COMBINATIONAL AND SEQUENTIAL LOGIC CIRCUITS

Dr. Atulkumar H. Patel
Combinational and Sequential Logic Circuits
Digital circuits can be classified into two types:

- Combinational logic circuits are made up from basic gates (AND, OR, NOT) or universal gates (NAND, NOR) that are "combined" or connected together to produce more complicated switching circuits.
These logic gates are the building blocks of combinational logic circuits.

An example of a combinational circuit is a decoder, which converts the binary code data present at its input into a number of different output lines, one at a time producing an equivalent decimal code at its output.

In these circuits “the outputs at any instant of time depends on the inputs present at that instant only.”
For the design of Combinational digital circuits Basic gates (AND, OR, NOT) or universal gates (NAND, NOR) are used.

Examples for combinational digital circuits are Half adder, Full adder, Half subtractor, Full subtractor, Code converter, Decoder, Multiplexer, De multiplexer, Encoder, ROM, etc.
Combinational And Sequential Logic Circuits

- **Classification of Combinational Logic Circuits**

  - **Combinational Logic Circuits**
    - Arithmetic & Logical Functions
    - Data Transmission
    - Code Converters
    - Adders
    - Subtractors
    - Comparators
    - Multiplexers
    - De multiplexers
    - Encoders
    - Decoders
    - Binary
    - BCD
    - 7 - Segment
Sequential Logic Circuits

Sequential logic differs from combinational logic in that the output of the logic device is dependent not only on the present inputs to the device, but also on past inputs; i.e., the output of a sequential logic device depends on its present internal state and the present inputs.

This implies that a sequential logic device has some kind of memory of at least part of its "history" (i.e. its previous inputs).
A simple memory device can be constructed from combinational devices with which we are already familiar.

By a memory device, we mean a device which can remember if a signal of logic level 0 or 1 has been connected to one of its inputs, and can make this fact available at an output.

A very simple, but still useful, memory device can be constructed from a simple OR gate, as shown in Figure below:
In this memory device, if A and Q are initially at logic 0, then Q remains at logic 0.

However if the single input A ever becomes a logic 1, then the output Q will be logic 1 ever after, regardless of any further changes in the input at A.

In this simple memory, the output is a function of the state of the memory element only; after the memory is "written" then it cannot be changed back.

However, it can be "read." Such a device could be used as a simple read only memory, which could be "programmed" only once.

Examples for sequential digital circuits are Registers, Shift register, Counters etc.
Multiplexers and Demultiplexers
A MUX is a digital switch that has multiple inputs (sources) and a single output (destination).

The select lines determine which input is connected to the output.

**MUX Types**
- 2-to-1 (1 select line)
- 4-to-1 (2 select lines)
- 8-to-1 (3 select lines)
- 16-to-1 (4 select lines)
Combinational And Sequential Logic Circuits

Typical Application of a MUX

Multiple Sources
- MP3 Player Docking Station
- Laptop Sound Card
- Digital Satellite
- Digital Cable TV

Selector
- B A Selected Source
  - 0 0 MP3
  - 0 1 Laptop
  - 1 0 Satellite
  - 1 1 Cable TV

Single Destination
- Surround Sound System
4-to-1 Multiplexer (MUX)

<table>
<thead>
<tr>
<th>B</th>
<th>A</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>D0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>D1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>D2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>D3</td>
</tr>
</tbody>
</table>
Combinational And Sequential Logic Circuits

4-to-1 Multiplexer Waveforms

D0
D1
D2
D3
A
B
Y

Input Data
Select Line
Output Data
Medium Scale Integration MUX

4-to-1 MUX
Inputs
Select
Enable

8-to-1 MUX
Output (Y) (and inverted output)

16-to-1 MUX
**Typical Application of a DEMUX**

- **Single Source**
- **Selector**
- **Multiple Destinations**

### B/W Laser Printer

### Fax Machine

### Color Inkjet Printer

### Pen Plotter

<table>
<thead>
<tr>
<th>B</th>
<th>A</th>
<th>Selected Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>B/W Laser Printer</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Fax Machine</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Color Inkjet Printer</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Pen Plotter</td>
</tr>
</tbody>
</table>
1-to-4 De-Multiplexer (DEMUX)

B | A | D0 | D1 | D2 | D3
---|---|---|---|---|---
0 | 0 | X | 0 | 0 | 0
0 | 1 | 0 | X | 0 | 0
1 | 0 | 0 | 0 | X | 0
1 | 1 | 0 | 0 | 0 | X
1-to-4 De-Multiplexer Waveforms
Medium Scale Integration DEMUX

Note: Most Medium Scale Integrated (MSI) DEMUXs, like the three shown, have outputs that are inverted. This is done because it requires few logic gates to implement DEMUXs with inverted outputs rather than no-inverted outputs.
Memory Element
A **sequential circuit** consists of a **feedback path**, and employs some **memory elements**.

**Sequential circuit** = **Combinational logic** + **Memory Elements**
There are two types of sequential circuits:
- **synchronous**: outputs change only at specific time
- **asynchronous**: outputs change at any time

*Multivibrator* a class of sequential circuits. They can be:
- **bistable** (2 stable states)
- **monostable** or **one-shot** (1 stable state)
- **astable** (no stable state)

Bistable logic devices: *latches* and *flip-flops*.

Latches and flip-flops differ in the method used for changing their state.
**Memory Element**

- **Memory element**: a device which can remember value indefinitely, or change value on command from its inputs.

- **Characteristic table**:

<table>
<thead>
<tr>
<th>Command (at time t)</th>
<th>Q(t)</th>
<th>Q(t + 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>Reset</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>Memorise / No Change</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

  - $Q(t)$: current state
  - $Q(t+1)$ or $Q^+$: next state
Memory element with clock. Flip-flops are memory elements that change state on clock signals.

- **Clock is usually a square wave.**

![Diagram](image_url)
Memory Element

- Two types of triggering/activation:
  - pulse-triggered
  - edge-triggered

- Pulse-triggered
  - latches
  - ON = 1, OFF = 0

- Edge-triggered
  - flip-flops
  - positive edge-triggered (ON = from 0 to 1; OFF = other time)
  - negative edge-triggered (ON = from 1 to 0; OFF = other time)
R S Flip Flop

- Complementary outputs: Q and Q’. 
- When Q is HIGH, the latch is in SET state. 
- When Q is LOW, the latch is in RESET state. 
- For active-HIGH input S-R latch (also known as NOR gate latch),
  
  \[ R=\text{HIGH} \text{ (and } S=\text{LOW}) \text{ a \text{ RESET state} } \]
  \[ S=\text{HIGH} \text{ (and } R=\text{LOW}) \text{ a \text{ SET state} } \]
  both inputs LOW a no change
  both inputs HIGH a Q and Q’ both LOW (invalid)!
For active-LOW input $S'$-$R'$ latch (also known as NAND gate latch),

$R'=$LOW (and $S'=$HIGH) a RESET state

$S'=$LOW (and $R'=$HIGH) a SET state

both inputs HIGH a no change

both inputs LOW a $Q$ and $Q'$ both HIGH (invalid)

Drawback of $S$-$R$ latch: invalid condition exists and must be avoided.
### Characteristics table for active-high input S-R latch:

<table>
<thead>
<tr>
<th>S</th>
<th>R</th>
<th>Q</th>
<th>Q'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>NC</td>
<td>NC</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

No change. Latch remained in present state.

### Characteristics table for active-low input S'-R' latch:

<table>
<thead>
<tr>
<th>S'</th>
<th>R'</th>
<th>Q</th>
<th>Q'</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>NC</td>
<td>NC</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>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

No change. Latch remained in present state.
[R S Flip Flop]

### Active-HIGH input S-R latch

- **10 1 0 0 R**
- **Q 1 1 0 0**
- **S R Q Q’**
  - initial: 1 0 1 0
  - (after $S=1$, $R=0$): 0 0 1 0
  - (after $S=0$, $R=1$): 1 1 0 0
  - invalid!

### Active-LOW input S’-R’ latch

- **10 0 0 1 S**
- **Q’ 0 0 1 10**
- **S’ R’ Q Q’**
  - initial: 1 0 0 1
  - (after $S’=1$, $R’=0$): 1 1 0 1
  - (after $S’=0$, $R’=1$): 0 1 1 0
  - invalid!
R S Flip Flop

- S-R latch + enable input (EN) and 2 NAND gates → gated S-R latch.

- Outputs change (if necessary) only when EN is HIGH.

- Under what condition does the invalid state occur?

- Characteristic table:

<table>
<thead>
<tr>
<th>Q(t)</th>
<th>S</th>
<th>R</th>
<th>Q(t+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>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>indeterminate</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</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>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>indeterminate</td>
</tr>
</tbody>
</table>

\[ S \cdot R = 0 \]

\[ Q(t+1) = S + R' \cdot Q \]

\[ EN = 1 \]
**Make R input equal to $S'$ → gated D latch.**

**D latch eliminates the undesirable condition of invalid state in the S-R latch.**

- **When** $EN$ **is HIGH,**
  - $D=HIGH$ → latch is SET
  - $D=LOW$ → latch is RESET
- **Hence when** $EN$ **is HIGH,** $Q$ 'follows' the $D$ (data) input.

**Characteristic table:**

<table>
<thead>
<tr>
<th>$EN$</th>
<th>$D$</th>
<th>$Q(t+1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0   Reset</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1   Set</td>
</tr>
<tr>
<td>0</td>
<td>X</td>
<td>$Q(t)$ No change</td>
</tr>
</tbody>
</table>

When $EN=1$, $Q(t+1) = D$
R S Flip Flop

- **Latch circuits are not suitable in synchronous logic circuits.**

- When the enable signal is active, the excitation inputs are gated directly to the output Q. Thus, any change in the excitation input immediately causes a change in the latch output.

- The problem is solved by using a special timing control signal called a *clock* to restrict the times at which the states of the memory elements may change.

- This leads us to the edge-triggered memory elements called **flip-flops**.
Edge Triggered Flip - Flops
**Edge Triggered Flip - Flops**

- **Flip-flops**: synchronous bistable devices

- Output changes state at a specified point on a triggering input called the clock.

- Change state either at the positive edge (rising edge) or at the negative edge (falling edge) of the clock signal.

![Diagram](https://via.placeholder.com/150)

- Positive edges
- Negative edges
- Clock signal
Edge Triggered Flip - Flops

- S-R, D and J-K edge-triggered flip-flops. Note the “>” symbol at the clock input.

Positive edge-triggered flip-flops

Negative edge-triggered flip-flops
Edge Triggered Flip - Flops

- **S-R flip-flop**: on the triggering edge of the clock pulse,
  - $S=$HIGH (and $R=$LOW) a SET state
  - $R=$HIGH (and $S=$LOW) a RESET state
  - both inputs LOW a no change
  - both inputs HIGH a invalid

- **Characteristic table of positive edge-triggered S-R flip-flop**:

<table>
<thead>
<tr>
<th>$S$</th>
<th>$R$</th>
<th>CLK</th>
<th>$Q(t+1)$</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>X</td>
<td>$Q(t)$</td>
<td>No change</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>↑</td>
<td>0</td>
<td>Reset</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>↑</td>
<td>1</td>
<td>Set</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>↑</td>
<td>?</td>
<td>Invalid</td>
</tr>
</tbody>
</table>

$X =$ irrelevant ("don’t care")

$↑ =$ clock transition LOW to HIGH
**Edge Triggered Flip - Flops**

- It comprises 3 parts:
  - a basic **NAND latch**
  - a **pulse-steering circuit**
  - a **pulse transition detector (or edge detector) circuit**
- The pulse transition detector detects a rising (or falling) edge and produces a very short-duration spike.

The pulse transition detector.

- Positive-going transition (rising edge)
- Negative-going transition (falling edge)
**Edge Triggered Flip - Flops**

- **D flip-flop**: single input $D$ (data)
  - $D=\text{HIGH}$ a SET state
  - $D=\text{LOW}$ a RESET state
- **$Q$ follows $D$ at the clock edge.**
- **Convert S-R flip-flop into a D flip-flop**: add an inverter.

A positive edge-triggered D flip-flop formed with an S-R flip-flop.

<table>
<thead>
<tr>
<th>$D$</th>
<th>$CLK$</th>
<th>$Q(t+1)$</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>↑</td>
<td>1</td>
<td>Set</td>
</tr>
<tr>
<td>0</td>
<td>↑</td>
<td>0</td>
<td>Reset</td>
</tr>
</tbody>
</table>

$\uparrow =$ clock transition LOW to HIGH
Edge Triggered Flip - Flops

- J-K flip-flop: Q and Q' are fed back to the pulse-steering NAND gates.
- No invalid state.
- Include a toggle state.
  - J=HIGH (and K=LOW) a SET state
  - K=HIGH (and J=LOW) a RESET state
  - both inputs LOW a no change
  - both inputs HIGH a toggle
- J-K flip-flop.

- Characteristic table.

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>CLK</th>
<th>Q(t+1)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>↑</td>
<td>Q(t)</td>
<td>No change</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>↑</td>
<td>0</td>
<td>Reset</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>↑</td>
<td>1</td>
<td>Set</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>↑</td>
<td>Q(t)'</td>
<td>Toggle</td>
</tr>
</tbody>
</table>

\[ Q(t+1) = J \cdot Q' + K' \cdot Q \]
5 5 5 5 Timers
The 555 timer is an integrated circuit specifically designed to perform signal generation and timing functions.

- The 555 timer consists of two voltage comparators, a bi-stable flip flop, a discharge transistor, and a resistor divider network.
555 Timers

Diagram of a 555 timer circuit, showing the connections for control, threshold, trigger, discharge, and output. The circuit includes a control flip-flop, trigger comparators, and a transistor.
555 Timers

- The voltage divider (blue) has three equal 5K resistors. It divides the input voltage (Vcc) into three equal parts.

- The two comparators (red) are op-amps that compare the voltages at their inputs and saturate depending upon which is greater.
  - The Threshold Comparator saturates when the voltage at the Threshold pin (pin 6) is greater than \((2/3)\text{Vcc}\).
  - The Trigger Comparator saturates when the voltage at the Trigger pin (pin 2) is less than \((1/3)\text{Vcc}\).
555 Timers

- The flip-flop (green) is a bi-stable device. It generates two values, a “high” value equal to Vcc and a “low” value equal to 0V.
  - When the Threshold comparator saturates, the flip-flop is Reset (R) and it outputs a low signal at pin 3.
  - When the Trigger comparator saturates, the flip-flop is Set (S) and it outputs a high signal at pin 3.

- The transistor (purple) is being used as a switch, it connects pin 7 (discharge) to ground when it is closed.
  - When Q is low, Qbar is high. This closes the transistor switch and attaches pin 7 to ground.
  - When Q is high, Qbar is low. This open the switch and pin 7 is no longer grounded
Applications include:

- precision timing,
- pulse generation,
- sequential timing,
- time delay generation and pulse width modulation (PWM).

Astable Multivibrator puts out a continuous sequence of pulses.

Monostable Multivibrator (or one-shot) puts out one pulse each time the switch is connected.