DUE IN CLASS

STUDENTS IDENTIFICATION:

Number:	Name:
93075	Gonçalo Azevedo
103425	Tomás Cruz
104139	Rodrigo Frias

2.1 Simple execution, without data forwarding techniques

e)

Clock cycles	18	Instructions	6	Average CPI	3.0

f)

Clock cycles	174
Instructions	61
Average CPI	2.852

Stalls:		
	- Data	101
	- Structural	0
	- Branch Taken	8

g) The branch policy adopted is Predict Not Taken: whenever a branch instruction is executed, it has already fetched the next sequential instruction. If the branch is not taken, the sequential instruction continues to be executed. However, if the branch is taken, the execution of the sequential instruction is cancelled and proceeds to execute the instruction with the address present in the branch instruction.

2.2 Application of data forwarding techniques

c)

Clock cycles	136
Instructions	61
Average CPI	2.230

Stalls:		
	- Data	63
	- Structural	9
	- Branch Taken	8

d)

$$\text{Speedup} = \frac{\text{clock cycles}_{old} \times \cancel{\text{clock rate}}}{\text{clock cycle}_{new} \times \cancel{\text{clock rate}}} = \frac{174}{136} \simeq 1.2794$$

2.3 Source code optimization: minimization of data and structural hazards

a) Attach a copy of the new assembly program. On dedicated page. (Further) →

c)

Clock cycles	118
Instructions	61
Average CPI	1.934

Stalls:		
	- Data	36
	- Structural	9
	- Branch Taken	8

d)

$$\text{Speedup} = \frac{174}{118} \simeq 1.4746$$

2.4 Source code optimization: loop unrolling

a) Attach a copy of the new assembly program. On dedicated page. (further)

c)

Clock cycles	100
Instructions	43
Average CPI	2.326

Stalls: - Data	42
- Structural	9
- Branch Taken	2

d)

$$\text{Speedup} = \frac{174}{100} = 1.74$$

2.5 Source code optimization: branch delay slot

a) Attach a copy of the new assembly program. On dedicated page. (further)

d)

Clock cycles	101
Instructions	61
Average CPI	1.656

Stalls: - Data	27
- Structural	9
- Branch Taken	0

e)

$$\text{Speedup} = \frac{174}{101} \simeq 1.7228$$

(2.2)

Table 1: 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 $12, 0($1)	F	D	X	M	W										F	D	X	M	W																					
2	dmul $12, $12, $9		F	D	X	X	X	X	X	X	X	X	M	W																											
3	dadd $9, $9, $12			F	D	D	X	X	X	X	X	X	X	M	W																										
4	daddi $5, $5, 1				F	F	D	D	D	D	D	D	D	X	M	W																									
5	daddi $1, $1, 8						F	F	F	F	F	F	F	D	X	M	W																								
6	bne $6, $5, loop													F	D	X	M	W																							
7	sw $9, mult($0)														F																										
8																																									
9																																									
10																																									
11																																									
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 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 $12, 0($1)	F	D	X	M	W								F	D	X	M	W																							
2	daddi $5, $5, 1		F	D	X	M	W																																		
3	dmul $12, $12, $9			F	D	X	X	X	X	X	X	X	M	W																											
4	daddi $1, $1, 8				F	D	X	M	W																																
5	dadd $9, $9, $12					F	D	X	X	X	X	X	X	M	W																										
6	bne $6, $5, loop						F	D	D	D	D	D	D	X	M	W																									
7	sw $9, mult($0)							F	F	F	F	F	F																												
8																																									
9																																									
10																																									
11																																									
12																																									
13																																									
14																																									
15																																									
16																																									
17																																									
18																																									
19																																									
20																																									
21																																									
22																																									
23																																									
24																																									
25																																									
26																																									
27																																									
28																																									
29																																									
30																																									

(2.4)

Table 3: 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 $12, 0($1)	F	D	X	M	W																										F	D	X	M	W					
2	lw $13, 8($1)		F	D	X	M	W																																		
3	dmul $12, $12, $9			F	D	X	X	X	X	X	X	X	M	W																											
4	lw $14, 16($1)				F	D	X	M	W																																
5	dadd $9, $9, $12					F	D	X	X	X	X	X	X	M	W																										
6	daddi $5, $5, 3						F	D	D	D	D	D	D	X	M	W																									
7	dmul $13, $13, $9							F	F	F	F	F	F	D	X	X	X	X	X	X	X	M	W																		
8	dadd $9, $9, $13													F	D	X	X	X	X	X	X	X	M	W																	
9	daddi $1, $1, 24														F	D	D	D	D	D	D	D	X	M	W																
10	dmul $14, $14, $9															F	F	F	F	F	F	F	D	X	X	X	X	X	X	X	M	W									
11	dadd $9, $9, $14																						F	D	X	X	X	X	X	X	X	M	W								
12	bne $6, $5, loop																							F	D	D	D	D	D	D	D	X	M	W							
13	sw $9, mult($0)																								F	F	F	F	F	F	F										
14																																									
15																																									
16																																									
17																																									
18																																									
19																																									
20																																									
21																																									
22																																									
23																																									
24																																									
25																																									
26																																									
27																																									
28																																									
29																																									
30																																									

(7.5)

Table 4: 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 $12, 0($1)	F	D	D	D	D	X	M	W			F	D	D	D	D	X	M	W																						
2	daddi $5, $5, 1		F	F	F	F	D	X	M	W																															
3	dmul $12, $12, $9						F	D	X	X	X	X	X	X	X	M	W																								
4	daddi $1, $1, 8							F	D	X	M	W																													
5	bne $6, $5, loop								F	D	D	X	M	W																											
6	dadd $9, $9, $12									F	F	D	X	X	X	X	M	W																							
7																																									
8																																									
9																																									
10																																									
11																																									
12																																									
13																																									
14																																									
15																																									
16																																									
17																																									
18																																									
19																																									
20																																									
21																																									
22																																									
23																																									
24																																									
25																																									
26																																									
27																																									
28																																									
29																																									
30																																									

(2 1)

Table 5: 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 $12, 0($1)	F	D	X	M	W														F	D	X	M	W																	
2	dmul $12, $12, $9		F	D	D	D	X	X	X	X	X	X	X	M	W																										
3	dadd $9, $9, $12			F	F	F	D	D	D	D	D	D	D	D	D	X	M	W																							
4	daddi $5, $5, 1						F	F	F	F	F	F	F	F	F	D	X	M	W																						
5	daddi $1, $1, 8															F	D	X	M	W																					
6	bne $6, $5, loop																F	D	D	X	M	W																			
7	sw $9, mult($0)																	F	F																						
8																																									
9																																									
10																																									
11																																									
12																																									
13																																									
14																																									
15																																									
16																																									
17																																									
18																																									
19																																									
20																																									
21																																									
22																																									
23																																									
24																																									
25																																									
26																																									
27																																									
28																																									
29																																									
30																																									

2.3 a)

```
            .data
A:          .word   1, 3, 1, 6, 4
            .word   2, 4, 3, 9, 5
mult:   .word   0

        .code
        daddi   $1, $0, A       ; *A[0]
        daddi   $5, $0, 1       ; $5 = 1 ;; i
        daddi   $6, $0, 10      ; $6 = N ;; N = 10
        lw      $9, 0($1)       ; $9 = A[0]  ;; mult
        daddi   $1, $1, 8       ;

loop:   lw      $12, 0($1)      ; $12 = A[i]
        daddi   $5, $5, 1       ; i++
        dmul    $12, $12, $9    ; $12 = $12*$9 ;; $12 = A[i]*mult
        daddi   $1, $1, 8       ;
        dadd    $9, $9, $12     ; $9 = $9 + $12  ;; mult = mult + A[i]*mult
        bne     $6, $5, loop    ; Exit loop if i == N

        sw      $9, mult($0)    ; Store result
        halt

;; Expected result: mult = f6180 (hex), 1008000 (dec)
```

2.4 a)

```
            .data
A:          .word   1, 3, 1, 6, 4
            .word   2, 4, 3, 9, 5
mult:   .word   0

        .code
        daddi   $1, $0, A       ; *A[0]
        daddi   $5, $0, 1       ; $5 = 1 ;; i
        daddi   $6, $0, 10      ; $6 = N ;; N = 10
        lw      $9, 0($1)       ; $9 = A[0]  ;; mult
        daddi   $1, $1, 8       ;

loop:   lw      $12, 0($1)      ; $12 = A[i]
        lw      $13, 8($1)      ; $13 = A[i+1]

        dmul    $12, $12, $9    ; $12 = $12*$9 ;; $12 = A[i]*mult
        lw      $14, 16($1)     ; $14 = A[i+2]
        dadd    $9, $9, $12     ; $9 = $9 + $12  ;; mult = mult + A[i]*mult

        daddi   $5, $5, 3       ; i+=3
        dmul    $13, $13, $9    ; $13 = $13*$9 ;; $13 = A[i+1]*mult
        dadd    $9, $9, $13     ; $9 = $9 + $13  ;; mult = mult + A[i+1]*mult

        daddi   $1, $1, 24      ;
        dmul    $14, $14, $9    ; $14 = $14*$9 ;; $14 = A[i+2]*mult
        dadd    $9, $9, $14     ; $9 = $9 + $14  ;; mult = mult + A[i+2]*mult

        bne     $6, $5, loop    ; Exit loop if i == N

        sw      $9, mult($0)    ; Store result
        halt

;; Expected result: mult = f6180 (hex), 1008000 (dec)
```

2.5 a)

```
            .data
A:          .word   1, 3, 1, 6, 4
            .word   2, 4, 3, 9, 5
mult:   .word   0

        .code
        daddi   $1, $0, A       ; *A[0]
        daddi   $5, $0, 1       ; $5 = 1 ;; i
        daddi   $6, $0, 10      ; $6 = N ;; N = 10
        lw      $9, 0($1)       ; $9 = A[0]  ;; mult
        daddi   $1, $1, 8       ;

loop:   lw      $12, 0($1)      ; $12 = A[i]
        daddi   $5, $5, 1       ; i++
        dmul    $12, $12, $9    ; $12 = $12*$9 ;; $12 = A[i]*mult
        daddi   $1, $1, 8       ;
        bne     $6, $5, loop    ; Exit loop if i == N
        dadd    $9, $9, $12     ; $9 = $9 + $12  ;; mult = mult • A[i]*mult

        sw      $9, mult($0)    ; Store result
        halt

;; Expected result: mult = f6180 (hex), 1008000 (dec)
```