Instruction Set for the problems below, unless otherwise specified, assume that the instruction set includes: (i) R-type instructions (ADD, SUB, AND, ORR), (ii) LDUR, (iii) STUR, (iv) CBZ, and (v) B.

Problem 1 (15 points) When silicon chips are fabricated, defects in materials (e.g., silicon) and manufacturing errors can result in defective circuits. A very common defect is for one signal wire to get broken and always register a logical 0. This is often called a stuck-at-0 fault. Answer the following questions based on **Section 4.4** of the textbook (see slide 19 in Chapter 4 on Canvas).

(a) [5] Which instructions fail to operate correctly if the MemToReg wire is stuck at 0?
   (i) Write data in the Register block will always get an address instead of the data stored in the address. STUR will fail.

(b) [5] Which instructions fail to operate correctly if the ALUSrc wire is stuck at 0?
   (i) Sign Extension will fail, so adding, subtracting, multiplying, and dividing will fail.

(c) [5] Which instructions fail to operate correctly if the Reg2Write wire is stuck at 0?
   (i) STUR and LDUR will fail

Problem 2 (15 points) Consider the addition of a multiplier to the CPU shown in Figure 4.23 (slide 19 in Chapter 4 on Canvas). This addition will add 300 ps to the latency of the ALU, but will reduce the number of instructions by 10% (because there will no longer be a need to emulate the multiply instruction). Answer the following questions based on **Section 4.4** of the textbook and the following table for the latencies of the stages of the pipeline.

IF	ID	EX	MEM	WB
250 ps	350 ps	150 ps	300 ps	200 ps

(a) [5] What is the clock cycle time with and without this improvement?
   (i) Without: 1500 ps; With: 2250 ps

(b) [5] What is the speedup achieved by adding this improvement?
   (i) The speedup is achieved by slowing down the clock rate to lower the total number of instructions,

(c) [5] What is the slowest the new ALU can be and still result in improved performance?
   (i) 450 ps

Problem 3 (30 points) In this exercise, we examine how pipelining affects the clock cycle time of the processor. Questions in this problem assume that individual stages of the datapath have the latencies shown in Problem 2 above. Also, assume that instructions executed by the processor are broken down as follows:

ALU/Logic	Jump/Branch	LDUR	STUR
45%	20%	20%	15%

Answer the following questions based on **Section 4.5** of the textbook.

(a) [5] What is the clock cycle time in a pipelined and non-pipelined processor?
   (i) Pipelined: 400 ps; Non-Pipelined: 1200 ps

(b) [10] What is the total latency of an LDUR instruction in a pipelined and non-pipelined processor?
   (i) Total Instructions = I, CPI = 1
   Pipelined: 1*(0.2*I)*400 ps = 80I ps; Non-Pipelined: 1*(0.2*I)*1200 ps = 250I ps

(c) [10] 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?

(i) Split the ID stage into two stages, making the new clock cycle time 1500 ps for a pipelined process

(d) [5] Assuming there are no stalls or hazards, what is the utilization of the data memory?

(i) To write data or read data to be written

Problem 4 (10 points) Consider the following loop.

```
LOOP:LDUR X10, [X1, #0]
     LDUR X11, [X1, #8]
     SUB X13, X11, X10
     ADD X12, X12, X13
     SUBI X1, X1, #162
     CBNZ X12, LOOP
```

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.

```
IF ED EX MEM WB
   IF ED EX  MEM WB
      IF ED  EX  MEM WB
         IF  ED  EX  MEM WB
             IF  ED  EX  MEM WB
                 IF  ED  EX  MEM WB
                     IF  ED  EX  MEM WB
                         IF  ED  EX  MEM WB
                             IF  ED  EX  MEM WB
                                 IF  ED  EX MEM WB
                                     IF  ED EX MEM WB
                                         IF ED EX MEM WB
```

Problem 5 (15 points) Questions in this problem refer to the following sequence of instructions, and assume that it is executed on a five-stage pipelined datapath:

ADD X5, X2, X1
LDUR X3, [X5, #4]
LDUR X2, [X2, #0]
ORR X3, X5, X3
STUR X3, [X5, #0]

Answer the following questions based on **Section 4.7** of the textbook.

(a) [5] If there is no forwarding or hazard detection, insert NOPs to ensure correct execution.

(i) **ADD X5, X2, X1**
Nop
Nop
LDUR X3, [X5, #4]
LDUR X2, [X2, #0]
ORR X3, X5, X3
Nop
Nop

```
STUR X3, [X5, #0]
```

(b) [10] 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.

(i)
```
ADD X5, X2, X1
LDUR X2, [X2, #0]
LDUR X7, [X5, #4]
Nop
Nop
ORR X3, X5, X7
STUR X3, [X5, #0]
```

Problem 6 (15 points) This problem examines the accuracy of various branch predictors for the following repeating pattern of branch outcomes: T, NT, T, NT, NT. Answer the following questions based on **Section 4.8** of the textbook.

(a) [5] What is the accuracy of always-taken and always-not-taken predictors for this sequence of branch outcomes?

(i) Always Taken: ⅖ = 40%; Always Not-Taken: ⅗ = 60%

(b) [10] 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.

(i) Assuming that the bottom left is encoded 11, bottom right 10, top right 01, and top left 00:
Predictor: 11 (start, miss), 10 (hit), 01 (hit), 00 (miss), 01 (miss)
⅖ = 40% accuracy.