**CPRE 381- Intro to Computer Organization & Implementation**

**HW6**

**Due Date: March 31, 2017**

Ningyuan Zhang

Section A

1. Problem 4.2; Patterson-Hennessy text-5th edition, Page 357. **[10 points]**

1). The existing blocks that can be used are:

Instruction Memory

ALU

Data Memory

Register File: all ports

2). No functional blocks are needed.

3). No new signals are needed.

1. Problem 4.3; Patterson-Hennessy text-5th edition, Page 358. **[10 points]**

1). the clock cycle time without improvement = I-Mem + Regs + Mux + ALU + D-Mem+ Mux

= 400 ps + 200 ps + 30 ps + 120 ps + 350ps + 30ps

= 1130 ps

ALU=120 ps+300 ps=420 ps ( Improvement latency is +300 for ALU )

the clock cycle time with improvement = I-Mem + Regs + Mux + ALU + D-Mem+ Mux

= 400 ps + 200 ps + 30 ps + 420 ps + 350ps + 30ps

= 1430 ps

2). There is no speedup in actual path. We assume, 1000 instructions at 1130 ps, so running time 1130000 ps. We have 950 instructions at 1430 ps running time is 1358500 ps. So, it is decreased at 225800 ps, no speedup is achieved.

3). Let increasing count to 1 million

Cost / (1 / execution time), where execution time = performance => 10000000 \* 1130 => 1130000000 => 0.00113 s

Cost performance = 3890 / ( 1 / 0.00113 ) = 4.4

a). cost 3890 + 600 = 4490

1000000 + 0.95 \* 1430 = 13585000000 ps => 0.0013585 s

4490 / ( 1 / 0.0013585 ) = 6.1

b). cost 3890 – 400 = 3490

10000000 \* 1130 = 1130000000 ps = 0.00113 s

3490 / ( 1 / 0.00113 ) = 3.94

To compare cost/performance these are no improvement in these comparsion.

1. Consider the following program sequence:

add r1, r2, r9

lw r3, 10(r31)

sub r5, r1, r3

add r10, r3, r5

What data hazards exist in this program? How will they be resolved? Are there any pipeline stalls in this program execution? If yes, can you reorder instructions to eliminate them? **[10 points]**

The main implication is that instruction 2 is not allowed to execute out-of-order in relation to instruction 1. There is a name-dependence between these instructions and if instruction 2 completes before instruction 1 there will be a WAW-hazard. Instruction 2 is stalled in the DI stage and the table must be modified:

Instruction IF DI EX WB

0 add r1, r2, r9 0 1 2 3

1 lw r3, 10(r31) 1 2 4 9

2 sub r5, r1, r3 2 3 10 11

3 add r10, r3, r5 3 4 11 12

Below is the re-order of instructions in order to eliminate pipeline stalls:

lw r3, 10(r31)

sub r5, r1, r3

add r1, r2, r9

add r10, r3, r5

1. Describe an RTL level implementation of beq r1, r2, target (or draw it pictorially as in textbook). Start with a version that resolves the branch in EX stage. Modify it to resolve the branch in ID stage. With 25% branch frequency, what is the CPI for branches (a) with EX resolution (b) with ID resolution? **[10 points]**

Beq r1, r2, target

|  |  |  |  |
| --- | --- | --- | --- |
| F | D | E | W |
| Ignore | F | F | F | D | E | W |

a). CPI with EX resolution = 1 + 25% \* 3 = 1 + 0.75 = 1.75

b). CPI with ID resolution = 1 + 75% \* 3 = 1 + 2.25 = 3.25