Static RAM
Ultra violet electrically programmable ROM.

RAM is a type of memory that computers use to store data and software to which it needs to access quickly. It is used to temporarily store data that is currently used.

RAM stores the program.

RAM
* Volatile mlm
* Data in ram is not permanently written when power off, data deleted
* Ram: DRAM & SRAM
* It requires flow of electricity to retain data
* Ram is the mlm available for the OS programs & process to use when the computer is running

ROM
* Non volatile
* Permanently, not erased
* ROM: PROM & EPROM
* It does not require electricity to retain data
* It is mlm that comes with your computer i.e. pre written to hold the instruction for booting up the computer.
SRAM vs DRAM.

- DRAM stores charges in capacitor.
  - Disappears over short period of time.
  - Must be refreshed.
- SRAM easier to use
  - Faster.
  - Most expensive per bit.
  - Smaller sizes.

30-11-11

Sequential Circuit.

In combinational circuits, the past input is not taken.

But in sequential, the present input is stored to next.
counter

do display 10 numbers.

1+1  2
prev 0\rightarrow
2+1  3

SR Latch

latch := storing each bit.
SR := set & reset.
reset means starting from 00.
set means getting op.

Two types of SR latch using NOR gate and using NAND gate.

SR Latch ← NAND.
   ← NOR
NOR (RS Latch)
NAND (SR Latch)

**NOR Latch**

\[ R \rightarrow Q \]
\[ S \rightarrow \overline{Q} \]

\[ S = 0 \quad R = 1 \quad Q = 0 \quad \overline{Q} = 1 \]

If we reset \( Q \) & \( \overline{Q} \) should be 0.
we have to prove, when

**case I**

\[ S = 0 \land R = 0 \] then
\[ Q = 0 \land \overline{Q} = 1 \]

Reset means \( R = 0, \overline{Q} = 1 \) (prev. 0lp) : \( 0 \land 1, Q = 0 \) in
first NOR gate. Also \( S = 0 \land Q = 0 \) .. \( \overline{Q} = 1 \) (2nd NOR)

\[ S = 0 \quad R = 0 \quad Q = 0 \quad \overline{Q} = 1 \]

ie, when we reset we get same value of \( Q \) & \( \overline{Q} \) as that of previous case (ie \( S = 0 \quad R = 0 \quad Q = 0 \quad \overline{Q} = 1 \))

**case II**

\[ S = 1 \quad R = 0 \quad Q = 1 \quad \overline{Q} = 0 \]

\[ S = 0 \quad R = 0 \]
\[ R = 0, \overline{Q} = 0 \] .. \( Q = 1 \)
\[ S = 0 \quad Q = 1 \quad \overline{Q} = 0 \] same 0lp. when we reset
Case III:

When $S = 1$ and $R = 1$, $Q = 0$ and $\bar{Q} = 0$.

When reset $S = 0$ and $R = 0$, $Q = 1$ and $\bar{Q} = 1$.

Which is not possible so we are not taking this case.

<table>
<thead>
<tr>
<th>R</th>
<th>S</th>
<th>Q</th>
<th>$\bar{Q}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>m/m</td>
<td>used</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>not used</td>
<td></td>
</tr>
</tbody>
</table>

NAND Latch

\[ A \quad B \quad Y \]
\[ 0 \quad 0 \quad 1 \]
\[ 0 \quad 1 \quad 0 \]
\[ 1 \quad 0 \quad 0 \]
\[ 1 \quad 1 \quad 0 \]
Case I

\[ S = 0 \quad R = 1 \quad Q = 1 \quad \bar{Q} = 0 \]
\[ S = 1 \quad R = 1 \quad Q = 1 \quad \bar{Q} = 0 \]

Case II

\[ S = 1 \quad R = 0 \quad Q = 0 \quad \bar{Q} = 1 \]
\[ S = 1 \quad R = 1 \quad Q = 0 \quad \bar{Q} = 1 \]

When \( S = 1 \) look at Table I. The flip-flop always
in case \( R = 0 \) the \( Q \) is always

\[ . \quad Q = 0 \quad \text{when} \quad Q = 0 \]
and NAND of \( D_2 \)

\[ . \quad Q = 1 \]

\[ \text{Case III} \]

\[ S = 0 \quad R = 0 \quad Q = 1 \quad \bar{Q} = 1 \]
\[ S = 1 \quad R = 1 \quad Q = 0 \quad \bar{Q} = 1 \]

2/11/17

Enabled SR Latch

```

<table>
<thead>
<tr>
<th>S</th>
<th>R</th>
<th>Q</th>
<th>\bar{Q}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>not used.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>mini used</td>
</tr>
</tbody>
</table>
```

In 10 latch

\[ \bar{Q} = 0 \]
\[ Q = 1 \]
The flip to change from high to low can be done by triggering. Clock is used to control by triggering clock signals are produced.

```
<table>
<thead>
<tr>
<th>E</th>
<th>S</th>
<th>R</th>
<th>Q</th>
<th>φ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>X</td>
<td>mlm</td>
<td>mlm</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mlm</td>
<td>mlm</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```

Clock signal.

Triggering

- Level Triggering
  - high stage triggering
  - low stage triggering

- Edge Triggering
  - positive edge triggering
  - negative edge triggering

- Edge tri in clock circuit
  - It is not high or low
  - The transition from low to high to low
flipflop

For a 8 bit number,

Each flipflop has 1 bit

1) SR flip-flop.
2) D flip-flop.
3) JK flip-flop
4) T flip-flop

SR flip flop.

Truth Table

<table>
<thead>
<tr>
<th>Clk</th>
<th>S</th>
<th>R</th>
<th>Qn+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>X</td>
<td>Qn</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Qn</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>invalid</td>
</tr>
</tbody>
</table>
### Characteristic Table

<table>
<thead>
<tr>
<th>$Q_n$</th>
<th>$S$</th>
<th>$R$</th>
<th>$Q_{n+1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$X$</td>
</tr>
</tbody>
</table>

### Excitation Table

<table>
<thead>
<tr>
<th>$Q_n$</th>
<th>$Q_{n+1}$</th>
<th>$S$</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$X$</td>
</tr>
</tbody>
</table>

### Calculation

$Q_{n+1} = S + Q \overline{R}$
D Flip Flop

Truth Table

<table>
<thead>
<tr>
<th>CLK</th>
<th>D</th>
<th>Qn+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>Qn</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Characteristic Table

<table>
<thead>
<tr>
<th>Qn</th>
<th>D</th>
<th>Qn+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Excitation Table

<table>
<thead>
<tr>
<th>Qn</th>
<th>Qn+1</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
JK Flip Flop

CLK  J  K  Qn  Qn+1
0   X  X  Qn
1   0  0  Qn
1   0  1  0
1   1  0  1
1   1  1  invalid

Truth Table
CLK  J  K  Qn+1
0   X  X  Qn
1   0  0  Qn
1   0  1  0
1   1  0  1
1   1  1  Qn

At J=1 & K=1 assume Q=0, Q̅=1 then
old will be 010101

Characteristic Table

\[
\begin{array}{cccc}
Q_n & I & K & Q_{n+1} \\
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 \\
\end{array}
\]

Excitation Table

\[
\begin{array}{cccc}
Q_n & Q_{n+1} & I & K \\
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 \\
\end{array}
\]

K map from Characteristic Table

\[
Q_{n+1} = \overline{Q_n} I + Q_n K
\]
Race Conditions.

If \( k=1 \) \( J=1 \) \( \& \) \( \text{clk}=1 \), it will be ON OFF ON...
when \( \text{clk}=0 \) only it will become a stable state. It is called race condition.

To remove this condition we can use:
1) Edge Triggering
2) Master Slave Method.
Master Slave Method.

At next cycle, \( CLK = 0 \), master and slave.

\( Clk \) at slave = \( I \) \( \oplus \) \( Q \) will change.

1. If \( J = 1 \), \( K = 1 \), \( CLK = 1 \), the old former master.
2. \( Q = 0 \) \( \Rightarrow \) \( \bar{Q} = 1 \).
3. \( \phi \) to slave \( CLK = 0 \).

\( CLK \)
\( J \)
\( K \)
\( Q \)

Master
Slave

Note: The diagram shows the timing and control signals for the master and slave sections of a master-slave flip-flop. The sequence of events is highlighted to illustrate the operation under specific conditions.
A flipflop (Toggle flipflop)

\[
\begin{array}{c}
\text{CLK} \quad \text{Q} \quad \text{Q}_{n+1} \\
0 \quad X \quad Q_n \\
1 \quad 0 \quad Q_n \\
1 \quad 1 \quad \overline{Q_n} \\
\end{array}
\]

Truth Table

\[
\begin{array}{ccc}
\text{Q}_n & \text{T} & \text{Q}_{n+1} \\
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
\end{array}
\]

Characteristic Table

\[
\begin{array}{ccc}
\text{Q}_n & \text{T} & \text{Q}_{n+1} \\
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
\end{array}
\]

Excitation Table

\[
\begin{array}{ccc}
\text{Q}_n & \text{Q}_{n+1} & \text{T} \\
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
\end{array}
\]

\[Q_{n+1} = Q_n \oplus T\]
Half Subtractor using NAND gate

\[ a \quad b \quad D \quad B_0 \quad \bar{a} \quad \bar{a}b. \]

- \[ \begin{array}{cccccc}
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 \\
\end{array} \]

D = a \oplus b.

B_0 = \bar{a}b.

\[ a \quad \bar{b} \quad \\uparrow \quad \bar{a} \quad \\downarrow \quad B_0. \]
\[ C_1 \quad \overline{a \cdot b} \]
\[ C_2 = \overline{a + \overline{a \cdot b}} \]
\[ = \overline{\overline{a}} + \overline{a \cdot b} \]
\[ = a + \overline{a \cdot b} \]
\[ = (\overline{\overline{a} + a}) \cdot (\overline{a} + b) \]
\[ = a + b \]

\[ C_{13} \quad \overline{b \cdot \overline{a \cdot b}} \]
\[ = \overline{b + \overline{a \cdot b}} \]
\[ = b + a \cdot b \]
\[ = (a + b) \cdot (b + b) \]
\[ = a \cdot b \]

\[ C_{14} \quad \overline{a + b} \cdot a + b \]
\[ = \overline{a + b} + a + b \]
\[ = (a \cdot b) + (\overline{a} \cdot b) \]
\[ = A \oplus B \]

\[ C_{15} \quad \overline{a + b} \cdot a + b \]
\[ = \overline{a + b} + a + b \]
\[ = a + b \]
\[ = a \cdot b \]

Counters (Module 5)

- Counters are a specific type of sequential circuit.
- A counter that follows the binary number sequence is called a binary counter.
- n-bit number binary counter: n flip flops, count in binary form 0 to \(2^n - 1\)

Counters are available in two types:
- Synchronous counters
- Ripple counters / Asynchronous counters