CSCI 4593: Homework 4

4.4.2

Add = 70ps

Sign extend = 15ps

Shift left 2 = 10ps

I-Mem = 200ps

Mux = 20ps

For an unconditional branch we need to first fetch the instruction from instruction memory, sign-extend the lower 16 bits of the instruction, shift those left by 2, add the result to create the branch target, and finally use a mux to replace the PC with the branch target.

Cycle time = 200ps + 15ps + 10ps + 70ps + 20ps = 315ps

Using the information from Problem 4.4.2, what is the clock cycles time if the only types of instructions we need to support are ALU instructions (ADD, AND, etc.)?

I-Mem = 200ps

Regs = 90ps

Mux = 20ps

ALU = 90ps

In order to support ALU (R-type) instructions, we need instruction memory, registers, 2 muxes, and an ALU.

Cycle time = 200ps + 90ps + 2×20ps + 90ps = 400ps

4.7.4

Instruction: 1010 1100 0110 0010 0000 0000 0001 0100

op (Inst[31:26])	rs (Inst[25:21])	rt (Inst[20:16])	address (Inst[15:0])
101011	00011	00010	0000000000010100
sw	3	2	20

Write register (RegDst = X)	ALU (ALUSrc = 1)	Write data (MemtoReg = X)	Branch (Branch = 0)	Jump (Jump = 0)
value = 0 (Inst[15-11]) or 2 (Inst[20-16])	value = 20 (Inst[15-0])	value = ALU result or Data memory read data	value = PC + 4	value = PC + 4

4.16.1

always-taken will hit 3/5 or be 60% accurate

always-not-take will hit 2/5 or be 40% accurate

4.16.2

Step	Predictor state	Branch state	result
1	not-taken	taken	miss
2	not-taken	not-taken	hit
3	not-taken	taken	miss
4	not-taken	taken	miss

The accuracy is 1/4 or 25%

4.16.3

Step	Predictor state	Branch state	result
1	not-taken (0)	taken	miss
2	not-taken (1)	not-taken	hit
3	not-taken (0)	taken	miss
4	not-taken (1)	taken	miss
5	taken (3)	not-taken	miss
6	not-taken (2)	taken	miss
7	taken (3)	not-taken	miss
8	not-taken (2)	taken	miss
9	taken (3)	taken	hit
10	taken (4)	not-taken	miss
11	taken (3)	taken	hit
12	taken (4)	not-taken	miss
13	taken (3)	taken	hit
14	taken (4)	taken	hit
15	taken (4)	not-taken	miss

It looks like the long term accuracy will be 3/5 or 60% when the predictor moves into the predict taken states. It will always hit on the taken branches and always miss on the not-taken branches. It will remain in the predict taken states move back and forth.