CSCI 202 Computer Organization II
Instructor: Gedare Bloom
Homework 1 (25 points)

Due: Friday, March 1

Do this assignment by yourself. Provide detailed calculation and justifications for your numerical answers. Always watch the course website for updates on the assignments.

Question 1 (10 points)

This is a three-part question about critical path calculation. Consider a simple single-cycle implementation of MIPS ISA with the datapath shown in **Figure 1**. The operation times for the major functional components for this machine are as follows:

Component	Latency
ALU	9 ns
Adder	7 ns
ALU Control Unit	1 ns
Shifter	2 ns
Control Unit/ROM	3 ns
Sign/Zero Extender	2 ns
2-1 Multiplexor	1 ns
Memory (R/W) (Insn/Data)	10 ns
PC Register (read)	1 ns
PC Register (write)	1 ns
Register file (read)	4 ns
Register file (write)	6 ns
Logic (1 or more levels of gates)	1 ns

In this implementation, the clock cycle is determined by the longest possible path in the machine. The critical paths for the different instruction types that need to be considered are: R-format, Load-word, and store-word. All instructions have the same instruction fetch and decode steps. The basic register transfer of the instructions are:

```
Fetch/Decode:
     Instruction <- IMEM[PC];
R-type:
     R[rd] <- R[rs] op R[rt]; PC <- PC + 4;
load:
     R[rt] <- DMEM[ R[rs] + signext(offset)]; PC <- PC +4;
store:
     DMEM[ R[rs] + signext(offset)] <- R[Rt]; PC <- PC +4;
```

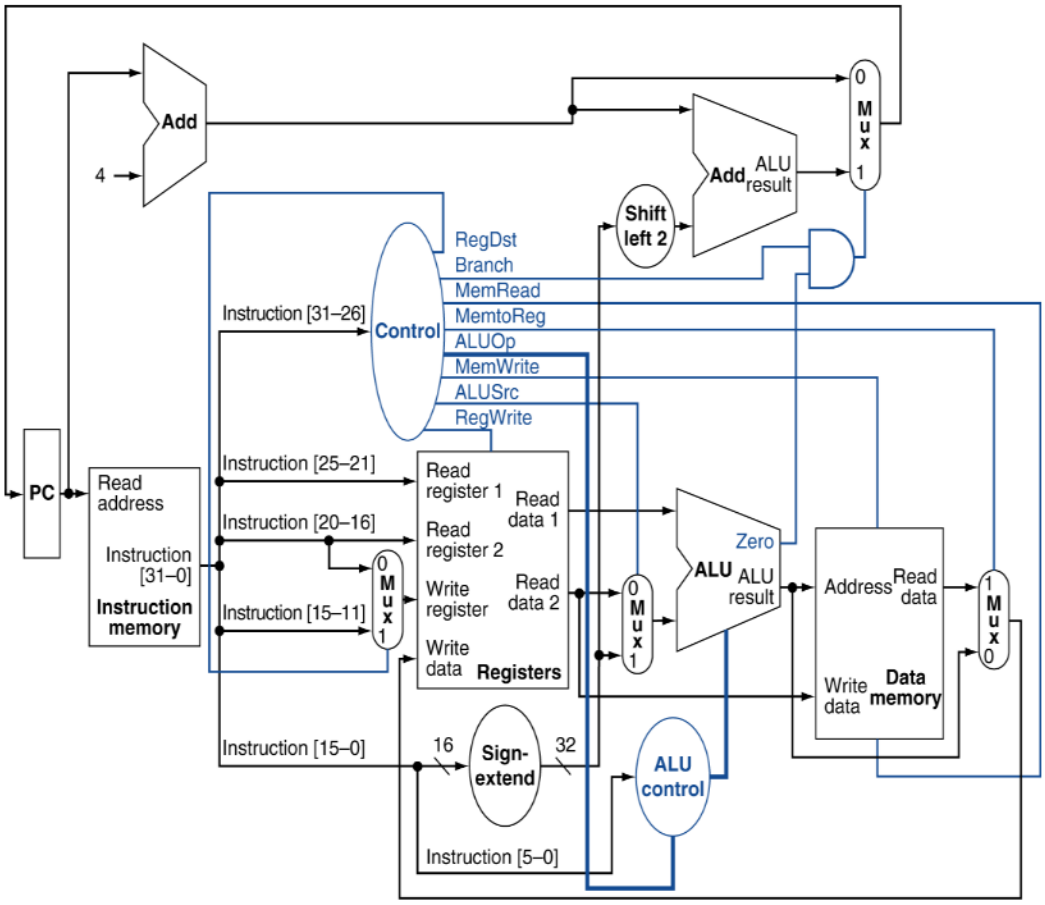

Figure 1 Single-Cycle Datapath (without jump)

In the table below, indicate the components that determine the critical path for the respective instruction, in **the order that the critical path occurs**. If a component is used, but not part of the critical path of the instruction (i.e. happens in parallel with another component), it should not be in the table. The register file is used for reading and for writing; it will appear twice for some instructions. All instructions begin by reading the PC register with a latency of 2ns. Put an x in any unused table entries.

Instruction Type	Hardware Elements Used by Instruction											
R-Format	PC	InsnMem										
Load	PC	InsnMem										
Store	PC	InsnMem										

Place the latencies in nanoseconds of the components that you have decided for the critical path of each instruction in the table below. Compute the sum of each of the component latencies for each instruction. Put a 0 (zero) in any unused table entries

Instruction Type	Hardware Latencies For Respective Elements											Total
R-Format	2 ns	10										
Load	2 ns	10										
Store	2 ns	10										

Use the total latency column to derive the following critical path information:
Given the data path latencies above, which instruction determines the overall machine critical path (latency)?

What will be the resultant minimum clock cycle time of the machine based on the critical path instruction?

What is the maximum frequency the machine can run?

Question 2 (10 points)

In reference to **Figure 1**, answer the questions for each of the following instructions.
Assume ($4) = 9231, ($5) = 5571, ($6) = 34271, ($8) = 80001006 at the beginning.

```
    add $3, $4, $5
    beq $3, $6, L
    sw  $7, 16($8)
L:  lw  $7, 16($8)
```

(a) Provide in the following table the values of control signals generated by the control unit for each instruction. Use an x for any don't-care signals.

	RegWrite	RegDst	MemRead	MemWrite	Branch	ALUSrc	MemtoReg	ALU Operation[1]
`add`	1	1	x	0	0	0	0	Add
`beq`								
`sw`								
`lw`								

(1) Identify the ALU operation with names e.g. add, subtract etc.

(b) For some instructions, some units produce output that is consumed by another unit, while some units produce outputs not consumed. Meanwhile, some storage units are written into, while some are not. Put the instruction operation symbol (i.e. `add`, `beq`, `sw` or `lw`) in the appropriate box in the following table.

	PC	PC Adder	Branch Adder	Register File	ALU	Instruction Memory	Data Memory
Output consumed	add, beq, sw, lw						
Outputs not consumed	x						
Written	add, beq, lw, sw						
Not	x						

written							

Question 3 (5 points)

Identify the components (using the table from Question 1) of the single-cycle datapath that must be modified to implement the following (non-MIPS) instruction:

```
swap $r1, $r2
```

This instruction exchanges the contents of registers $r1 and $r2.

Submission Instructions (read carefully)

Create a single PDF file that contains all your answers, and name it your HowardU username plus .pdf, for example my submission would be named gedare.bloom.pdf
Upload your PDF file to Homework2 on BlackBoard and input your answers there too.