23

10

2.3

STUDENTS IDENTIFICATION:

Number:	**Name:**
Maria Ramos	ist1105875
Guilherme Campos	ist1106906
Enzo Nunes	ist1106336

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) By analysing the program execution, we conclude that the simulator adopts a “Predict Not Taken” branch prediction policy. This is demonstrated by the fact that, upon encountering the bne instruction, the simulator proceeds to execute the next sequential instruction outside the loop (the halt), under the assumption that the branch will not be taken. The halt instruction is cut off after the decode stage of the bne instruction, as the processor initially predicts the branch as not taken, but once it realizes the branch should be taken, the halt is interrupted at the fetch stage.

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) The speedup is calculated by comparing the old execution time to the new execution time. The formula for speedup is: Speedup = Old Time / New Time.
Since execution time is a product of the number of cycles and the cycle time, we can expand the formula to: Speedup = #Old Cycles x Old Cycle Time / #New cycle x New Cycle Time.
Since the old and new cycle times are the same, the cycle time cancels out, leaving us with: Speedup = #Old Cycles / #New Cycles = 377 / 297 = 1.2693.

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	1.5

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

d) Using the same approach as before, the speedup formula simplifies to: #Old Cycles / #New Cycles = 377 / 249 = 1.514.

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)

Using the same approach as before, the speedup formula simplifies to: #Old Cycles / #New Cycles = 377 / 153 = 2.464.

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	32
	- Structural	16
	- Branch Taken	0

e)

Using the same approach as before, the speedup formula simplifies to: #Old Cycles / #New Cycles = 377 / 218 = 1.729.