University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Sciences

EECS151/251A J. Wawrzynek
Spring 2023 5/10/23

Final Exam

Name: _____

Student ID number: _____

Class (EECS151 or EECS251A): _____

Before solving the problems, write your student ID number on all pages.

You have three hours to take the exam. This exam comprises a set of questions with 1 point per approximate expected minute of completion—with a total of 118 points.

For each problem if you find yourself taking excessive time to work out a solution consider skipping the problem or a fresh approach. Also, start by answering the easier questions and then move on to the more difficult ones.

No calculators, phones, or other devices allowed.

Neatness counts. We will deduct points if we need to work hard to understand your answer.

1 Parallel Prefix [8 pts]

Draw a block diagram for a circuit that reads a string of integers $(x_0, x_1, x_2, \dots)$ from a FIFO_A and generates the parallel prefix of that string $(x_0, x_0 + x_1, x_0 + x_1 + x_2, \dots)$ writing the result into FIFO_B. Design your circuit to maximize F_{max}, and write down your F_{max} value.

Your building blocks are listed below, along with their timing characteristics:

block	setup for write	clk-q for read	hold time
FIFO	100 ps	100 ps	0 ps
register	10 ps	10 ps	0 ps

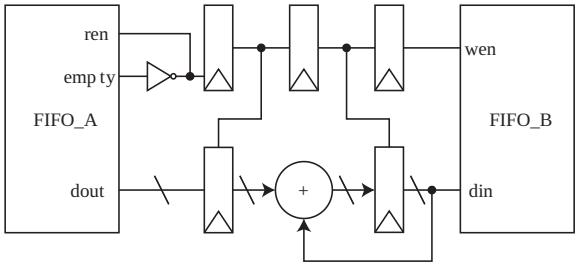
block	delay
adder	100 ps
logic gate	10 ps

FIFO_A takes a read enable as input and outputs an empty flag and data output. FIFO_B takes a write enable and data input as input—assume it never becomes full. Also assume data input of FIFO_B and data output of FIFO_A are of the same bit-width (the result of addition may wrap around).

The register has a clock enable and a synchronous reset input (sets its value to all 0). Your circuit should keep operating until FIFO_A indicates empty. Assume an external reset signal is asserted before the clock starts and then is held high for at least one clock cycle, then lowered.

Solution:

F_{max} = 1/(120 ps) = 8.3 GHz

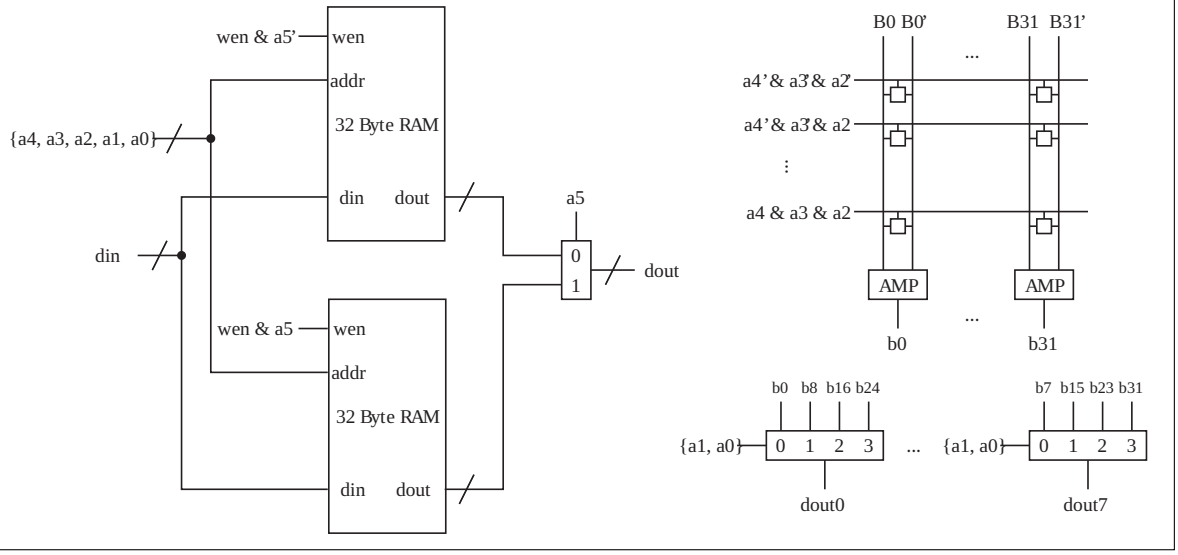


2 Memory Composition [8 pts]

Suppose we have a single-port Byte-wide memory that holds 64 Bytes. It comprises multiple blocks where each block can hold 256 bits (32 bits per row). You are asked to show details of internal memory circuitry, and in particular, show how the address bits {a5, a4, a3, a2, a1, a0} are used within the memory.

Draw two diagrams. The first showing how memory sub-blocks are put together (and how address bits are assigned), and the second details of the internals of one sub-block (and how address bits are assigned). Don't show details of storage cell, decoder internals, column drivers, sense amps., etc.

Solution:



3 Memory Access Time [6 pts]

Using big O notation:

1. How would you expect the best case read access time to grow with N for one large memory block of size N?

2. Now assume we restrict the block size to smaller than N, say to a constant size M, but still want to scale up the size of the memory to size N, by using multiple blocks. Now, how does the best case access time scale with N? Assume wire resistance and capacitance are negligible.

Justify your answers.

Solution:

1. $O(\sqrt{N})$. For word lines, we need to charge/discharge gate capacitances of access transistors, where the delay increases in proportion to the number of transistors per row. For bit lines, we need to charge/discharge the drain capacitances of access transistors, where the delay increases in proportion to the number of transistors per column. Although gate capacitance and drain capacitance may not be the same, these delay will be balanced by placing cells in a rectangular with some aspect ratio. Therefore, the delay will increase in proportion to the square root of memory size. (The decoders will take $O(\log N)$ delay assuming they are composed as a balanced tree of AND gates or multiplexers, but this is smaller.)

2. On the other hand, if it is composed of small sub-blocks hierarchically, delay will be $O(\log N)$.

4 Dual-Port Memory [5 pts]

John has the idea that you could build an SRAM memory using the standard 6T cell, but instead of a single read or write operation per cycle, he could use the two bit lines independently to have a *dual ported read* of two locations or a single write of one location per clock cycle. With a standard SRAM the two bit lines are used together (with one signaling d and the other $\overline{d}$). In this case, each of the two bit lines could signal a different read value from two different rows. If this scheme can work it has obvious area advantages over having more bit lines for simultaneous reads, but some disadvantages also. What changes to the bit-cell and the memory array would be needed to implement John's idea. What are the dissadvantages of this approach over the usual approach?

Solution:

Besides we need two word lines per row (two decoders), we need to precharge bit lines to $V_{dd}/2$ and use normal sense amplifier instead of differential sense amplifier, which takes more time to be safe and charging to 1 is slow (and actually goes only up to $V_{dd} - V_{th}$) because it is through NMOS.

5 FinFET [3 pts]

Briefly describe the primary advantage(s) and disadvantage(s) of FinFET transistors over traditional planar transistors.

Solution:

Since the gate is surrounding the channel, when biased more holes or electrons can move between source and drain and thus it can achieve smaller delay. In other words, it has smaller subthreshold (leakage) current for the same on current. However, it has more complex structure and hard to build physically.

6 Energy Efficient Design [10 pts]

You are given a very large combinational logic block with registered inputs and registered outputs. The latency for moving data from the input register to the output register is allowed to be at most 320 ps, and the throughput must be at least one data item per 100 ps. The combinational logic block has 80 ps delay, and a register setup time is 10 ps and clk-to-q is 10 ps at a nominal V_{dd}. Computation on successive elements of the data stream are independent.

With emphasis on low cost, devise a scheme that improves the switching energy efficiency of this circuit as much as possible. Assume that registers come at no cost and add no switching energy. Sketch your solution and estimate the energy efficiency of your solution compared to the original scheme. Assume that circuit delay scales linearly with V_{dd}.

Solution:

We are going to lower V_{dd} as much as possible while pipelining the block to meet the requirements. With N stage pipeline, the latency is $80+20N$ ps and the throughput is one data per $20+80/N$ ps at a nominal V_{dd}. Assuming the delay is inversely proportional to V_{dd}, if we lower V_{dd} by a factor of $1/k$, to meet the latency and throughput requirements,

$$(80+20N)k \le 320, \;\; (20+80/N)k \le 100$$

For $N = 1$, k must be 1 to meet the throughput requirement.
For $N = 2$, $k \le 5/3$ to meet the throughput requirement:

$$120k \le 320 \rightarrow k \le 8/3, \;\; 60k \le 100 \rightarrow k \le 5/3$$

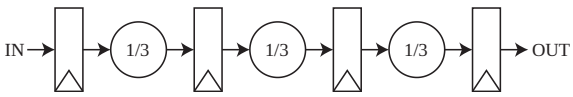
For $N = 3$, $k \le 15/7$ to meet the throughput requirement:

$$140k \le 320 \rightarrow k \le 16/7, \;\; 140k/3 \le 100 \rightarrow k \le 15/7$$

For $N = 4$, $k \le 2$ to meet the latency requirement:

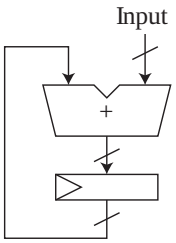
$$160k \le 320 \rightarrow k \le 2, \;\; 40k \le 100 \rightarrow k \le 5/2$$

For larger N, k must be less than 2 to meet the latency requirement. Therefore, V_{dd} is minimized when $N = 3$ and its value is 15/7 of the nominal value. Given that pipeline registers do not consume energy, the energy efficiency (data per energy = cycle per energy $= 2/(\alpha C V_{dd}^2)$) becomes $(15/7)^2$ times higher than the original one.



7 Accumulator [6 pts]

You have a simple accumulator circuit (shown below) that is used to iteratively calculate the arithmetic sum of 1024 8-bit integers. It accepts one integer per clock cycle.



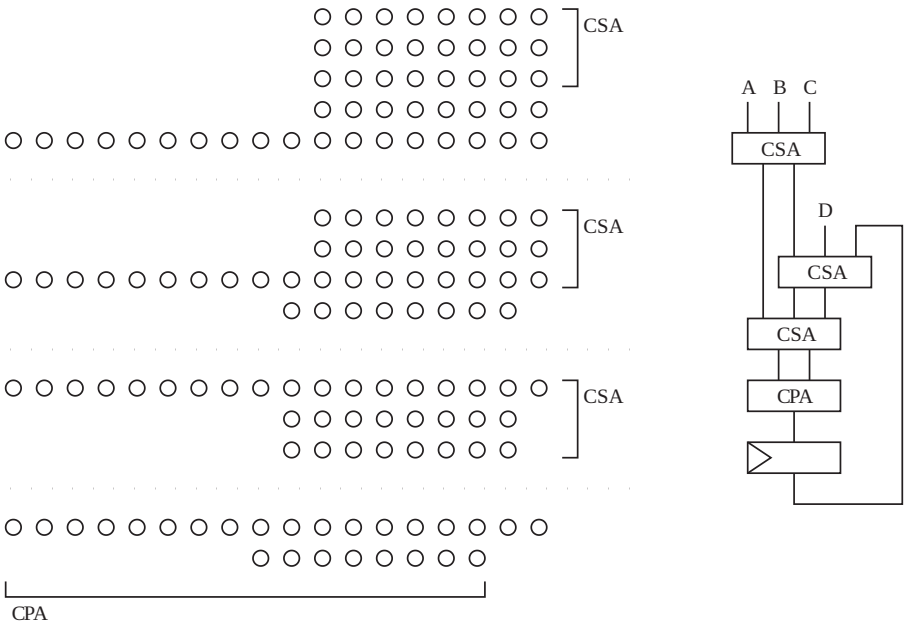
a) What is the minimal size of the register and adder that does not cause overflow?

b) Suppose we can arrange to supply 4 input integers per clock cycle. How would you use carry-save adders to improve its performance over the original (one input at a time) scheme? Sketch a new circuit.

Solution:

a) $8 + \log_2(1024) = 18$ bits.

b)



8 Digital Filter [6 pts]

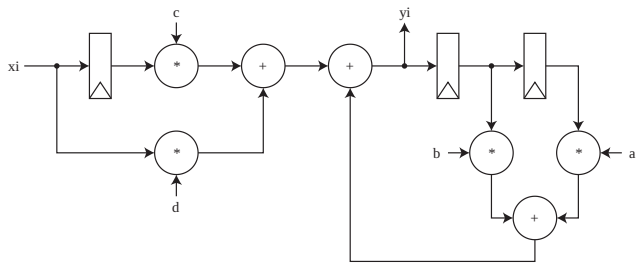
Suppose you need to design a circuit for a digital filter (streaming input/output) with four variable coefficients (a, b, c, and d) and this function:

$$y_i = ay_{i-2} + by_{i-1} + cx_{i-1} + dx_i$$

Assume all values are N-bit wide and overflow is ignored.

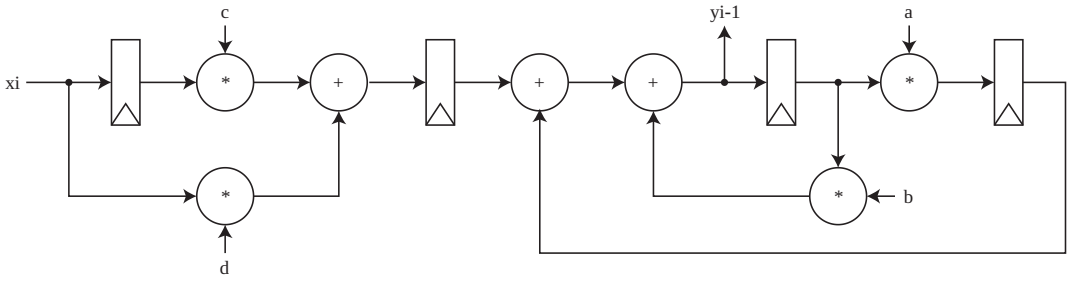
1. Draw a circuit diagram for an *unpipelined version* of this filter using three N-bit registers, and adders and multipliers.

Solution:



2. Now suppose you are given *one* extra N-bit register. Using the extra register how would you reorganize your circuit to acheive higher throughput?

Solution:



9 Greater-Than Comparator [8 pts]

Consider the design of a combinational logic circuit that does a "greater than" comparison between two N-bit unsigned integers, A and B. The circuit outputs a 1 iff $A > B$. It is possible to do this operation with a subtractor, but for this problem we want to design a dedicated comparator. The structure of one solution is similar to that of a ripple adder, with N cascaded stages (except in this case, no sum outputs). Stage i takes as input a_i and b_i, along with a signal from the previous stage, y_i, and generates y_{i+1}. y_0 is set to 0 and y_n is the final output.

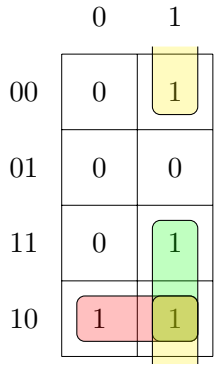
a) Using simple logic gates, design the details of stage i.

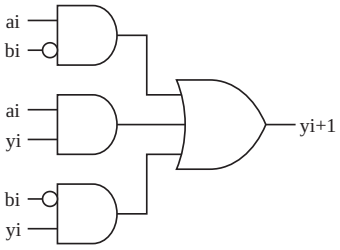
b) Obviously, the solution above will have delay proportional to N. Based on that solution, devise a new circuit with delay proportional to $\sqrt{N}$.

Solution:

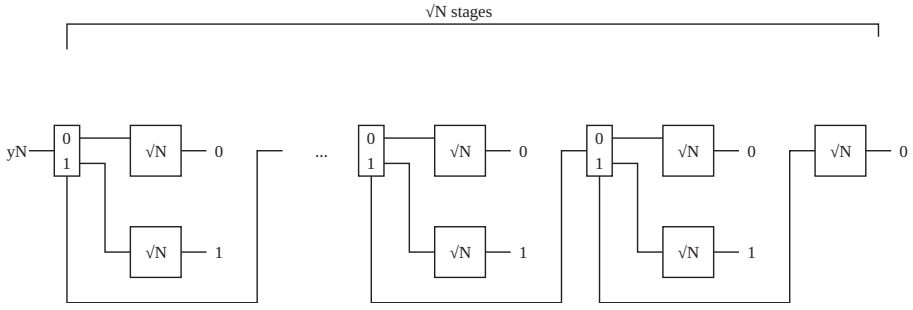
a) $y_{i+1} = a_i b_i' + a_i y_i + b_i' y_i$

a_i	b_i	y_i	y_{i+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1





b) $\sqrt{N}$ stages of $\sqrt{N}$ comparators. The mamximum delay is of $\sqrt{N}$ comparators and $\sqrt{N}$ multiplexors.

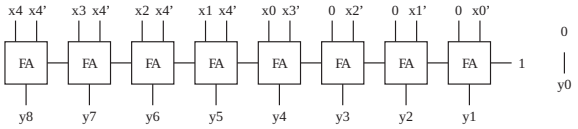


10 Multiplication by a Constant [6 pts]

Using only full-adder cells (FAs), design a combinational logic circuit that produces $14_{10} * X$, where X is a 5-bit signed integer. Minimize the number of FAs and don't worry about delay.

Solution:

$$\begin{aligned}14X &= 16X - 2X = (X << 4) - (X << 1)\\ &= \{x_4, x_3, x_2, x_1, x_0, 0, 0, 0, 0\} - \{x_4, x_3, x_2, x_1, x_0, 0\}\\ &= \{x_4, x_3, x_2, x_1, x_0, 0, 0, 0, 0\} + \{x_4', x_4', x_4', x_4', x_3', x_2', x_1', x_0', 1\} + 1\\ &= \{\{x_4, x_3, x_2, x_1, x_0, 0, 0, 0\} + \{x_4', x_4', x_4', x_4', x_3', x_2', x_1', x_0'\} + 1, 0\}\end{aligned}$$



11 Cross-Bar Switch [6 pts]

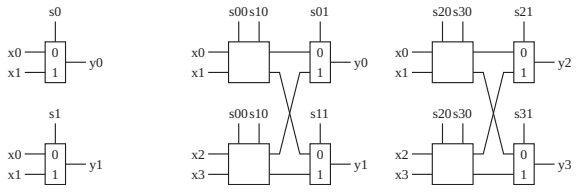
A cross-bar switch is a combinational logic circuit with N-inputs and N-outputs where each output y_i is equal to one of the inputs $x_0, x_1, \ldots, x_{N-1}$.

a) Draw the circuitry for a 2-bit cross-bar switch, using simple logic elements (gates, 2-to-1 multiplexors, etc.)

b) Using your answer from part a) as a building block, and, if needed, additional simple logic elements, design a 4-bit cross bar switch.

Solution:

a) b)



12 Clock [5 pts]

Some circuit has a pair of registers separated by a combinational logic (CL) circuit. The register delay characteristics and the CL circuit delay are as follows:

$$\tau_{setup} = 1 \text{ ns}, \quad \tau_{hold} = 2 \text{ ns}, \quad \tau_{clk-to-q\_MIN} = 1 \text{ ns},$$
$$\tau_{clk-to-q\_MAX} = 2 \text{ ns}, \quad \tau_{CL\_MIN} = 1 \text{ ns}, \quad \tau_{CL\_MAX} = 4 \text{ ns}.$$

Answer the following questions considering clock skew, which in this case is $|\delta| \leq 1$ ns assigned *randomly* (the clock skew between the registers could be plus or minus and up to 1 ns).

a) What is the minimum clock period for safe operation?

b) Is there a potential for a hold time volation (data race)? Explain.

Solution:

a) In the worst case $\delta = -1$ ns and we get the critical path delay $2 + 4 + 1 + 1 = 8$ ns.

b) Yes. With $\delta = 1$, the minimum delay is $1 + 1 - 1 = 1$ ns, which is smaller than the hold time.

13 Power Supply [5 pts]

A logic gate somewhere on a chip is connected to a power supply providing $V_{dd} = 1$ V through a wire that is 1 μm wide and 100 μm long. No other circuits connect to the wire between our logic gate and the power source. However, after our logic gate a large number of other circuits connect. In operation 1A of current is drawn through the power wire.

For our IC process and the particular thickness of the wire, it has a "sheet resistance", R_{sq}, of 1 mΩ per square. (That means—looking down onto the layout—a square segment of a wire has resistance of 1 mΩ, and total resistance $R = R_{sq} * L/W$.)

Is the delay of our logic gate significantly worse than the ideal case? If so, why and approximately by how much?

Solution:

Yes, IR drop happens with $\Delta V = 1 \text{ A} \cdot 1 \text{ m}\Omega/\text{sq} \cdot 100 \text{ sq} = 0.1$ V, where our wire has 100 sq because it is 100 squares in a series. With 0.1 V voltage drop, delay increases by a factor of 1/0.9 and the maximum frequency drops 10%.

14 Pulse-Based Clocking [5 pts]

In the early days designers of digital computers tried using *pulse-based* clocking and latches instead of what we use today—edge-triggered clocking and flip-flops. This was used in the CRAY-1 supercomputer, for instance.

Those systems were organized very much as we organize our systems with combinational logic blocks seperated by state elements. The difference then was that the state elements were level-sensitive latches (and all of the same polarity—for instance, all positive level sensitive latches), and the clock signal was a series of narrow pulses, instead of the square wave we use today.

a) Assuming it works correctly, what would be two advantages of the pulse-based method over our edge-triggered method?

b) What would be two disadvantages?

Solution:

a) Latches take smaller area. The clock signal has smaller load capacitance.

b) If the pulse is too wide, a singnal could go through consequtive latches in a single cycle (similar to what happens in hold time violation). On the other hand, if it is too narrow, the pulse may be filtered out in the clock tree (wire RC causes low pass).

15 ASIC / FPGA [4 pts]

You work as a Engineering Manager at a scooter startup and are in charge of the electronics for the controller. You think using an FPGA is the solution that will cost the company the least amount, but your boss (the CTO) thinks that your team should design an ASIC instead.

The FPGA part you need costs \$100 per chip and you expect NRE costs of \$500K. For the ASIC, the per chip cost will be \$5, and The NRE cost estimate is \$10M. Which approach should you use to save costs—FPGA or ASIC? Explain your reasoning.

Solution:

It depends on the number of chips to manufacture. FPGA cost is $100N + 500,000$, while ASIC cost is $5N + 10,000,000$ for manufacturing N chips. Threshold is $N = 100,000$, where FPGA is better when manufacturing less than that, ASIC is better otherwise.

16 Saturating Counter [6 pts]

Consider the design and Verilog implementation of a module that implements an 8-bit *saturating counter*. When enabled, the counter increments its output, but never goes beyond all ones (255_{10}). On reset it starts at 0. The catch is that for the state element, you are only allowed to use a simple register with no clock enable or reset input signal.

In the space provided below, write the Verilog for the saturating counter and in such a way as to minimize the critical path.

```
module sat_count(input rst, ce, clk, output [7:0] out);
  reg [7:0] in;
  REGISTER #(.N(8)) a(.q(out), .d(in), .clk(clk));
```

Solution:

```
  wire sat;
  assign sat = &out;

  always @(*) begin
  if(sat || (!ce || rst)) begin
    if(rst) begin
      in = 0;
    end else begin
      in = out;
    end
  end else begin
    in = out + 8'd1;
  end
```

```
endmodule
```

17 Boolean Algebra [6 pts]

The exclusive-OR of three variables (expressed as $a \oplus b \oplus c$) computes the *parity* of the three variables—i.e., is equal to 0 for an even number of 1s and 1 for an odd number.

a) Derive the canonical SOP form for $a \oplus b \oplus c$ from truth table.

b) Using the algebraic definition of exclusive-or ($x \oplus y = xy' + x'y$) and algebraic manipulation, derive the same SOP for $a \oplus b \oplus c$.

Solution:

(a) $a'b'c + a'bc' + ab'c' + abc$

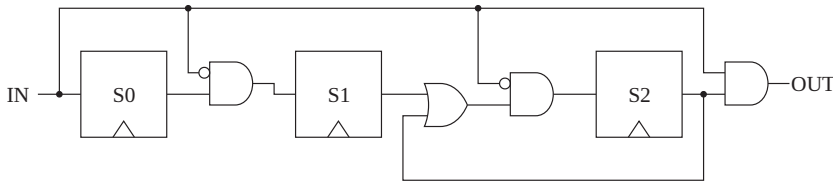
a	b	c	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(b)

$$\begin{aligned} a \oplus b \oplus c &= (ab' + a'b) \oplus c \\ &= (ab' + a'b)c' + (ab' + a'b)'c \\ &= ab'c' + a'bc' + ((a' + b)(a + b'))c \\ &= ab'c' + a'bc' + (a'b' + ab)c \\ &= ab'c' + a'bc' + a'b'c + abc \end{aligned}$$

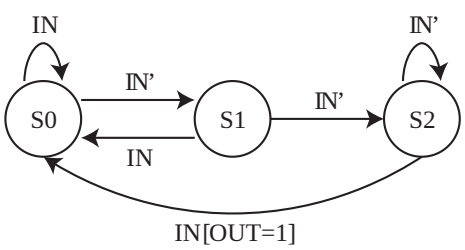
18 Finite State Machine Implementation [8 pts]

Shown below is a one-hot encoded Mealy FSM with a single input, IN, and single output, OUT. On reset it starts in state S0. Convert this to an binary-encoded state machine with the following encoding: S0=00, S1=01, S2=11. You don't need to show the final circuit diagram, but must derive the next state and output logic equations.



Solution:

Let {s1,s0} be the current state and {n1, n0} be the next state.



s1	s0	IN	n1	n0	OUT
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	1	1	0
0	1	1	0	0	0
1	0	0	-	-	-
1	0	1	-	-	-
1	1	0	1	1	0
1	1	1	0	0	1

	0	1
00	0	0
01	1	0
11	1	0
10	-	-

n1

	0	1
00	1	0
01	1	0
11	1	0
10	-	-

n0

	0	1
00	0	0
01	0	0
11	0	1
10	-	-

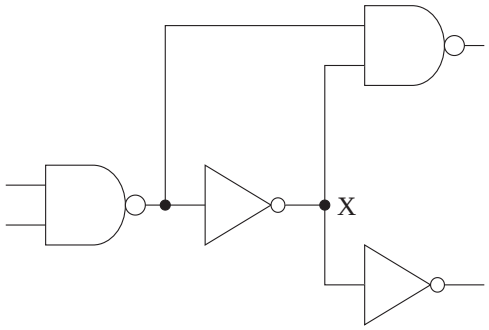
OUT

$$n1 = s0 \cdot IN', \quad n0 = IN', \quad OUT = s1 \cdot IN$$

19 Gate Delay [7 pts]

Assume an inverter in our technology has input capacitance of $3c$, and has equal delay pulling up and pulling down of $r(3c + C_L)$, where C_L is the output load capacitance. Likewise, a NAND-gate has input capacitance $2c$ (on each input) and delay $2r(3c + C_L)$. It also has equal pullup and pulldown delay (in this case assuming only one transistor pulling up).

1. For the circuit shown below, write an expression for the delay from the inputs of the leftmost NAND to the point "X", in terms of r and c.



Solution:

The first NAND gate has $C_L = 3c + 2c$, and the following inverter has $C_L = 3c + 2c$. The delay at the point X is $2r(3c + 5c) + r(3c + 5c) = 24rc$.

2. Suppose now that we would like to scale up the NAND gate to give it twice the drive strength. Write an expression for the delay of this scaled up NAND gate. Also, what is its input capacitance of one input?

Solution:

It has the same parasitic delay but a half fanout delay, so delay is $2r(3c + C_L/2)$. All internal transistors are twice wider, so it has $4c$ for input capacitance.

Build timestamp: Tuesday 9th May, 2023 21:22