EECS 151/251A Homework 9

Due Monday , November 25th, 2019

Problem 1: Carry-Lookahead

In this problem, we would like to build an 8-bit carry-lookahead adder

a) You are given the following building blocks. Write down the expression for each building block.

a)

$$G = A \cdot B$$

$$P = A \oplus B$$

$$S = A \oplus B \oplus C$$

b)

$$P = p0 \cdot p1$$

$$G = g0 \cdot p1 + g1$$

c)

$$C_{out} = C_{in} \cdot P + G$$

b) Construct a 8-bit CLA adder with the building blocks, and highlight its critical path on the diagram.

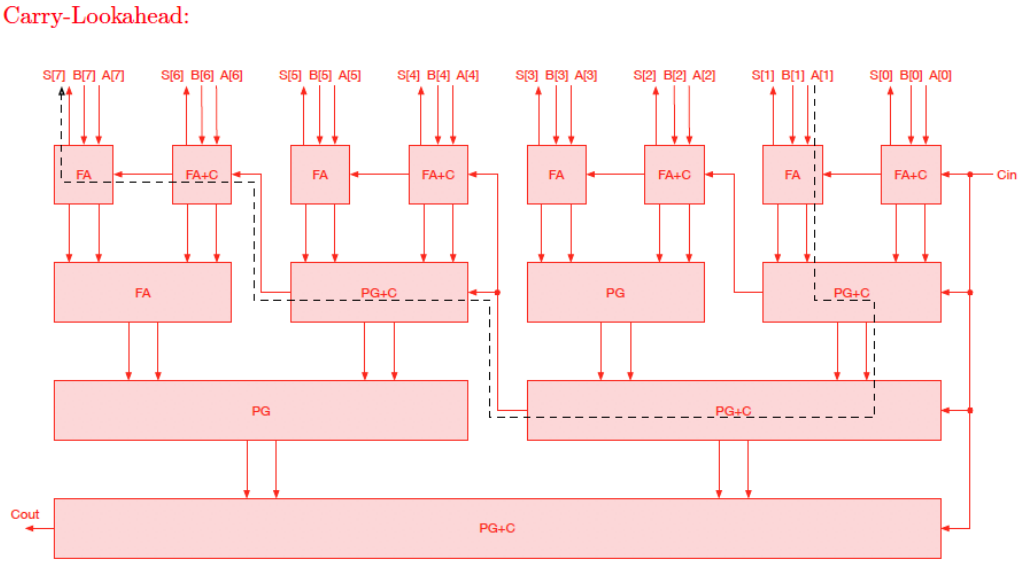

Figure 1: Carry-Lookahead adder

c) Now we would like to build a 8-bit Kogge-Stone adder to optimize for delay. Again you are given the building blocks shown below. Notice that $P_{i:j}$ and $G_{i:j}$ denote the propagate and generate functions for a group of bits from i to j. Derive the expression for each building block.

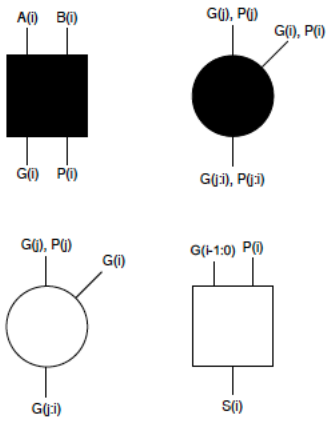

Figure 2: Building Blocks for Kogge-Stone Adder

Black square: $G_i = A_i \cdot B_i$, $P_i = A_i \oplus B_i$
Black circle: $G_{j:i} = G_j + P_j \cdot G_i$, $P_{j:i} = P_j \cdot P_i$
White circle: $Gj : i = G_j + P_j \cdot G_i$
White square: $S_i = P_i \oplus G_{i-1:0}$

d) Using the logic blocks, construct an 8-bit Kogge-Stone adder with a carry input and a carry output. User a radix-2 implementation, and highlight the critical path of your design. How does this compare to the previous design?

The first stage is the black boxes: here we generate the bit propagate (P_i) and generate (G_i) signals that will be used by the tree. For the actual tree, the Kogge-Stone implementation first groups the (P_i,G_i) in groups of 2, therefore generating ($P_{1:0}$,$G_{1:0}$),($P_2 : 1$,$G_{2:1}$) etc. Then those signals are grouped again in groups of 2 to form ($P_{3:0}$,$G_{3:0}$),($P_{4:1}$,$G_{4:1}$) etc.

The key here is that you need to incorporate the C_{in} signal into the tree. Remember that in order to get a sum bit you need $S_i = P_i \oplus C_i = P_i \oplus G_{i-1:0}$. Therefore for S_0 you need P_0 and C_{in}, for S_1 you need P_1 and $G_0 + P_0 \cdot C_{in}$ etc. So we add the white circles to generate the carries needed for the final sum, including the C_{in}.

The critical path of this design is shorter than the previous implementation.

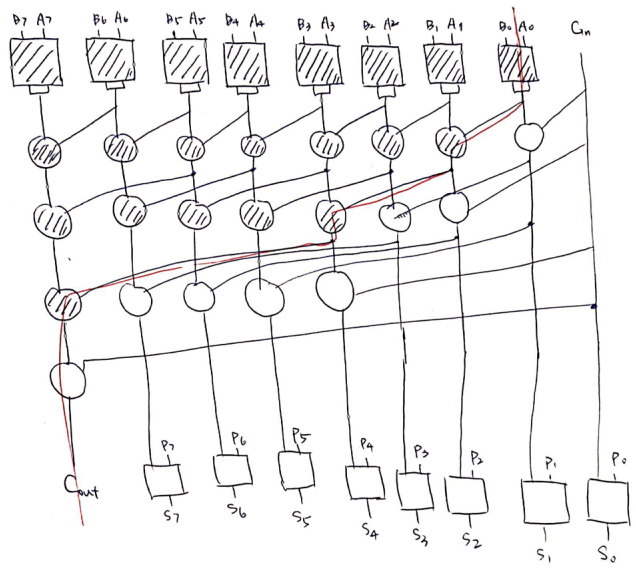

Figure 3: Radix 2 Kogge-Stone Adder

Problem 2: Multipliers

In this problem we will explore several structures of 4x4 multiplier. You are allowed to use Full Adders, Half Adders and AND gates.

a) Draw the array multiplier, and calculate its critical path using t_{carry}, t_{sum} and t_{and}.

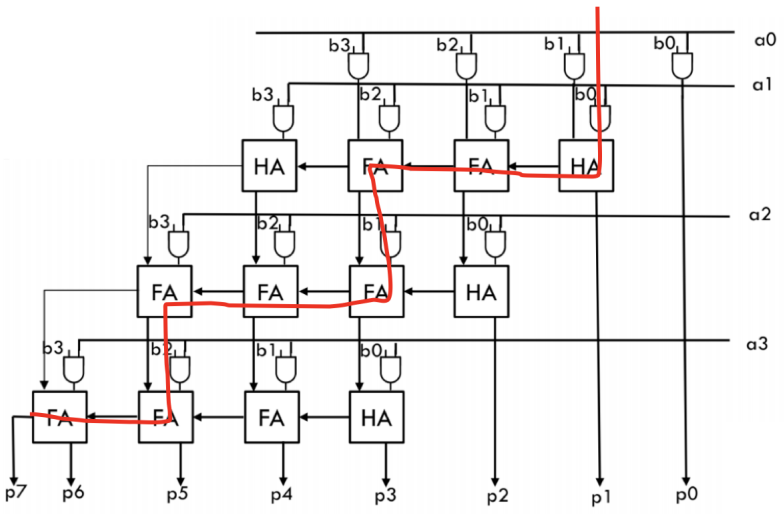

Figure 4: 4x4 Array Multiplier

$t_{critical} = t_{and} + 5 \times t_{carry} + 3 \times t_{sum}$

b) Implement the same adder using the Carry-Save structure, and compute the new critical path. (gray blocks are critical path)

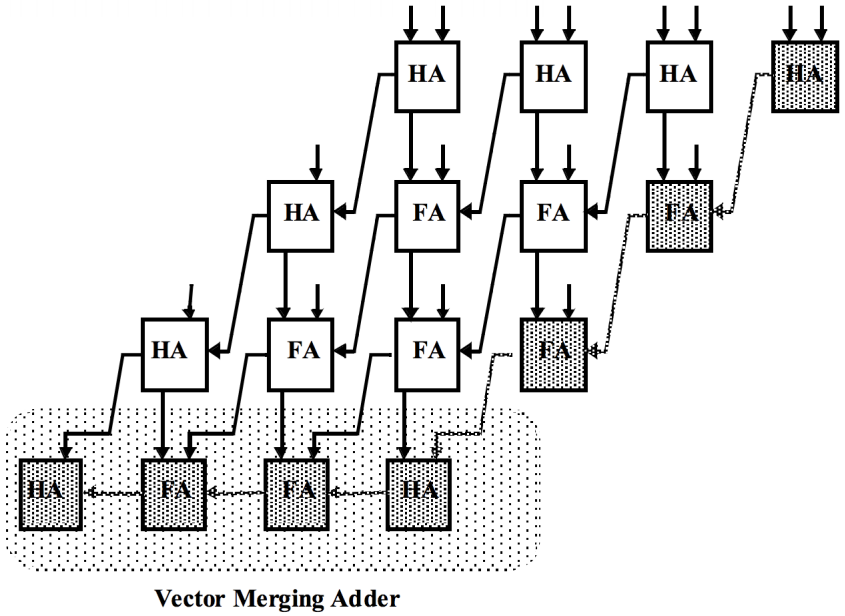

Figure 5: 4x4 Carry-save Multiplier

$t_{critical} = t_{and} + 6 \times t_{carry} + t_{sum}$

c) Draw the wallace tree dot diagram, and implement the tree structure. What is the critical path now?

Refer to diagrams on lecture 16 slides 16 and 17
$t_{critical} = t_{and} + 3 \times t_{carry} + 3 \times t_{sum}$

Problem 3: Latches/FF

1. Derive the truth table for the following circuit.

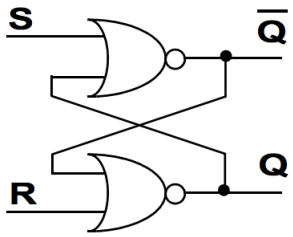

Figure 6: SR latch

S R	Q $\bar{Q}$
0 0	Q $\bar{Q}$
0 1	0 1
1 0	1 0
1 1	x x

2. This circuit can be used as a latch, but it has a potential problem. Explain why it can be used as a latch, and what is the problem?

   This circuit can be used as a latch because when S and R are both 0, Q and $\bar{Q}$ hold their previous values. 0 and 1 can be written by setting S to 0 and 1, and R to 1 and 0. However, if both inputs are 1, the output is undefined.

3. We propose a solution for the problem above. Rederive the truth table for the structure below, and explained how it can be used as a latch.

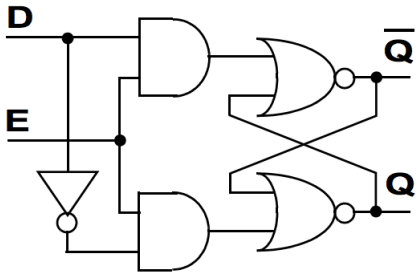

Figure 7: Transparent D latch

E D	Q $\bar{Q}$
0 0	Q $\bar{Q}$
0 1	Q $\bar{Q}$
1 0	0 1
1 1	1 0

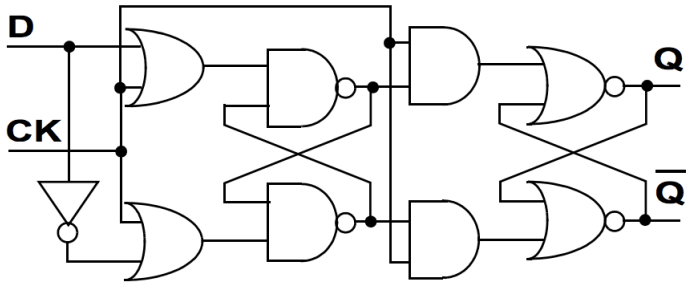

Figure 8: Master-slave flip flop

4. Finally, we have come up with the following flip-flop structure. Is this flip flop posedge triggered or negedge triggered?

   This flip flop is positive edge triggered. Because in the slave latch, the AND gates pass the values at their input to the output on the rising edge of the clock.

5. In this structure, we call the first SR latch the master latch, and the later one slave latch. Assume Q was originally storing 0. When a 1 appears at D, describe what happens for the master and slave during one clock cycle.

   In one clock cycle, while CK is low, the OR gates in the master latch pass the value at D to the output of the cross-coupled NAND gate pair. The NAND gates in the slave latch cannot pass any value thus Q remains unchanged. While clock goes high, the master latch maintains its output value, and the slave latch passes 1 to Q.

Problem 4: Timing

Consider the following logic function. The minimum and maximum delays of the logic modules are annotated on the figure. The flip-flops have the following timing properties: $t_{clk-q} = 50ps$, $t_{setup} = 50ps$, and $t_{hold} = 25ps$. You may assume that the clock has no jitter.

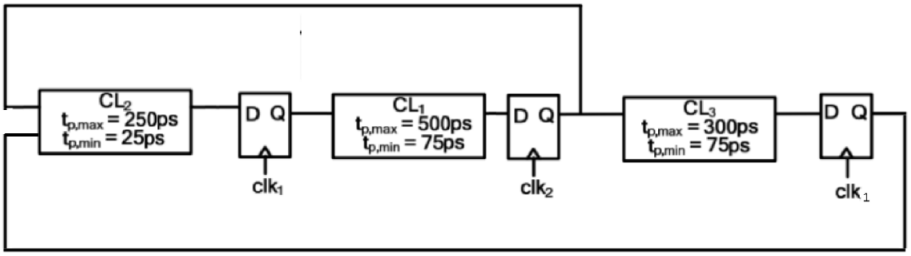

Figure 9: Circuit diagram

1. Assuming there is no skew between clocks, what is the minimum clock cycle time for this pipeline?

   The longest critical path is CL1.

   $$t_{clk-q} + t_{p,max,CL1} + t_{setup} \leq t_{clk}$$

   $$T_{clk} \geq 50ps + 500ps + 50ps$$

   $$T_{clk} \geq 600ps$$

2. Under the condition established so far, does the circuit meet all hold time requirements?

   The shortest delay is CL2.

   $$t_{clk-q} + t_{p,min,CL2} + \geq t_{hold}$$

   $$50ps + 25ps \geq 25ps$$

   $$75 \geq 25ps$$

   No hold-time violations.

3. Now assume that $clk1$ and $clk2$ can be randomly skewed relative to each other by up to +/- 60ps, what is the minimum clock cycle time? Does this solution cause hold time violations?

   The first thing to note is that the skew is given relative to each other. This means that the maximum skew that can be added in a cycle is 60ps (not 120ps). Since the max delay of path CL1 is much longer than that of the other paths (i.e. much more than 60ps), this is going to be the worst case for delay. The worst-case occurs when the skew shortens the cycle. i.e. if we use clk1 as a reference ($t_{skew,clk1} = 0$) then $t_{skew,clk2} = -60ps$ would give us the worst case.

   $$t_{clk-q} + tp, max, CL1 + t_{setup} \leq T_{clk} - t_{skew,max}$$

   $$T_{clk} \geq 50ps + 500ps + 50ps + 60ps$$

   $$T_{clk-q} \geq 660ps$$

For hold time:

$$t_{clk} + t_{p,min,CL1} \geq t_{hold} + t_{skew,max}$$

$$50ps + 250s \geq 25ps + 60ps$$

$$75ps \geq 85ps$$

So there is hold-time violation.

Problem 5: SRAM

Consider the 8T SRAM cell given below. With this design, there is a Write Word Line (WWL) that is used to write the values of Write Bit Line (WBL) and $\overline{WBL}$ into the cell, and a separate Read Word Line (RWL) that is used to read the content of the cell on the Read Bit Line (RBL)

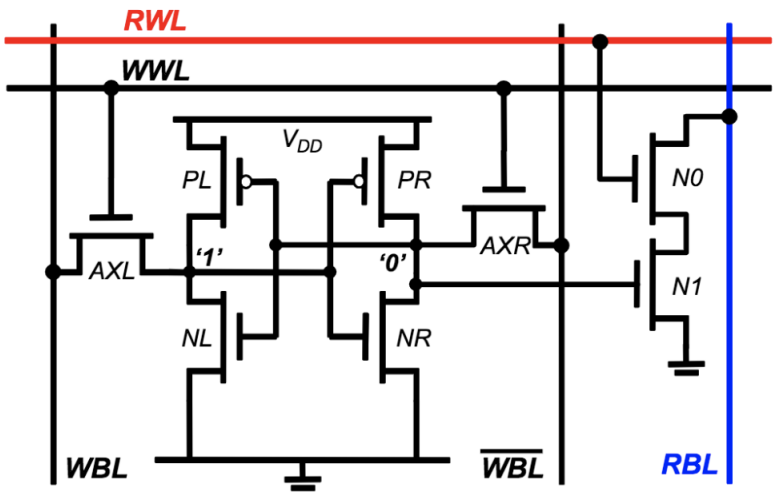

Figure 10: 8T SRAM

1. Determine which transistors are involved in a Write operation, and comment on their relative sizing.

   Same as 6T SRAM cell, AXL and AXR need to be stronger than PL and PR.

2. Determine which transistors are involved in a Read operation, and comment on their relative sizing.

   Read is done through N0 and N1; no sizing constraints for stability.

3. Compare this structure with the 6T SRAM cell. What are the advantages and disadvantages?

   Decoupled read and write operations poses less constraints on cell sizing. In a normal 6T SRAM, the pull down (PD) must be stronger than then access transistor /pass gate (PG), and PG stronger than PU is required for writability. Therefore,changing PG will trade off read stability for writability. By not using any of these transistors for reads, this trade-off disappears. In this case, both the PD and PU can be minimum sized.

4. Qualitatively explain how increasing cell supply voltage (without changing other signal levels) affect the read stability, read access time and writability of the cell.

   Increasing VDD strengthens the SRAM cell's ability to retain data. Therefore, read stability is improved. Read access time is thus increased, and writability decreases.