Brown University
School of Engineering
EN1640 Design of Computing Systems
Professor Sherief Reda
HW 02. 115 points. Due Date: Wednesday April 6th in class.

1. [10 points]
   a. [2 points] Assume a single-cycle processor design with a clock period *S*. If the design is pipelined to *n* stages, express the clock period and frequency as a function of *n*. Assume that the delay overhead of adding a pipeline register is *h*.
   b. [8 points] If the single-cycle design has a total capacitance of *C*, what is the dynamic power consumption of the pipelined design as a function of *n*? Assume that each pipeline register adds an additional capacitance of value *a*, and that the operating voltage of the pipeline design is equal to *V*.

2. [10 points]

   a. [5 points] You are going to enhance a computer, and there are two possible improvements: either make multiply instructions run four times faster than before, or make memory access instructions run two times faster than before. You repeatedly run a program that takes 100 seconds to execute. Of this time, 20% is used for multiplication, 50% for memory access instructions, and 30% for other tasks. What is speedup if you improve only multiplication? What will the speedup be if you improve only memory access? What is the speedup if both improvements are made?

   b. [5 points] You are going to change the program described in (a) so that the percentages are not 20%, 50%, and 30% anymore. Assuming that none of the new percentages is 0, what sort of program would result in a tie (with regard to speedup) between the two individual improvements? Provide both a formula and some examples.

3. [10 points] Assume the following datapath latencies for the individual stages.
   IF: 250 ps
   ID: 350 ps
   EX: 150 ps
   MEM: 300 ps
   WB: 200 ps

   a. [2 points] What is the clock cycle time in a pipelined and non-pipelined processor?
   b. [3 points] What is the total latency of an LDR instruction in a pipelined and non-pipelined processor? Ignore the overhead of the pipeline registers.
   c. [5 points] 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?

4. [15 points]

```
I1:     LDR     R1, [R1, #0]
I2:     AND     R1, R1, R2
I3:     LDR     R2, [R1, #0]
I4:     LDR     R1, [R3, #0]
```

a. [5 points] Find all data dependences in this instruction sequence.
b. [5 points] Find all hazards in this instruction sequence for a 5-stage pipeline with and then without forwarding
c. [5 points] To reduce clock cycle time, we are considering a split of the MEM stage into two stages, where reading completes in the second MEM stage. Repeat parts (a) and (b) for this 6-stage pipeline

5. [12 points] The pipelined ARM processor is running the following program.

```
ADD   R0, R0, R1
SUB   R7, R2, R3
AND   R8, R6, R7
OR    R9, R4, R5
ADD   R10, R8, R9
```

a. [3 points] Which registers are being written, and which are being read in the sixth clock cycle?
b. [3 points] Identify all Read After Write (RAW) hazards.
c. [3 points] Modify the program to eliminate the hazards by inserting nop instructions in the program to eliminate any RAW hazards. How many cycles does it need to complete the modified program?
d. [3 points] Reorder the instructions in the program, while maintaining correctness, to eliminate the RAW hazards. How many cycles does it need to complete the modified program?

6. [25 points] The following program is executed on the classical 5-stage pipeline design. The program searches an area of memory and counts the number of times a memory work is equal to a key word:

```
SEARCH:   LDR   R5, [R3, #0]      /I1 Load item
          CMP   R5, R2            /I2 Compare with Key
          BNE   NOMATCH           /I3 Check for match
          ADD   R1, R1, #1        /I4 Count for matches
NOMATCH:  ADD   R3, R3, #4        /I5 Next item
          CMP   R4, R3            /I6 Did we finish all items?
          BNE   SEARCH            /I6 Continue until all items
```

Branches are predicted untaken always and are taken in EX if needed. Hardware support for branches is included in all cases. Consider several possible pipeline interlock designs

for data hazards and answer the following questions for each loop iteration, except for the last iteration.

a. [5 points] Assume first that the pipeline has no forwarding unit and no hazard detection units. Values are not forwarded inside the register file. Re-write the code by inserting NOPs wherever needed so that the code will execute correctly.
b. [5 points] Next, assume no forwarding at all, but a hazard detection unit that stalls instructions in ID to avoid hazards. How many clock cycles does it take to execute one iteration of the loop (1) on a match and (2) on a no match?
c. [5 points] Next, assume internal register forwarding and a hazard detection unit that stalls instructions in ID to avoid hazards. How many cycles does it take to execute one iteration of the loop (1) on a match and (2) on a no match?
d. [5 points] Next, assume full forwarding and a hazard detection unit that stalls instructions in ID to avoid hazards. How many clocks does it take to execute one iteration of the loop (1) on a match and (2) on a no match?
e. [5 points] A basic block is a sequence of instructions with one entry point and one exit point. Identify basic blocks (using instruction numbers) in the code. Is it safe for the compiler to move I5 up across the BNE. Does this help? How? Is it always safe to move instructions across basic block boundaries?

7. [13 points] The importance of having a good branch predictor depends on how often conditional branches are executed. Together with branch predictor accuracy, this will determine how much time is spent stalling due to mispredicted branches. In this exercise, assume that the breakdown of dynamic instructions into various instruction categories is as follows:

R-Type: 40% BEQ: 25% JMP: 5% LDR: 25% STR: 5%

Also assume the following branch predictor accuracies:

Always-Taken: 45% Always-Not-taken: 55% 2-Bit: 85%

a. [4 points] Stall cycles due to mispredicted branches increase the CPI. What is the extra CPI due to mispredicted branches with the always-taken predictor? Assume that branch outcomes are determined in the EX stage, that there are no data hazards, and that the no delay slot are used.
b. [4 points] Repeat (a) for the "Always-not-taken" predictor.
c. [5 points] Repeat (a) for the 2-bit predictor.

8. [20 points] Assume the following repeating pattern (e.g., in a loop) of branch outcomes:
T, NT, T, T, NT

a. [5 points] What is the accuracy of always-take and always-not-taken predictors for this sequence of branch outcomes?
b. [5 points] What is the accuracy of the two-bit predictor for the first four branches in this pattern, assuming that the predictor starts off in (predict not taken) state (bottom-left dark-grey state in the lecture slides)?
c. [5 points] What is the accuracy of the two-bit predictor if this pattern is repeated forever?

d. [5 points] Design a predictor that would achieve a perfect accuracy if this pattern is repeated forever. Your predictor should be a sequential circuit with one output that provides a prediction (1 for taken, 0 for not taken) and no inputs other than the clock and the control signal that indicates that the instruction is a conditional branch.