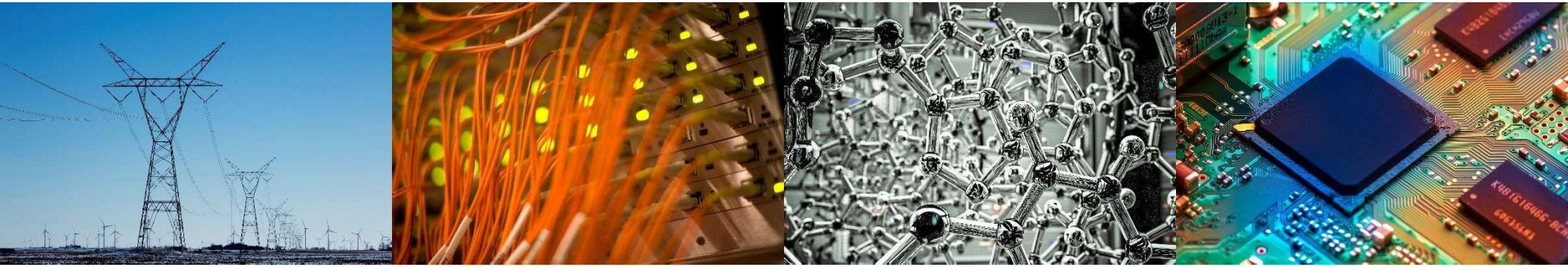
Fall 2017
ECE 120 Midterm 2
HKN Review Session



Abhinav Das, Aamir Hasan, Utsav Kawrani

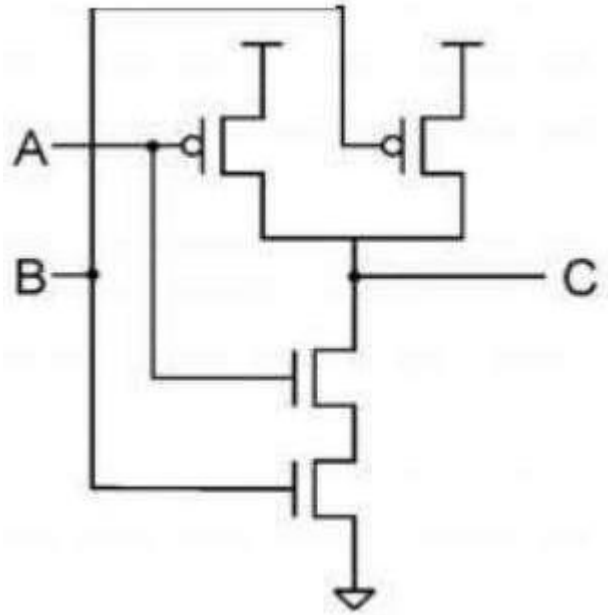
Topics

- Transistors
- Boolean Logic and Logic Gates
- Karnaugh Maps
- Bit sliced design
- Multiplexers and Decoders
- Sequential logic
- Registers

CMOS

CMOS: Complementary Metal-Oxide-Semiconductor Transistor.

CMOS circuits contain PMOS transistors and NMOS transistors.



NMOS & PMOS

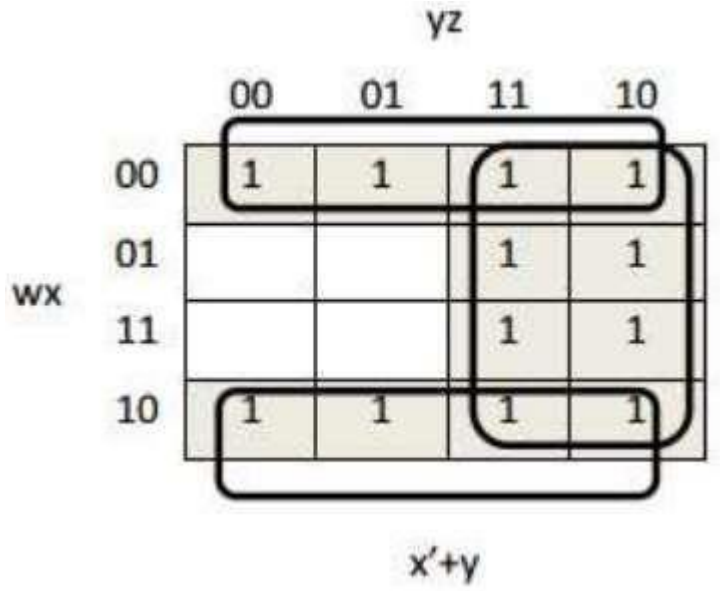
- When a high voltage is applied to the gate, NMOS will conduct.
- When a high voltage is applied to the gate, PMOS will not conduct
- Parallel and series duality. (DeMorgan's)

Boolean Algebra Properties

- Suppose A and B are literals. Then, A·B (AB) reads "A AND B", A+B reads "A OR B", A' reads "NOT A"
- To find the dual:
  - Swap all 0's and 1's (just the values, do not complement the variables).
  - Swap AND with OR.
- DeMorgan's laws:
  - (A+B)' = A'B'
  - (AB)' = A' + B'
- Alternatively, to complement an expression, you can find its dual and swap variables and complemented variables.
- A(B+C) = AB + AC

Karnaugh maps

- Gray code order: to put adjacent cells together
- Minimal expression can be derived by grouping neighboring cells into powers of 2



- Implicant—a rectangular cover of 1's or X's
- Prime implicant—an implicant (containing at least one minterm) that is not wholly covered by a single other implicant
- Cover the remaining 1's using as few prime implicants as possible
- In other words, find minimum number of rectangles to cover all 1's in K-map, each rectangle as large as possible

- POS to SOP:

  (A+B)(C+D+E) =AC+AD+AE+BC+BD+BE
- SOP to POS:
- use Boolean algebra distributivity property: A+BC=(A+B)(A+C)
- Example:

  wx'+y+vz

  = (w+y)(x'+y)+vz

  = [(w+y)(x'+y)+v][(w+y)(x'+y)+z]

  = (w+y+v)(x'+y+v)(w+y+z)(x'+y+z)

Canonical forms

- Boolean function expressed as a sum of minterms is termed the *canonical sum of products* form of the function
- Example:

  f = x'y'z' + x'y'z + x'yz' + x'yz + xyz'

- Boolean function expressed as a product of maxterms is termed the *canonical product of sum* form of the function
- Example:

  $f = (x' + y + z)(x' + y + z')(x' + y' + z')$

- JUST a binary to decimal
- Max = 0, Min = 1

Don’t Cares

- Implicants can include Don’t cares but cannot be made up of only don’t cares.
- Use don’t cares to make your SOP or POS simpler. Include them only when it reduces the SOP/POS expression.

CD \ AB	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	d	1	0	d
10	d	d	d	d

SOP/POS to CMOS

- SOP => AND/OR
- AND/OR => NAND
- Example: F = A'B + BC + D

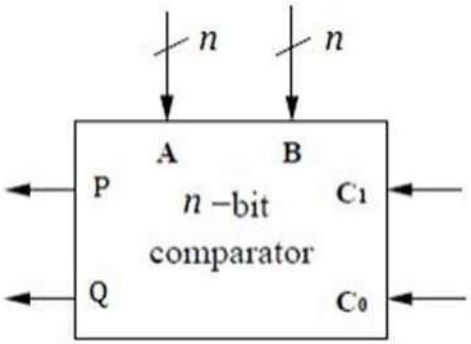
- POS => OR/AND
- OR/AND => NOR
- Example: F = (A + B + C')(A + C)(B')

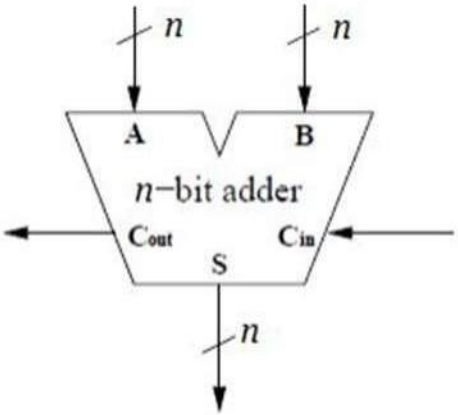
Bit Slice Design

- Multiple units work together to get a task done.
- Each unit takes in the output of the previous unit, performs its computation and sends its output to the next unit.
- Common units that have bit slice design:
  - Adders - Ripple Adder
  - Comparators

Bit-Slice Design

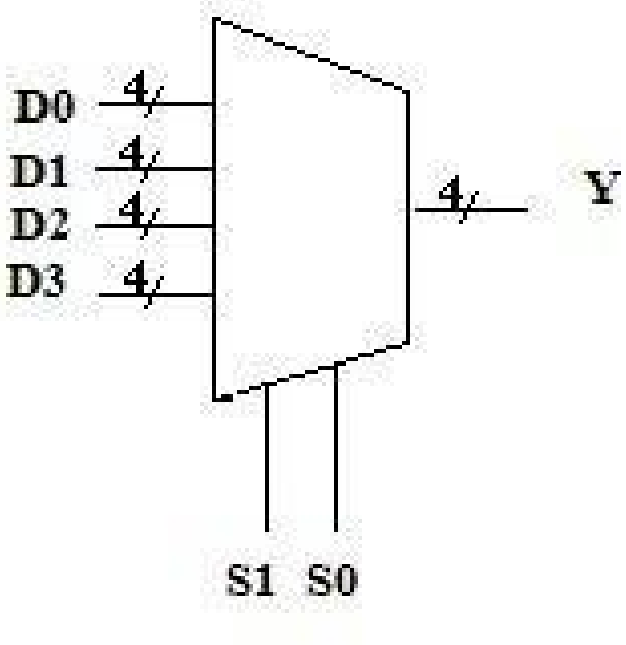
- Comparator has two outputs
  - Represent A=B, A>B, A<B (4th combination is a dont care)
- Adder produces sum bit and carry bit

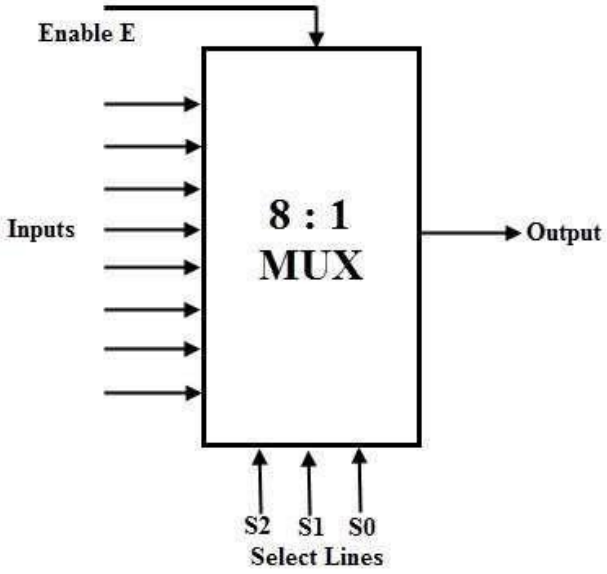


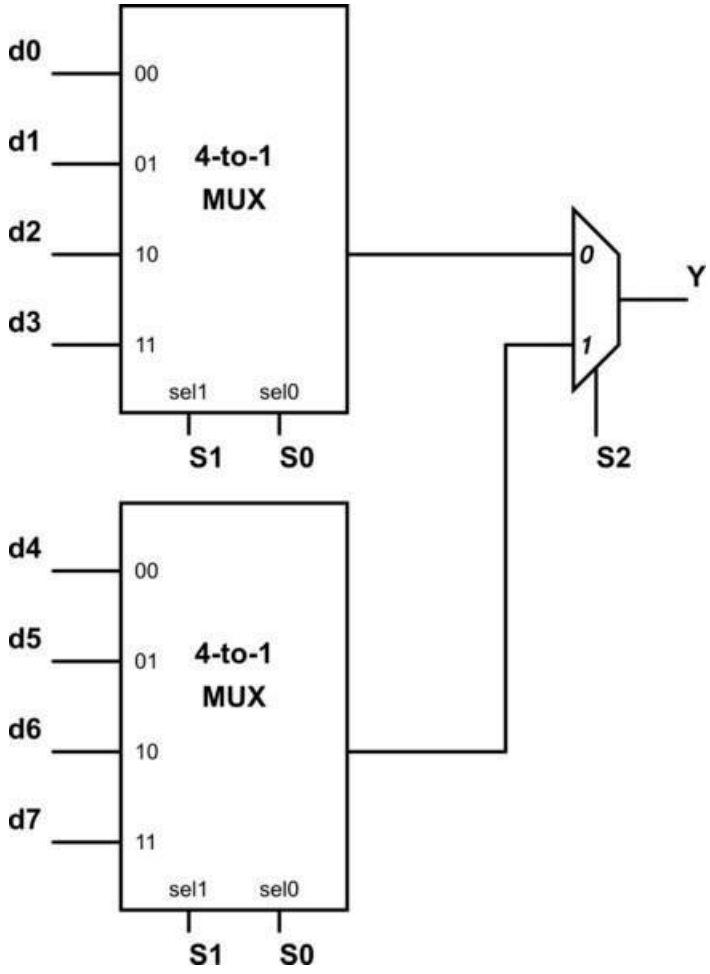


MUX

2^ns inputs for n select bits

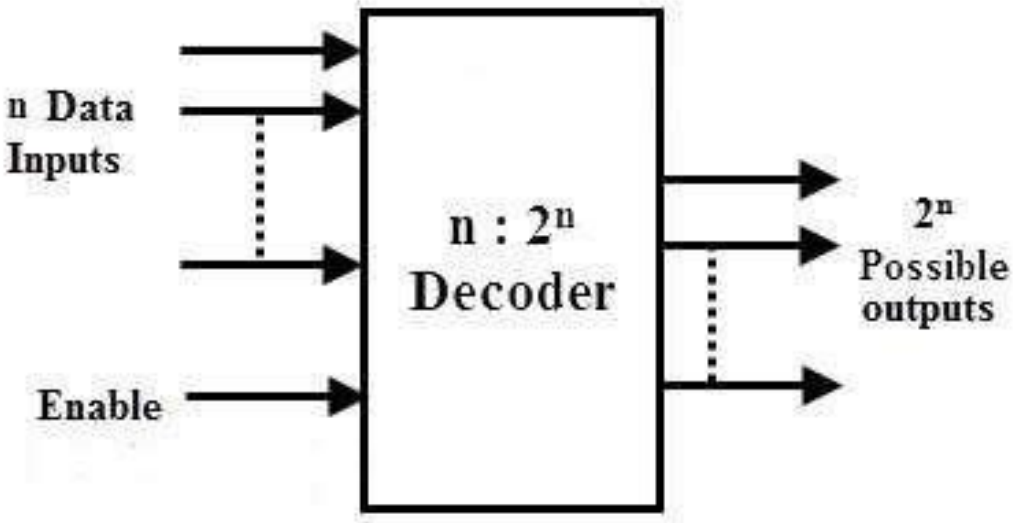






Decoders

- Decoder is a minterm generator. Any n-variable function can be implemented using a 2^n decoder and a single OR gate.
- N inputs 2^n outputs.

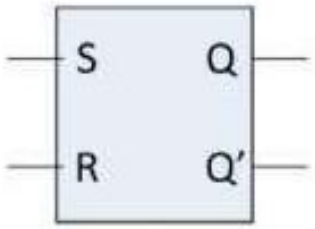


Combinational and Sequential Logic

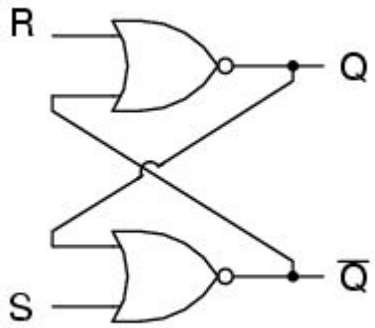
- Combinational circuit: Output is a function of its input ONLY
- Sequential circuit: Depends on the current and past inputs.

Latches

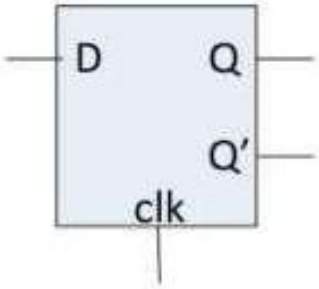
- **SR Latch**



S R	Q^+
0 0	Q (hold)
0 1	0 (reset)
1 0	1 (set)
1 1	forbidden



- **D Latch**

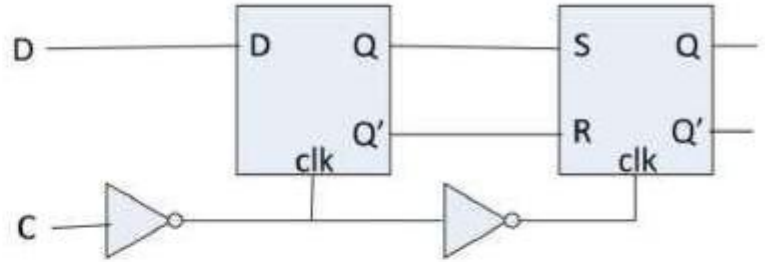


C	D	R	S	Q^+	
1	0	1	0	0	reset
1	1	0	1	1	set
0	0	0	0	Q	no change
0	1	0	0	Q	no change

SP16 Midterm

Flip-flops

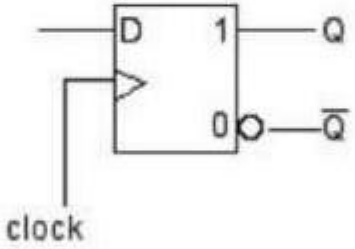
- D flip-flop



- Master D latch — Slave SR or D latch

D flip-flop

D	Q+
0	0
1	1



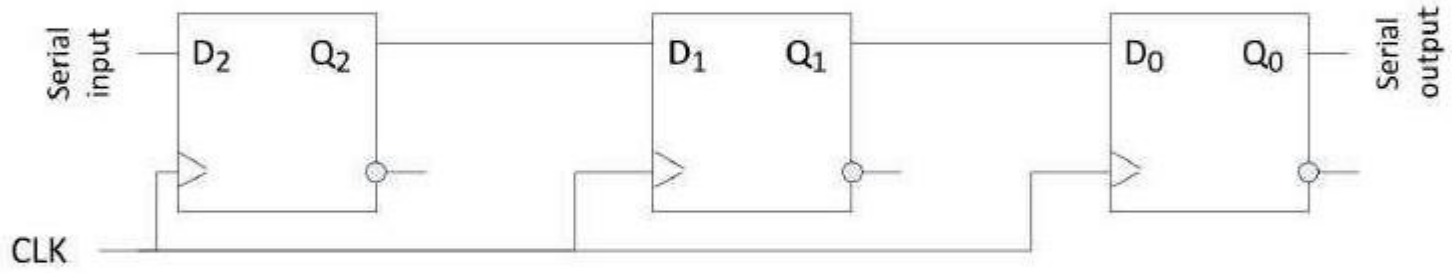
Changes happens only on the edges of clock (positive edge or negative edge)

- Other kinds of flip-flops: T, SR, JK....

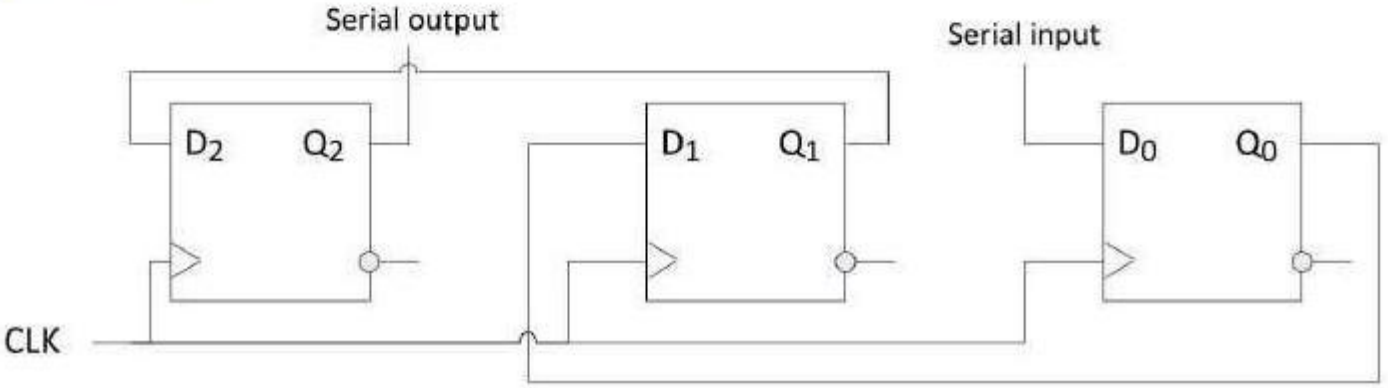
Registers

It seems like a group of flip-flops

- **Logical Shift Right:**

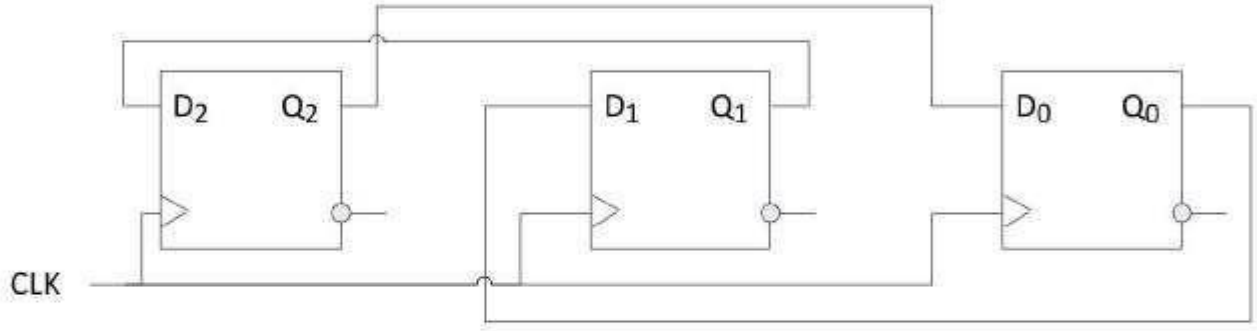


- **Logical Shift Left:**

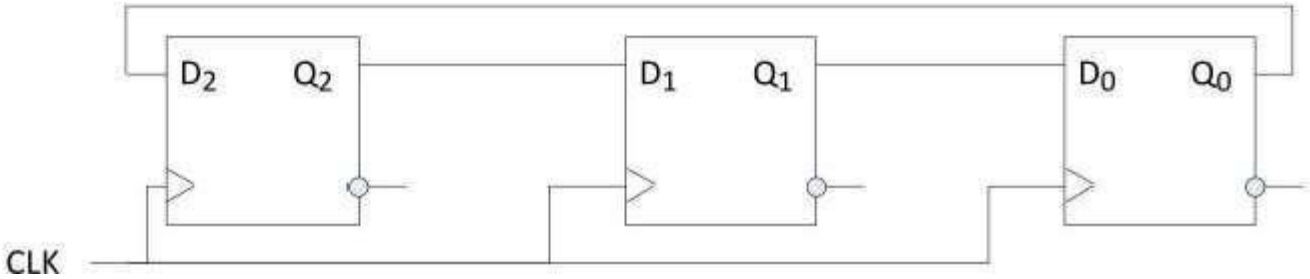


Registers

- **Circular Shift Left:**

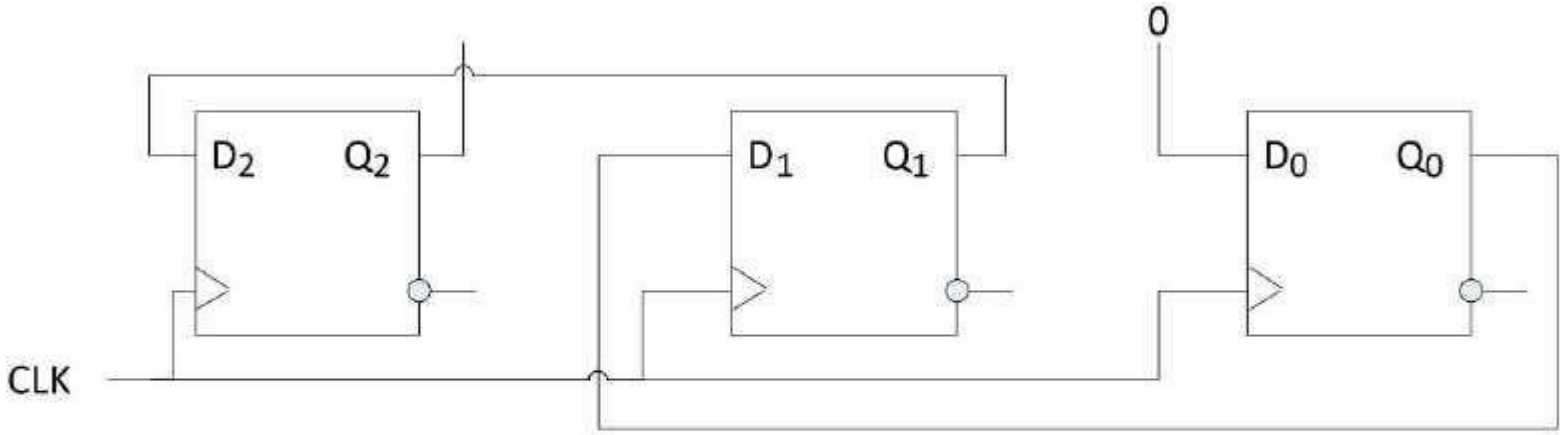


- **Circular Shift Right:**

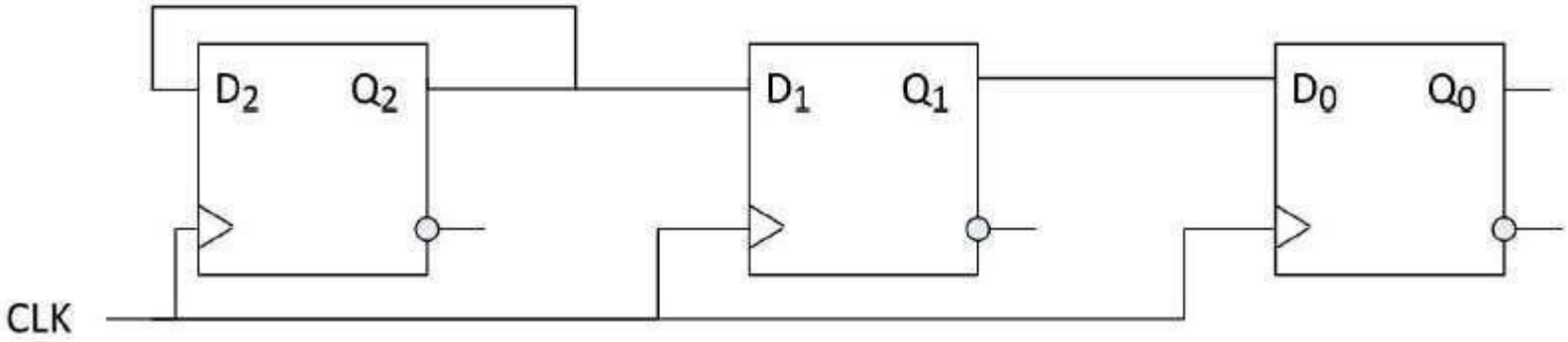


Registers

- Arithmetic Shift Left: Must shift in 0



- Arithmetic Shift Right: Can't change sign bit



Questions

- What is the dual of:

  A + (BC) + (0 (D+1))?

- What is the complement of

  F = AB (C + (DL'G(B' + A + E))) (H + (J'A'B))

- Let f(w,x,y,z)=m_9. Find its dual and write it in Mi notation.

Previous Exam Questions

Problem 2 (14 pts): Canonical forms

A committee has members A, B, and C. Variables a, b, c have value 1 iff A, B, C respectively vote in favor of a proposal. Design a combinational circuit whose output g is 1 iff there is a majority in favor.

1. Fill in truth table.

a	b	c	g
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

- Write the canonical SOP using literals and using minterm notation.
- Write the canonical POS using literals and using maxterm notation.

- For function f(a,b,c,d)=a'bc+a'cd', write corresponding canonical SOP.
- For function g(w,x,y,z)=(w+x')(w+x'+y+z'), write corresponding canonical POS.

wx \ yz	00	01	11	10
00	X	0	1	1
01	1	0	1	X
11	1	1	0	0
10	0	X	0	1

wx \ yz	00	01	11	10
00	X	0	1	1
01	1	0	1	X
11	1	1	0	0
10	0	X	0	1

1. List the **essential prime implicants**.

   Answer: _____

2. Give a **minimal SOP** expression for `f(w,x,y,z)` and **show the corresponding loops** on the left map.

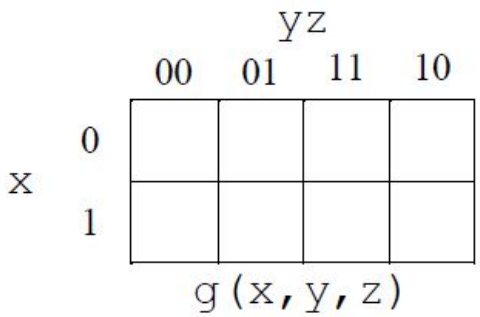
   Answer: _____

3. Give a **minimal POS** expression for `f(w,x,y,z)` and **show the corresponding loops** on the right map.

4. Do your answers to **Part 2** and **Part 3** represent the same Boolean function? Justify your answer.

5. Let $g(x, y, z) = x(y \oplus z)$. Fill in its K-map on the right and write **minimal POS** below. Show the corresponding loops.

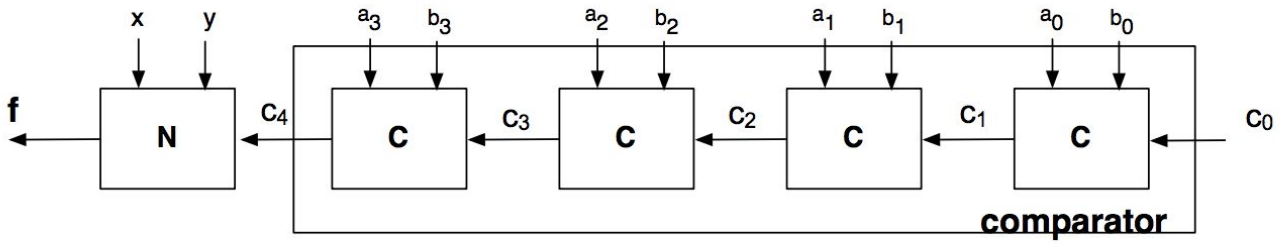
Answer: _____



Problem 4 (14 points)

In this problem you will complete the design of the circuit shown below, which compares two 4-bit unsigned binary numbers $A=a_3a_2a_1a_0$ and $B=b_3b_2b_1b_0$ and outputs

$$f = \begin{cases} x \text{ NAND } y & \text{if } A < B \\ x' \text{ OR } y & \text{if } A \geq B \end{cases}$$



1. (8 points) Design cell C so that the comparator portion of the above circuit operates correctly and outputs

$$c_4 = \begin{cases} 0 & \text{if } A < B \\ 1 & \text{if } A \geq B \end{cases}$$

a. Specify the input c_0.

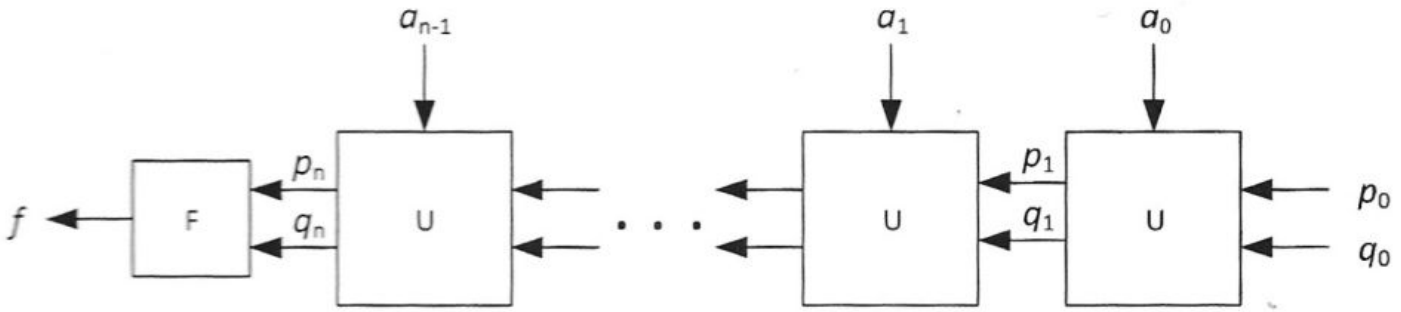
c_0 = _____

b. Express c_{i+1} in terms of c_i, a_i, b_i.

c_{i+1} = _____

SP16 Midterm

Design a *bit-slice circuit* that checks whether an unsigned integer $A=a_{n-1}a_{n-2}..a_1a_0$ is a power of 2, starting with the least significant bit.



The U cell should be designed so that the *n*-bit network shown above correctly checks if the unsigned integer number A is a power of two. The final output f should be 1 iff A is a power of two. (*Hint*: If a number is a power of two, how many 1s are in its binary representation?)

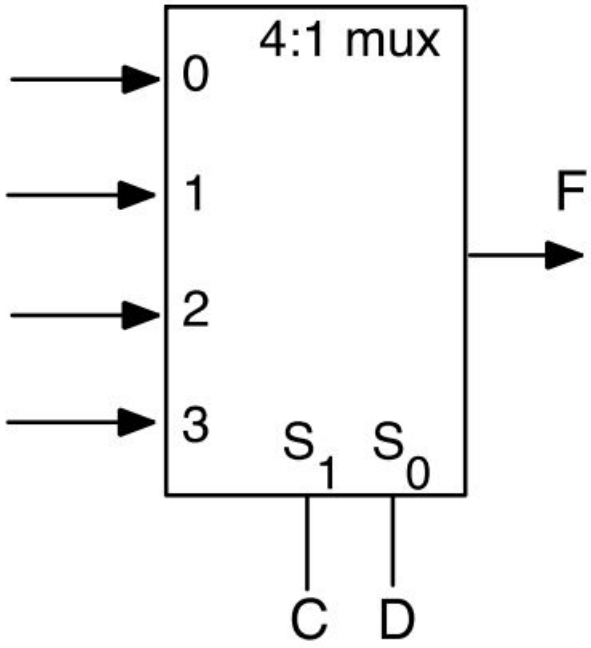
1. List the possible 'answers' (or information) that your bit slice may need to communicate to the next bit slice (and receive from the previous bit slice). One such answer is already provided for you to get started.

p_i	q_i	**Meaning**
0	0	Have not seen any 1s so far
0	1	
1	0	
1	1	

FA13 Midterm

2. What are initial values for p_0 and q_0?

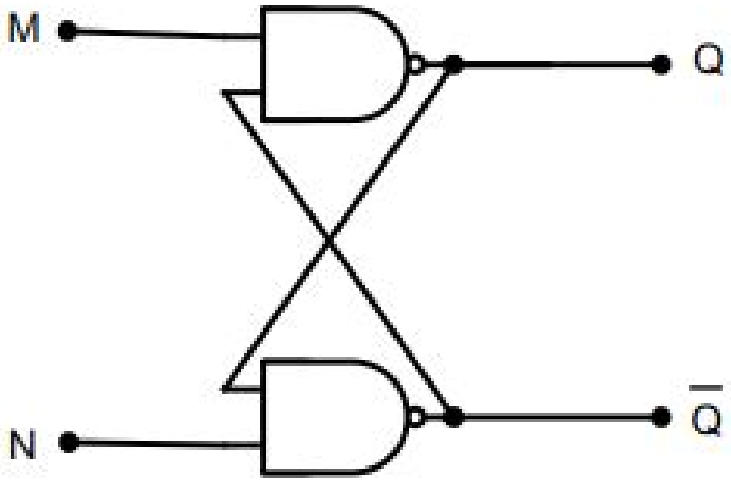
2. **(8 points)** Consider function F(A,B,C,D) as defined in the following K-map. Implement F using **only** the multiplexer given below. **You may not add any components, nor any additional gates.** Complemented inputs are **not** available.



AB \ CD	00	01	11	10
00	0	x	1	x
01	1	0	x	1
11	1	0	x	1
10	x	x	1	0

F(A,B,C,D)

SP16 Midterm



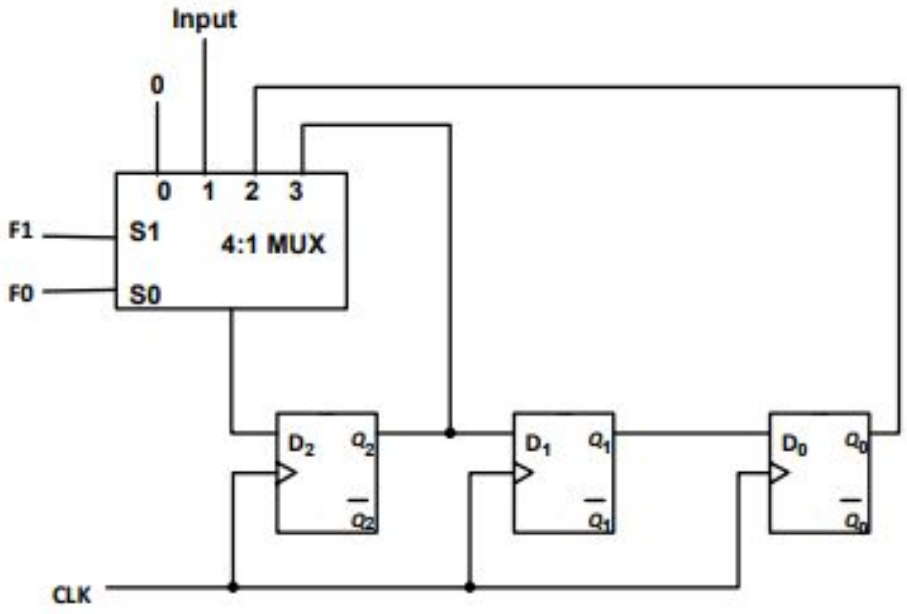
a. Complete the next-state table for this circuit

M	N	Q^+
0	0	Forbidden
0	1	
1	0	
1	1	

b. Express the next state Q^+ as a function of *M*, *N*, and *Q* in **SOP form**.

Q^+ = _____

2. (9 points) Consider a 3-bit shift register that has the following diagram:



a. Determine the functionality of the register by completing the following table

F_1	F_0	Operation
0	0	Unused
0	1	
1	0	
1	1	

b. If the shift register initially stores $Q_2Q_1Q_0$=100 and *Input*=0, what is stored in the register after one clock pulse and

$F_1 F_0 = 01$? _____

$F_1 F_0 = 10$? _____ (Assume again that 100 is stored before the operation.)

$F_1 F_0 = 11$? _____ (Assume again that 100 is stored before the operation.)

Useful link

- http://lumetta.web.engr.illinois.edu/120-S17/ (search Steve Lumetta on google and click the first link).
- https://www.reddit.com/r/aww/ (procrastination)