STUDENTS IDENTIFICATION:

Number:	Name:
ist1106221	João Lucas Morais Cardoso Tavares Rodrigues
ist1106180	Vasco Alexandre Amado Rodrigues
ist1107316	Afonso Maria Pereira Mateus

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

e) clockcycles = 23
Instructions = 10
AVG CPI = $\frac{23}{10}$ = 2,3

g) Num total de 16 iterações programadas para o ciclo do programa, ocorreram 15 stalls de Branch Taken e a instrução imediatamente a seguir a "bne", que já tinha entrado no pipeline e estava a meio da sua execução, é flushed sempre que o branch for taken. Logo estamos perante uma branch prediction policy de 'predict not taken".

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)

$$Exe\_time = num\ cycles \times cycle\_time$$
$$Exe\_time_{old} = num\ cycles_{old} \times cycle\_time$$
$$Exe\_time_{new} = num\_cycles_{new} \times cycle\_time$$
$$Speedup = \frac{Exe\_time_{old}}{Exe\_time_{new}} = \frac{num\ cycles_{old} \times \cancel{cycle\_time}}{num\_cycles_{new} \times \cancel{cycle\_time}} = \frac{377}{297} = 1{,}269$$

2.3 Source code optimization: minimization of data and structural hazards

a) Attach a copy of the new assembly program.

c)

Clock cycles	249
Instructions	166
Average CPI	1.500

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

d) Pela fórmula descrita em 2.2 d), o speedup comparado com 2.1 é

$$Speedup = \frac{377}{249} = 1{,}514$$

2.4 Source code optimization: loop unrolling

a) Attach a copy of the new assembly program.

c)

Clock cycles	161
Instructions	126
Average CPI	1,278

Stalls:	- Data	8
	- Structural	16
	- Branch Taken	7

d)

$$\text{Speedup} = \frac{377}{161} = 2,342$$

2.5 Source code optimization: branch delay slot

a) Attach a copy of the new assembly program.

d)

Clock cycles	234
Instructions	166
Average CPI	1,410

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

e)

$$\text{Speedup} = \frac{377}{234} = 1,611$$

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_n	M	W																																			
2	lw $11, 0($3)		F	D	X_n	M	W																																		
3	dmul $12,$10,$11			F	D	Raw	Raw	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M	W																									
4	dadd $12,$12,$10				F	F	F	D	Raw	Raw	Raw	Raw	Raw	Raw	Raw	Raw	X_n	M	W																						
5	sw $12, 0($4)							F	F	F	F	F	F	F	F	F	D	Raw	Raw	X_n	M	W																			
6	daddi $1,$1,1																F	F	F	D	X_n	M	W																		
7	daddi $2,$2,8																			F	D	X_n	M	W																	
8	daddi $3,$3,8																				F	D	X_n	M	W																
9	daddi $4,$4,8																					F	D	X_n	M	W															
10	bne $1,$5, loop																						F	D	X_n	M	W														
11	halt																							F																	
12																																									
13																																									
14																																									
15																																									
16																																									
17																																									
18																																									
19																																									
20																																									
21																																									
22																																									
23																																									
24																																									
25																																									
26																																									
27																																									
28																																									
29																																									
30																																									

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_n	M	W																																			
2	lw $11, 0($3)		F	D	X_n	M	W																																		
3	dmul $12,$10,$11			F	D	M_0	Raw	M_1	M_2	M_3	M_4	M_5	M_6	M	W																										
4	dadd $12,$12,$10				F	D	Raw	X_n	Raw	Raw	Raw	Raw	Raw	Str	M	W																									
5	sw $12, 0($4)					F	F	D	D	D	D	D	D	D	X_n	M	W																								
6	daddi $1, $1, 1							F	F	F	F	F	F	F	D	X_n	M	W																							
7	daddi $2,$2, 8														F	D	X_n	M	W																						
8	daddi $3,$3, 8															F	D	X_n	M	W																					
9	daddi $4, $4, 8																F	D	X_n	M	W																				
10	bne $1, $5, loop																	F	D	X_n	M	W																			
11	halt																		F																						
12																																									
13																																									
14																																									
15																																									
16																																									
17																																									
18																																									
19																																									
20																																									
21																																									
22																																									
23																																									
24																																									
25																																									
26																																									
27																																									
28																																									
29																																									
30																																									

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	daddi $1,$1,1			F	D	X	M	W																																	
4	dmul $12,$10,$11				F	D	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M	W																										
5	daddi $2,$2,8					F	D	X	M	W																															
6	daddi $3,$3,8						F	D	X	M	W																														
7	daddi $12,$12,$10							F	D	X	Raw	Raw	Raw	Str	M	W																									
8	sw $12,0($4)								F	D	D	D	D	D	X	M	W																								
9	daddi $4,$4,8									F	F	F	F	F	D	X	M	W																							
10	bne $1,$5,loop														F	D	X	M	W																						
11	halt															F																									
12																																									
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.

	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	Lw $13, 8($2)			F	D	X	M	W																																	
4	Lw $14, 8($3)				F	D	X	M	W																																
5	dmul $12, $10, $11					F	D	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M	W																									
6	daddi $1, $1, 2						F	D	X	M	W																														
7	dmul $15, $13, $14							F	D	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M	W																							
8	daddi $2, $2, 16								F	D	X	M	W																												
9	daddi $3, $3, 16									F	D	X	M	W																											
10	dadd $12, $12, $10										F	D	X	RAW	STR	M	W																								
11	dadd $15, $15, $13											F	D	D	D	X	STR	M	W																						
12	sw $12, 0($4)												F	F	F	D	D	X	M	W																					
13	sw $15, 8($4)															F	F	D	X	M	W																				
14	daddi $4, $4, 16																	F	D	X	M	W																			
15	bne $1, $5, loop																		F	D	X	M	W																		
16	halt																			F																					
17																																									
18																																									
19																																									
20																																									
21																																									
22																																									
23																																									
24																																									
25																																									
26																																									
27																																									
28																																									
29																																									
30																																									

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	ld $10,0($2)	F	D	X	M	W																																			
2	ld $11,0($3)		F	D	X	M	W																																		
3	daddi $1,$1,1			F	D	X	M	W																																	
4	dmul $12,$10,$11				F	D	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M	W																										
5	daddi $2,$2,8					F	D	X	M	W																															
6	daddi $3,$3,8						F	D	X	M	W																														
7	daddi $12,$12,10							F	D	X	Raw	Raw	Raw	Str	M	W																									
8	sd $12,0($4)								F	D	D	D	D	D	X	M	W																								
9	bne $1,$5,loop									F	F	F	F	F	D	X	M	W																							
10	daddi $4,$4,8														F	D	X	M	W																						
11																																									
12																																									
13																																									
14																																									
15																																									
16																																									
17																																									
18																																									
19																																									
20																																									
21																																									
22																																									
23																																									
24																																									
25																																									
26																																									
27																																									
28																																									
29																																									
30																																									

Code for 2.3

```
loop:
    lw $10, 0($2)           # Load A[i] into $10
    lw $11, 0($3)           # Load B[i] into $11
    daddi $1, $1, 1         # Increment i by 1
    dmul $12, $10, $11      # Multiply A[i] and B[i], store result in $12
    daddi $2, $2, 8         # Move to the next element in A (increment $2 by 8 bytes)
    daddi $3, $3, 8         # Move to the next element in B (increment $3 by 8 bytes)
    dadd $12, $12, $10      # Add A[i] to the product (A[i] * B[i] + A[i])
    sw $12, 0($4)           # Store the result in C[i] (store $12 at address pointed by $4)
    daddi $4, $4, 8         # Move to the next element in C (increment $4 by 8 bytes)
    bne $1, $5, loop        # If i ≠ N (stored in $5), repeat the loop

halt                        # Stop execution
```

Code for 2.4

```
loop:
            lw $10, 0($2)           # Load A[i] into $10
            lw $11, 0($3)           # Load B[i] into $11
            lw $13, 8($2)           # Load A[i[redacted]] into $13
            lw $14, 8($3)           # Load B[i[redacted]] into $14

            dmul $12, $10, $11      # Multiply A[i] and B[i], store in $12
            daddi $1, $1, 2         # Increment i by 2
            dmul $15, $13, $14      # Multiply A[i[redacted] and B[i[redacted]], store in $15

            daddi $2, $2, 16        # Move to the next set of elements in A
            daddi $3, $3, 16        # Move to the next set of elements in B

            dadd $12, $12, $10      # Add A[i] to the product, store in $12
            dadd $15, $15, $13      # Add A[i[redacted] to the product, store in $15

            sw $12, 0($4)           # Store result in C[i]
            sw $15, 8($4)           # Store result in C[i[redacted]

            daddi $4, $4, 16        # Move to the next set of elements in C

            bne $1, $5, loop        # If i ≠ N, repeat loop

            halt                    # Stop execution
```

A[i+1]

B[i+1]

A[i+1] B[i+1]

A[i+1]

C[i+1]

Code for 2.5

```
loop:
    lw $10, 0($2)           # Load A[i] into $10
    lw $11, 0($3)           # Load B[i] into $11
    daddi $1, $1, 1         # Increment i by 1
    dmul $12, $10, $11      # Multiply A[i] and B[i], store result in $12
    daddi $2, $2, 8         # Move to the next element in A (increment $2 by 8 bytes)
    daddi $3, $3, 8         # Move to the next element in B (increment $3 by 8 bytes)
    dadd $12, $12, $10      # Add A[i] to the product (A[i] * B[i] + A[i])
    sw $12, 0($4)           # Store the result in C[i] (store $12 at address pointed by $4)
    bne $1, $5, loop        # If i ≠ N (stored in $5), repeat the loop
    daddi $4, $4, 8         # Move to the next element in C (increment $4 by 8 bytes)
    halt                    # Stop execution
```