

CS383 - Homework 3 - "I pledge my honor that I have abided by the Stevens Honor System."**Problem 1—****A. Which instructions fail to operate correctly if the MemToReg wire is stuck at 0?**

When MemToReg stuck at 0, it is unable to read from memory...

LDUR Instructions will fail because they must read value from memory

B. Which instructions fail to operate correctly if the ALUSrc wire is stuck at 0?

While stuck at 0, you cannot get the value from any instructions, only the available registers...

I-instructions fail including STUR, LDUR, ORR

C. Which instructions fail to operate correctly if the RegWrite wire is stuck at 0?

When stuck at 0, no register can be written in...

Instructions that fail include arithmetic instructions & LDUR because you cannot write the value in

Problem 2—**A. What is the clock cycle time with and without this improvement?****B. What is the speedup achieved by adding this improvement?****C. What is the slowest the new ALU can be and still result in improved performance?****Problem 3—****A. What is the clock cycle time in a pipelined and non-pipelined processor?**

Pipeline needs to account for the slowest and can use the ID that = **350ps**

Non-Pipeline counts all stages: $250+350+150+300+200 = 1250ps$

B. What is the total latency of an LDUR instruction in a pipelined and non-pipelined processor?

Pipeline: LDUR = 20% which would account for 5 cycles while the ID is 350ps....

$$5 \times 350ps = 1750ps$$

Non-Pipeline: $250+350+150+300+200 = 1250ps$

C. If we can split one stage of the pipelined datapath into two new stages, each with half the latency of the original stage, which stage would you split and what is the new clock cycle time of the processor?

largest stage = 350ps

$$350/2 = 175ps$$

now longest is 300ps so new clock cycle time = **300ps**

D. Assuming there are no stalls or hazards, what is the utilization of the data memory?

data memory is only being used currently by LDUR and STUR which account for 20% and 15%

$$\text{data memory utilization} = 20 + 15 = 35\%$$

Problem 4— Assume that perfect branch prediction is used (no stalls due to control hazards) and that the pipeline has full forwarding support. Show a pipeline execution diagram for the first two iterations of this loop based on Section 4.7 of the textbook. Making a table is not required. Just align stages of difference instructions that take place concurrently.

LDUR and LDUR forward data to SUB

SUB forwards data to ADD

ADD forwards data to CBNZ

LDUR and LDUR forward data to SUB

SUB forwards data to ADD

ADD forwards data to CBNZ

Problem 5—

A. If there is no forwarding or hazard detection, insert NOPs to ensure correct execution.

```

ADD x5, x2, x1      // line 1
NOP
NOP
LDUR x3, [x5, #4]   // line 2
LDUR x2, [x2, #0]   // line 3
NOP
ORR x3, x5, x3      // line 4
NOP
NOP
STUR x3, [x5, #0]   // line 5

```

B. Now, change and/or rearrange the code to minimize the number of NOPs needed. You can assume register X7 can be used to hold temporary values in your modified code.

line 2 depends on line 1 for x5 // line 4 depends on line 2 for x3 // line 5 depends on line 4 for x3

only no depended on line is line 3... so we could potentially move it around without changing code..

moved it up to place of first NOP Not sure if this is legal because no x7

```

ADD x5, x2, x1
LDUR x2, [x2, #0]
NOP
LDUR x3, [x5, #4]
NOP
ORR x3, x5, x3
NOP
NOP
STUR x3, [x5, #0]

```

Problem 6—

- A. What is the accuracy of always-taken and always-not-taken predictors for this sequence of branch outcomes?**

SEQ: T, NT, T, NT, NT

ALWAYS-TAKEN: T, T, T, T, T $\Rightarrow 2/5 = 0.4 = \mathbf{40\% \textit{ accurate}}$

ALWAYS-NOT-TAKEN: NT, NT, NT, NT, NT $\Rightarrow 3/5 = 0.6 = \mathbf{60\% \textit{ accurate}}$

- B. What is the accuracy of the 2-bit predictor for this pattern, assuming that the predictor starts off in the bottom left state from Figure 4.62 (predict not taken)? Show the state of the predictor at each step.**

SEQ: T, NT, T, NT, NT

using more accurate ALWAYS-NOT prediction

SEQ STATES: 00 - 01 - 10 - 11 - 10 - 10

60%