6.823 Computer System Architecture
Prerequisite Self-Assessment Test

Assigned Sept. 7, 2005 ***Due Sept. 14, 2005***

This self-assessment test is intended as a prerequisite assessment to determine if you have the background necessary to take 6.823. This course assumes that you have a solid understanding of the material presented in 6.004 or an equivalent course.

For each question, we ask that you fill out the table at the end of the problem set handout indicating your level of confidence with each assigned problem and hand this in with your solutions. If you have never seen the material before, then please enter "0". If you have seen the material, and think you should know it, but can't answer the question without spending time studying your old notes, then please enter "1". If you are comfortable with the material, then enter "2". You should turn in solutions for problems where you entered "1" and "2", but do not have to turn in solutions for problems for which you entered a "0".

We will only have a short review of this material in 6.823. If there are too many "0"s in the table then it is unlikely you will be able to keep up with 6.823 this term, as the pace is such that you will not be able to take prerequisite material concurrently. If you are not sure whether you have the background to take this class, please feel free to discuss your particular situation with a TA.

For this test only, you must work individually and turn in your own solutions. Do not discuss the problems with others.

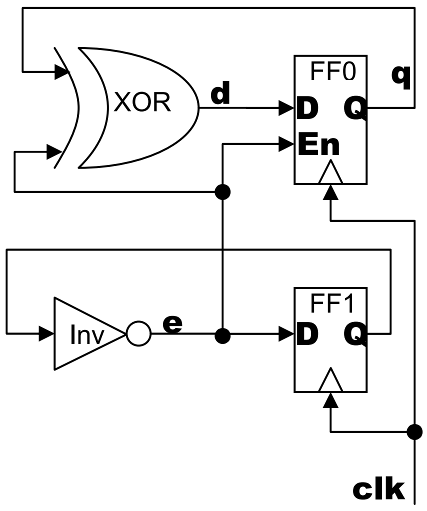
Problem 1

Construct the following logic functions using only two-input NAND gates (please use hierarchy where possible to simplify your designs):

a) inverter
b) two-input XOR gate
c) 2-to-1 multiplexer
d) 2-to-4 decoder
e) transparent latch (transparent on clock low)
f) flip-flop (positive-edge triggered)

Problem 2

The questions below refer to the following circuit and its associated timing parameters. The flip-flops are positive-edge triggered, and FF0 has an enable input (Q only changes if En is high). Assume that all timing parameters are positive.

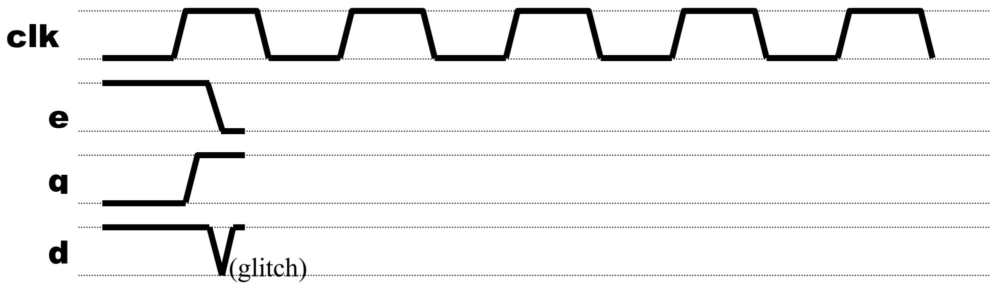


t_{xorMin}	Minimum delay through XOR gate
t_{xorMax}	Maximum delay through XOR gate
t_{invMin}	Minimum delay through inverter
t_{invMax}	Maximum delay through inverter

Flip-flop parameters (for both flip-flops)

t_{setupD}	Minimum time before clock for arrival of D input
$t_{setupEn}$	Minimum time before clock for arrival of En input
$t_{clkQMin}$	Minimum delay from clock to Q output
$t_{clkQMax}$	Maximum delay from clock to Q output
t_{holdD}	Minimum time after clock for change of D input
t_{holdEn}	Minimum time after clock for change of En input

a) Fill in the following timing diagram for the circuit:



b) What is the maximum clock frequency for the circuit in terms of the given timing parameters?

c) What are the minimum values for t_{xorMin} and t_{invMin} in terms of the given timing parameters? What might happen if these restrictions are violated?

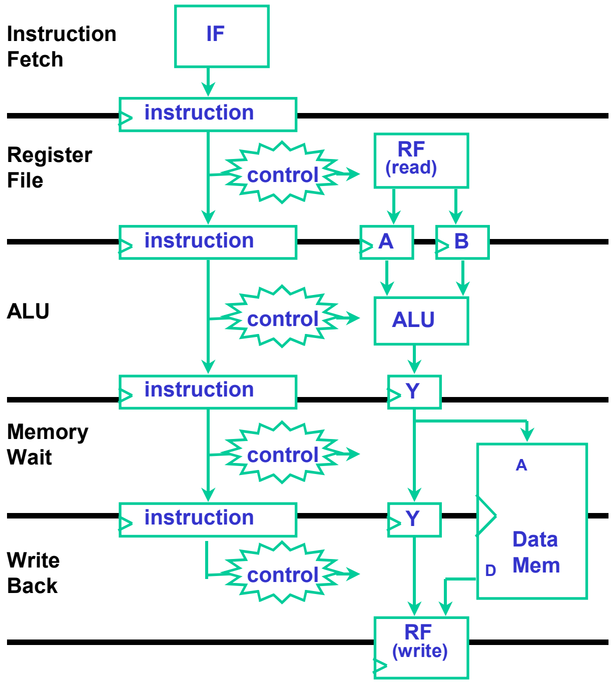
d) Assume that the circuit is clocked at the maximum frequency computed in (b) and that the XOR and Inv gates have the minimum delays computed in (c). Describe a specific example of how clock skew could cause the circuit to fail.

Problem 3

The following figure shows a 5-stage pipelined processor. The pipelined processor should always compute the same results as an unpipelined processor. Answer the following questions for each of the instruction sequences below:

- Why does the sequence require special handling (what could go wrong)?
- What are the minimal hardware mechanisms required to ensure correct behavior?
- What additional hardware mechanisms, if any, could help preserve performance?

Assume that the architecture does not have any branch delay slots, and assume that branch conditions are computed by the ALU.



a)	BEQ	r1, r0, 200	# branch to PC+200 if r1 == r0
	ADD	r2, r3, r5	# r2 $\leftarrow$ r3 + r5
	SUB	r4, r5, r6	# r4 $\leftarrow$ r5 + r6
	…		
b)	ADD	r1, r0, r2	# r1 $\leftarrow$ r0 + r2
	SUB	r4, r1, r2	# r4 $\leftarrow$ r1 - r2
	…		
c)	LD	r1, 0(r2)	# r1 $\leftarrow$ Mem[r2+0]
	ADD	r3, r1, r2	# r3 $\leftarrow$ r1 + r2
	…		

Problem 4

Describe the operation of a data cache. Your description should include discussion of the following:

a) Spatial and temporal locality.
b) Valid bits.
c) Direct mapped versus set-associative structures. Show how cache indexing and tag match works for both direct mapped and 2-way set-associative cache configurations assuming one word per cache line. What are the advantages and disadvantages of direct mapped versus set-associative structures?
d) Multiple-word cache lines. What are the advantages and disadvantages of multiple-word cache lines? Describe how they are implemented for a direct mapped cache.
e) LRU and random replacement policies. What are their relative advantages and disadvantages?

Problem Ratings

		subproblem						
		A	B	C	D	E	F	G
problem	1							■
	2					■	■	■
	3				■	■	■	■
	4						■	■

0	No idea
1	Used to know it
2	Know it