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 makes use 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 Combinational logic is only used in the instruction decode stage.

2 Single-Cycle CPU

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

(a) On the datapath, fill in each **round** box with the name of the datapath component, and each **square** box with the name of the control signal.

(b) Explain what happens in each datapath stage.

IF Instruction Fetch

ID Instruction Decode

EX Execute

MEM Memory

WB Writeback

2.2 Fill out the following table with the control signals for each instruction based on the datapath on the previous page. Wherever possible, use * to indicate that what this signal is does not matter (as in, letting the value be whatever it wants won't affect the execution of the instruction). If the value of the signal does matter for correct execution, but can vary, list all of the values (for example, for a signal that matters 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											
bltu											

2.3 **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, assume the delay for each stage in the datapath is as follows:

IF: 200 ps   ID: 100 ps   EX: 200 ps   MEM: 200 ps   WB: 100 ps

(a) Mark the stages of the datapath that the following instructions use and calculate the total time needed to execute the instruction.

	IF	ID	EX	MEM	WB	Total Time
add						
ori						
lw						
sw						
beq						
jal						
bltu						

(b) Which instruction(s) exercise the critical path?

(c) What is the fastest you could clock this single cycle datapath?

(d) Why is the single cycle datapath inefficient?

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