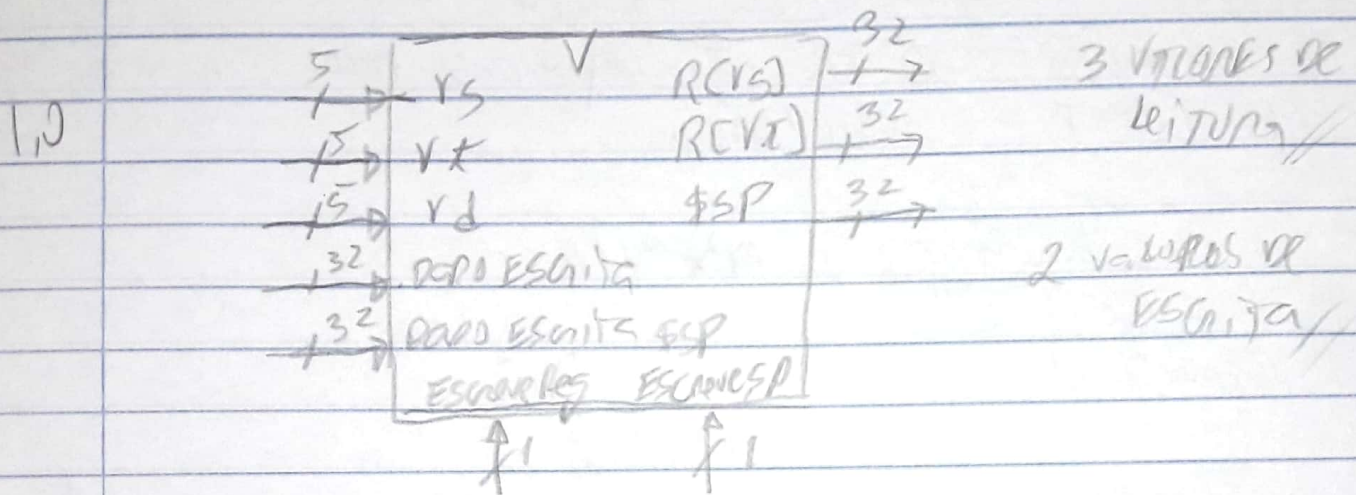


INSTRUÇÃO [5:0]

INSTRUÇÃO	TABELA VERDADE	
	ENTRADA FUNCT	Saída OPFPUA
add.s	0x00	0010
sub.s	0x01	0011
mul.s	0x02	0100
div.s	0x03	0101
SRT.s	0x04	0110
ABS.s	0x05	0111
NEG.s	0x07	1000
FLOOR.W.S	0x0F	1001
CEIL.W.S	0x0E	1010
ROUND.W.S	0x0C	1011
OUTROS		XXXX don't care

OBS: OPCODE = 0x11 para TODAS

2) Para implementação da instrução POP é necessário  
 Escrever o valor lido da pilha em  $V_d$  E  
 Atualizar o registrador  $\$SP$   
 (Obs.: Se  $V_d = \$SP$  pode gerar exceção ou gravar o  
 valor lido gravado, a critério do Aluno)



OBS.: PUSH.W e POP.W  $\rightarrow$  TIPO-R OPCODE = 0

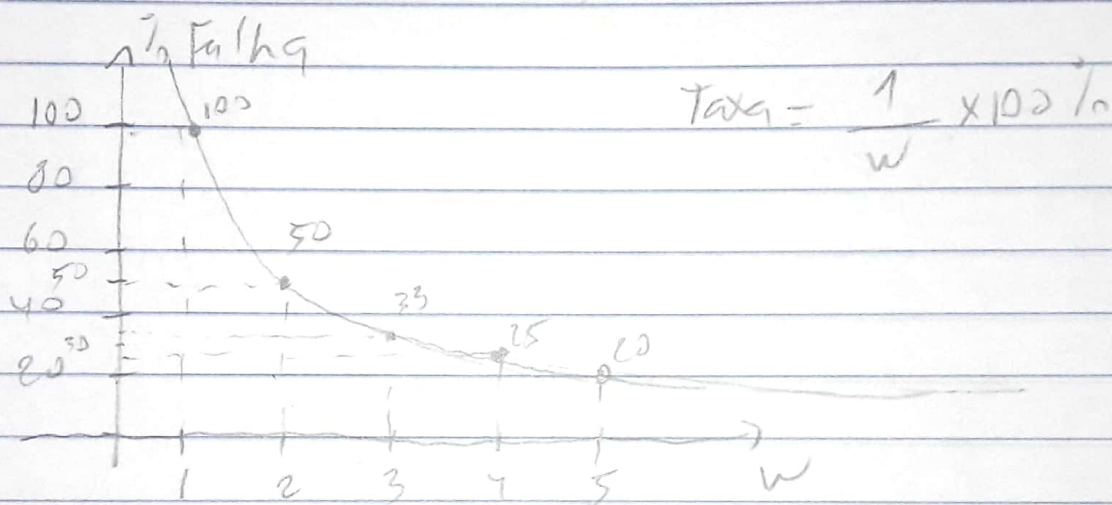
Logo precisa colocar FUNCJ como e, vale a

Nos blocos de controle P/ poder identificá-las



3) Aplicação  $\rightarrow$  leitura puramente sequencial da memória  
 sem reverso 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ...

9) Bloco		Taxa de Falhas
$W=1$	1 Falha a cada leitura	100%
$W=2$	1 Falha a cada 2 leituras	50%
$W=3$	1 Falha a cada 3 leituras	33%
$W=4$	" " 4 " "	25%
$W=5$	" " 5 " "	20%



b) dado ser sequencial  $\rightarrow$  explorar ao máximo a localidade espacial

$\hookrightarrow$  maior tamanho de bloco possível  $N=1$

$$4^b = 32 \times 2^{10}$$

$$2^{2b} = 2^5 \times 2^{10}$$

$$2b = 15$$

$$b = 7,5 \text{ e } N=1$$

OBS: b é só um

parâmetro!

não necessariamente inteiro!

c) tamanho da cache não interfere neste caso logo as taxas de falhas permanecem as mesmas

d) tamanho de papos não influencia na eficiência da cache

Logo, taxas de falhas permanecem as mesmas