Rodney Wototn
Shudong Hao
CS 382
Homework 3
11/12/21

1.
    a. If the Branch wire were stuck at zero, the instruction that would fail to operate would be the CBZ instructions.
    b. If the ALUSrc wire were stuck at zero, the instructions that would fail to operate would be the LDUR and STUR instructions.
    c. If the RegWrite wire were stuck at zero, the instructions that would fail to operate would be the LDUR and R-format instructions.
2. The necessary changes that would need to be made would be that the two LDUR instructions need to run at the same time in order to set them without an extra swap. This needs to be done to set Reg[Rd] equal to Reg[Rn]. Rd and Rn will load into each other at the same time, but for this to be done, we need to add another ALU and Data Memory block.
3.
    a.
        i. Clock Cycle with improvement: 300+250+150+300+200 = **1100ps**
        ii. Clock Cycle without new improvement:(300+250+150+300+200+200)*.8 = **1040ps**
    b. The speedup achieved is 60 ps improvement. Which is obtained by 1100ps we calculated minus the 1040ps from before improvements were made.
    c. The slowest the new ALU can be while still improving performance is 1100/.8 = 1375 ps and 1375 - 1100 = **275ps**
4.
    a.
        i. Pipeline Clock Cycle Time: **300 ps**
        ii. Non-Pipeline Clock Cycle Time: **1100 ps**
    b.
        i. Pipeline Clock Cycle Time Latency: **300 ps**
        ii. Non-Pipeline Clock Cycle Time Latency : **220 ps**
    c. I would split the MEM into two stages. This is because it is the stage with the largest latency, and therefore if we want better clock time, it should be cut. The old clock time would have been 300ps, but the new clock time would be **250ps**

5. The MEM stage of LDUR forwards X21 to the ID stage ofADD. The it forwards X22 from MEM stage of ADD to the EX stage of SUB

LDUR	IF	ID	EX	MEM	WB				
LDUR		IF	ID	EX	MEM	WB			
NOP			IF	ID	EX	MEM	WB		
ADD				IF	ID	EX	MEM	WB	
SUB					IF	ID	EX	MEM	WB

6.

a. LDUR x1, [x6, #8]

NOP

NOP

ADD x0, x1, x0

NOP

NOP

STUR x0, [x10, #4]

LDUR x2, [x6, #12]

NOP

NOP

SUB x3, x0, x2

NOP

NOP

STUR x3, [x8, #24]

CBZ x2, #40

b. LDUR x1, [x6, #8]

LDUR x2, [x6, #12]

```
NOP

ADD x0, x1, x0

CBZ x2, 40

NOP

STUR x0, [x10, #4]

SUB x3, x0, x2

NOP

STUR x3, [x8, #24]
```