## 1  Pre-Check

This section is designed as a conceptual check for you to determine if you conceptually understand and have any misconceptions about this topic. Please answer true/false to the following questions, and include an explanation:

1.1  The single cycle datapath uses the outputs of all hardware units for each instruction.

1.2  It is possible to execute the stages of the single cycle datapath in parallel to speed up execution of a single instruction.

1.3  If the logic delay of reading from IMEM is reduced, then any (non-empty) program using the single cycle datapath will speed up.

1.4  Stores and loads are the only instructions that require input/output from DMEM.

1.5  It is possible to feed both the immediate generator's output and the value in rs2 to the ALU in a single instruction.

# 2    Single-Cycle CPU

2.1  For this worksheet, we will be working with the single-cycle CPU datapath provided the last page.

(a) Explain what happens in each datapath stage, and which hardware units in the datapath are used.

**IF** Instruction Fetch

**ID** Instruction Decode

**EX** Execute

**MEM** Memory

**WB** Writeback

(b) On the datapath on the last page of the worksheet, fill in each **round** box with the name of the datapath component, and each **square** box with the name of the control signal.

(c) List all possible values that each control signal may take on for the single cycle
datapath, then briefly describe what each value means for each signal.

| Signal Name | Values | Signal Name | Values |
|---|---|---|---|
| PCSel | | RegWEn | |
| ImmSel | | BrEq | |
| BrLt | | ALUSel | |
| MemRW | | WBSel | |

2.2 Fill out the following table with the control signals for each instruction based on the
datapath on the last page. If the value of the signal does not affect the execution of
an instruction, use the * (don't care) symbol to indicate this. If the value of the
signal **does** affect the execution, but can be different depending on the program, list
all possible values (for example, for a signal with possible values of 0 and 1, write
0/1).

| | BrEq | BrLT | PCSel | ImmSel | BrUn | ASel | BSel | ALUSel | MemRW | RegWEn | WBSel |
|---|---|---|---|---|---|---|---|---|---|---|---|
| add | | | | | | | | | | | |
| ori | | | | | | | | | | | |
| lw | | | | | | | | | | | |
| sw | | | | | | | | | | | |
| beq | | | | | | | | | | | |
| jal | | | | | | | | | | | |
| blt | | | | | | | | | | | |

# 3   Timing the Datapath

**Clocking Methodology**

- A **state element** is an element connected to the clock (denoted by a triangle at the bottom). The **input signal** to each state element must stabilize before each **rising edge**.

- The **critical path** is the longest delay path between state elements in the circuit. The circuit cannot be clocked faster than this, since anything faster would mean that the correct value is not guaranteed to reach the state element in the alloted time. If we place registers in the critical path, we can shorten the period by **reducing the amount of logic between registers**.

For this exercise, the times for each circuit element is given as follows:

| Clk-to-Q | RegFile Read | PC/RegFile Setup | Mux | Adder |
|----------|--------------|------------------|-----|-------|
| 5ns | 35ns | 20ns | 15ns | 20ns |

| ALU | Branch Comp | Imm Gen | MEM Read | DMEM Setup |
|-----|-------------|---------|----------|------------|
| 100ns | 50ns | 45ns | 300ns | 200ns |

(a) Mark an **X** for the stages used by each instruction:

|     | IF | ID | EX | MEM | WB | |
|-----|----|----|----|-----|----|--|
| add |    |    |    |     |    |  |
| ori |    |    |    |     |    |  |
| lw  |    |    |    |     |    |  |
| sw  |    |    |    |     |    |  |
| beq |    |    |    |     |    |  |
| jal |    |    |    |     |    |  |

(b) How long does it take to execute each instruction? Ignore the length of a clock cycle based off of the critical path, and assume that the setup times to the RegFile and the PC are the same.

1. jal

2. lw

3. sw

(c) Which instruction(s) are responsible for the critical path?

(d) What is the highest clock frequency for this single cycle datapath?

(e) Why is the single cycle datapath inefficient?

(f) How can you improve its performance? What is the purpose of pipelining?

RegWriteData

ALU

PC+4

+4

PC+4

ALU

PC

0

1

RegWriteData

PC

RegReadData1

inst

inst[11:7]

RegWriteIndex

inst[19:15]

RegReadIndex1

RegReadData2

inst[24:20]

RegReadIndex2

RegWEn

1

0

A

0

1

B

MemAddress

ALU

PC+4

Mem

1

2

0

MemReadData

MemWriteData

MemRW

inst[31:0]