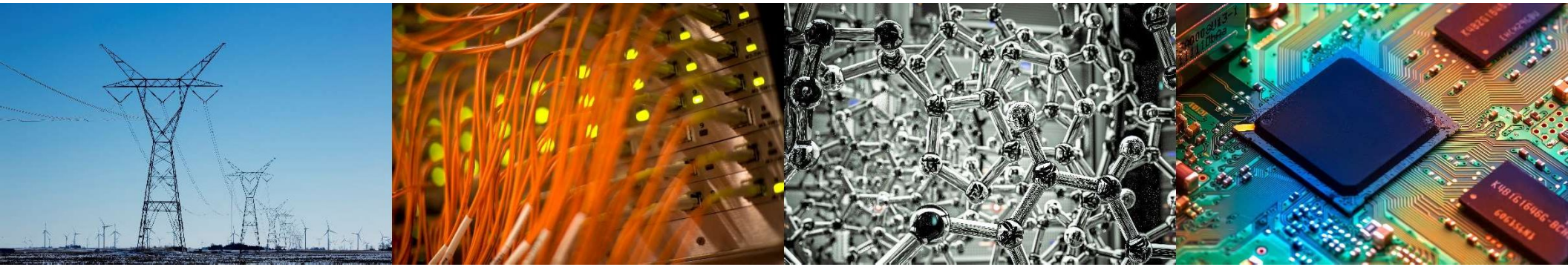


Spring 2017

ECE 120 Midterm 2

HKN Review Session



Reviewers: Quanhua Huang, Janish Yerra, Raymond Mai

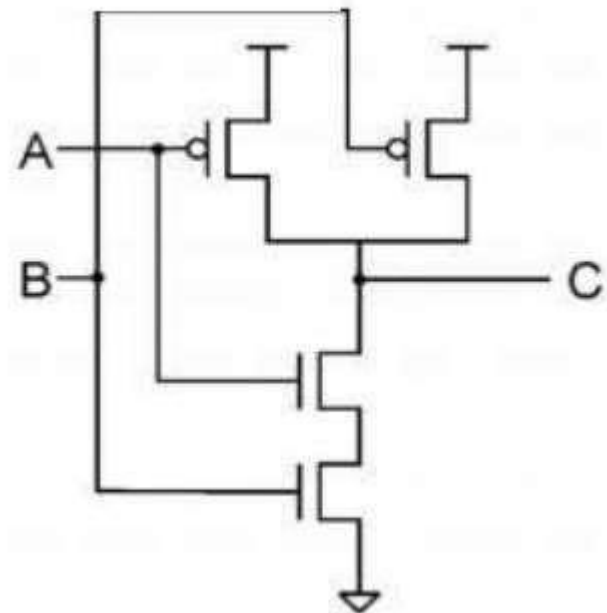
Location: ECEB 1013

Time: 2-4pm

CMOS

CMOS means Complementary Metal-Oxide-Semiconductor Transistor.

CMOS circuit contains PMOS transistor and NMOS transistor.

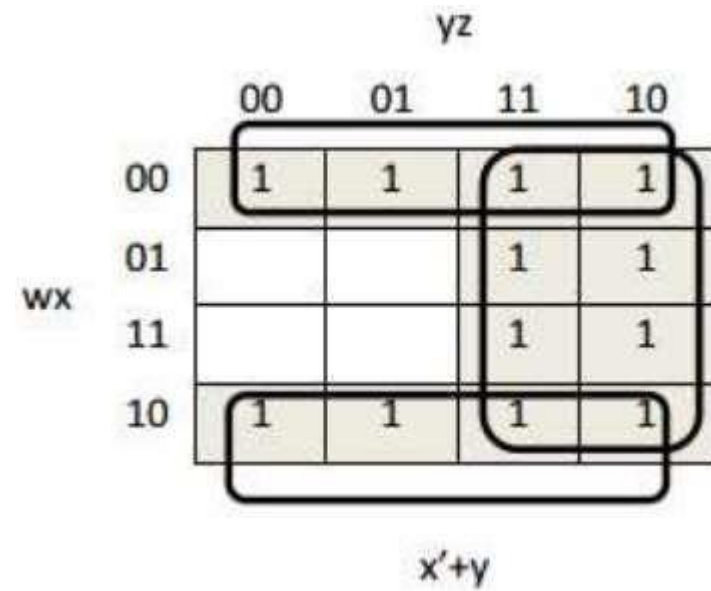


NMOS & PMOS

- In a NMOS, carriers are electrons
- When a high voltage is applied to the gate, NMOS will conduct
- In a PMOS, carriers are holes.
- When a high voltage is applied to the gate, PMOS will not conduct
- Parallel and series duality.

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

Boolean Algebra Properties

- Suppose a and b are literals. Then, $a \cdot b$ as well as ab reads “ a AND b ”, $a+b$ reads “ a OR b ”, a' reads “NOT a ” (same as \bar{a})
- Writing truth tables
- POS: $(a+b) \cdot (x'+y)$
 - Circle all rows where $f(x, y, \dots) = 0$ SOP: $ab+x'y+abxy$
 - Circle all rows where $f(x, y, \dots) = 1$

- 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.

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

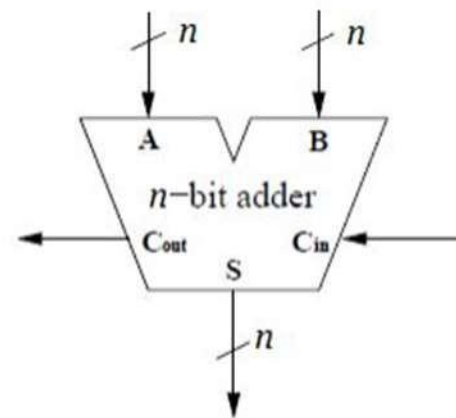
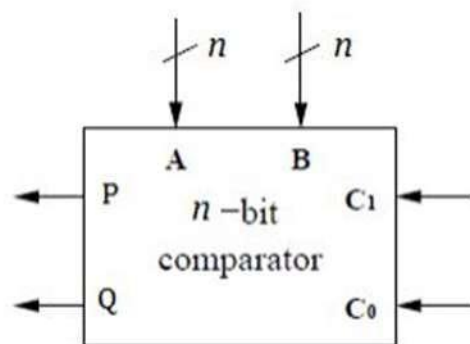
SOP/POS to CMOS

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

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

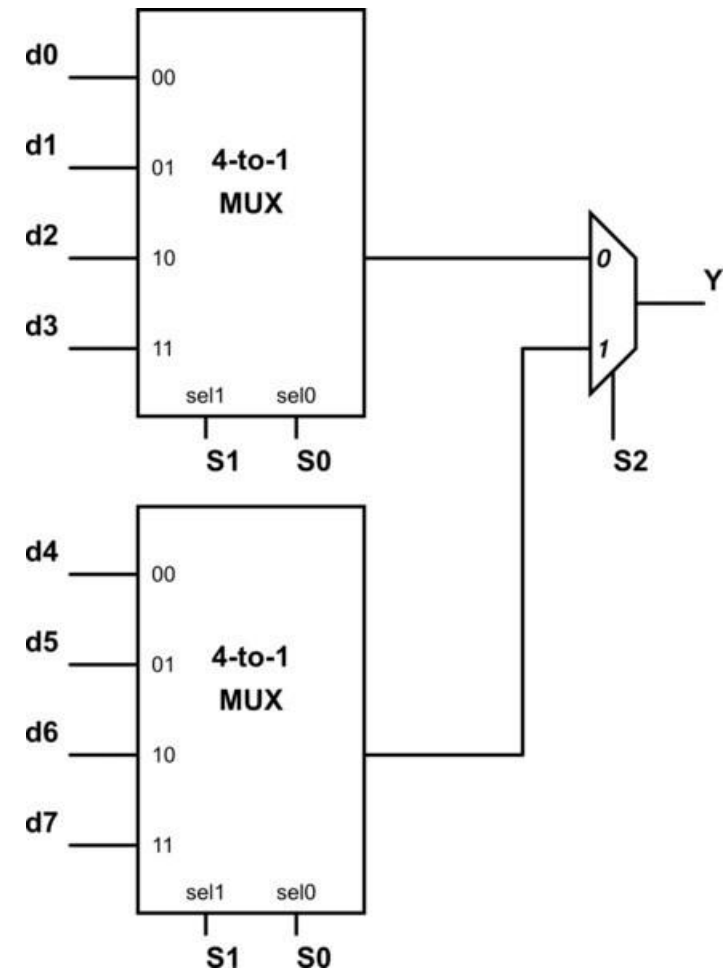
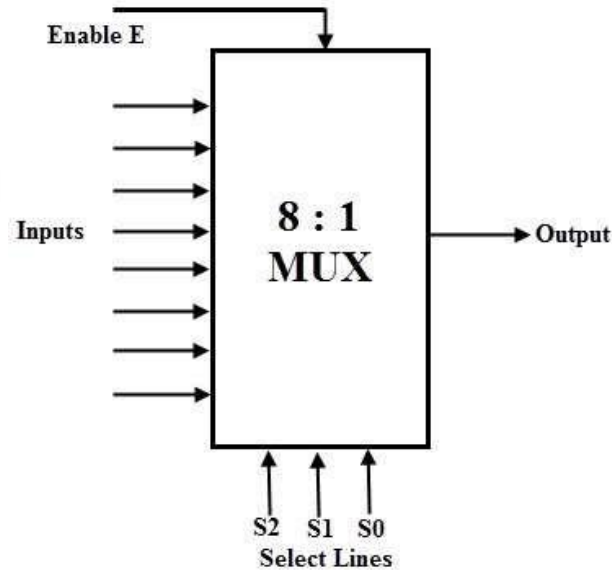
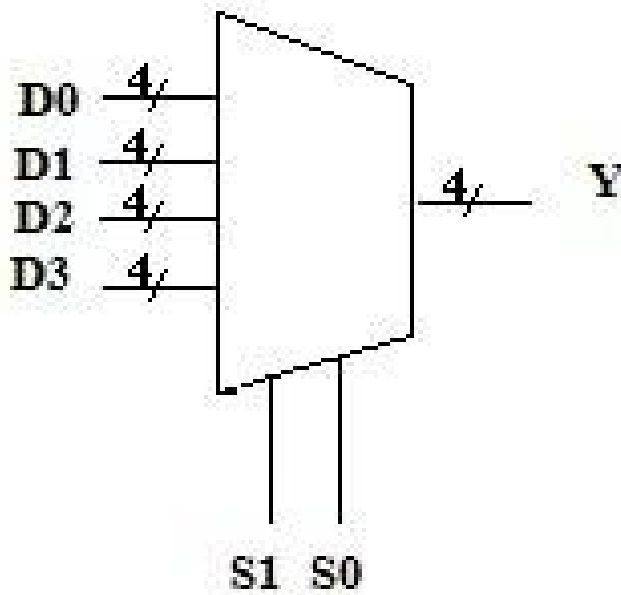
Bit-Slice Design

- Comparator has two outputs
 - Represent $A=B$, $A>B$, $A<B$ (4th comb. unused)
- Adder produces sum bit and carry bit



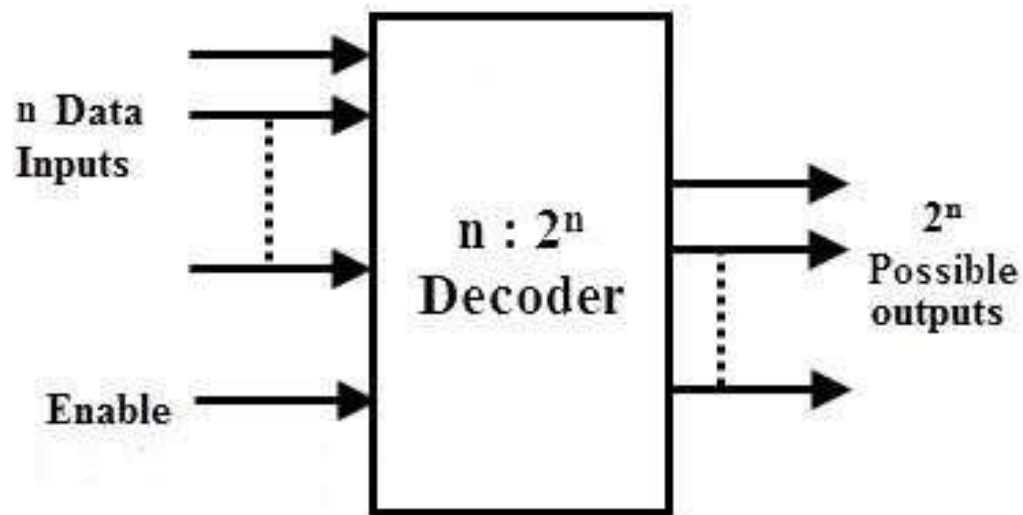
MUX

2^n s 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.

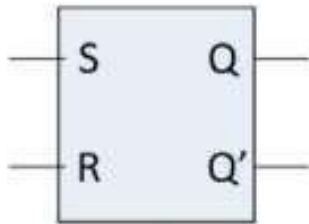


Combinational and Sequential Logic

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

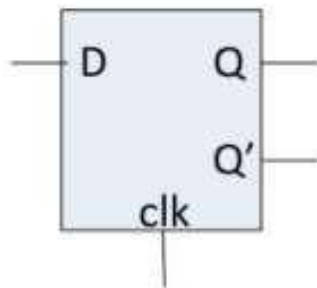
Latches

■ SR Latch



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

■ D Latch



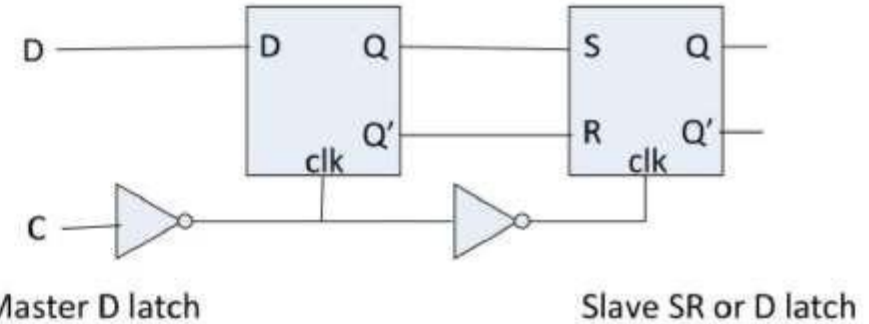
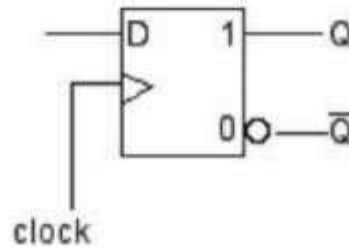
| C | D | R | S | Q ⁺ | |
|---|---|---|---|----------------|-----------|
| 1 | 0 | 1 | 0 | 0 | reset |
| 1 | 1 | 0 | 1 | 1 | set |
| 0 | 0 | 0 | 0 | Q | no change |
| 0 | 1 | 0 | 0 | Q | no change |

Flip-flops

■ D flip-flop

D flip-flop

| D | Q+ |
|---|----|
| 0 | 0 |
| 1 | 1 |



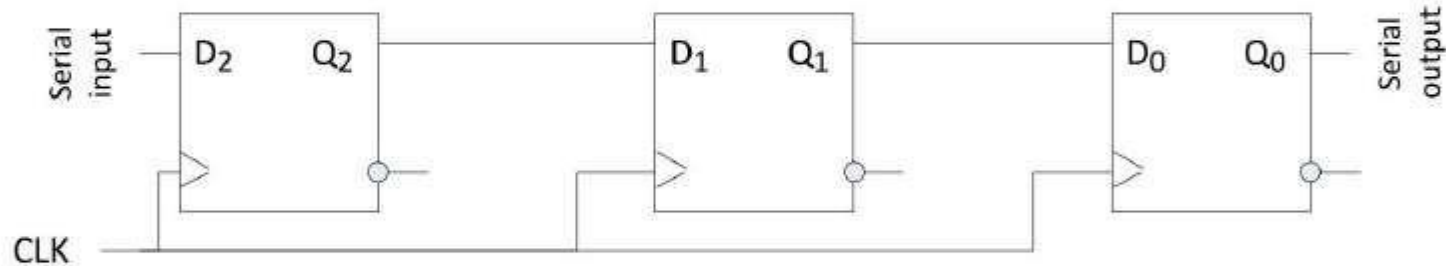
Changes happens only on the edges of clock (positive edge or negative edge)

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

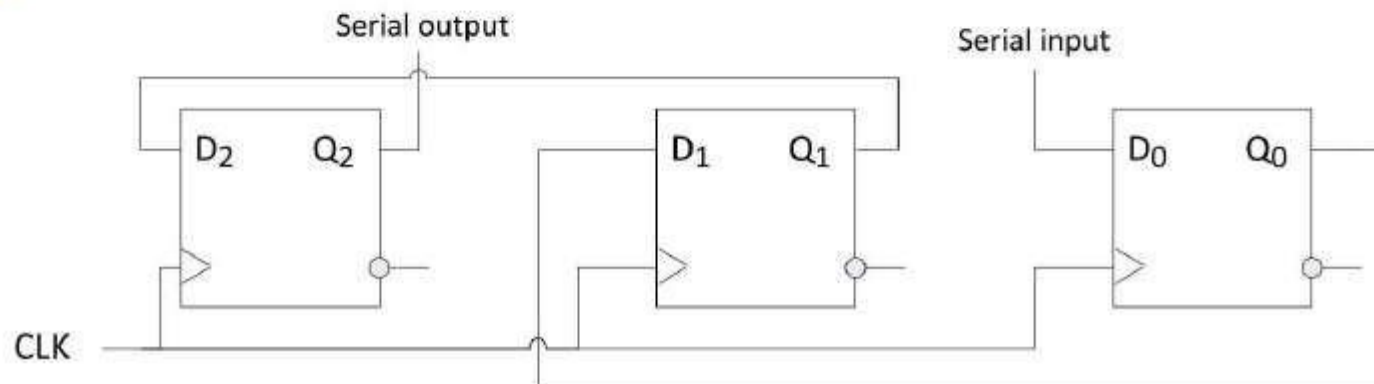
Registers

It seems like a group of flip-flops

- **Logical Shift Right:**

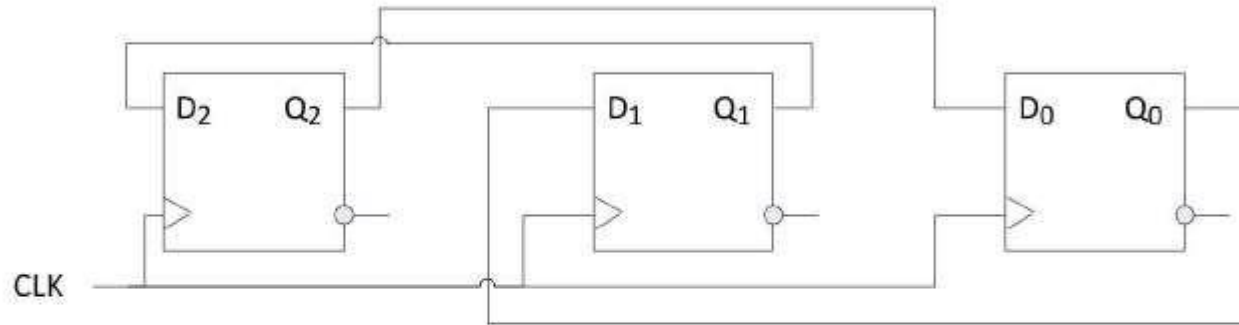


- **Logical Shift Left:**

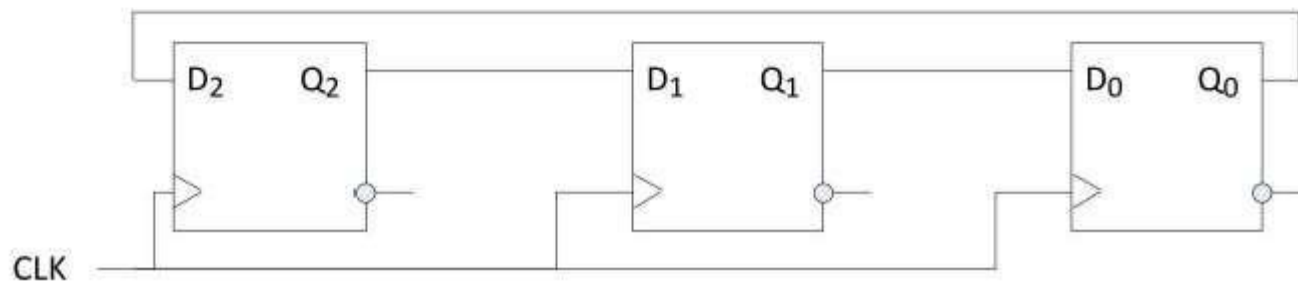


Registers

■ Circular Shift Left:

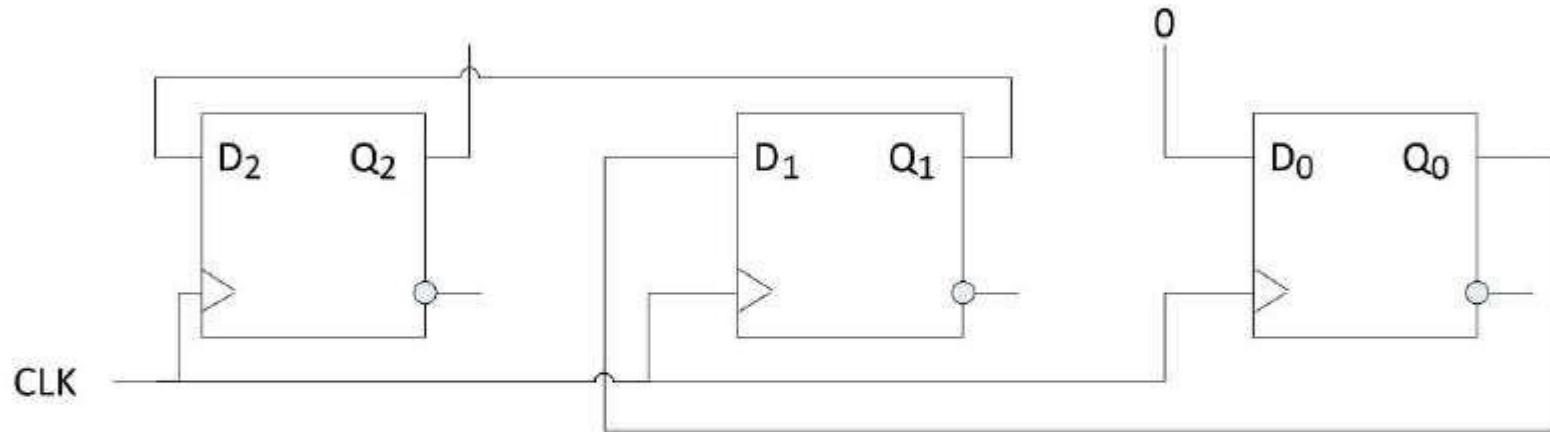


■ Circular Shift Right:

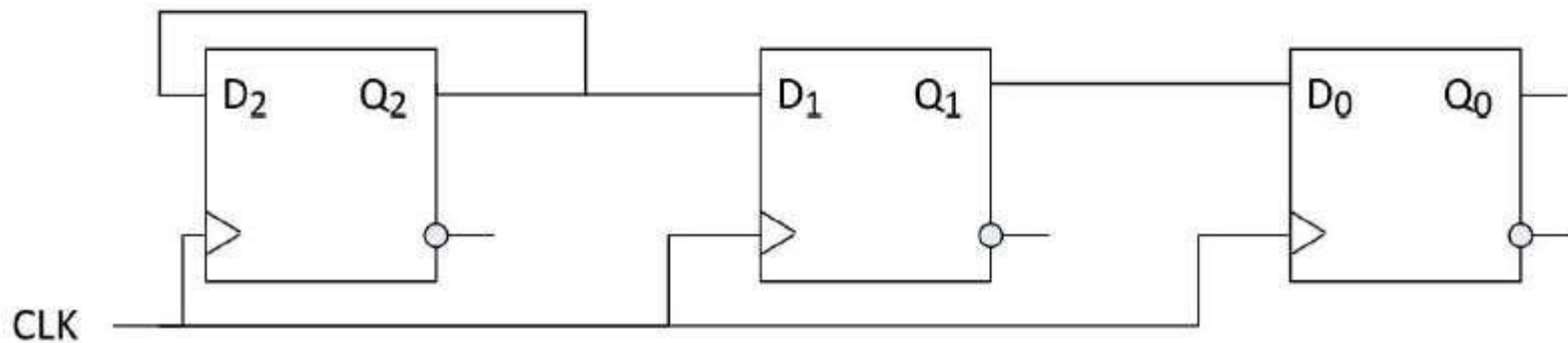


Registers

- **Arithmetic Shift Left: Must shift in 0**



- **Arithmetic Shift Right: Can't change sign bit**



Exam Review Questions