CDA 4213/CIS 6930 CMOS VLSI
Fall 2019

Final Project
Design Proposal (10%)

Due date
11:59PM, Wednesday, 6th November

Today's Date:	11/07/2019
Your Team Name:	Joseph?
Team Members:	Joseph Daniel Moreno
No. of Hours Spent:	
Exercise Difficulty: (Easy, Average, Hard)	
Any Other Feedback:	

(1) **(10 pts)** From the project description, write down the requirements in the form of R1, R2, etc…
Add as many bullets as you need below.

a) Requirement 1 (R1):
Create an N-bit by N-bit multiplier; N = 8 is the minimum

b) Requirement 2 (R2):
Implement the rectangular layout of the multiplier.

c) Requirement 3 (R3):
Inputs and final output are fed in/out serially.

d) Requirement 4 (R4):
The multiplicand, multiplier, and product should be registered.

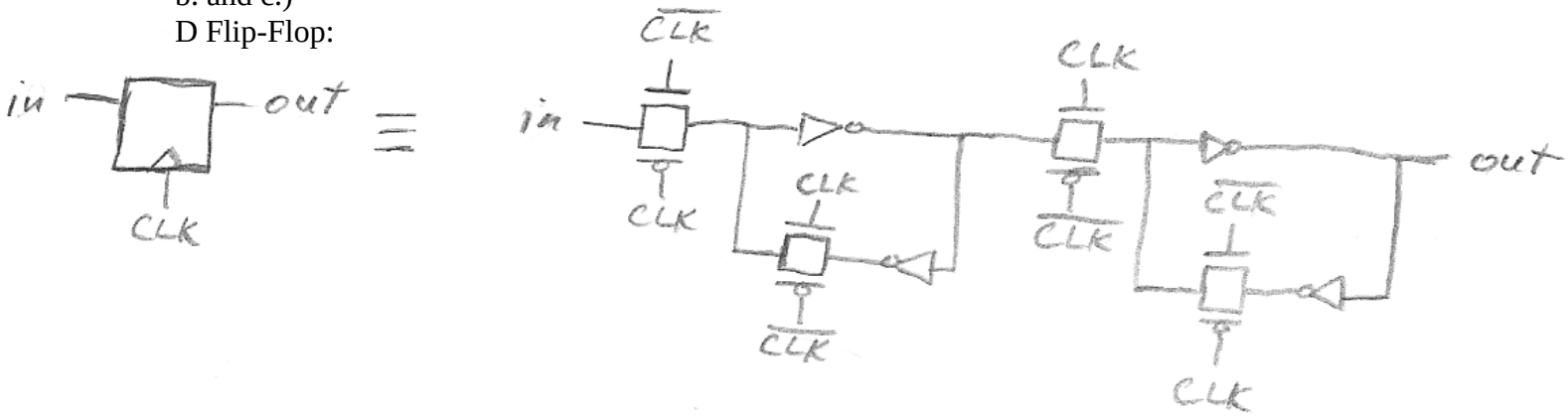
e) Requirement 5 (R5):
The three registers are clocked by an external clock signal.

(2) **(50 pts.)** Proposed Design:

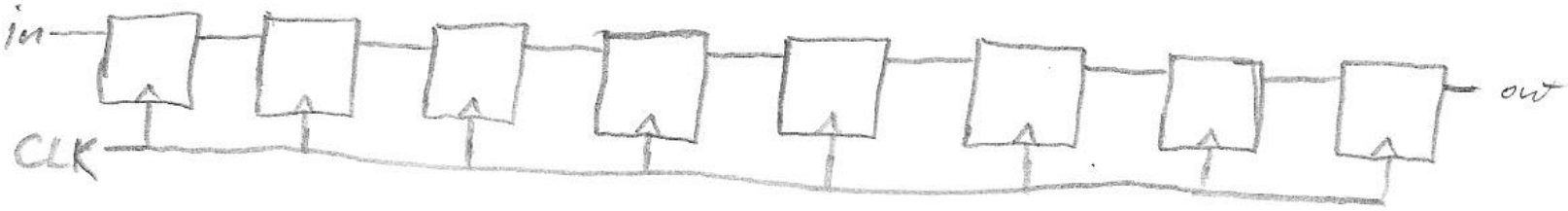
(a) List all module bit-slices you will need for your design.
(b) For each bit slice, show the gate-level design.
(c) Show the block level design of your register. NOTE: Use Flip-flops for your registers. Do not use Latches.
(d) Show an overall block-level diagram of your design. Briefly, explain how each of the above requirements will be met by your design.

a.) 8-bit shift register, carry-save adder, full adder, and D flip-flop.

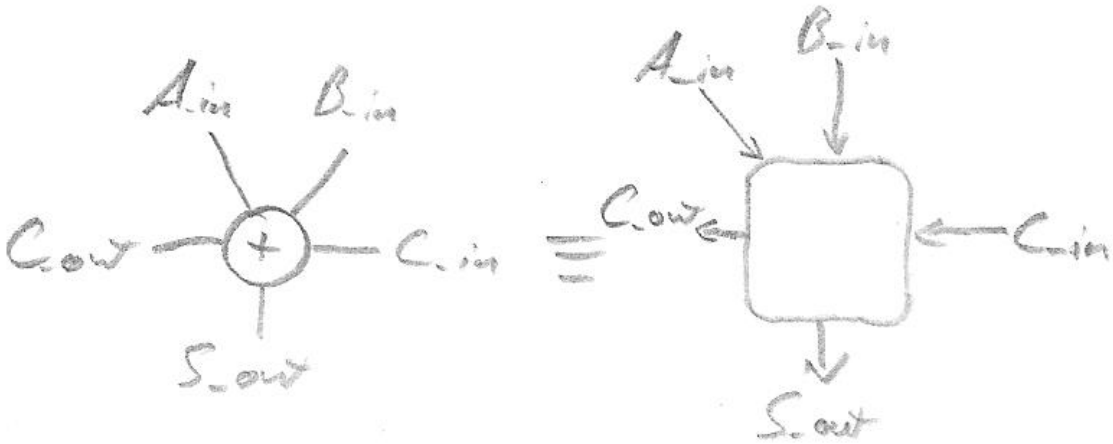
b. and c.)
D Flip-Flop:



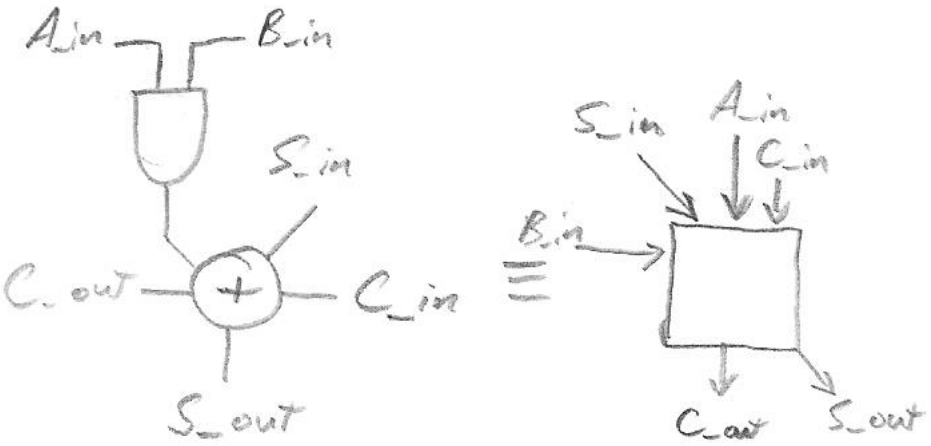
8-Bit Shift Register:



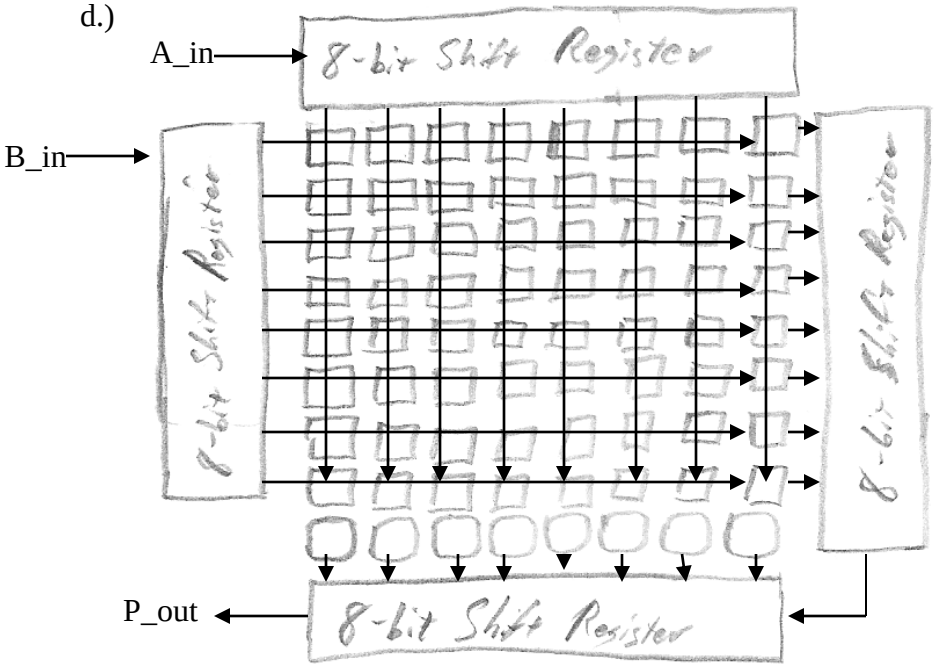
Full Adder:



Carry-Save Adder:



d.)



There are 64 carry-save adders and 8 full adders to calculate the product of two unsigned 8-bit numbers (R1). The inputs/outputs of each CSA and full adder will be fed in/out so that the design is rectangular (R2). A_in, B_in, and P_out will be fed in/out of their shift registers serially (R3, R4). An external clock signal will pulse into the shift registers (R5).

(3) **(10 pts)** *Area Estimation:*

(a) Estimate the area of your design. You already have the area numbers for the cells you have designed in lab assignments. Add more rows if needed.
Remember that you have to maximize N.

No.	Cell Name	Individual Cell Area (Sq. Micron)	No. of instances	Total Area	Remarks
1	inverter	145.8624	3	437.5872	CLK, RST, and TEST need to be inverted.
2	D Flip-Flop	3,277.45	2	6,554.9	
3	8-bit shift reg.	30,264.66	4	121,058.64	
4	full adder	732.501	8	5,860.008	
5	carry-save adder	1,506.0475	64	96,387.04	
6					
7					
Total Estimated Area				287,872.719	25% of 230,298.1752 added for routing

(b) Estimate the size of the multiplier you can fit in a die area of 960 um x 960 um.

Based on my total estimated area, a 25 by 25 multiplier could fit in an area of 960 µm x 960 µm; about 230,300 μm^2 is for 8 bits, so 230,300 μm^2 * 3.125 = 719,687.5 μm^2 plus 25% of that total equals 899,609.375 μm^2.

(4) **(10 pts)** Work Distribution:

	Team Member Name	*Briefly explain the tasks responsible for*
1	Joseph Moreno	Drawing/planning, layout creation, simulating and correcting layout, writing up reports
2		
3		

(5) **(10 pts)** Project Schedule & Milestones. *Provide a weekly schedule with important milestones.*

Week of November 11th:
Create a functioning 8 by 8 multiplier without the shift registers.

Week of November 18th:
Add shift registers on top of functioning 8 by 8 multiplier; input needs to be clocked in properly and output needs to be clocked out properly. Demo the partial project.

Week of November 25th:
Add RST (reset) and TEST functions to the layout.

Week of December 2nd:
Add internal clock to the layout. Run test cases and record results for the project report.

Week of December 9th:
Demo the final project and submit the report.

(6) **(10 pts)** Design Validation Plan:

Explain the plan for your design validation by simulation. Consider the following:

a) What input vectors will you test your design with?

b) Which input vector(s) will exercise the critical path (longest path)?

c) How will you estimate the maximum clock speed?

a.)

A_in	**B_in**
0000 1000	0001 0000
1111 0000	0000 1111
0000 0110	1100 0000
0001 0010	0000 0011
1111 1111	0000 0001
0100 0000	0000 0111
0010 0000	0000 0010

b.) A_in = 1111 0000, B_in = 0000 1111
A_in = 1111 1111, B_in = 0000 0001

c.) Add up the delay along the critical path:
csa_d = carry-save adder delay, fa_d = full adder delay, cp_d = critical path delay
f = fastest clock speed

(8 * csa_d) + (8 * fa_d) = cp_d => **1 / cp_d = f**