Name ______________________________ SID ______________________________

CS 152 Computer Architecture and Engineering
Fall 1998
R. W. Brodersen

Midterm # 2

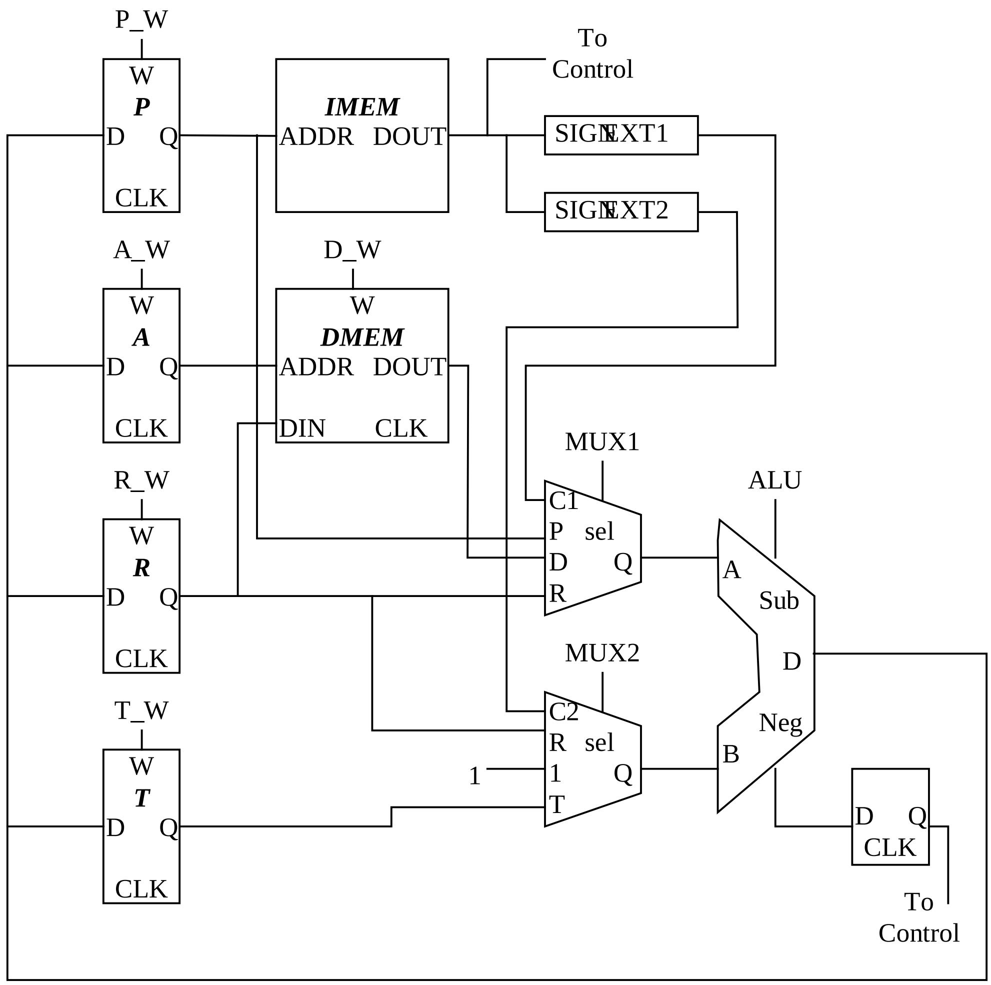
Please write your name and SID on *each page* of this exam. The number of points for each problem is shown below. You have three hours for this exam – budget your time accordingly.

#	Possible	Score
1	35	
2	15	
3	25	
4	25	
Total	100	

Name SID

Problem 1: Multi-cycle Datapath [40 points]

For this problem, please refer to the multi-cycle datapath drawn below. The ALU can perform either A+B or A-B and the output NEG indicates if the result is negative. The NEG signal, after being delayed one cycle, is fed through the controller and can be used to conditionally write to any of the four main registers (P, A, R, T). The control state machine is reset to 0 every time the P register (program counter) is written; otherwise, the state machine advances by one every cycle. Each instruction for this multi-cycle datapath contains two immediates, termed C1 and C2, each of which is independently sign-extended. Registers and memories are clocked and have a "CLK" input shown (but not connected). The clock only applies to the write operation of DMEM, and has no effect on the read operation. All write enables are active high.



Name____SID______________________________

a) Micro-Programming

Fill in the values missing in the table below for the given instructions. The RTL specification for the instructions is given. The phrase "if NEG" indicates that the register write should be preformed only if the result of the previous ALU operation was negative, use the keyword "NEG" in the table below to indicate this behavior.

ADD R,C1 (add immediate)
R<-R+C1
P<-P+1
SBN R,C1,C2 (subtract and branch if neg)
R<-C1-R
P<-P+C2 if NEG
Else P<-P+1
SWP [R+C2],R (swap memory data)
A<-R+C2
T<-R-R
T<-M[A]+T
M[A]<-R
P<-P+1
IIG R,C1 (increment R if >C1)
_<-C1-R (the result is not written to register)
R<-R+1 if NEG
P<-P+1

Inst./Cycle	P_W	A_W	R_W	T_W	D_W	MUX1	MUX2	ALU	
ADD	0	0	1	0	0	C1	R	ADD	
2	1	0	0	0	0	P	1	ADD	
SBN	0	0	1	0	0				
2	NEG	0	0	0	0				
3	1	0	0	0	0				
SWP	0				0	R	C2	ADD	
2	0				0	R	R	SUB	
3	0				1	D	T	ADD	
4	1				0	P	1	ADD	
IIG									
2									
3									

b) Datapath Delay

Using the delay values given below, calculate the minimum cycle time for the multi-cycle datapath. Assume there is no stall time between the completion of one instruction and the initiation of the following instruction.

Component	Delay
Sign-Extender	1ns
4-1 Mux	2ns
ALU	4ns
IMEM	2ns
DMEM	4ns
Register Clk-Q	1ns
Register Setup	2ns
Register Hold	4ns
Control Logic	3ns

Minimum cycle time:_______________

Name______________________________________ SID_____________________________

c) Complex Micro-programming

Fill in the given table with micro-code implementation of the BEQ. The definition of BEQ is as follows:

BEQ R, C1, C2: Branch to P+C2 if register R equals constant C1. The value in register R remains unchanged; the value in register T is not preserved.

Hint: the RTL operation ‘T<-R-R’ will always place the value 0 in register T.

Inst./Cycle	P_W	A_W	R_W	T_W	D_W	MUX1	MUX2	ALU	
BEQ									
2									
3									
4									
5									
6									
7									
8									
9									
10									

For parts d) and e), we consider a single-cycle implementation of the multi-cycle datapath that is only capable of executing the following four instructions:

Inst./Cycle	P_W	A_W	R_W	T_W	D_W	MUX1	MUX2	ALU	
BAD	0	0	0	1	0	R	T	ADD	
2	1	0	0	0	0	P	C2	ADD	
BLE	0	0	0	0	0	R	T	SUB	
2	NEG	0	0	0	0	P	C2	ADD	
3	1	0	0	0	0	P	1	ADD	
LAD	0	1	0	0	0	R	T	ADD	
2	0	0	1	0	0	D	C2	ADD	
3	1	0	0	0	0	P	1	ADD	
SIG	0	0	0	0	0	R	T	SUB	
2	0	0	NEG	0	0	R	T	SUB	
3	1	0	0	0	0	P	1	ADD	

d) Multi-Cycle CPI

Assuming the instruction mix in the table below, how much slower can the minimum cycle time of the single cycle implementation be compared to the multicycle implementation while maintaining equal performance? Give your answer relative to the cycle time of the multi-cycle version (i.e. “The single-cycle datapath can have a cycle-time that is 100 times slower”)

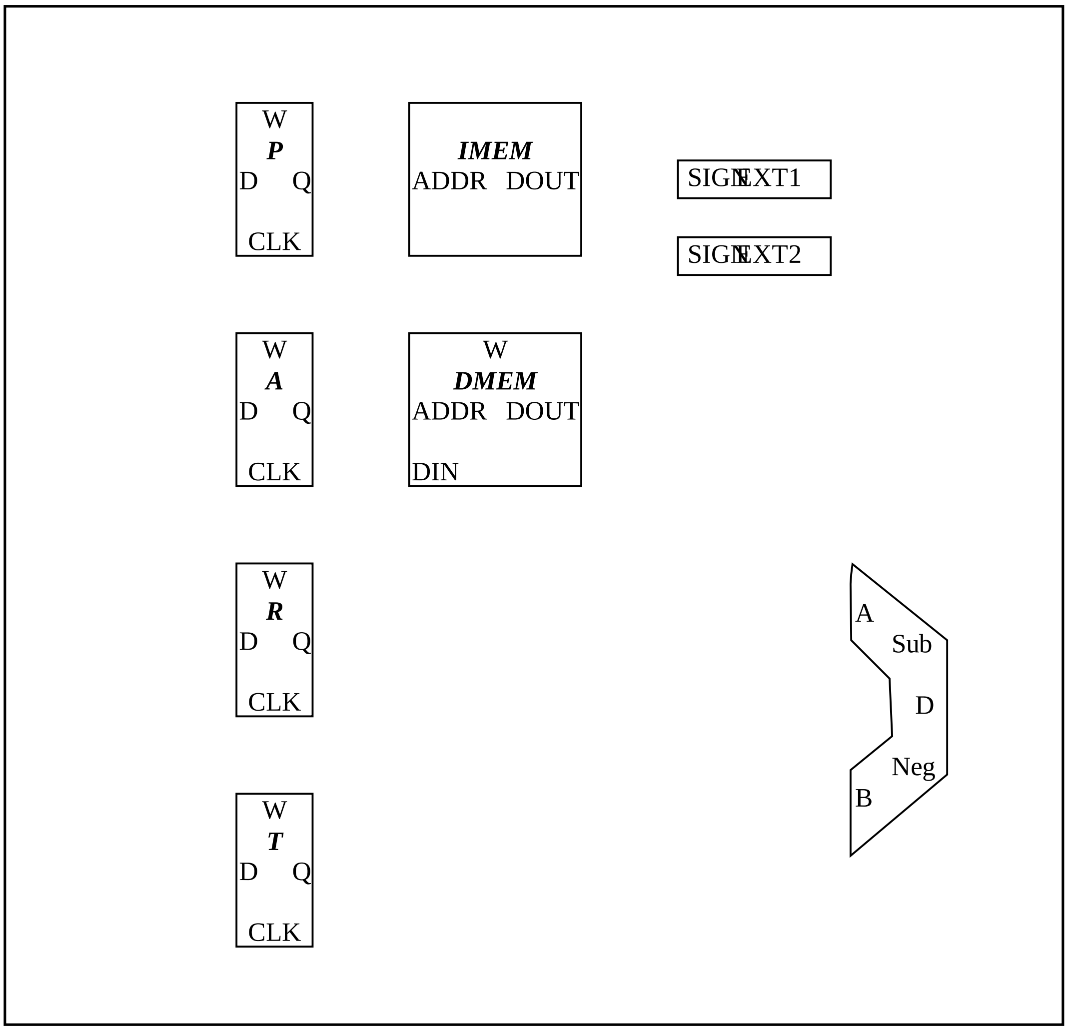
Instruction	Frequency
BAD	10%
BLE taken	30%
BLE not-taken	20%
LAD	30%
SIG	10%

Allowable increase in cycle time:_____________

Name SID

e) Multi-Cycle To Single-Cycle Conversion:

Fill in the missing pieces of the new single-cycle data path below so that it is capable of executing the four multi-cycle instructions (BAD, BLE, LAD, and SIG), each in a single cycle. Use the micro-code to determine the function of each instruction, as necessary. If you do not need a component that is already shown in the diagram, cross it out. You may only add MUXes (any size), adders, and wires. Use the minimum hardware required (you do not need to support any instructions given in part a); consider adders and MUXes to be of equivalent area). You do not need to show any control signals.



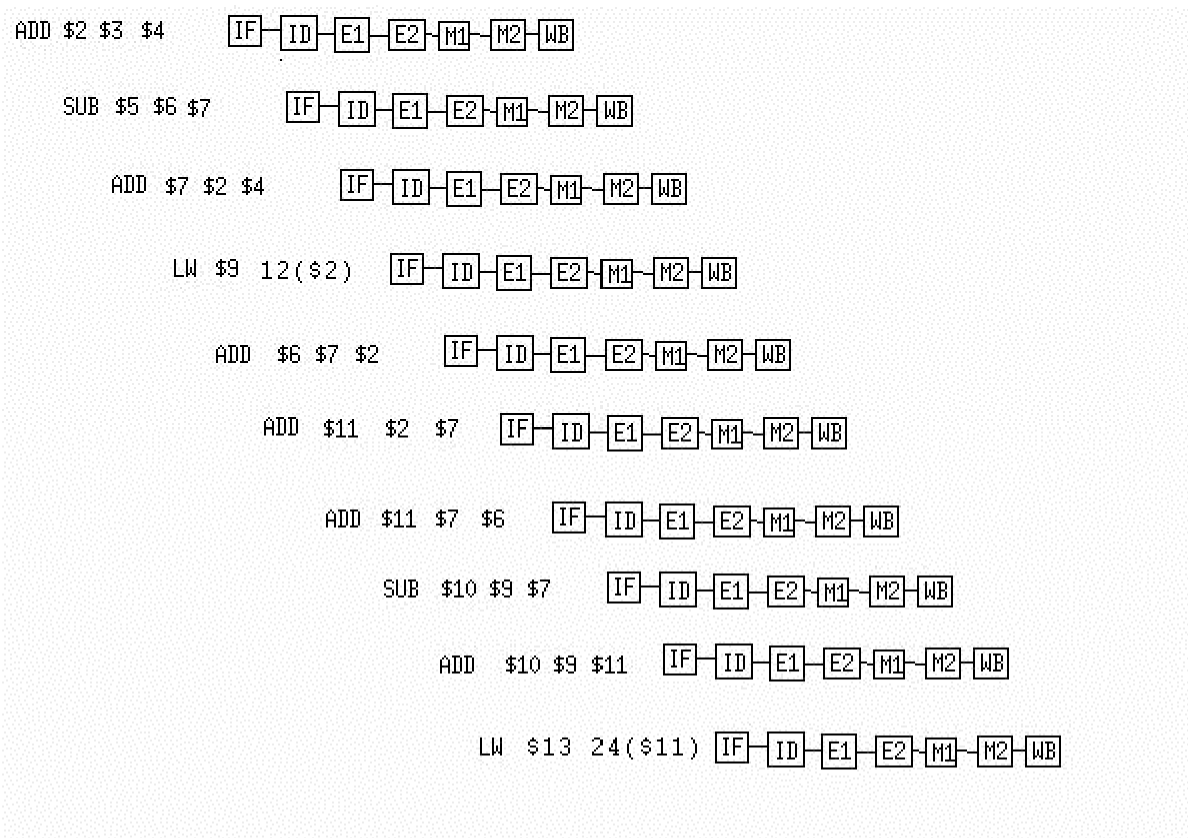
Name______________________________________ SID______________________________

Problem 2: Pipelining, Datapath Hazards, Forwarding

You work for the CPU giant, NoTell. The marketing department has decided that, even though the flagship CPU has the best performance on the market, it is difficult to sell because it has a low clock rate. The other engineers have decided that the best way to increase clock rate is to split the standard MIPS-style 5-stage pipeline into a 7-stage pipeline. This will be accomplished by splitting the EXE stage into EXE1 and EXE2. Likewise for the MEM stage, there is now MEM1 and MEM2. Assume you can not access any useful data in the position between the newly split stages, and you cannot forward into the middle of EXE or MEM stages. The new pipeline, repeated a few times, is shown below.

This, of course, expands the number of forwarding paths. As a new designer, you get the job of designing the forwarding scheme. You decide to first do some test cases.

a) On the pipelines shown below, draw the forwarding paths that are exercised each cycle. The instruction shown to the left of each pipeline is the instruction that is issued (enters the IF stage) in the cycle where the IF stage resides in the picture, e.g. the first ADD is issued in cycle 1, the first SUB is issued in cycle 2, etc. Show only the forwarding paths that are used by the code shown, not all the possible forwarding paths in the processor. It is possible that stalls will be required to resolve the hazard, which you will enumerate in part b). Assume that the register file can do a write before a read in the same cycle.



Name ______________________________ SID ______________________________

b) In the course of working through the above instructions and the forwarding paths that are needed, you realize that there are still some cases where no amount of forwarding can resolve the dependencies; the data is simply not ready by the time the next instruction needs it. The only way to resolve the hazard is to insert a bubble into the pipeline, stalling the next instruction, which needs the data, while allowing the current instruction that is using the data, to continue processing. You have come up with some instruction sequences with these problems, and now you have to figure out how many bubbles need to be inserted in each case. One bubble is equivalent to stalling for one clock cycle. The cases of interest are shown in the table below. Fill in the total number of bubbles that need to be inserted for the blocks of code shown, in order to resolve all hazards.

Instruction Sequence	Total Number of Bubbles
ADD $2, $3, $4 SUB $7, $2, $3	

LW $5, 24($6)
ADD $9, $8, $5
SUB $6, $9, $5	

LW $5, 12($13) LW $5, 16($8)	

LW $4, 32($3)
LW $5, 32($4)
LW $6, 32($5)	

ADD $7, $4, $2 LW $4, 0($7)	
SW $7, 0($12) XOR $2, $8, $4	

c) The hardware group has finally built an actual CPU, based on your testing from above. It is your job to exhaustively test the chip. However, the test board for the CPU has only a small ROM to store the test code. Write the test code that exercises every forwarding path in the CPU. You do not need to test the bubble cases.

Name ______________________________________ SID ______________________________

Problem 3: Cache and Virtual Memory

a) Design a 3-way set associative cache given the following constraints:

- Total size of 24 words
- Block size of 4 words
- Word length of 16 bits
- Total addressable memory space is 1024 (1K) word-addressed words
- Write-back Write-allocate policy
- Least frequently used (LFU) replacement policy

i.) How many comparators are required in this design, and what are their widths (in bits)? Explain your reasoning.

ii.) How many registers are required in this design, and what are their widths (in bits)? Explain your reasoning.

iii.) What status bits are needed for each block? Explain your reasoning.

iv.) Draw a diagram of the address bits in this machine and indicate what the bits are used for in this cache.

v.) Fill in the table below, indicating whether each request is a cache hit or miss:

Address	3	1	7	9	5	18	13	11	2	6	27	15	22	30
H/M														

vi.) Draw a table below that represents the final state of the cache.

Name ______________________________________ SID ______________________________

b) What page size(s) would allow parallel cache and TLB lookups, therefore speeding up physical address translation? Explain your answer.

c) Assume your cache is slower than your processor critical path, so that a cache hit will stall the pipeline, requiring two clock cycles. On a miss, you suffer a penalty of eight cycles. Calculate the CPI for a 40% miss rate if all other instructions take one clock cycle and the total number of data and instruction memory requests compose 75% of the total instruction count.

d) We decide to pare the hit delay in your cache by reverting to a direct mapped organization. Compute the new CPI if hits take one cycle, misses take six cycles, the miss rate is 80%, and the number of data and instruction requests still total 75% of the instruction count.

e) Explain how changing to direct mapped cache would cause the above changes in hit and miss delay and miss rate.

Name ______________________________________ SID ______________________________

f) Consider a machine with a three-tiered virtual memory hierarchy as described below. Note that the sum of hit rates is 100%, so no misses occur at the physical disk level.

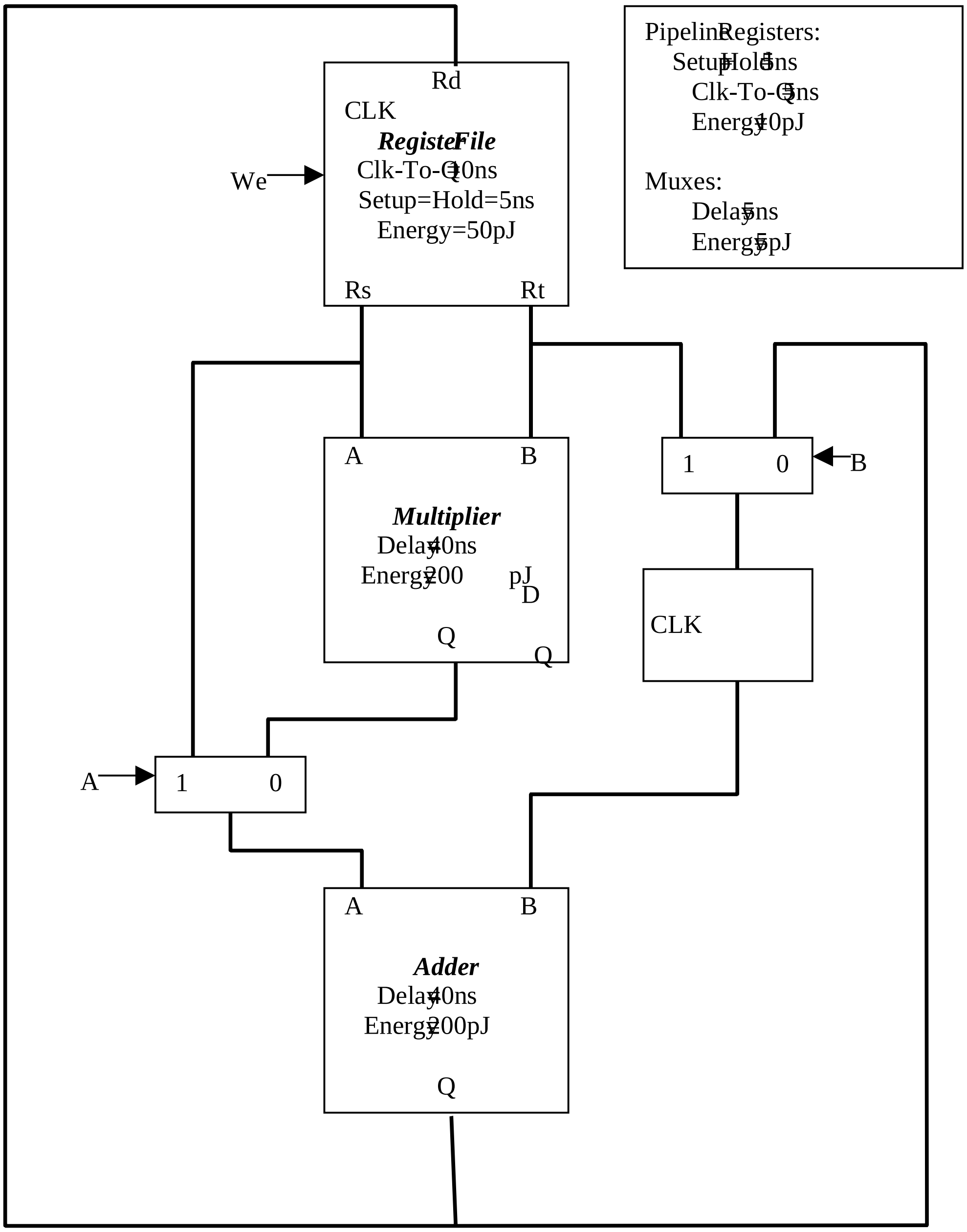
Level / Memory Type	**L1 Cache On-die SRAM**	**L2 Cache Off-chip SRAM**	**Main Memory DRAM**	**Physical Disk IDE Hard Drive**
Hit Delay	1 cycle	4 cycles	15 cycles	500,000 cycles
Hit Rate	15.7%	28.8%	55.4%	0.1%

i.) Calculate the CPI if memory accesses total 20% of the instruction count and all other instructions take one cycle.

ii.) Calculate the time required to load 1000 words from disk if the CPU frequency is 100MHz, all instructions are LW, and there is 8ms fixed access time for any request from the physical disk. Assume that the hit rates above apply.

Name ______________________________ SID ______________________________

Problem 4



The energy and delay is for 1 operation and the energy is the dissipation per cycle.

This problem uses the datapath above to execute the following program:

```
sum1 = 0
sum2 = 0
FOR I = 1,2 DO
{sum1 = sum1 + a[i]b[i]
  sum2 = sum2 + a[i]
```

Name SID

Assume the register file is already loaded with all necessary data

```
$s1 = a[1]    $s2 = a[2]
$t1 = b[1]    $t2 = b[2]
$s3 = sum1    $s4 = sum2
```

a) Compile this program into machine language, using the following format. Unroll the loop and reorder the instructions to minimize the execution time

Amux	Bmux	Rs	Rt	Rd	We		Comments
1/0	1/0	$Rs	$Rt	$Rd	1/0		

b) What is the minimum cycle time and execution time for this datapath to execute the above program?

c) What is the energy necessary to execute this program. If operated at its maximum rate, what is the power dissipated?

d) Add a single pipeline register to this datapath to optimally decrease its cycle time - show the position of the register on the datapath on the previous page.

e) What is the new minimum cycle time, execution time, energy, and power assuming it is operated at its maximum rate to execute the program you wrote in section a)? (Ignore the fact that the program no longer works)

Name ______________________________ SID ______________________________

f) If the voltage is reduced, all the delays, setup and hold times increase by the following formula:

$$T_{delay}(V_{supply}) = T_{delay}(5\ volts)*[4/(V_{supply} - 1)]$$

By reducing the voltage to the point where the execution time is the same as before the pipeline register was added, what is the energy consumed. What is the power dissipation?

Name SID