Name :

CS / ECE 6810 — Midterm Exam — March 5th 2018

Notes: This is an open notes and open book exam. If necessary, make reasonable assumptions and clearly state them. The only clarifications you may ask for during the exam are definitions of terms. You may use calculators. Laptops are allowed if you want to browse through class material (textbook CD, notes, and slides), but you are **NOT** allowed to access other websites. Complete your answers in the space provided (including the back-side of each page). Confirm that you have 7 questions on 6 pages, followed by a blank page. Turn in your answer sheets before 1:10PM.

1. **Processor Performance.** Two different software implementations (programs) are proposed for a particular scientific function that are called *prog-A* and *prog-B*. The programs are executed on *comp-1*, a scalar RISC processor, one at a time to collect the following statistics. The total numbers of executed instructions for *prog-A* and *prog-B* on *comp-1* are 1000 and 2000, respectively.

comp-1			
Instruction Type	Cycles	*prog-A*	*prog-B*
ADD	2	30%	60%
MULT	8	40%	25%
DIV	40	15%	5%
Branch	1	15%	10%

   (a) Assuming that *comp-1* operates at 2GHz, find the execution times of the programs and identify which one runs faster. **(10 points)**

   (b) A newer version of the processor is *comp-2* that provides a 6-cycle fused multiply and add (FMA) instruction in addition to the instructions supported by *comp-1* (with the same number of cycles per instruction). Assuming that *comp-2* operates at 1.8GHz, find the execution times of the programs and identify which one runs faster. **(10 points)**

      i. Every ADD in *prog-A* is followed by a MULT, which can be replaced by an FMA.

      ii. Every MULT in *prog-B* follows an ADD, which can be replaced by an FMA.

2. **Instruction Set Architecture.** The initial values for parts of the register file and main memory used in an 8-bit processor are shown in the table below.

Register File		**Main Memory**	
Address	*Value*	*Address*	*Value*
R1	100	200	250
R2	150	250	200
R3	200	251	250
R4	250	350	100
R5	300	400	300

Compute the effective address and final result for each of the following instructions. Register value changes are considered when moving from one instruction to another. All of the instructions are executed serially. **(20 points)**

```
ADD R5, R1, R2
LD R5, +(R5)
LD R4, 100(R5)
LD R1, @(R3)
ADD R4, R4, R1
LD R2, (R3+R1)
```

3. **Pipelining.** Consider an un-pipelined processor where it takes 20ns to go through the circuits and 1ns for the latch overhead. Assume that the point of production and point of consumption in the un-pipelined processor are separated by 10ns. Assume that one fourth of the instructions do not introduce a data hazard and the rest depend on their preceding instruction. What is the throughput—in million instructions per second (MIPS)—for this processor with (i) an un-pipelined architecture, and (ii) a 10-stage pipeline? **(10 points)**

4. **Data Hazards.** Identify all the data hazards in the following code. **(10 points)**

```
SD R1, 0(R3)
ADD R1, R2, R4
LD R2, 0(R5)
ADD R3, R1, R6
```

5. **Branch Prediction.** Consider executing a scientific application on a scalar pipelined processor with a local branch predictor.

   (a) Find the average number of stall cycles per instruction caused by branches in the pipeline. Assume that the branch misprediction penalty is 16 cycles, branch predictor accuracy is 80%, and branch target buffer hit rate is 60%. Also, every fourth instruction of the application is a branch and 75% of branches are actually taken. **(10 points)**

   (b) The branch predictor employs 12 bits of the program counter (PC) for indexing into 16-bit history registers. Moreover, 2-bit saturating counters are used for individual predictors. What is the total capacity of the branch prediction system without the target buffer? **(10 points)**

6. **Register Renaming.** Consider an out-of-order processor that has 8 logical registers (denoted by R) and 10 physical registers (denoted by P). On power up, assume that logical register R1 is mapped to physical register P1, R2 is mapped to P2, and so on; the following program starts executing:

```
SD R1, 0(R1)
ADD R4, R3, R1
SD R4, 8(R3)
LD R1, 16(R4)
SUB R3, R4, R1
ADD R1, R2, R3
SD R1, 8(R1)
```

Show the renamed version of this code. **(10 points)**

7. **Static Instruction Scheduling.** Consider the following code including NOP instructions to produce the necessary stall cycles between producers and consumers. The following delays are necessary between dependent instructions:

   (a) Load feeding any instruction: 1 stall cycle
   (b) FP ADD feeding any instruction : 2 stall cycles
   (c) FP MULT feeding store: 4 stall cycles
   (d) Integer ADD feeding any instruction: 0 stall cycles

```
LD F1, 0(R1)
LD F2, 0(R2)
NOP
ADD F3, F1, F2
NOP
NOP
MULT F4, F3, F5
NOP
NOP
NOP
NOP
SD F4, 0(R3)
ADD F4, F3, F4
ADD F3, F2, F1
ADDI R3, R3, #-8
```

Show an optimized schedule with minimum number of NOPs for this code by reordering instructions and (if necessary) modifying immediate values. Notice that the optimized code MUST result in the same final values for registers and memory as those in the original code. **(10 points)**