

University of California, Berkeley  
College of Engineering  
Department of Electrical Engineering and Computer Science

Spring 2000 Prof. Bob Brodersen

**Midterm 1**  
March 15, 2000  
CS152: Computer Architecture

This midterm consists of four problems, each of which has multiple parts, so budget your time accordingly. The exam is closed-book, but calculators and one sheet of notes are allowed. Good luck!

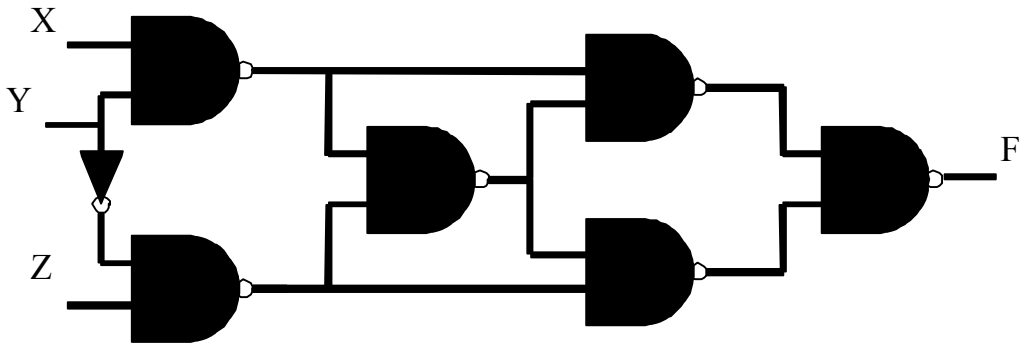
Name	
SID	
Discussion	

1	
2	
3	
4	
Total	

### Problem 1: Critical Path and Delay (25 points)

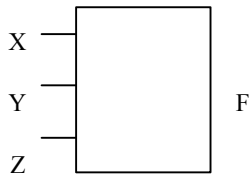
Throughout this problem, use the simple linear delay model presented in class. For the circuit below, assume the following delay parameters:

- NAND:  $t_{plh} = 0.5\text{ns}$ ,  $t_{phl} = 0.5\text{ns}$ ,  $t_{plhf} = 0.002\text{ns/fF}$ ,  $t_{phlf} = 0.002\text{ns/fF}$   
 Input capacitance: 100fF
- Inverter:  $t_{plh} = 0.2\text{ns}$ ,  $t_{phl} = 0.2\text{ns}$ ,  $t_{plhf} = 0.001\text{ns/fF}$ ,  $t_{phlf} = 0.001\text{ns/fF}$   
 Input capacitance: 50fF
- Wiring Capacitance: (Equal for all nodes) 5fF



a) What is the worst case delay? Assume there is no delay at the inputs X, Y and Z.

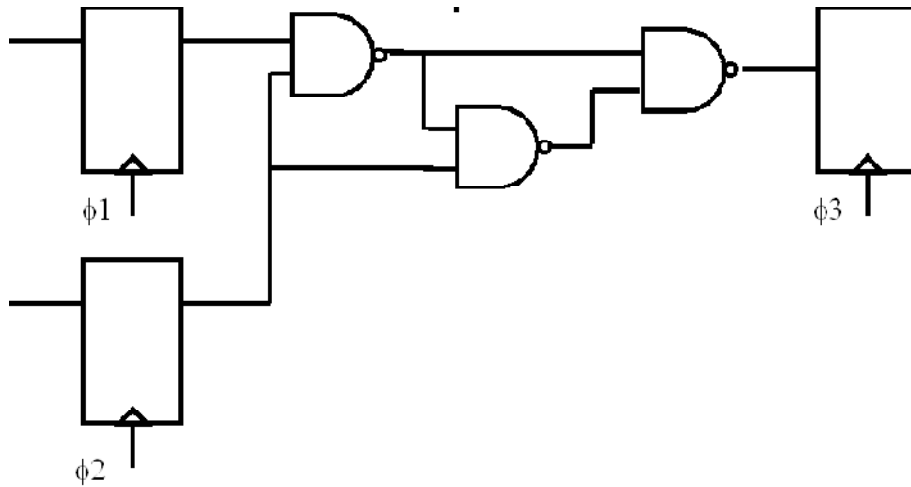
b) Now assume that you want to generate a symbol for the circuit in part (a). Determine the following parameters for your symbol:  $t_{plh}$ ,  $t_{phl}$ , and the load dependant delay (in ns/fF).



Now consider the following circuit and the following parameters:

Register:  $t_{\text{clk-to-Q}} = 0.6\text{ns}$ ,  $t_{\text{setup}} = 0.5\text{ns}$ ,  $t_{\text{hold}} = 0.2\text{ns}$

NAND:  $t_{\text{pth}} = 0.5\text{ns}$ ,  $t_{\text{phl}} = 0.5\text{ns}$



- c) What is the maximum frequency at which this circuit will operate correctly? Ignore any load dependent delay.
- d) If the clock signals are skewed so that  $\phi_1$  arrives 0.2ns before  $\phi_2$  which arrives 0.2ns before  $\phi_3$  (such that  $\phi_1 - \phi_2 = \phi_2 - \phi_3 = 0.2\text{ns}$ ), what is the maximum frequency at which this circuit will operate correctly?
- e) Now assume that  $\phi_1$  and  $\phi_2$  arrive at the same time, and that  $\phi_3$  arrives later. What is the maximum tolerable skew to ensure that there are no hold time violations?

## Problem 2: Single-cycle Processors (25 points)

The following MIPS code finds the maximum integer within a bounded array, where \$4 contains a pointer to the beginning of the array, \$5 contains the length of the array and \$3 contains the pointer to store the result at the end. (Assuming there is no branch delay slot.)

```
LW$2,0($4) //assumethefirstnumberisthelargest
ADDI$4,$4,4
```

```
ADDI$5,$5,-1
```

max:

```
LW$6,0($4)//loadarrayelementandincrementpointer
ADDI$4,$4,4
```

```
SLT$7,$2,$6//update$2if$6islarger
BEQ$7,$0,next
ADD$2,$0,$6
```

next:

```
ADDI$5,$5,-1//continuethesearchuntilendoffarray
BEQ$5,$0,finish
Jmax
```

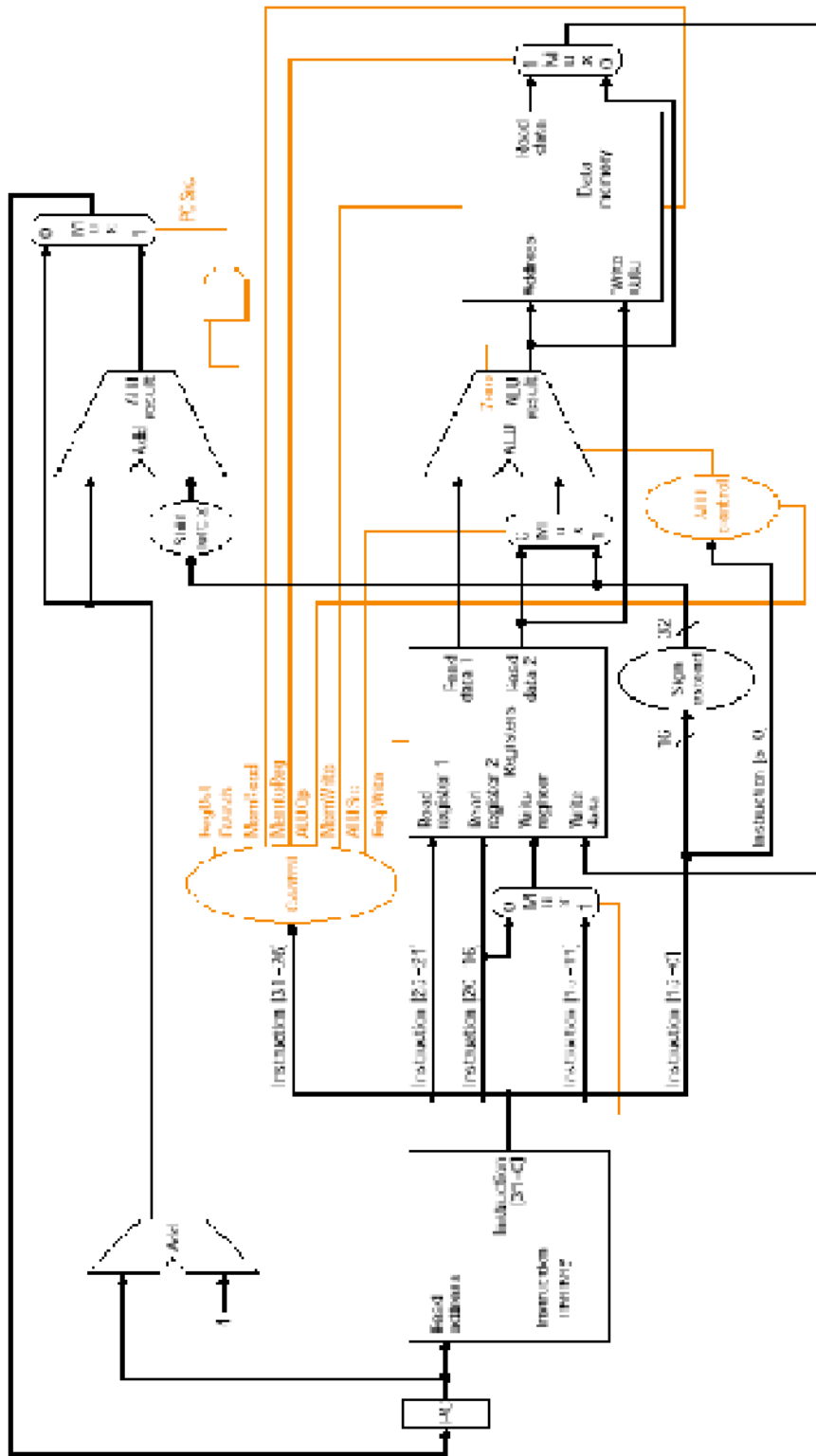
```
finish:      SW$2,0($3)//storeresult
```

The single-cycle datapath and control unit are shown on the next page. Assume that the delay and energy consumption per operation for each functional unit is as follows:

- Memory (read or write): 3 ns, 3 pJ
- ALU and adder: 2 ns, 2 pJ
- Register file (read or write): 1 ns, 1 pJ
- All other units: 0 ns, 0 pJ

- What is the minimum clock cycle time for this processor?
- For an array of length  $N$ , what is the range of execution time for this program (e.g. the minimum possible execution time and the worst case execution time)?
- What is the energy consumption (per instruction) for each type of instruction in the program? Assume that components are completely turned off and do not consume energy when they are not needed.

Diagram and scratch space for Problem 2:



### Problem 3: Single-cycle Datapath Design (25 points)

The task is to design a single cycle processor with the minimum number of functional units that can perform the following standard MIPS instructions plus a new rotate instruction. The rotate instruction does a rotate of \$RS to the right by the IMMED value and stores it in \$RD (e.g. a 2 shift rotate of a 5 byte word would turn [a b c d e] into [d e a b c]). The blocks that you can use are given below along with their control signals and delay values. All blocks are similar to those we used in class, except for the addition of a 64 bit shifter.

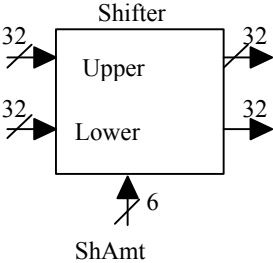
Instructions:

- ADDIU \$RD \$RS IMMED
- ADD \$RD \$RS \$RT
- SRL \$RD \$RS IMMED
- Rotate \$RD \$RS IMMED

Name

Components (and control signals):

- ALU (ALUcontrol, Zero) => 32 bit ALU with Zero status bit  
ALUcontrol = 00 for ADD, 01 for AND, 10 for SUB, 11 for OR  
Delay = 4
- EXTENDER (Sign/Zero) => Sign extender  
Delay = 1
- MUX (Select) => 2 input mux  
Delay = 1
- MEMORY (WrEnable, Addr) => Ideal memory  
Delay = 1
- REGISTER (Enable) => Clocked register  
Clk-to-Q Delay = 1
- REGISTER FILE (RD, RS, RT, WrEnable) => Register file  
Read delay = 1, Setup time = 1, Hold time = 1
- CONSTANTS => A 32 bit constant can be defined as an input to any block (no delay)
- SHIFTER (ShAmt) => 64 bit shifter (see symbol below)  
Input and output is through 2 buses which connect to the upper and lower 32 bits of a 64 bit word.  
Delay = 1



a) Draw the datapath showing all interconnections and components (including the controller).

b) What is the critical path ?

c) What is the delay of the critical path?

d) Show the values of all the control points for each instruction. (The Enable for the PC is given as an example)

	PCEnable					
ADDIU	1					
ADD	1					
SRL	1					
Rotate	1					

## Problem 4: Multi-cycle Processors (25 points)

For this problem you will be working with the multi-cycle datapath components on the next page. All inputs for the functional units are labeled, and registers only have one data input and one data output (you should not draw the clock lines). You will not need to deal with control in this problem, so the control inputs to each block are not shown.

- a) Given the datapath components on the next page, determine the register transfer language description for each of the standard MIPS instructions in the table below. You do not need to fill in every row. Hint: Do not write to the register file at the end of a cycle (i.e. only write directly from a register, not a functional unit).
- b) You'll notice that some components need to be reused during execution of an instruction. Wire the datapath to support all four instructions, adding only muxes as needed. You may provide constants as inputs to any component. Be sure to label special buses, such as instruction fields. You do not need to draw any control signals (including mux select signals) just assume they will be correctly generated in all cases.
- c) For each instruction in part (a), calculate the CPI and indicate on the table above which operations occur during each cycle.
- d) The table below indicates the worst case delay through each of the functional units used in the datapath. Given these delays, calculate the execution time of this processor for a program consisting of 400,000 adds, 250,000 loads, 250,000 stores, and 100,000 branches.

Functional Unit	Worst-case Delay
Memory	50ns
Register File (read)	25ns
Register File (write)	15ns
ALU	30ns
All others	0ns



