LAB 3 - OC - G50

STUDENTS IDENTIFICATION:

Number:	Name:
105894	Eduardo Cardoso
106329	Tiago Santos
106427	Martim Rito

2.1 Simple execution, without data forwarding techniques

f)

Clock cycles	377
Instructions	166
Average CPI	2.271

Stalls:	- Data	192
	- Structural	0
	- Branch Taken	15

g) Ao analisar a execução do programa, podemos concluir que a branch prediction policy é branch not taken porque, pela pipeline, antes de saber o resultado do branch, começamos a executar a instrução 'halt' que é a instrução que se executaria caso o salto não fosse tomado.

2.2 Application of data forwarding techniques

c)

Clock cycles	297
Instructions	166
Average CPI	1.789

Stalls:	- Data	112
	- Structural	16
	- Branch Taken	15

d)

$$Speedup = \frac{Time_{base}}{Time_{com\ forwarding}} = \frac{(CPI_{base} \times \#inst_{base}) / clock\ rate_{base}}{(CPI_{forwarding} \times \#inst_{forwarding}) / clock\ rate_{forwarding}} = \rightarrow$$

como a arquitetura, o compilador e o programa são iguais, #inst e clock rate são iguais

$$= \frac{CPI_{base}}{CPI_{forwarding}} = \frac{2.271}{1.789} = 1.269$$

2.3 Source code optimization: minimization of data and structural hazards

a) Attach a copy of the new assembly program.

c)

Clock cycles	233
Instructions	166
Average CPI	1.404

Stalls:	- Data	48
	- Structural	16
	- Branch Taken	15

d)

$$Speedup = \frac{Time_{base}}{Time_{otim}} = \frac{(CPI_{base} \times \#inst_{base}) / clock\ rate_{base}}{(CPI_{otim} \times \#inst_{otim}) / clock\ rate_{otim}} = \rightarrow$$

como a arquitetura, o compilador e o programa são iguais, #inst e clock rate são iguais

$$= \frac{CPI_{base}}{CPI_{otim}} = \frac{2.271}{1.404} = 1.618$$

2.4 Source code optimization: loop unrolling

a) Attach a copy of the new assembly program.

c)

Clock cycles	153
Instructions	126
Average CPI	1.214

Stalls:	- Data	0
	- Structural	16
	- Branch Taken	7

d)

$$Speedup = \frac{Time_{base}}{Time_{otim}} = \frac{(CPI_{base} \times \#inst_{base}) / clock\ rate_{base}}{(CPI_{otim} \times \#inst_{otim}) / clock\ rate_{otim}} = \rightarrow$$

como a arquitetura é igual, clock rate é igual

$$= \frac{CPI_{base} \times \#inst_{base}}{CPI_{otim} \times \#inst_{otim}} = \frac{2.271 \times 166}{1.214 \times 126} = 2.465$$

2.5 Source code optimization: branch delay slot

a) Attach a copy of the new assembly program.

d)

Clock cycles	218
Instructions	166
Average CPI	1.313

Stalls:	- Data	48
	- Structural	16
	- Branch Taken	0

e)

$$Speedup = \frac{Time_{base}}{Time_{otim}} = \frac{(CPI_{base} \times \#inst_{base}) / clock\ rate_{base}}{(CPI_{otim} \times \#inst_{otim}) / clock\ rate_{otim}} = \rightarrow$$

como arquitetura, compiladores e programa são iguais, #inst e clock rate são iguais

$$= \frac{CPI_{base}}{CPI_{otim}} = \frac{2.271}{1.313} = 1.730$$

e)

$CPI = \frac{23}{11} = 2,091$

Table 1: Pipeline time diagram, without data forwarding techniques.

	INSTRUCTIONS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	lw $10, 0($2)	F	D	X	M	W																																			
2	lw $11, 0($3)		F	D	X	M	W																																		
3	dmul $12, $10, $11			F	D	–	–	X_0	X_1	X_2	X_3	X_4	X_5	X_6	M	W																									
4	daddi $12, $12, $10				F	–	–	D	–	–	–	–	–	–	–	–	X	M	W																					
5	sw $12, 0($4)							F	–	–	–	–	–	–	–	–	D	–	–	X	M	W																		
6	daddi $1, $1, 1																F	–	–	D	X	M	W																
7	daddi $2, $2, 8																			F	D	X	M	W															
8	daddi $3, $3, 8																				F	D	X	M	W														
9	daddi $4, $4, 8																					F	D	X	M	W													
10	bne $1, $5, loop																						F	D	X	M	W												
11	halt																							F															
12	lw $10, 0($2)																								F	D	X	M	W										
13																																							
14																																							
15																																							
16																																							
17																																							
18																																							
19																																							
20																																							
21																																							
22																																							
23																																							
24																																							
25																																							
26																																							
27																																							
28																																							
29																																							
30																																							

próxima iteração

Table 2: Pipeline time diagram, with data forwarding techniques.

	INSTRUCTIONS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	lw $10, 0($2)	F	D	X	M	W																																			
2	lw $11, 0($3)		F	D	X	M	W																																		
3	dmul $12, $10, $11			F	D	X_0	-	X_1	X_2	X_3	X_4	X_5	X_6	M	W																										
4	dadd $12, $12, $10				F	D	-	X	-	-	-	-	-	-	M	W																									
5	sw $12, 0($4)					F	-	D	-	-	-	-	-	-	X	M	W																								
6	daddi $1, $1, 1							F	-	-	-	-	-	-	D	X	M	W																							
7	daddi $2, $2, 8														F	D	X	M	W																				
8	daddi $3, $3, 8															F	D	X	M	W																		
9	daddi $4, $4, 8																F	D	X	M	W																	
10	bne $1, $5, loop																	F	D	X	M	W																
11	halt																		F																				
12	lw $10, 0($2)																			F																		
13																																						
14																																						
15																																						
16																																						
17																																						
18																																						
19																																						
20																																						
21																																						
22																																						
23																																						
24																																						
25																																						
26																																						
27																																						
28																																						
29																																						
30																																						

próxima iteração

Table 3: Pipeline time diagram, with minimization techniques to reduce the data and structural hazards.

	INSTRUCTIONS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	lw $10, 0($2)	F	D	X	M	W																																			
2	lw $11, 0($3)		F	D	X	M	W																																		
3	dmul $12, $10, $11			F	D	X_0	-	X_1	X_2	X_3	X_4	X_5	X_6	M	W																										
4	daddi $1, $1, 1				F	D	X	M	W																																
5	daddi $2, $2, 8					F	D	X	M	W																															
6	daddi $3, $3, 8						F	D	X	M	W																														
7	daddi $4, $4, 8							F	D	X	M	W																												
8	dadd $12, $12, $10								F	D	X	-	-	-	M	W																								
9	sw $12, -8($4)									F	D	-	-	-	X	M	W																							
10	bne $1, $5, loop										F	-	-	-	D	X	M	W																						
11	halt														F																								
próxima iteração	12	lw $10, 0($2)															F	D	X	M	W																			
	13																																						
	14																																						
	15																																						
	16																																						
	17																																						
	18																																						
	19																																						
	20																																						
	21																																						
	22																																						
	23																																						
	24																																						
	25																																						
	26																																						
	27																																						
	28																																						
	29																																						
	30																																						

Table 4: Pipeline time diagram: usage of loop unrolling minimization techniques to reduce the control hazards.

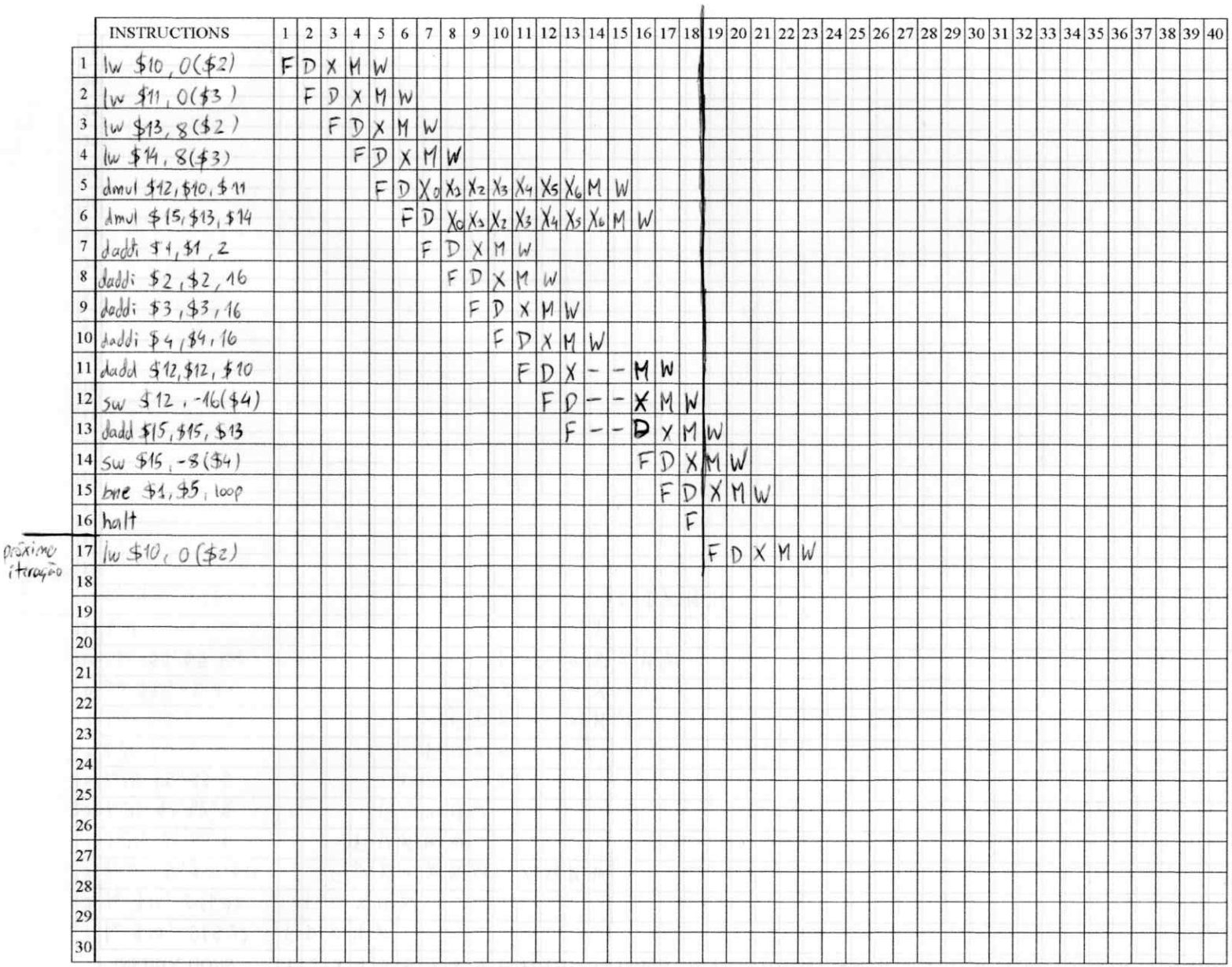

Table 5: Pipeline time diagram: usage of branch delay slot techniques to reduce the control hazards.

	INSTRUCTIONS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	lw $10, 0($2)	F	D	X	M	W																																			
2	lw $11, 0($3)		F	D	X	M	W																																		
3	dmul $12, $10, $11			F	D	X_0	-	X_1	X_2	X_3	X_4	X_5	X_6	M	W																										
4	daddi $1, $1, 1				F	D	X	M	W																																
5	daddi $2, $2, 8					F	D	X	M	W																															
6	daddi $3, $3, 8						F	D	X	M	W																														
7	daddi $4, $4, 8							F	D	X	M	W																												
8	dadd $12, $12, $10								F	D	X	-	-	-	M	W																								
9	bne $1, $5, loop									F	D	-	-	-	X	M	W																							
10	sw $12, -8($4)										F	-	-	-	D	X	M	W																						
11	lw $10, 0($2)														F	D	X	M	W																				
12																																								
13																																								
14																																								
15																																								
16																																								
17																																								
18																																								
19																																								
20																																								
21																																								
22																																								
23																																								
24																																								
25																																								
26																																								
27																																								
28																																								
29																																								
30																																								

próximas iterações