UF Herbert Wertheim College of Engineering
UNIVERSITY of FLORIDA

POWERING THE NEW ENGINEER TO TRANSFORM THE FUTURE

Department of Electrical & Computer Engineering

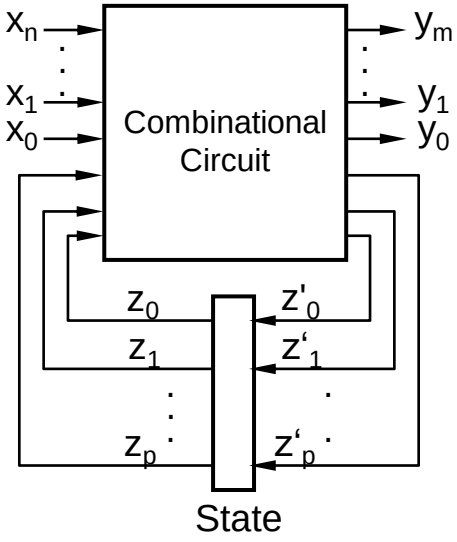
Digital Logic And Computing Systems

Chapter 03 – Latches, FlipFlops, Memory

Dr. Christophe Bobda

Sequential Circuit

- Definition: A sequential circuit is a circuit whose outputs depend on the current and previous inputs.
- Memory is required to store the circuit's state.
- The state of a circuit is the set of all values needed for future computation.

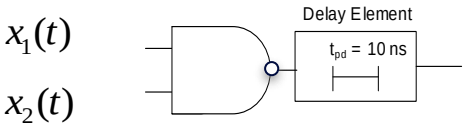


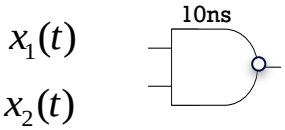
Agenda

- Delays
- Feedback Loop
- Latches and Flip-Flops
- Synchronization
- Registers und Counters
- RAM and ROM

Delays

- Modelling of delays
  - Gates and wires have delays.
  - Only gate delays t_{pd} (propagation delay) are considered in most tools.
    - Today with transistor miniaturization, wire delays are beginning to be more important and cannot be ignored anymore → timing closure (accurate timing estimation) as consequence.
- Inputs and outputs are time-dependent
  - Modeling of a real gate is done using an ideal gate and a delay element.
  - The output of a gate uses the input values at (t – delay).
  - alternative

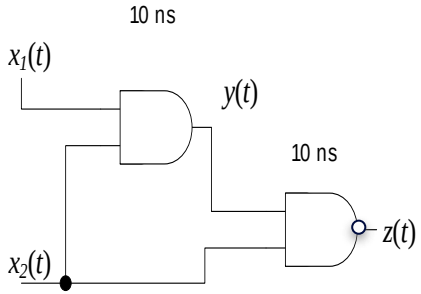


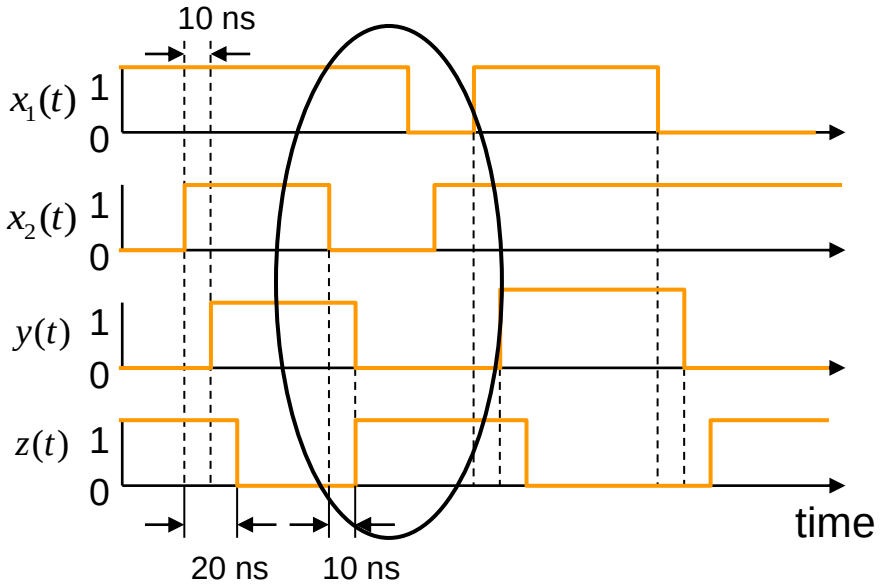


$$z(t) = \overline{x_1(t-10) \cdot x_2(t-10)}$$

Delays

Circuit analysis





Observed effects

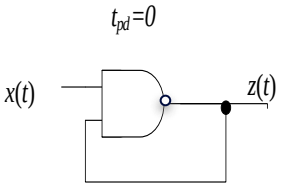
- The reaction of the circuit to 1-0-transition of x_2 is faster than the reaction to 0-1-transition
- Inertial-Delays: input impulse is too short to have an affect on the output
- Transient impulses (glitches) can be observed at the output before the correct value become stable

Feedback

- Sequential circuits need memory elements
  - Requirement: A memory element must hold one of two values (0,1) and stable (bistable) for a long period of time.
    - Combinational circuits: "memorize" input signal through delay but only for a very short time

  Idea: combinational circuit with feedback

Ideal NAND-Gate:



$$z(t) = \overline{x(t) \cdot z(t)}$$

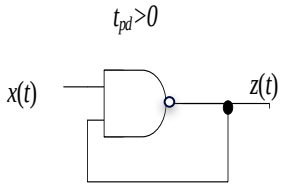
Are the states stable?

Case 1: $x(t) = 0 \Rightarrow z(t) = \overline{0 \cdot z(t)} = 1$ ✓

Case 2: $x(t) = 1 \Rightarrow z(t) = \overline{1 \cdot z(t)} = \overline{z(t)}$ Logic contradiction!

Feedback

Real NAND-Gate:

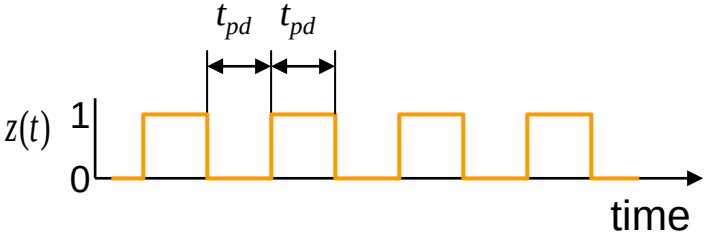


$$z(t) = \overline{x(t-t_{pd}).z(t-t_{pd})}$$

Are the states stable ?

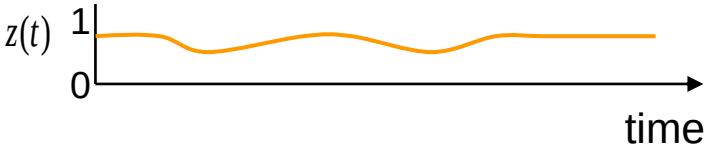
Case 2: $x(t) = 1 \Rightarrow z(t) = \overline{z(t-t_{pd})}$

→ Solution of the equation:



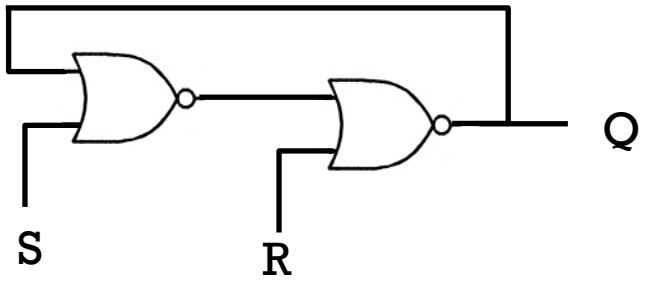
Unstable behavior (Oscillation)

→ Real gates: possible transition in metastable states



SR-Latch

- Using two gates

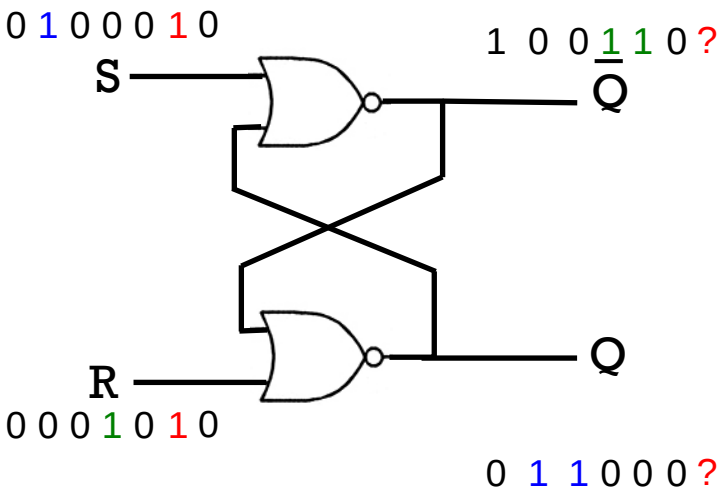


S	R	Q	$\overline{Q}$	Stable?	
0	0	0	1	✓	Init (Start)
1	0	1	0	✓	Set
0	0	1	0	✓	Keep (no change)
0	1	0	1	✓	Reset
0	0	0	1	✓	Keep
1	1	0	0	✓	Forbidden
0	0	?	?		

time

SR-Latch

- The SR-Latch (Set-Reset)- is a bistable element



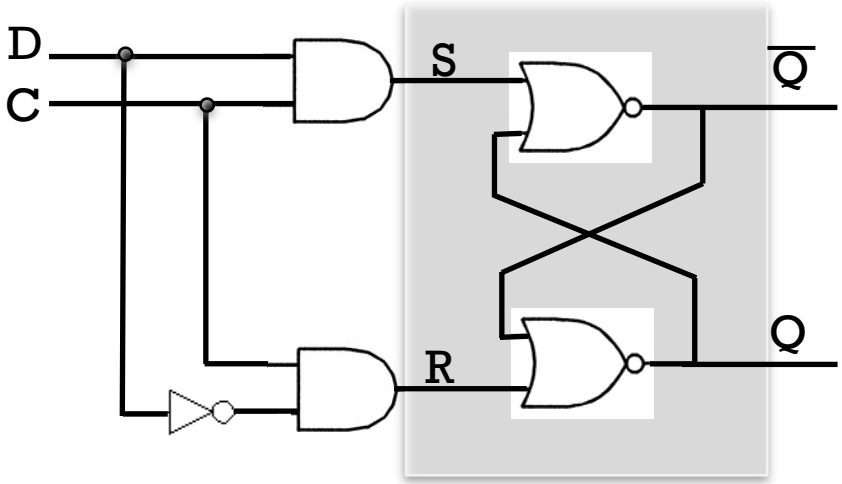
S	R	Q	$\overline{Q}$	Stable?	
0	0	0	1	✓	Init (Start)
1	0	1	0	✓	set
0	0	1	0	✓	No Change
0	1	0	1	✓	reset
0	0	0	1	✓	No Change
1	1	0	0	✓	
0	0	?	?		

time

- The output for (S,R)=(1,1) stable, however in case of (S,R)=(0,0) transition
  - Identical gate delay → the outputs oscillate
  - Different gate delay → stable output but not predictable

D-Latch

- D-Latch: Gated inputs to eliminates "undefined" states of SR-Latch
  - D for "delay" or "data".
  - C - Enable

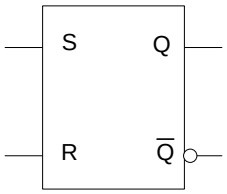


  - The inverter ensures that the combination (S,R)=(1, 1) never appears
  - C=1 → transparent state: Changes at the input are available after t_{pd} (propagation delay)
  - C=0 → input keeps the previously stored value

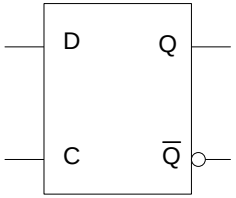
SR/D-Latch

- Symbol

SR-Latch



D-Latch



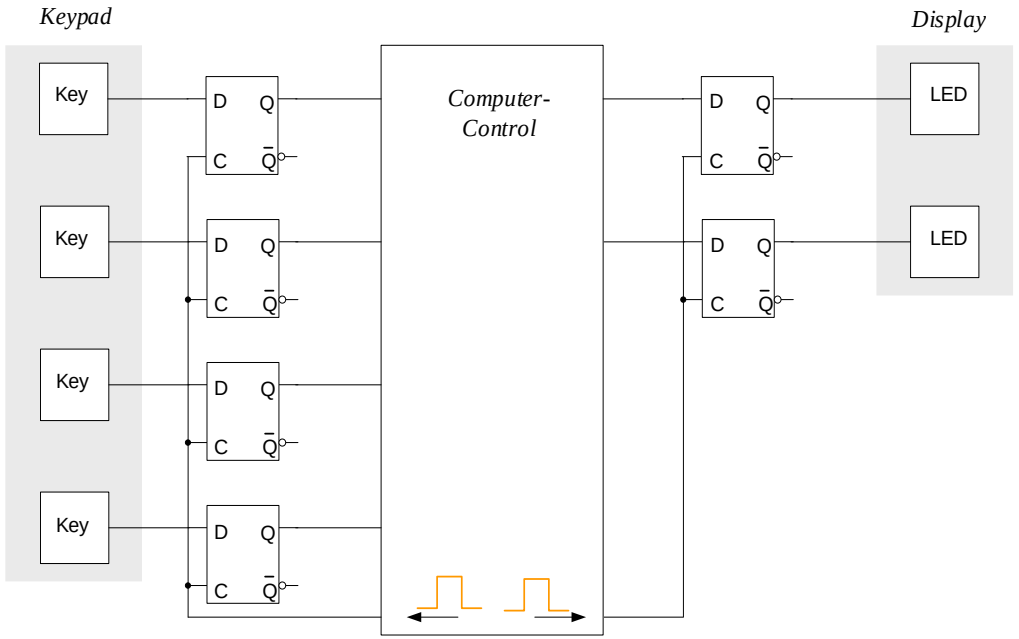
(bubble on $\overline{Q}$ some time ignored when it's clear that we have an inverter)

- SR- and D-Latches are basic memory elements that can be built with logic gates
- SR-Latches and D-Latches are also known as level-driven
- The term flip-flop is used for bistable elements without "transparent state" (clock-driven)
- Bistable elements with a transparent state are known as Latches

SR-Latch, D-Latch

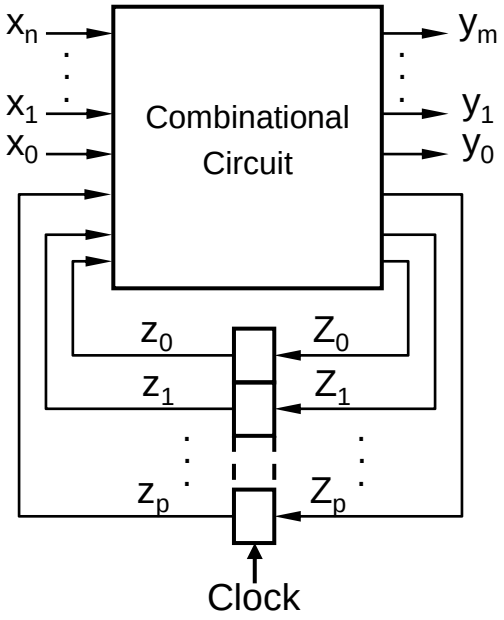
- Application of Latches
  - In circuits without combinational feedback from Latch-output to latch input
    - Transparent phase could lead to instable or metastable states

Example



Synchronization

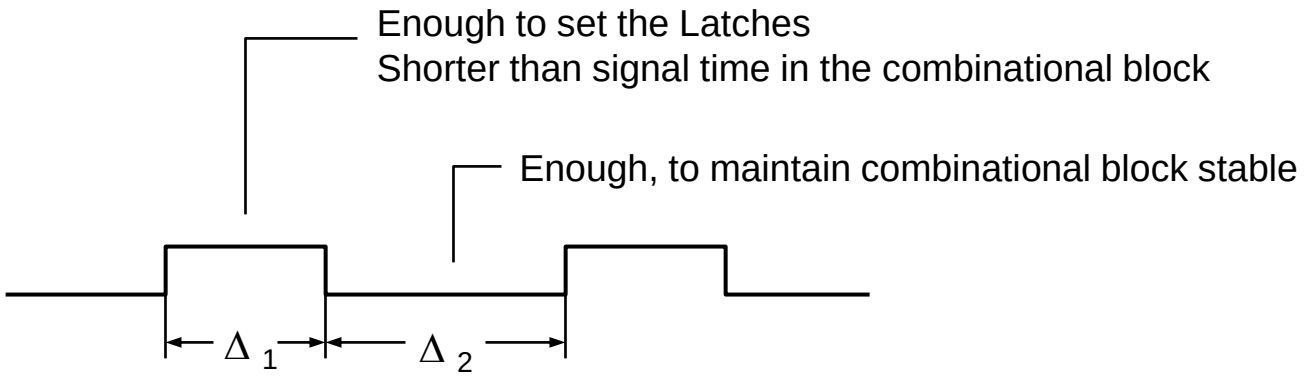
- Synchronization
  - Huffman Model



  - Appropriate clock for circuit stability

1-Phase Clocking

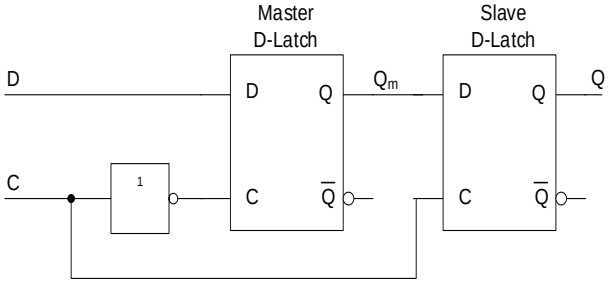
- Assumption:
  - **Memory elements are based on D-Flipflops**
  - All gates (including those in the flip flops) have internal propagation delays
- Consequence:
  - Clock scheme with restriction

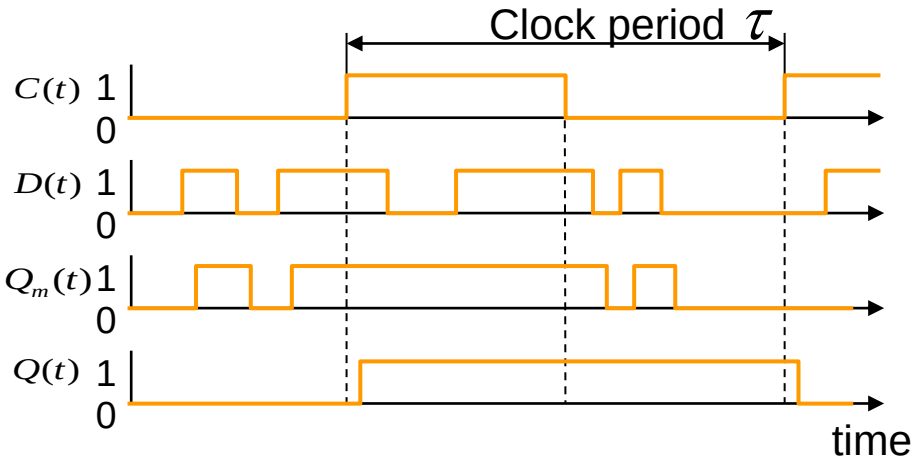


  - Problem case: too long Δ_1 as well as too short $\Delta_2 \rightarrow$ instability

Master-Slave D-Latches

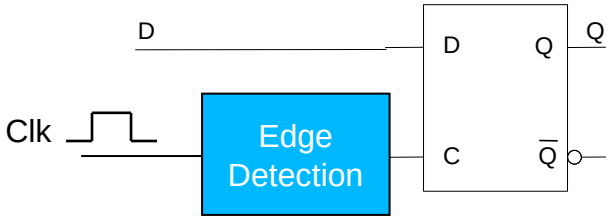
- Basic Idea: Interrupt feedback loop by using two latches serially with complementary enable
- Result: Master-Slave-D-Latch

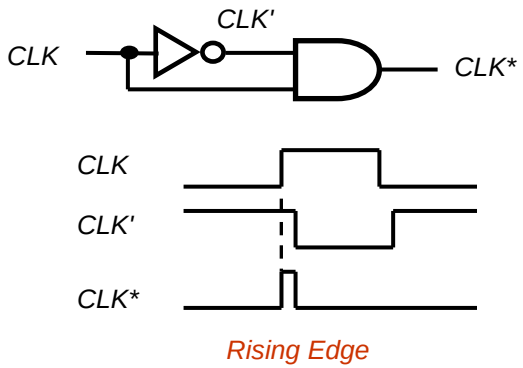




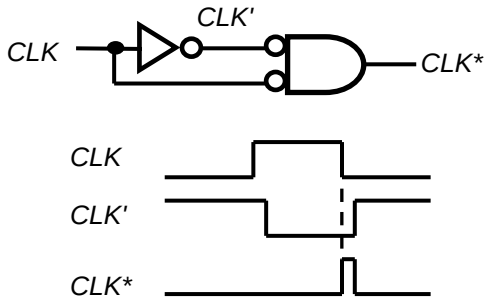
Edge-Triggered Latches (Flip-Flops)

- Flip-Flops: Edged-triggered latches.
  - React to signal edge.
  - Edge detection circuit recognizes positive/negative edge and generates corresponding impulse for a short period.





Rising Edge

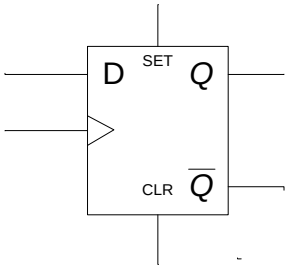


Falling Edge

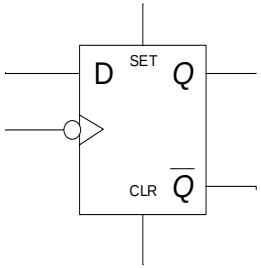
D-Flipflops

- Symbol

Rising (positive) edge D-FF

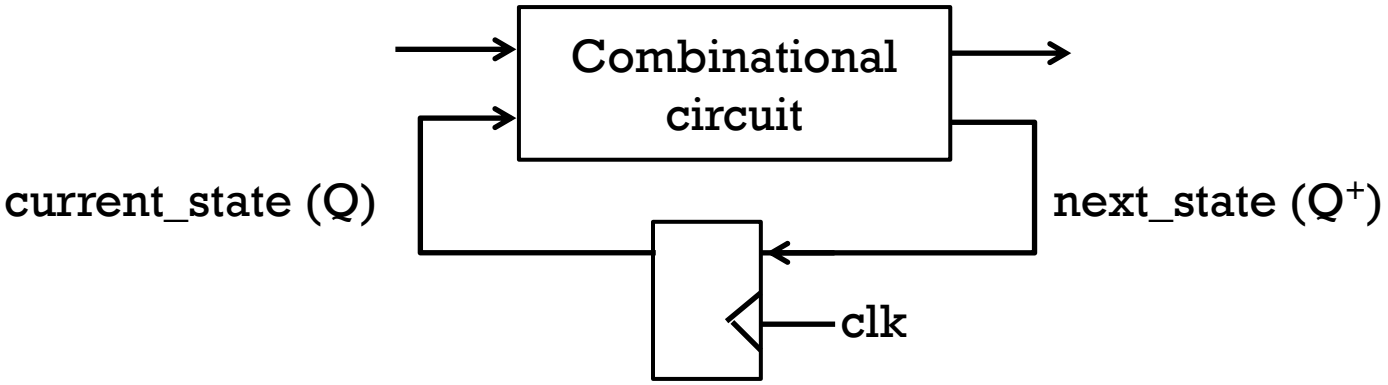


Falling (negative) edge D-FF

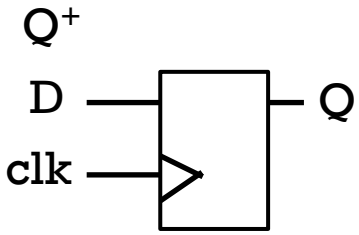


- Definition: A circuit is called sequential synchronous if all flipflops are triggered with the same clock

Flip-flops – Equation



- D-Flip-Flops
  - Q is current state, Q+ is next state

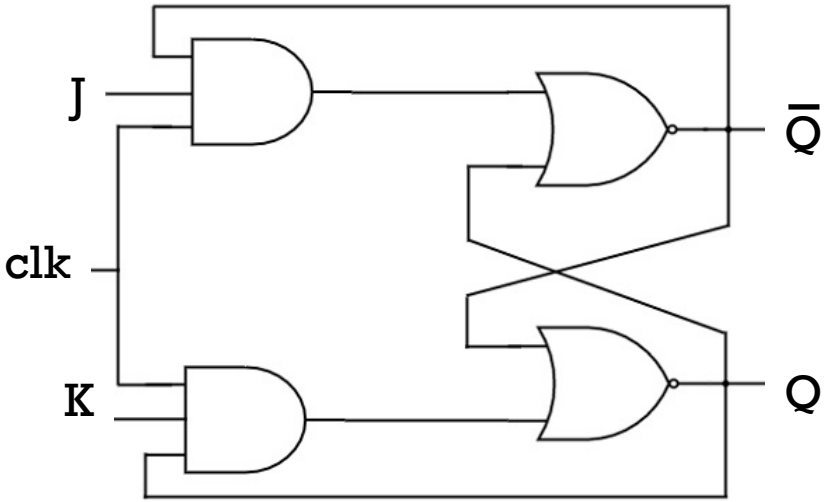


D	Q	Q+
0	0	0
0	1	0
1	0	1
1	1	1

$Q^+ = D$

JK-Flip-flops

- The JK-Flipflop is an extension SR flip-flop.
  - The J input corresponds to S and K corresponds to R.
  - Difference between SR and JK
    - J = K = 1 cause Q to toggle (0->1 and vice-versa)



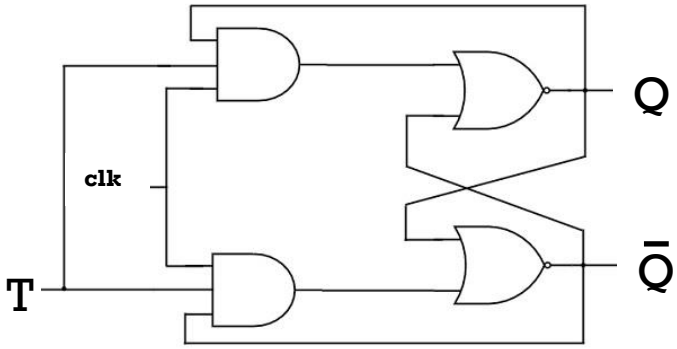
J	K	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

$$Q+ = J\overline{Q}+\overline{K}Q$$

The equation for Q+ can be derived using a K-Map

T-Flip-flops

- The Toggle-Flip-flop
  - Single input T that replaces J and K.
  - T = 1 causes the flip-flop to toggle (0->1 and vice-versa)
  - T = 0, no state change occurs.

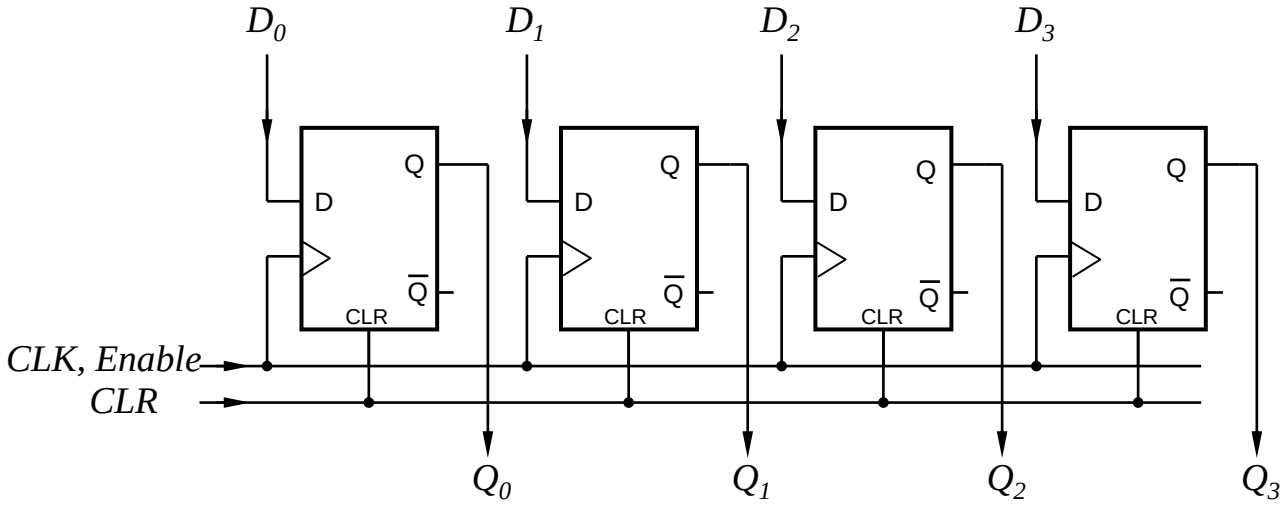


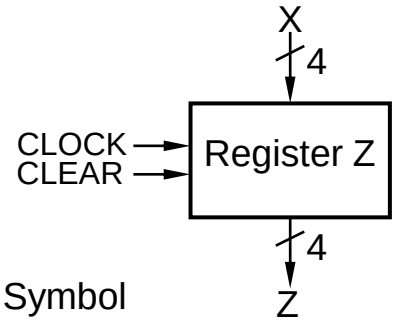
T	Q	Q^+
0	0	0
0	1	1
1	0	1
1	1	0

$$Q+ = T\bar{Q}+\bar{T}Q$$

Register and Counters

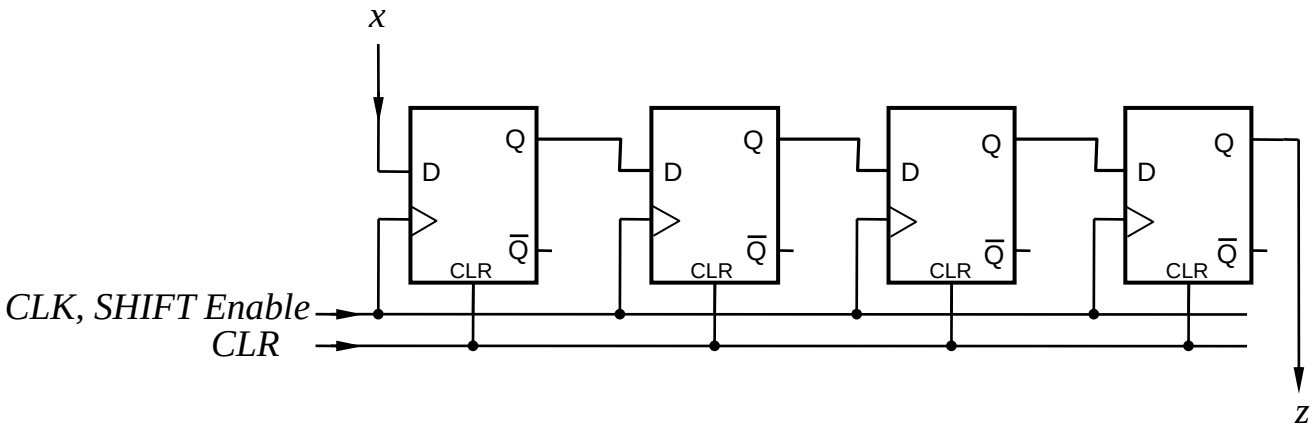
- a n-bit register consists of n flipflops that can store one n-bit word
  - Also known as parallel register or buffer register
  - Application example: state registers in finite state machines (automata) and in CPUs

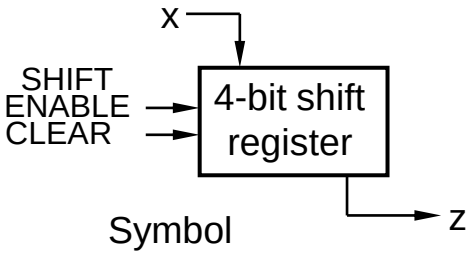




Shift Register

- a n-bit shift register produces the input value at the output after n pulses
  - Shift registers have FIFO (First-In, First-Out) behavior

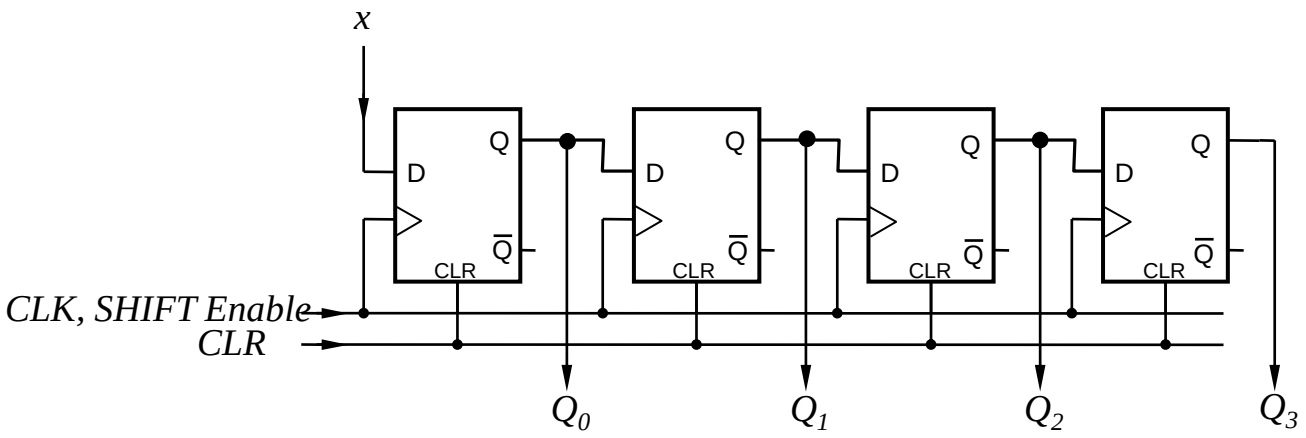


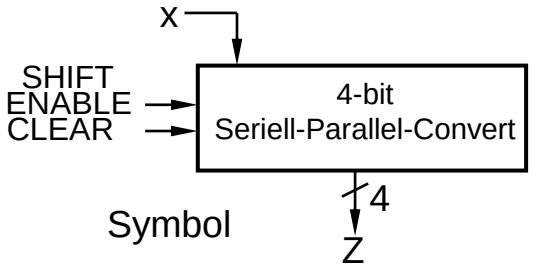


  - Exercise: add shift left to this register design
    Use a signal DIR to control the direction

Seriell Parallel Converter

- Serial-to-Parallel Converter
  - n-bit shift register (in writing) and parallel register in reading

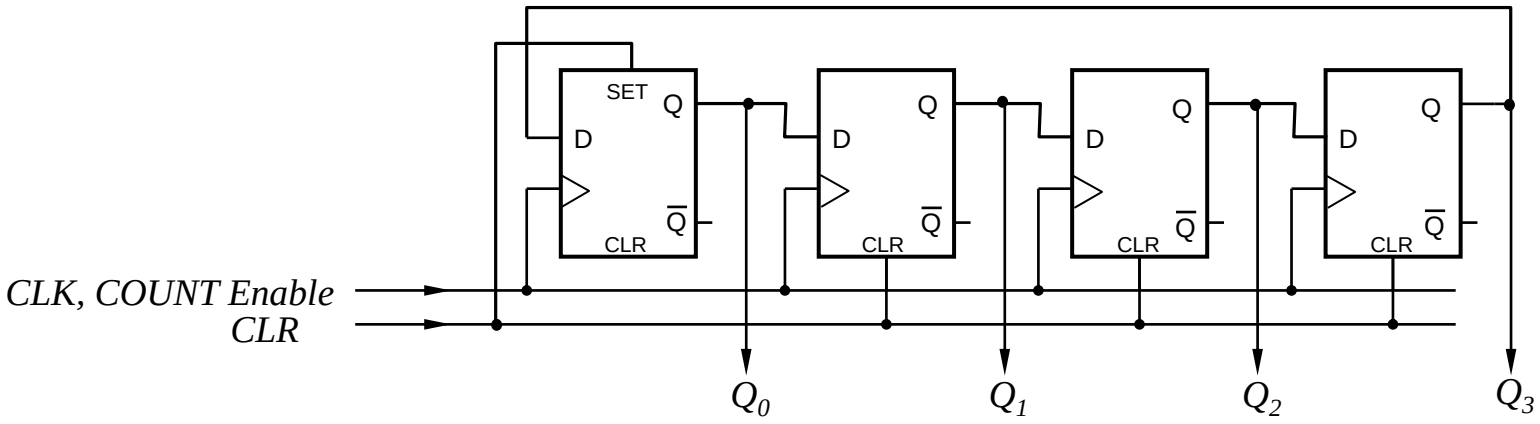




  - Exercise: Draw the circuit of a parallel-to-serial converter
    - Use a signal LOAD (LOAD=1 → parallel write, LOAD=0 → serial read out)

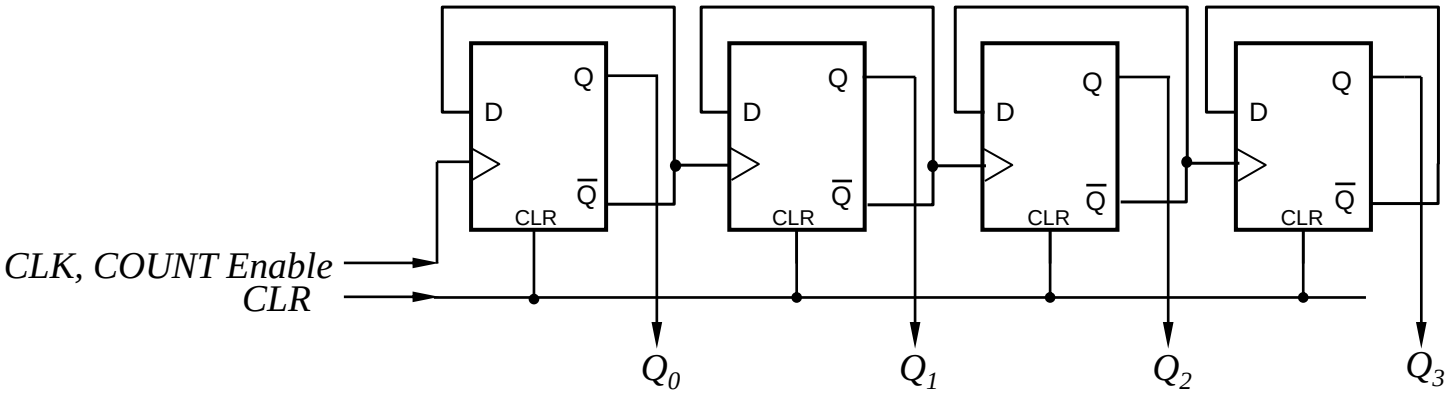
Ring Counter

- One the simplest counters
  - A n-bit Ring counter produces the following sequence in a cycle
    $(Q_{n-1} \ldots Q_0)_2 = 1, 2, 4, 8, \ldots 2^{n-1}$
  - Can be realized using a n-bit shift register with feedback



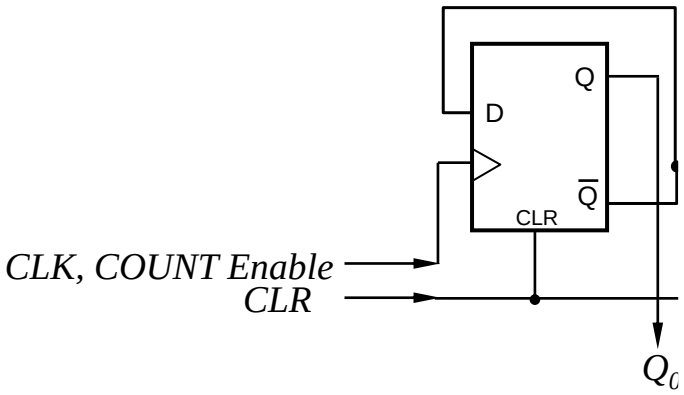
Asynchronous Counter

- Mod 2^n can be easily realized using asynchronous counters
  - Each flip-flop is clocked with half the frequency of the previous flipflop
  - A mod 2^n counter produces the following values in a cycle
    $(Q_{n-1} \ldots Q_0)_2 = 0 \ldots 2^n - 1$



Asynchronous Counter

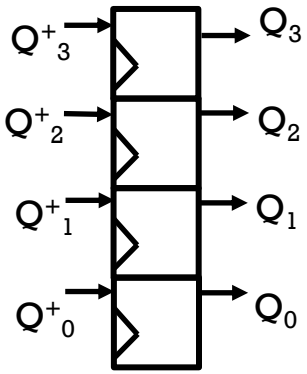
- Mod 2^n can be easily realized using asynchronous counters
  - Each flip-flop is clocked with half the frequency of the previous flipflop
  - A mod 2^n counter produces the following values in a cycle
    $(Q_{n-1} \ldots Q_0)_2 = 0 \ldots 2^n - 1$



Mod 2^n Counter

- Count sequences (0, 1, 0), (0, 1, …, 3), (0, 1, …., 2^{n-1}, 0)

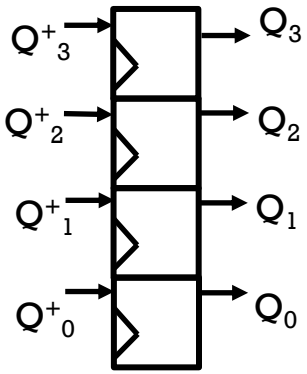
Q_3	Q_2	Q_1	Q_0	Q^+_3	Q^+_2	Q^+_1	Q^+_0
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

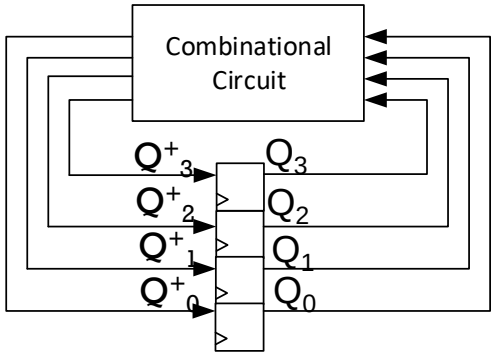


Synchronous Counter

- Mod 2^n (2, 4, 8, 16)

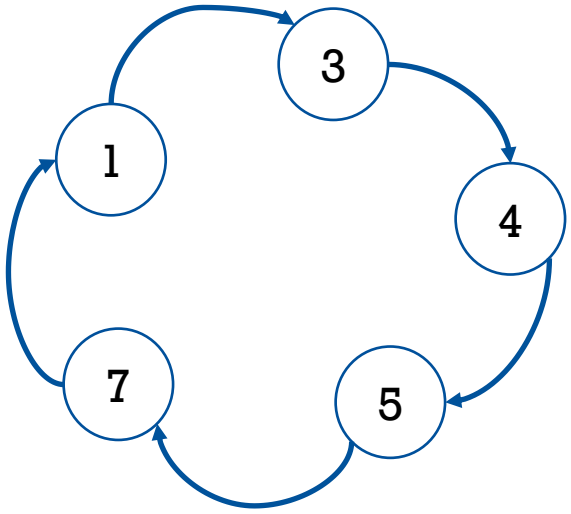
Q_3	Q_2	Q_1	Q_0	Q^+_3	Q^+_2	Q^+_1	Q^+_0
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0



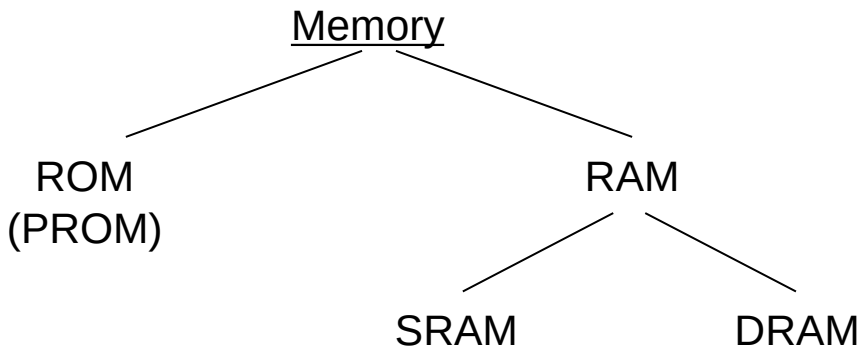


Synchronous Counter

- Any random counting sequence can be realized using a synchronous counter
  - For the design of synchronous counter, use the Huffmann‘s model along with the methods learned in circuit synthesis



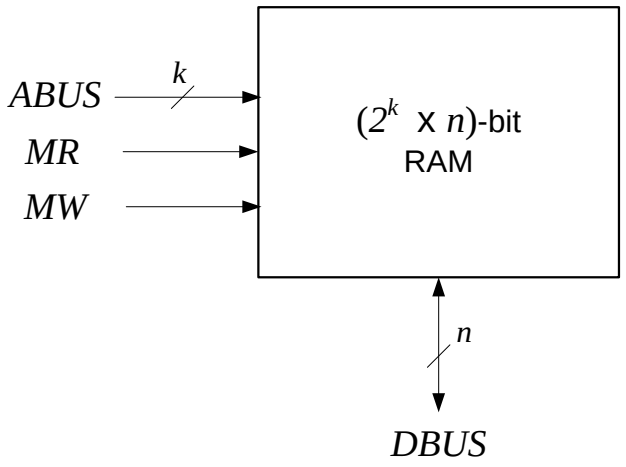
Memory



- Memory is a collection of memory cells
  - Cells can be D-FFs
    - Very expensive for large-size memories
  - RAM: random access memory
  - ROM: read-only memory

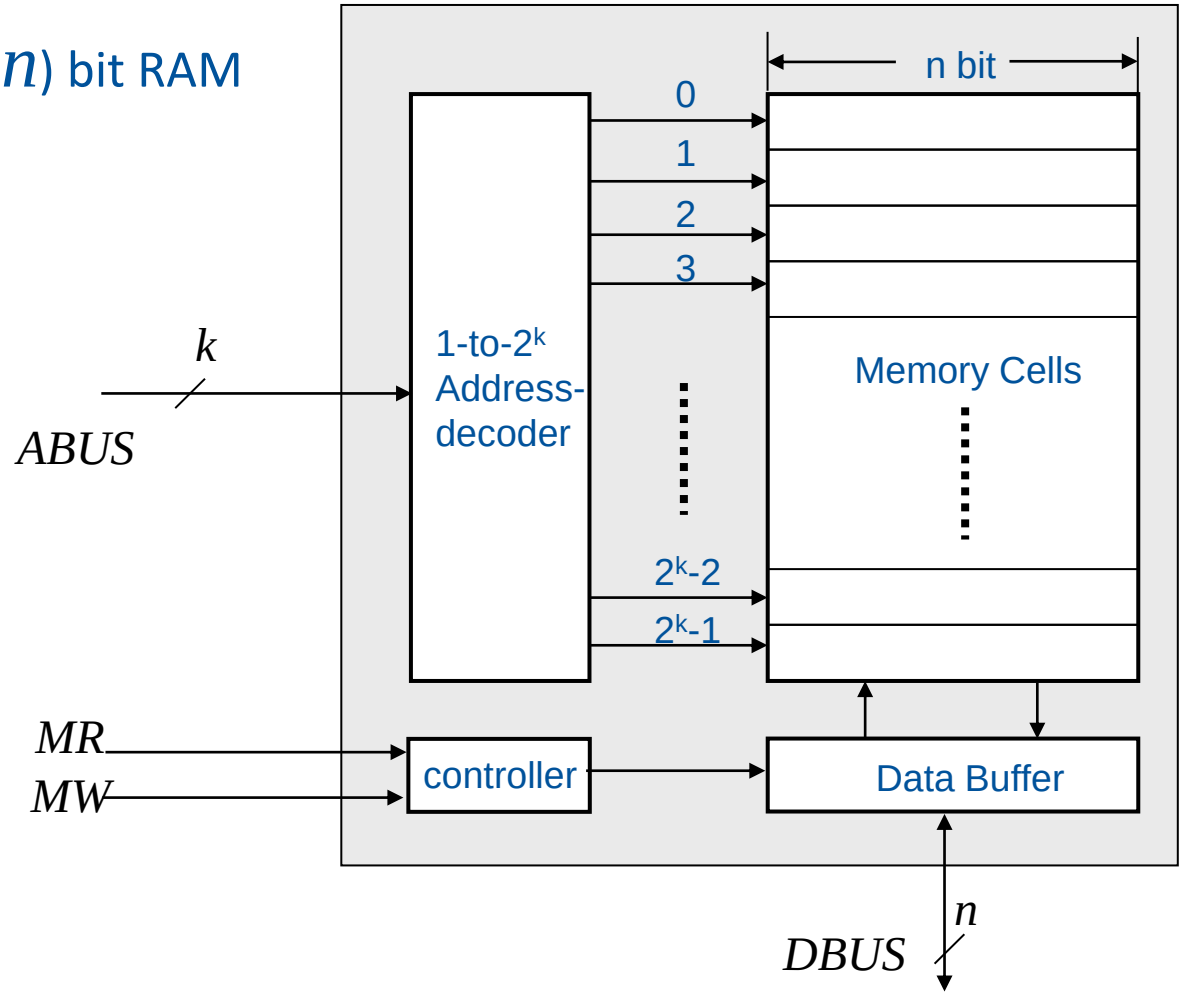
RAM

- RAM (write-read memory) can be randomly written and randomly read
- Inputs and Outputs
  - k bit address bus $ABUS$
  - n bit data bus $DBUS$
  - control signals MR (memory read) and MW (memory write)
    - $MR=1$ means that value at address location defined by ABUS must be place on the bus (read)
    - $MW=1$ means that the value on the bus DBUS must be stored at the location defined by ABUS (write)



RAM

- Architecture of a (2^k x n) bit RAM



UF Herbert Wertheim College of Engineering
UNIVERSITY of FLORIDA

POWERING THE NEW ENGINEER TO TRANSFORM THE FUTURE