Computer Organization
Third Lab Assignment: Instruction Level Parallelism

December 7th 2018

Baltasar Dinis 89416; Artur Fortunato 86388; João Oliveira 86441

November 30, 2018

Contents

2 Answers

2.1 Simple Execution, no data forwarding

We observe that the number of clock cycles is 22. Given that there are 11 instructions in the loop (including the control), the CPI is:

$$CPI = \frac{22}{11} = 2$$

The program only has backward branches. In all of the executions, the CPU predicts that the branch is not taken - which leads us to conclude that it uses a Backward Branch Not Taken policy. We cannot conclude anything with regards to the Forward Branch policy. However, it is costumary to have (in terms of static prediction techniques) a Forward Branch Not Taken paired with a Backward Branch Taken policy; we can then extrapolate that the CPU has a generalistic Branch Not Taken Policy.

Global Results

- Cycles: 169
- Instructions: 87
- CPI: 1.943
- Stalls:
  - Data: 72
  - Structural: 0
  - Branch Taken: 6

2.2 Simple Execution, with data forwarding

We observe that the number of clock cycles is 22. Given that there are 11 instructions in the loop (including the control), the CPI is:

$$CPI = \frac{20}{11} = 1.81818181818$$

The speedup is:

$$speedup = \frac{22}{20} = 1.1$$

Global Results

- Cycles: 132
- Instructions: 87
- CPI: 1.517
  - Data: 28
  - Structural: 7
  - Branch Taken: 6

2.3 Source Code Optimization: minimization of data and structural hazards

We observe that the number of clock cycles in a loop cycle is 16 . We manage this by interweaving instructions without data dependencies between them. Given that there are 11 instructions in the loop (including the control), the CPI is:

$$CPI = \frac{16}{11} = 1.45454545455$$

The speedup is:

$$speedup = \frac{22}{16} = 1.375$$

Global Results

- Cycles: 104
- Instructions: 87
- CPI: 1.195
  - Data: 0
  - Structural: 7
  - Branch Taken: 6

2.4 Source Code Optimization: minimization of data and structural hazards

We observe that the number of clock cycles is 38. We manage this by repeating the body of the cycle and moving the control to the begining of the loop. Given that there are 32 instructions in the loop (including the control), the CPI is:

$$CPI = \frac{38}{32} = 1.1875$$

The speedup is:

$$speedup = \frac{16}{11}\frac{38}{32} = \frac{19}{11} = 1.7272727272727272$$

Global Results

- Cycles: 100
- Instructions: 85
- CPI: 1.176
  - Data: 1
  - Structural: 7
  - Branch Taken: 3

2.5 Source Code Optimization: branch delay slot

We observe that the number of clock cycles is 38. We manage this by repeating the body of the cycle and moving the control to the begining of the loop. Given that there are 32 instructions in the loop (including the control), the CPI is:

$$CPI = \frac{16}{11} = 1.45454545455$$

The speedup is:

$$speedup = \frac{22}{16} = 1.375$$

Global Results

- Cycles: 98
- Instructions: 87
- CPI: 1.126
  - Data: 0
  - Structural: 7
  - Branch Taken: 0