



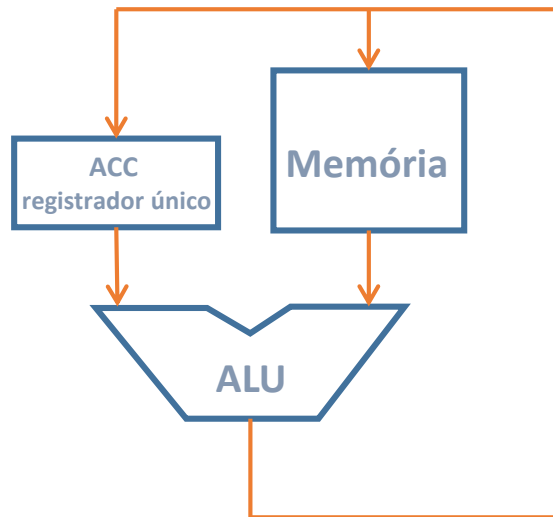
UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO CIÊNCIA DA COMPUTAÇÃO

BRENO DE CASTRO PIMENTA
RA: 2017114809

Trabalho: Lista 01
Disciplina: Organização de Computadores II

Belo Horizonte
2020

1 a) Arquitetura:



Equivalência de código:

Para realizar a equivalência de código será levado em conta os seguintes aspectos:

1. ACC é o registrador único, também chamado de acumulador.
2. O t0 do código MIPS será equivalente a zero para a correspondência com o código do acumulador.
3. O espaço de memória da arquitetura do acumulador com endereço 116 será equivalente ao registrador A0 do MIPS.

Logo o código correspondente é:

LoadACC 108

MulACC 112

SubACC 100

SubACC 104

StoreACC 116

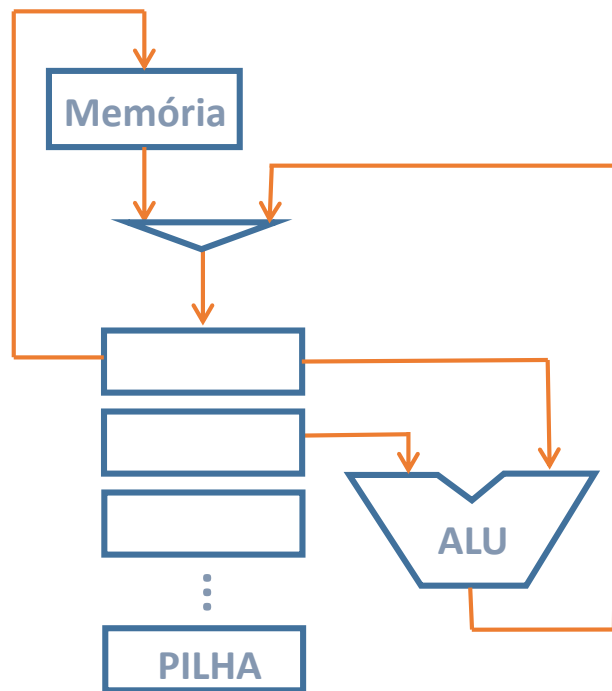
1 b) Vantagens:

- Complexidade espacial melhor, sendo necessário apenas um registrador.
- Quantidade menor ou igual de instruções, pois as instruções de Load são juntas às de operação.

Desvantagens:

- Complexidade temporal pior, pois há que realizar acesso a memória em praticamente toda instrução.

1 c) Arquitetura:



Equivalência de código:

Para realizar a equivalência de código será levado em conta os seguintes aspectos:

1. O t0 do código MIPS será equivalente a zero para a correspondência com o código da pilha.
2. O espaço de memória da arquitetura da pilha com endereço 116 será equivalente ao registrador A0 do MIPS.

Logo o código correspondente é:

```
PUSH 100
PUSH 104
PUSH 108
PUSH 112
MUL
SUB
SUB
POP 116
```

1 d) Vantagens:

- Menor complexidade de endereçamento em instruções, em comparação a uma arquitetura com registradores como o MIPS.

Desvantagens:

- Difícil acesso aos dados, pois em uma arquitetura simples de pilha só acessamos o topo, diferentemente do MIPS onde teremos um endereçamento diferente para cada registrador.
- Maior quantidade de instruções comparado ao MIPS.

2 a) Dependência:

A leitura da questão abre espaço para interpretação de alteração da resposta a partir de uma arquitetura específica, por exemplo caso forsse uma arquitetura de ciclo único não haveria depedência dos dados.

Sendo assim, a interpretação aplicada foi a da dependência dos dados de forma independente da arquitetura.

ADD	R15	R2	R3
SUB	R1	R12	R16
ADDIU	R11	R10	1
MUL	R5	R1	R4
MUL	R7	R5	R6
ADDIU	R18	R11	1
ADDIU	R14	R18	1
ADDIU	R13	R18	2
SW	R5	0 (R14)	
SW	R7	0 (R14)	

2 b)

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10	CC11	CC12	CC13	CC14
ADD R15, R2, R3	IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB									
SUB R1, R12, R16		IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB								
ADDIU R11, R10, 1			IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB							
MUL R5, R1, R4				IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB						
MUL R7, R5, R6					IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB					
ADDIU R18, R11, 1						IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB				
ADDIU R14, R18, 1							IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB			
ADDIU R13, R18, 2								IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB		
SWR5, 0(R14)									IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB	
SWR7, 0(R14)										IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB

2 c)

OUT-OF-ORDER (Y é a unidade funcional de 4 estágios responsável pela multiplicação)

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10	CC11	CC12	CC13	CC14	CC15	CC16	CC17	CC18
ADD R15, R2, R3	IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB													
SUB R1, R12, R16		IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB												
ADDIU R11, R10, 1			IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB											
MUL R5, R1, R4				IF Acesso Mem.	ID	Y0	Y1	Y2	Y3	M Acesso Mem.	WB							
MUL R7, R5, R6					IF Acesso Mem.	ID	-	-	-	Y0	Y1	Y2	Y3	M Acesso Mem.	WB			
ADDIU R18, R11, 1						IF Acesso Mem.	-	-	-	ID	Exec.	M Acesso Mem.	WB					
ADDIU R14, R18, 1							-	-	-	IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB				
ADDIU R13, R18, 2										IF Acesso Mem.	ID	Exec.	-	M Acesso Mem.	WB			
SWR5, 0(R14)											IF Acesso Mem.	ID	-	Exec.	M Acesso Mem.	WB		
SWR7, 0(R14)												IF Acesso Mem.	-	ID	Exec.	M Acesso Mem.	WB	

3 a) Dependência:

A leitura da questão abre espaço para interpretação de alteração da resposta a partir de uma arquitetura específica, por exemplo caso forsse uma arquitetura de ciclo único não haveria depedência dos dados.

Sendo assim, a interpretação aplicada foi a da dependência dos dados de forma independente da arquitetura.

ADD	R1	R2	R3
LW	R4	100(R1)	
SUB	R2	R5	R4
DIV	R5	R2	R1
ADD	R4	R1	R4
SUB	R5	R6	R7
MUL	R2	R5	R4
MUL	R6	R7	R8
ADD	R1	R2	R3
LW	R4	100(R1)	

3 b)

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10	CC11	CC12	CC13
ADD R1, R2, R3	IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB								
LW R4, 100(R1)		IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB							
SUB R2, R5, R4			IF Acesso Mem.	ID	-	Exec.	M Acesso Mem.	WB					
DIV R5, R2, R1				IF Acesso Mem.	-	ID	Exec.	M Acesso Mem.	WB				
ADD R4, R1, R4					-	IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB			
SUB R5, R6, R7							IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB		
MUL R2, R5, R4								IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB	
MUL R6, R7, R8									IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB

3 c) OUT-OF-ORDER

(Y é a unidade funcional de 4 estágios responsável pela multiplicação e D é a responsável pela divisão)

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10	CC11	CC12	CC13	CC14	CC15	CC16	CC17	CC18	CC19	CC20	CC21	CC22
ADD R1, R2, R3	IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB																	
LW R4, 100(R1)		IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB																
SUB R2, R5, R4			IF Acesso Mem.	ID	-	Exec.	M Acesso Mem.	WB														
DIV R5, R2, R1				IF Acesso Mem.	-	ID	D0	D1	D2	D3	D4	D5	D6	D7	M Acesso Mem.	WB						
ADD R4, R1, R4					-	IF Acesso Mem.	ID	Exec.	M Acesso Mem.	WB												
SUB R5, R6, R7							IF Acesso Mem.	ID	-	-	-	-	-	-	Exec.	M Acesso Mem.	WB					
MUL R2, R5, R4								IF Acesso Mem.	-	-	-	-	-	-	ID	Y0	Y1	Y2	Y3	M Acesso Mem.	WB	
MUL R6, R7, R8									-	-	-	-	-	-	IF Acesso Mem.	ID	Y0	Y1	Y2	Y3	M Acesso Mem.	WB

4 a) Processador I2O2

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10	CC11	CC12	CC13	CC14	CC15	CC16	CC17	CC18	CC19	CC20	CC21	CC22
MUL R1, R2, R3	F	D	I	Y0	Y1	Y2	Y3	W														
ADD R4, R2, R1		F	D	I	-	-	-	X0	W													
MUL R2, R7, R8			F	D	-	-	-	I	Y0	Y1	Y2	Y3	W									
LW R10, 0(R12)				F	-	-	-	D	I	M0	M1	W										
MUL R4, R10, R1								F	D	I	-	Y0	Y1	Y2	Y3	W						
SW R4, 0(R2)									F	D	-	I	-	-	-	M0	M1	W				
ADD R5, R6, R7										F	-	D	-	-	-	I	X0	-	W			
ADD R4, R3, R1											F	-	-	-	-	D	I	-	X0	W		
BNE R4, R0																F	D	-	I	X0	W	
ADDI R3, R1, 10																	F	-	D	I	X0	W

4 b) Scoreboard

	P	F	4	3	2	1	0
R1	1	Y					1
R2	1	Y					
R3							
R4	1	X				1	

4 c) Processador IO3

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10	CC11	CC12	CC13	CC14	CC15	CC16	CC17	CC18	CC19
MUL R1, R2, R3	F	D	I	Y0	Y1	Y2	Y3	W											
ADD R4, R2, R1		F	D	i	i	i	I	X0	W										
MUL R2, R7, R8			F	D	I	Y0	Y1	Y2	Y3	W									
LW R10, 0(R12)				F	D	i	i	I	M0	M1	W								
MUL R4, R10, R1					F	D	i	i	I	Y0	Y1	Y2	Y3	W					
SW R4, 0(R2)						F	D	i	i	i	i	i	I	M0	M1	W			
ADD R5, R6, R7							F	D	i	I	X0	W							
ADD R4, R3, R1								F	D	i	i	i	i	i	I	X0	W		
BNE R4, R0									F	D	i	i	i	i	i	I	X0	W	
ADDI R3, R1, 10										F	D	i	i	i	i	i	I	X0	W

- 4 d)** Ao gerar a interrupção no estágio M1 da instrução SW, terá-se um problema quanto a instrução seguinte, número 7 (ADD R5, R6, R7) que já foi executada e já escreveu no banco de registradores, logo se reiniciarmos o pipeline a partir da instrução 6, a instrução 7 será executada duas vezes.

4 e/f) Podemos utilizar do Issue Queue e do Storeboard no momento da exceção para verificar quais instruções já foram executadas e evitar execuções repetidas.

Scoreboard:

	P	4	3	2	1	0
R3	1					
R4	1					
R5						

Issue Queue:

OP	Imm	S	V	Dest	V	P	Src0	V	P	Src1
6	0	1	x	x	1	0	R4	1	0	R2
7	0	0	1	R5	1	0	R6	1	0	R7
8	0	0	1	R4	1	0	R3	1	0	R1
9	x	1	x	x	1	1	R4	1	0	R0
10	10	x	1	R3	1	0	R1	x	x	x

Ou seja, vemos através do IQ que a instrução 7 já iniciou sua execução antes da exceção de desalinhamento de memória. E conferindo com o Scoreboard, confirmamos que a instrução já terminou de ser executada e escreveu seu resultado no banco de registradores, pois o registrador R5 não estava pendente. Logo ao tratar a exceção notificamos que não será necessário executar a instrução 7 novamente.

5 a) Processador I2O2

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10	CC11	CC12	CC13	CC14
MUL R1, R2, R3	F	D	I	Y0	Y1	Y2	Y3	W						
ADD R4, R2, R1		F	D	I	-	-	-	X0	W					
LW R3, 20(R4)			F	D	-	-	-	I	Y0	Y1	Y2	Y3		
ADD R4, R3, R2				F	-	-	-	D	I	-	-	-	X0	W

CC1	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

CC2	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

CC3	P	F	4	3	2	1	0
R1	1	Y					
R2							
R3							
R4							

CC4	P	F	4	3	2	1	0
R1	1	Y	1				
R2							
R3							
R4	1	X					

CC5	P	F	4	3	2	1	0
R1	1	Y		1			
R2							
R3							
R4	1	X					

CC6	P	F	4	3	2	1	0
R1	1	Y			1		
R2							
R3							
R4	1	X					

CC7	P	F	4	3	2	1	0
R1	1	Y				1	
R2							
R3							
R4	1	X					

CC8	P	F	4	3	2	1	0
R1	1	Y					1
R2							
R3							
R4	1	X				1	

CC9	P	F	4	3	2	1	0
R1							
R2							
R3							
R4	1	X					1

CC10	P	F	4	3	2	1	0
R1							
R2							
R3							
R4	1	X					

CC11	P	F	4	3	2	1	0
R1							
R2							
R3							
R4	1	X					

CC12	P	F	4	3	2	1	0
R1							
R2							
R3							
R4	1	X					

CC13	P	F	4	3	2	1	0
R1							
R2							
R3							
R4	1	X				1	

CC14	P	F	4	3	2	1	0
R1							
R2							
R3							
R4	1	X					1

6 a) Processador I2OI

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10	CC11	CC12	CC13	CC14	CC15
MUL R5, R2, R3	F	D	I	Y0	Y1	Y2	Y3	W	C						
MUL R4, R2, R5		F	D	i	i	i	I	Y0	Y1	Y2	Y3	W	C		
SUB R3, R1, R4			F	D	i	i	i	i	i	i	I	X0	W	C	
ADDI R4, R6, 1				F	D	i	i	I	X0	W					C

6 b) CICLO 10:

ROB

Estado	Especulativo	Store Bit	Destino Válido	P-Reg	A-Reg	P P-Reg
F	0	0	1	P9	R4	P7
P	0	0	1	P8	R3	P3
P	0	0	1	P7	R4	P4
--						

FL

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
0	0	0	0	0	0	0	0	0	1	1	1

RT

Registrador	R1	R2	R3	R4	R5	R6
Pendente	0	0	1	1	0	0
P-Reg	P1	P2	P8	P9	P5	P6