

1. Analytical study in the pipelined processor from [2]

Timing diagram:

START OF THE PROGRAM

	Cycle 1	Cyc 2	Cyc 3	Cyc 4	Cyc 5	Cyc 6	Cyc 7	Cyc 8	Cyc 9	Cyc 10	Cyc 11	Cyc 12	Cyc 13	Cyc 14	Cyc 15	Cyc 16	Cyc 17	Cyc 18	Cyc 19	Cyc 20	Cyc 21	Cyc 22	Cyc 23	Cyc 24	Cyc 25	Cyc 26
LUI \$t0, 0x8000	Fetch	Deco	Ex	Mem	WB																					
ADDIU \$t0, \$t0, test_array		Fetch	Deco	Ex	Mem	WB																				
SUB \$t1,\$t1,\$t1			Fetch	Deco	Ex	Mem	WB																			
SUB \$t3,\$t3,\$t3				Fetch	Deco	Ex	Mem	WB																		
ADDI \$t4,\$0,1000					Fetch	Deco	Ex	Mem	WB																	
								Ex BUBBLE	Mem BUBBLE	WB BUBBLE																
Loop1:BEQ \$t3,\$t4,OutLoop1						Fetch	Deco STALL	Deco	Ex	Mem	WB															
LW \$t5,0(\$t0)							Fetch STALL	Fetch	Deco	Ex	Mem	WB														
											Ex BUBBLE	Mem BUBBLE	WB BUBBLE													
ADD \$t1,\$t1,\$t5									Fetch	Deco STALL	Deco	Ex	Mem	WB												
ADDI \$t0,\$t0,4										Fetch STALL	Fetch	Deco	Ex	Mem	WB											
ADDI \$t3,\$t3,1												Fetch	Deco	Ex	Mem	WB										
B Loop1													Fetch	Deco	Ex	Mem	WB									
OutLoop1:LUI \$t3, 0x8000													Deco	Fetch FLUSH	Deco	Ex	Mem BUBBLE	WB BUBBLE								
Loop1:BEQ \$t3,\$t4,OutLoop1															Deco BUBBLE	Deco	Ex	Mem	WB							
LW \$t5,0(\$t0)															Fetch	Deco	Ex	Mem	WB							
																Fetch	Deco	Ex	Mem	WB						
ADD \$t1,\$t1,\$t5																	Fetch	Deco	Ex	Mem	WB					
ADDI \$t0,\$t0,4																		Fetch STALL	Fetch	Deco	Ex	Mem	WB			
ADDI \$t3,\$t3,1																			Fetch	Deco	Ex	Mem	WB			
B Loop1																				Fetch	Deco	Ex	Mem	WB		
OutLoop1:LUI \$t3, 0x8000																					Fetch FLUSH	Deco BUBBLE	Ex BUBBLE	Mem BUBBLE	WB BUBBLE	

END OF THE PROGRAM

B Loop1	Fetch	Deco	Ex	Mem	WB																					
OutLoop1: LUI \$t3, 0x8000		Fetch FLUSH	Deco	BUBBLE	Ex	BUBBLE	Mem	BUBBLE	WB	BUBBLE																
Loop1: BEQ \$t3, \$t4, OutLoop1			Fetch	Deco	Ex	Mem	WB																			
LW \$t5, 0(\$t0)				Fetch FLUSH	Deco	BUBBLE	Ex	BUBBLE	Mem	BUBBLE	WB	BUBBLE														
OutLoop1: LUI \$t3, 0x8000					Fetch	Deco	Ex	Mem	WB																	
ADDIU \$t3, \$t3, Addition						Fetch	Deco	Ex	Mem	WB																
SW \$t1, 0(\$t3)							Fetch	Deco	Ex	Mem	WB															

Detailed analysis:

- The first instruction is fetched in cycle 1.
- The first iteration starts in cycle 6 and ends in cycle 18. Thus, the first iteration takes 13 cycles.
- The remaining 999 iterations take 8 cycles (for example, the second iteration starts in cycle 19 and ends in cycle 26).
- The end of the loop (`beq` and flushed `lw`) plus the three external instructions takes 5 more cycles.

Taking all this into account we have:

- Number of cycles = $5 + 13 + 999 \cdot 8 + 5 = 8015$ cycles
- Number of instructions = $5 + 1000 \cdot 6 + 4 = 6009$ instructions
- **CPI = $8015/6009 \approx 1.33$**

2. Analytical study in MIPSfpga

Timing diagram:

START OF THE PROGRAM

	Cycle 1	Cyc 2	Cyc 3	Cyc 4	Cyc 5	Cyc 6	Cyc 7	Cyc 8	Cyc 9	Cyc 10	Cyc 11	Cyc 12	Cyc 13	Cyc 14	Cyc 15	Cyc 16	Cyc 17	Cyc 18	Cyc 19	Cyc 20	Cyc 21	Cyc 22	Cyc 23	Cyc 24	Cyc 25	Cyc 26	Cyc 27	
LUI \$t0, 0x8000	Fetch	Deco	Ex	Mem	WB																							
ADDIU \$t0, \$t0, test_array		Fetch	Deco	Ex	Mem	WB																						
SUB \$t1,\$t1,\$t1			Fetch	Deco	Ex	Mem	WB																					
SUB \$t3,\$t3,\$t3				Fetch	Deco	Ex	Mem	WB																				
ADDI \$t4,\$0,1000					Fetch	Deco	Ex	Mem	WB																			
Loop1:BEQ \$t3,\$t4,OutLoop1						Fetch	Deco	Ex	Mem	WB																		
NOP							Fetch	Deco	Ex	Mem	WB																	
LW \$t5,0(\$t0)								Fetch	Deco	Ex	Mem	WB																
										Ex BUBBLE	Mem BUBBLE	WB BUBBLE																
ADD \$t1,\$t1,\$t5									Fetch	Deco	STALL	Deco	Ex	Mem	WB													
ADDI \$t0,\$t0,4										Fetch	STALL	Fetch	Deco	Ex	Mem	WB												
ADDI \$t3,\$t3,1												Fetch	Deco	Ex	Mem	WB												
B Loop1													Fetch	Deco	Ex	Mem	WB											
NOP														Fetch	Deco	Ex	Mem	WB										
Loop1:BEQ \$t3,\$t4,OutLoop1															Fetch	Deco	Ex	Mem	WB									
NOP																Fetch	Deco	Ex	Mem	WB								
LW \$t5,0(\$t0)																	Fetch	Deco	Ex	Mem	WB							
																		Fetch	Deco	Ex	Mem	WB						
																			Fetch	Deco	STALL	Deco	Ex	Mem	WB			
ADD \$t1,\$t1,\$t5																				Fetch	Deco	STALL	Fetch	Deco	Ex	Mem	WB	
ADDI \$t0,\$t0,4																					Fetch	STALL	Fetch	Deco	Ex	Mem	WB	
ADDI \$t3,\$t3,1																						Fetch	Deco	Ex	Mem	WB		
B Loop1																							Fetch	Deco	Ex	Mem	WB	
NOP																								Fetch	Deco	Ex	Mem	WB

END OF THE PROGRAM

B Loop1	Fetch	Deco	Ex	Mem	WB																						
NOP		Fetch	Deco	Ex	Mem	WB																					
Loop1: BEQ \$t3, \$t4, OutLoop1			Fetch	Deco	Ex	Mem	WB																				
NOP				Fetch	Deco	Ex	Mem	WB																			
OutLoop1: LUI \$t3, 0x8000					Fetch	Deco	Ex	Mem	WB																		
ADDIU \$t3, \$t3, Addition						Fetch	Deco	Ex	Mem	WB																	
SW \$t1, 0(\$t3)							Fetch	Deco	Ex	Mem	WB																

Detailed analysis:

- The first instruction is fetched in cycle 1.
- The first iteration starts in cycle 6 and ends in cycle 18. Thus, the first iteration takes 13 cycles.
- The remaining 999 iterations take 9 cycles (for example, the second iteration starts in cycle 19 and ends in cycle 27).
- The end of the loop (`beq` and `nop`) plus the three external instructions takes 5 more cycles.

Taking all this into account we have:

- Number of cycles = $5 + 13 + 999 \cdot 9 + 5 = 9014$ cycles
- Number of instructions = $5 + 1000 \cdot 6 + 4 = 6009$ instructions
- **CPI = $9014/6009 \approx 1.5$**

3. Empirical study in MIPSfpga

a. Original program (no optimizations)

The results provided by the performance counters are the following:

- **Number of cycles = 9047**
- **Number of instructions = $8017 - 2000$ (2 `nops` per iteration) = 6017**
- **CPI = $9047/6017 \approx 1.5$**

b. Original program (-O3 option)

The results provided by the performance counters are the following:

- **Number of cycles = 8046**
- **Number of instructions = $7011 - 1000 = 6011$**
- **CPI = $8046 / 6011 \approx 1.34$**

Note that the compiler fills the delay slot of the `b` instruction, but not the delay slot of the `beq`.

c. Reordered program

New program:

```
".set noreorder;"
"LUI      $t0, 0x8000;"
"ADDIU    $t0, $t0, V;"
"SUB      $t1, $t1, $t1;"
```

```

"SUB $t3,$t3,$t3;"
"ADDI $t4,$0,1000;"
"Loop1:    BEQ $t3,$t4,OutLoop1;"
    "ADDI $t3,$t3,1;"
    "LW $t5,0($t0);"
    "ADDI $t0,$t0,4;"
    "B Loop1;"
    "ADD $t1,$t1,$t5;"
"OutLoop1: LUI    $t3, 0x8000;"
"ADDIU $t3, $t3, Addition;"
"SW $t1,0($t3);"
".set reorder;"

```

The results provided by the performance counters are the following:

- **Number of cycles = 6043**
- **Number of instructions = 6016**
- **CPI = 6043 / 6016 \approx 1**