CpE2210
Introduction to Computer Engineering

Dr. Minsu Choi

CH 6: Memory Elements & Arrays

Definition

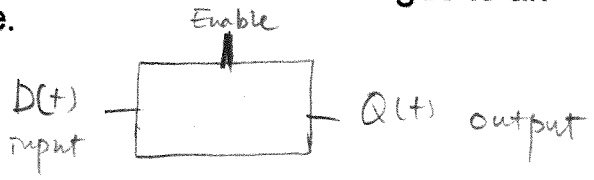
- A memory element is a circuit that can hold the value of a binary variable as required by the system.
- 3 modes: write, hold and read.

Memory Types

- RAM (Random Access Memory): read/write can be done. Loose contents when the power supply is disconnected.
- ROM (Read-Only Memory): Information is permanently stored. Only read operation can be done.
- Variations:
  - PROM (Programmable ROM) – write procedure requires a special electronics setup.
  - Flash: holds contents even though the power supply is disconnected.
  - Registers: commonly used for fast internal storage. Each resister can usually hold one word.

Latches

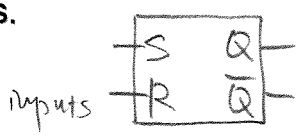
- A latch is a logic element that can follow data variations and transfer these changes to an output line.



- It is transparent in that the output Q(z) follows changes at the input at least part of the time.
- The storage is achieved using a bi-stable circuit, in which either Q=0 or Q=1 can be held in the cell.

SR Latch (Set-Reset Latch)

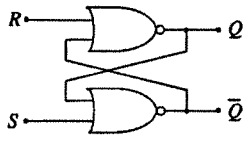
- A transparent bi-stable element that is sensitive to changes in the inputs.



- Set (S) operation: the output is forced to a value of Q = 1.
- Reset (R) operation: the output is forced to a value of Q = 0.
- There are two different ways to implement: NOR-based and NAND-based.

NOR-Based SR Latch Implementation

- Two cross-coupled NOR gates used.
- Feedback interconnects used to retain stored logic value.
- 0 to 1 to 0 transition at S sets Q.
- 0 to 1 to 0 transition at R resets Q.
- S=0 and R=0 to hold the current value of Q.
- S=1 and R=1 case is not used.



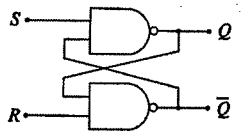
(a) Logic diagram

S	R	Q	$\bar{Q}$	Operation
0	0	Q	$\bar{Q}$	Hold
⎍	0	1	0	Set (Q → 1)
0	⎍	0	1	Reset (Q → 0)
1	1	0	0	Not used

(b) Function table

NAND-Based SR Latch Implementation

- Two cross-coupled NAND gates used.
- Feedback interconnects used to retain stored logic value.
- 1 to 0 to 1 transition at S sets Q.
- 1 to 0 to 1 transition at R resets Q.
- S=1 and R=1 to hold the current value of Q.
- S=0 and R=0 case is not used.



(a) Logic diagram

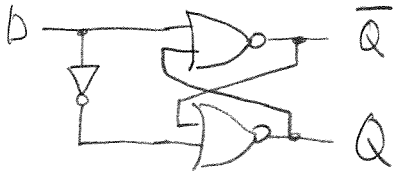
S	R	Q	$\bar{Q}$	Operation
0	0	0	0	Not used
⊓	1	1	0	Set ($Q \rightarrow 1$)
1	⊓	0	1	Reset ($Q \rightarrow 0$)
1	1	Q	$\bar{Q}$	Hold

(b) Function table

D Latch

- Input D determines Q and /Q.
- An inverter can be added to the NOR-based SR latch to implement a D latch.

D	Q
0	0
1	1



NOR TT.

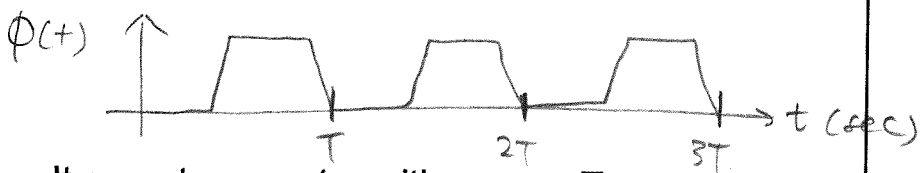
a	b	f
0	0	1
0	1	0
1	0	0
1	1	0

When D = 0, $\bar{D}$ input to the bottom NOR gate is 1. This forces the output $Q = 0$. ⇒ First stable state

When D = 1, then the output of the upper NOR gate is forced to $\bar{Q} = 0$. ⇒ second stable state

Clocks and Synchronization

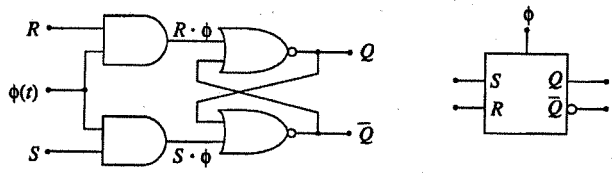
- A clock is a control signal that periodically makes a transition from 0 to 1 then back to 0.



- It repeats same transition every T seconds.
- Recall f = 1/T and T = 1/f.
- Ex)

$T = 1ns \Rightarrow f = 1/1ns = 1MHz$

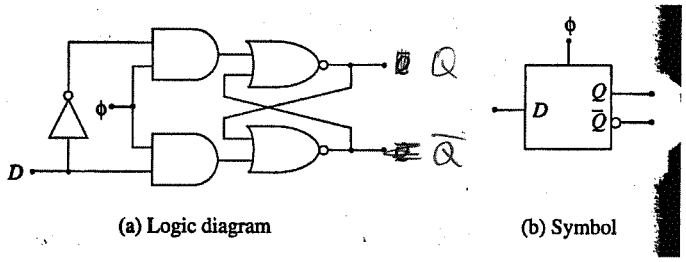
Clocked SR Latch



(a) Logic diagram (b) Symbol

- If Φ=0, then S Φ=0 & R Φ=0. So, the latch is in the hold state.
- If Φ=1, then set/reset operations can be done by S and R.
- This defines a "level triggered" or "level-sensitive" latch.

Clocked D Latch

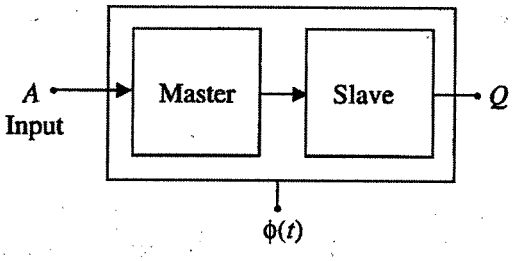


(a) Logic diagram (b) Symbol

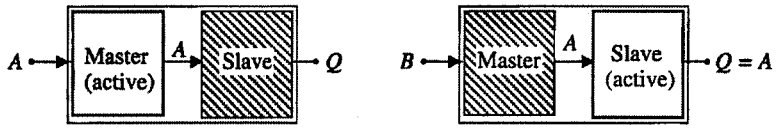
- The input D is active only when Φ=1.
- If Φ=0, then it is in hold state.

Master-Slave and Edge-Triggered Flip-Flops

- A flip-flop is non-transparent latch that is controlled by the clock.
- Master-slave structure is shown below.



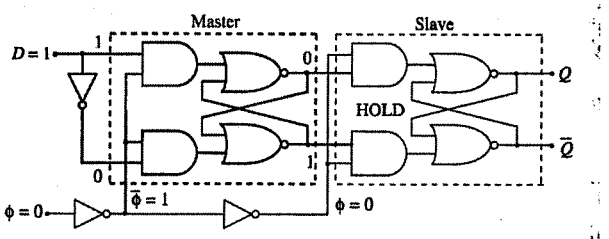
Why Master-Slave Structure Makes It "Non-Transparent"?



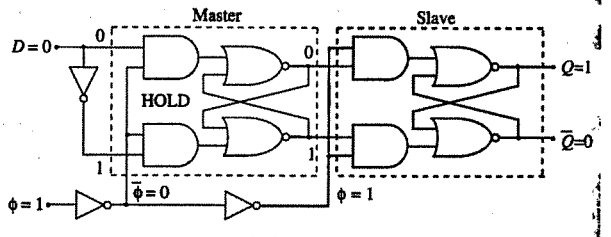
(a) Master active (b) Slave active

- Fig (a): The input to the master circuit is activated and the value of A is stored there. During the time, the slave is inactive (hold state) and cannot accept input.
- Fig (b): Then, the master is placed in a hold state while the slave is active. The value of A (not B!) is then accepted by the slave and is available at the output of the flip-flop giving Q=A.
- So, it is non-transparent (current output is not determined by the current input).

Master-Slave D-Type Flip-Flop (DFF)

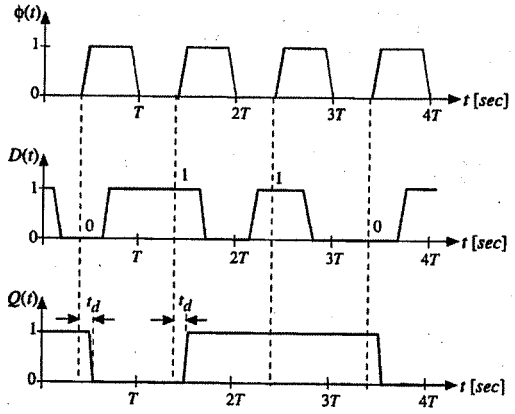


(a) Load master



(b) Transfer to slave

Timing Diagram

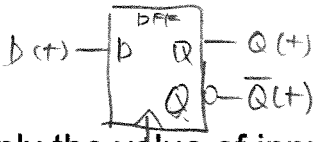


- The value that is transferred to the slave circuit is the value that is in the master latch when the clock makes a transition from 0 to 1.
- So, DFF is "edge-sensitive".

Edge sensitive DFF's master is transparent when $\Phi = 0$, but edge triggered DFF's master is transparent when $\Phi = 0 \to 1$.

Edge-Triggered

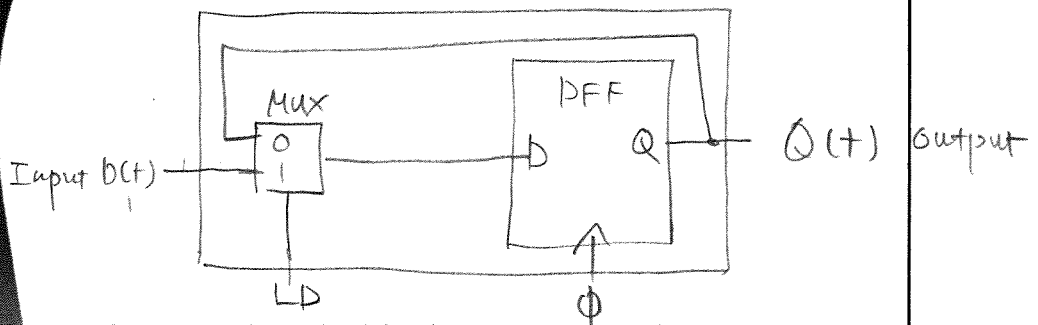
- Similar to edge-sensitive. However, it allows input data D to be loaded only when the clock is making a transition.



D(t)	Q(t+T)
0	0
1	1

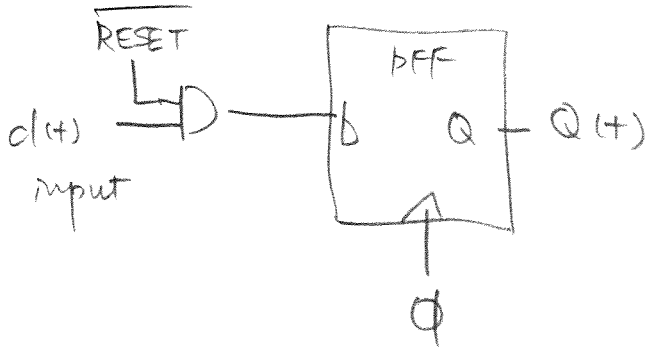
- The output Q is simply the value of input D delayed by one clock cycle -> Q(t+T)=D(t).
- The delay (=T) is an important aspect since we can store the present value for use during the next clock cycle.
- Then, how can we store a data bit for several clock cycles? => we must route the output Q back to the input D.

DFF with Feedback Loop



- If LD=0, then holds the present value.
- If LD=1, then new value D is routed to the DFF storage element.
- Then, how to reset the flip-flop? (force Q to 0)

DFF with Reset (force Q to 0)

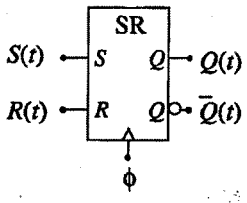


- If RESET=0, D=d at the input.
- If RESET=1, forces D=0 at the input.

SR FF (SR latch is used)

JK FF (modified SR FF) w/ Toggle operation

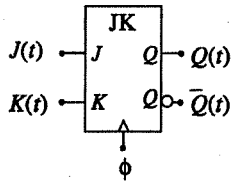
Other Types of Flip-Flops



(a) Symbol

$S(t)$	$R(t)$	$Q(t+T)$	Operation
0	0	$Q(t)$	Hold
1	0	1	Set
0	1	0	Reset
1	1	?	Not used

(b) Operation summary

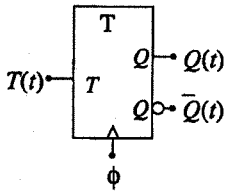


(a) Symbol

$J(t)$	$K(t)$	$Q(t+T)$	Operation
0	0	$Q(t)$	Hold
1	0	1	Set
0	1	0	Reset
1	1	$\bar{Q}(t)$	Toggle

(b) Operation summary

Continued,



(a) Symbol

$T(t)$	$Q(t+T)$
0	$Q(t)$
1	$\bar{Q}(t)$

(b) Operation summary

T FF (when T changes from 0 to 1, then $Q(t+T) = \overline{Q(t)}$)

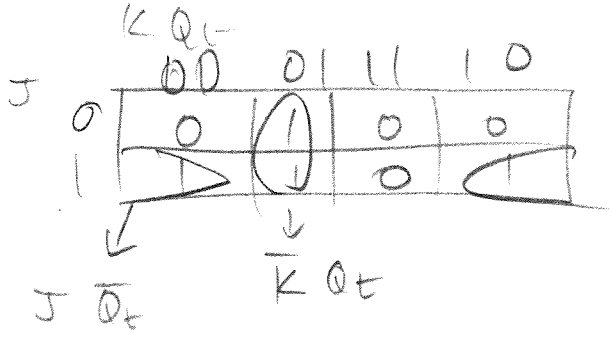
Example)

Make JK FF using D FF and logic gates.

	J	K	Q_t	Q_{t+1}	D (→ D FF input)
Hold	0	0	0	0	0
	0	0	1	1	1
Reset	0	1	0	0	0
	0	1	1	0	0
Set	1	0	0	1	1
	1	0	1	1	1
Toggle	1	1	0	1	1
	1	1	1	0	0

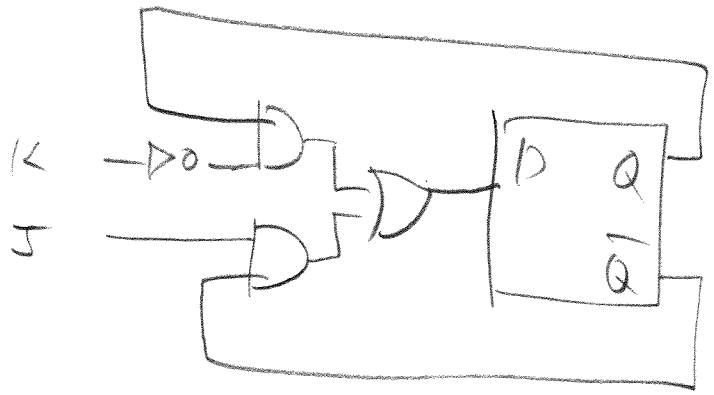
Review

b



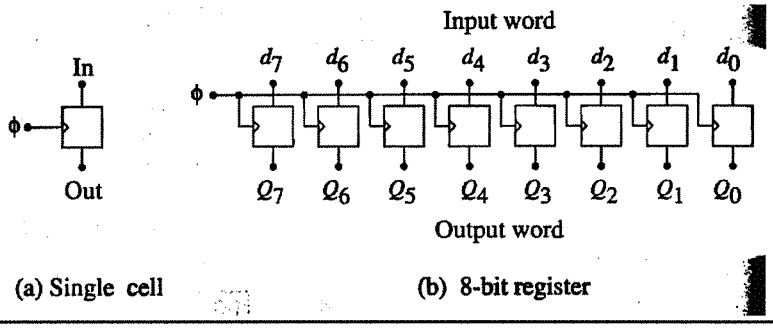
$D = K Q_t + J \overline{Q_t}$

logic diagram:



Registers

- A storage for n-bit binary word.
- Single-bit cells are cascaded to build multi-bit register.
- Mostly edge-triggered. Other control signals (e.g., LD) are not shown explicitly.
- Register symbol:

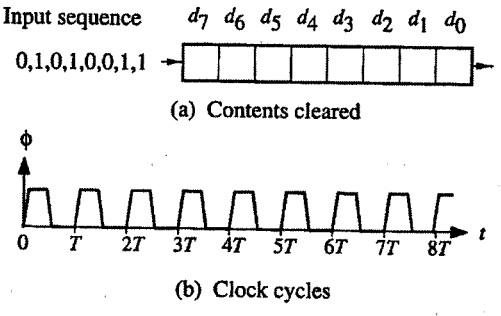


(a) Single cell (b) 8-bit register

Shift Registers

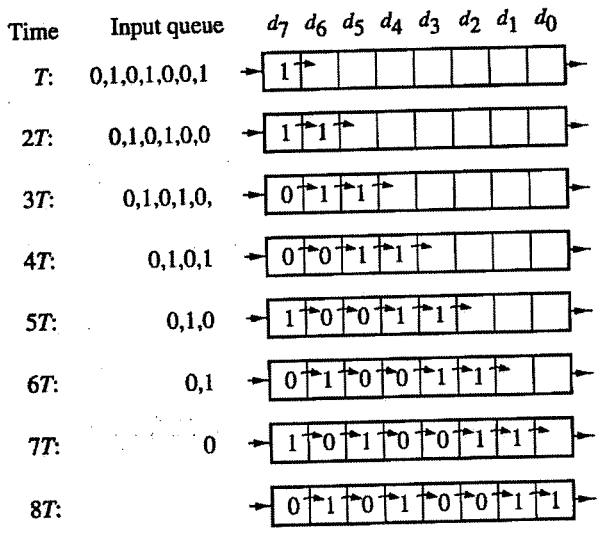
- Designed to move bits to neighboring cells as the clock pulses are applied.

- Ex)

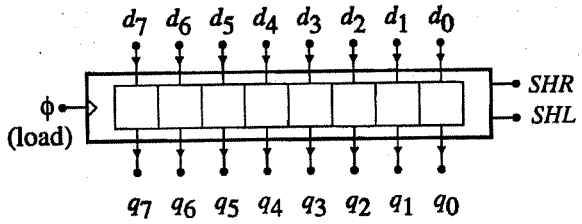


→ This data structure is called "FIFO" First in First out.

Continued,



Shift Register w/ Parallel Load Capability



- 8-bit data word can be loaded/ accessed at the same time.
- SHR (shift right) & SHL (shift left) control signals are used to shift contents.

Example

- N=00011100 $= 28_{10}$

SHL1 ⇒ $NL_1 = 00111000 = 56_{10}$

SHL2 ⇒ $NL_2 = 01110000 = 112_{10}$

⇒ The SHLn operation is equivalent to multiplying the contents of the register by 2^n.

SHR1 ⇒ $NR_1 = 00001110 = 14_{10}$

SHR2 ⇒ $NR_2 = 00000111 = 7_{10}$.

⇒ SHRm is to divide contents by 2^m.

Rotation Operations

- ROR (Rotation Right) and ROL (Rotation Left): Rotation does not lose a bit if it is shifted out, but transfers it to the other side.
- Ex) N=10101100

ROR=1 $NR_1 = 01010110$

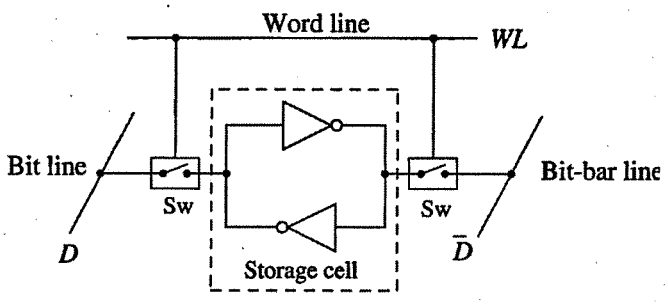
ROL=1 $NL_1 = 01011001$

Random Access Memory (RAM)

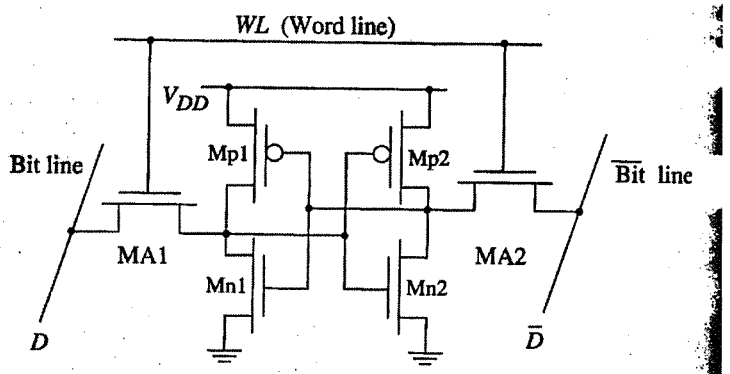
- Commonly found as the main system memory in a desktop computer unit.
- It provides the ability to store all of the important data needed to run the computer such as programs and the OS.
- There are two major types:
  - SRAM (Static RAM) for faster operation, but expensive.
  - DRAM (Dynamic RAM): cheap, large but slow.

SRAM

- Closed loop cascaded inverters can be used to build a SRAM cell.
- If WL=0, hold state (SWs open).
- If WL=1, read & write ops possible (SWs closed).



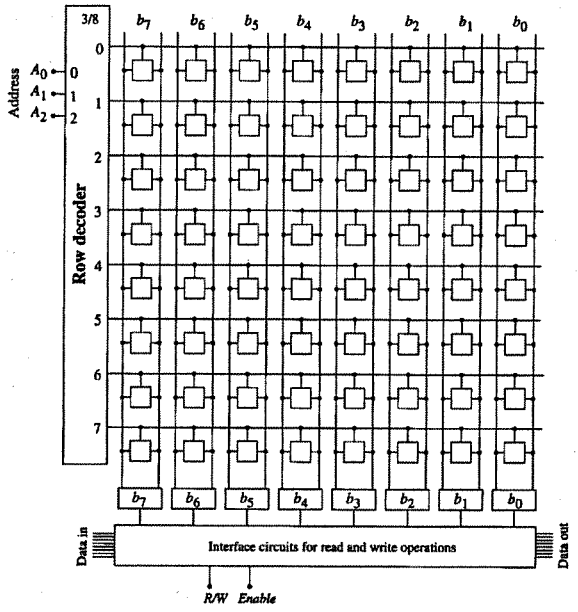
CMOS Implementation



- Two cross-coupled CMOS inverters used.
- Two nFET used as switches.

M X N SRAM Array

(m rows & n columns)



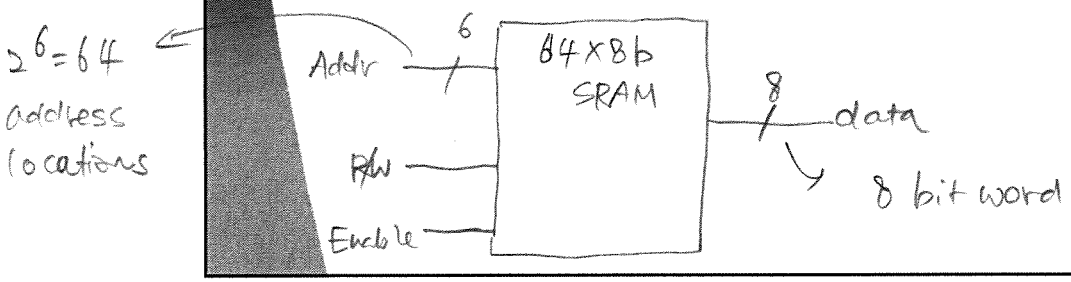
8x8 example

(Enable = 0 then Hold
Enable = 1 then r/w ops can be done & R/W bit is used to determine read / write op.

Continued,

- Row decoder is 3/8 active-high decoder to select desired row.
- 3-bit word, $A_2A_1A_0$ is used to select -> called "address".
- It specifies the location of a particular word in memory.
- Ex) Larger SRAM array

64 x 8b SRAM
# of words
bits per word



Parity & Error Detection Codes

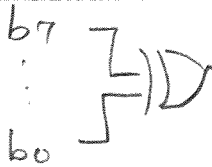
- Even a single bit error can cause an entire program to crash -> reliability is critical.
- EDC (Error Detection Code) & ECC (Error-Correction Code) are used.
- Ex) Parity bit (the most simple EDC)

① even parity scheme : P_{even} is chosen to make (B + P_{even}) have an even # of 1s ex) B = 01101101 P_{even} = 1

② Odd parity
(B + P_{odd}) has an odd # of 1s.
ex) B = 11110000 then P_{odd} = 1

- Parity bit generator

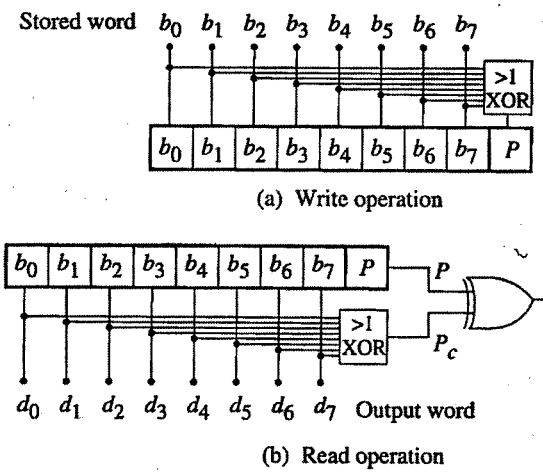
XNOR gate to generate Odd parity.



P = 1 (odd # of 1s in B)
0 (even # of 1s in B)

=> even parity bit generator.

Parity Bit Generation and Checking

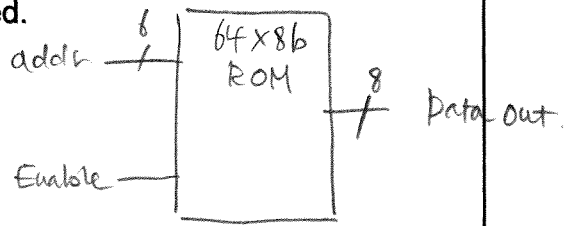


→ no error

→ error.

ROM (Read-Only Memory)

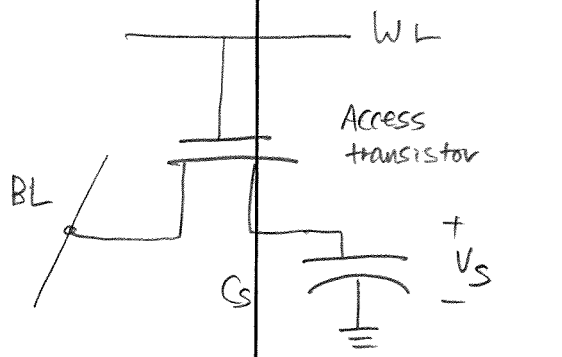
- Memory array that used for permanent data storage (=non-volatile) => data remains even if the power supply is disconnected.
- Ex) 64x8b ROM



1. The address word $A_5A_4A_3A_2A_1A_0$ is sent to a row decoder.
2. The data word is send to the output circuits, which are activated by the Enable control.

DRAM (Dynamic RAM)

- Can be built with much higher densities than SRAM cells.
- The descriptive adjective "Dynamic" reflects the fact that the contents of a cell will change in time during the hold state.
- DRAM cell is slower than SRAM cell. However, the cost per bit is much lower than SRAM cell -> attractive in system design where large arrays are required.
- Since many companies use their most advanced technology for making DRAMs, it is called "technology driver".



⊛ Simple. So, very high integration is possible

DRAM Operation Detail: Write Operation

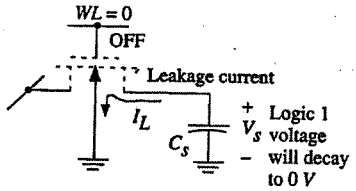
- When WL=1, the nFET acts as a closed switch allowing write or read operation.
- To write to the cell:
  1. Data voltage V_D is applied to the data line D.
  2. The resulting current charges the storage capacitor C_S to the voltage level V_S.
  3. The value of the stored bit is defined by the charge, $Q_S = C_S \cdot V_S$, on the capacitor.
  4. If $V_S = V_0 = 0v$, then $Q_S = 0$ corresponding to a logic '0' value. If $V_S = V_1$, then a logic 1 charge. $Q_S = C_S \times V_1$ is stored on the capacitor.

V_0 – logic 0

V_1 – logic 1

DRAM: Hold Operation

- WL=0 for hold operation.
- The charge Q_S would be held, since nFET is open.
- However, nFET cannot block all of the current flow.
- Even with a zero gate voltage, nFET admit a small "leakage current" I_L that removes charge from the capacitor -> the charge Q_S cannot be stored indefinitely -> logic 1 voltage will decay to 0v (logic 0) over the time.



(a) Circuit model

Hold Time Estimation

- Suppose that we place a logic 1 voltage VS = V1 on the capacitor, and we wish to calculate the time that the voltage can be held on the cell when the nFET is open, so that we can refresh the stored logic 1 to keep it.

$I_L = -\frac{d(Q_S)}{dt}$ → derivative of the charge over the time

$= -C_S \cdot \frac{dV_S}{dt}$ (since $Q_S = C_S \cdot V_S$)

- is used since the charge is decreasing in time.

latest charge in time ←

$\Rightarrow dt = -\left(\frac{C_S}{I_L}\right) \cdot dV_S$

$\Rightarrow \int_0^{t_H} dt = \left(-\frac{C_S}{I_L}\right) \cdot \int_{V_1}^{V_{min}} dV_S$ ← Integrating both sides gives

t_H is the "Hold time" ⇒ the time interval that the capacitor can maintain the voltage > V_{min}. (V_{min} is the lowest voltage level to represent logic 1).

$\Rightarrow t_H = \frac{C_S}{I_L}(V_1 - V_{MIN})$

(ex) Storage capacitance $C_S = 50\,fF = 50 \times 10^{-15}\,F$
leakage current $I_L = 1\,pA = 1 \times 10^{-12}\,A$
Say $(V_1 - V_{min}) = 1V$.

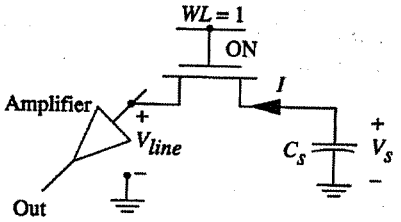
then $t_H = \frac{50 \times 10^{-15}}{10^{-12}} = 50 \times 10^{-3} \Rightarrow 50ms.$

$\Rightarrow$ So, data must be refreshed in every 50ms.

Read & Refresh Operations

- WL=1 to turn on the nFET, which allows the current to flow to the bit line.
- Then, amp provides strengthened signal to output.
- In case of hold op, periodical read-out, amplification and rewrite needed -> called refresh.
- Say the max hold time is 10^-3sec. Then, the min refresh frequency is:

$f_{min} = 1/t_H = 1000Hz$



So, each cell must be refreshed at 1000Hz frequency, min.

Program Completed

University of Missouri-Rolla

© 2003 Curators of University of Missouri

UMR UNIVERSITY OF MISSOURI-ROLLA
The Name. The Degree. The Difference.