EE-421: Digital Systems Design

Expressing Sequential Circuits

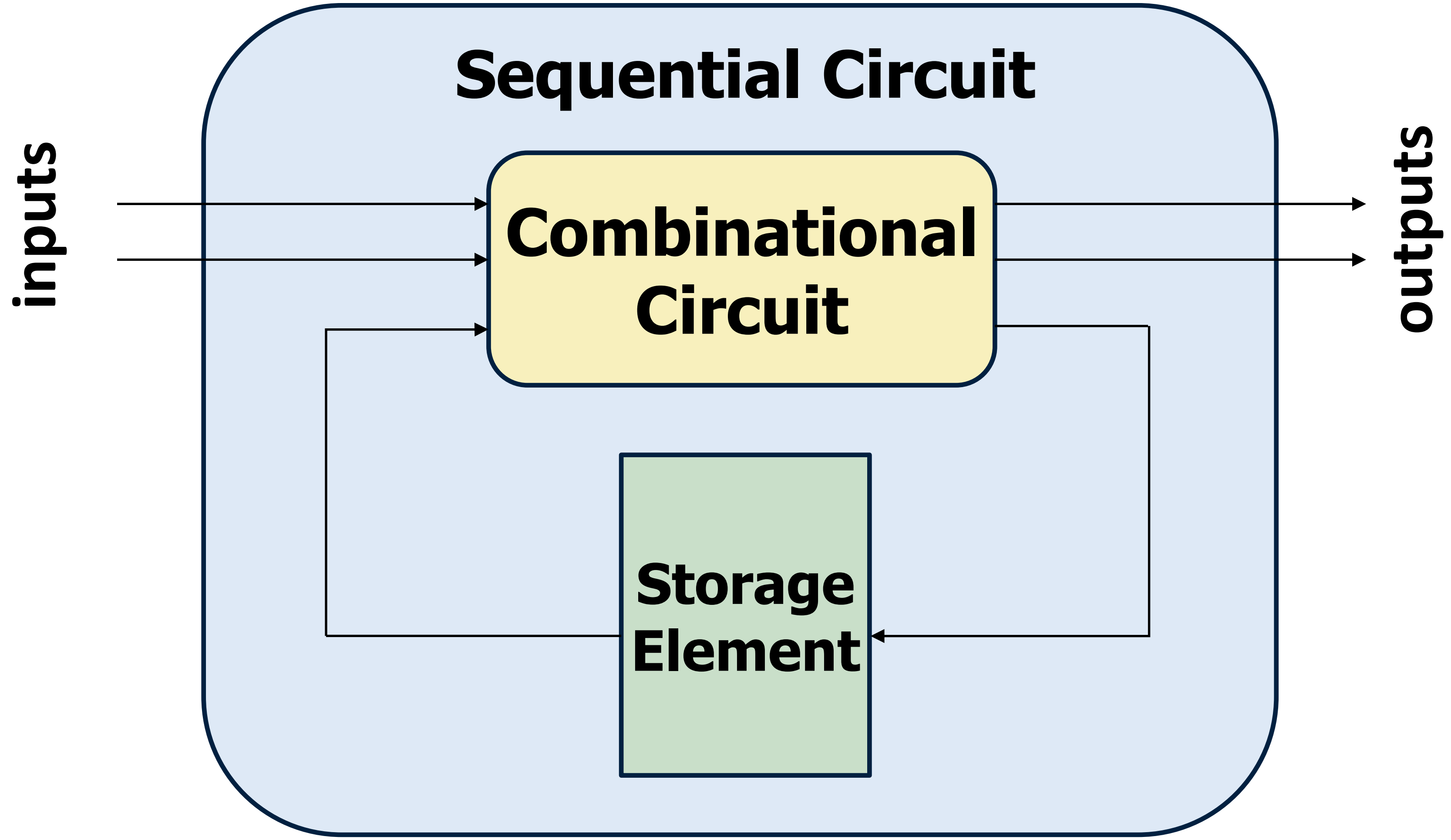
Instructor: Dr. Rehan Ahmed [rehan.ahmed@seecs.edu.pk]

NUST

SCHOOL OF ELECTRICAL ENGINEERING &
COMPUTER SCIENCE (SEECS)

Quick Review: Sequential Circuits

Combinational + Memory = Sequential

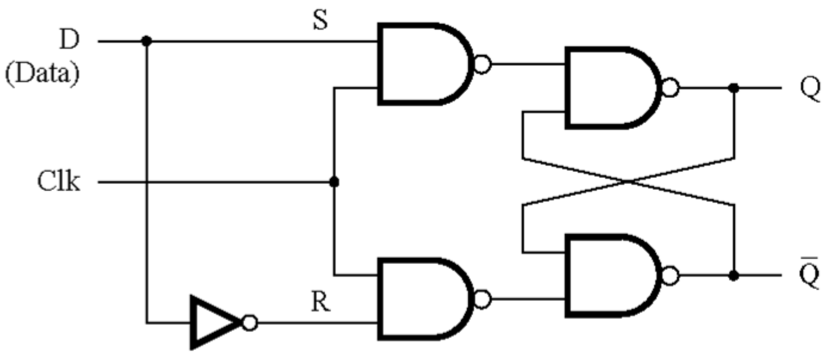


Review: Sequential Logic

- In a combinational circuit, the values of the outputs are determined solely by the present values of its inputs.

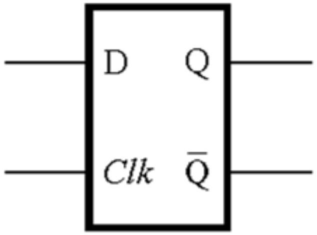
- A sequential circuit has states, which in conjunction with the present values of inputs determine its behavior:
  - Such circuits include storage elements that store the values of logic signals
  - The contents of the storage elements are said to represent the state of the circuit

Review: Gated D Latch

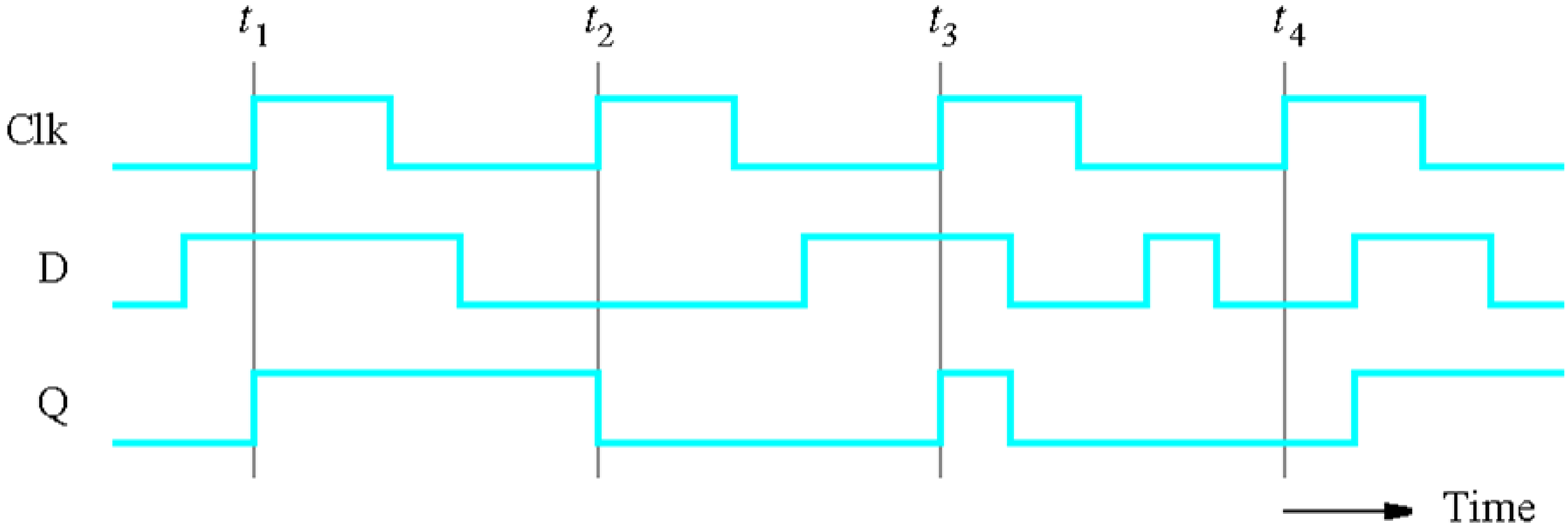


Clk	D	$Q(t+1)$
0	x	$Q(t)$
1	0	0
1	1	1

(b) Characteristic table

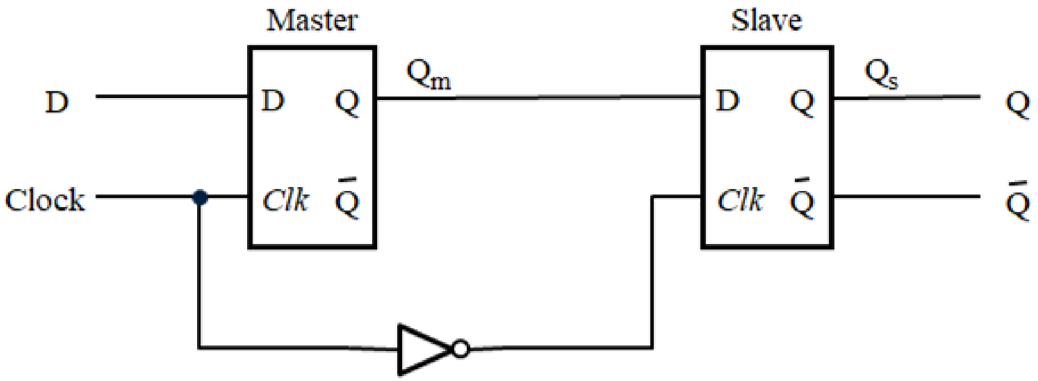


(c) Graphical symbol

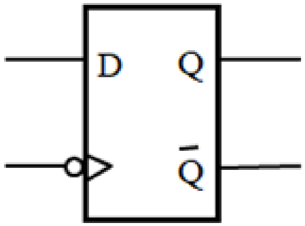


(d) Timing diagram

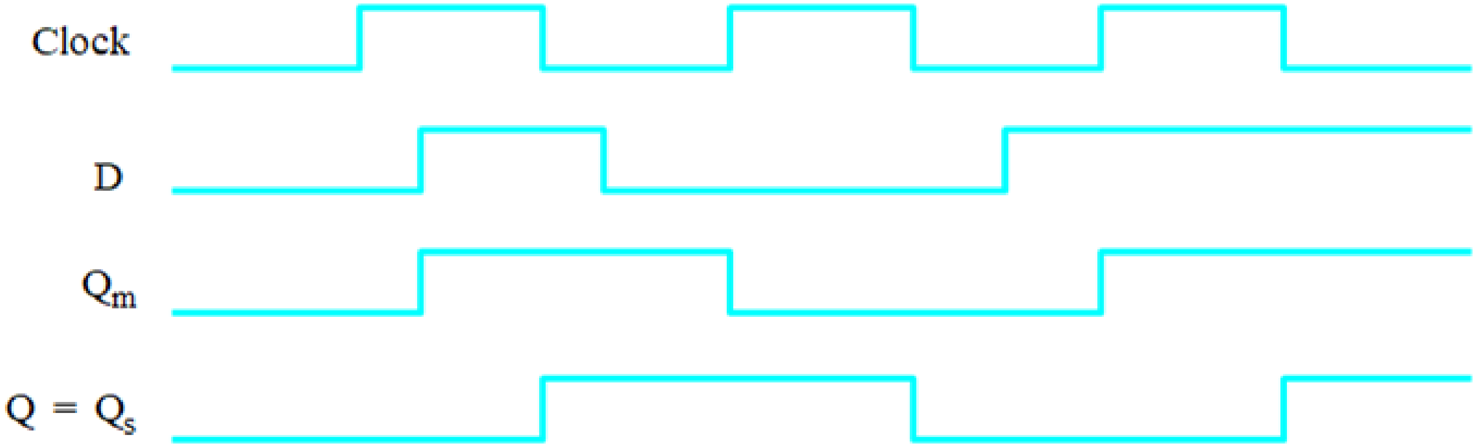
Review: Master Slave D-flip flop



(a) Circuit

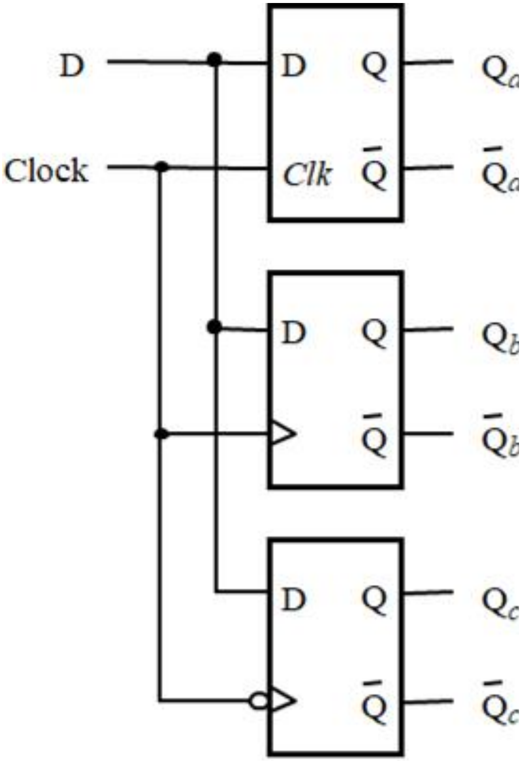


(c) Graphical symbol

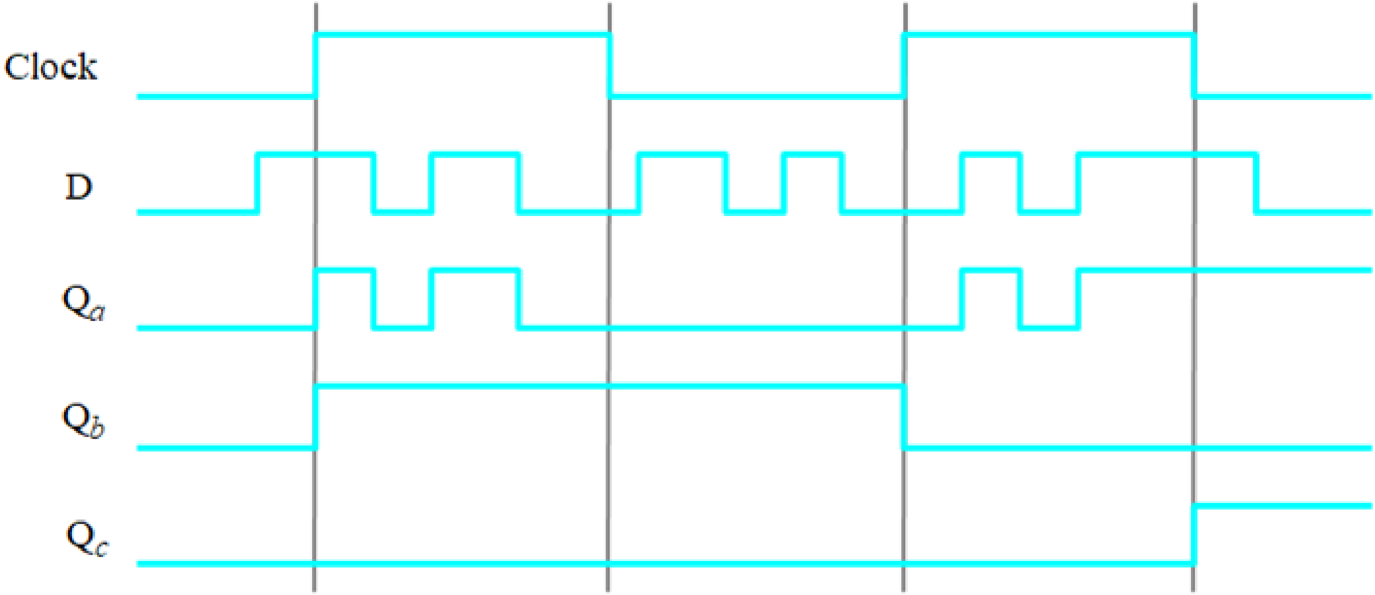


(b) Timing diagram

Level-Sensitive vs Edge-Triggered Storage Elements



(a) Circuit



(b) Timing diagram

A flip-flop is a storage element that can have its output state changed only on the edge of the controlling clock signal

Discussion: Level-Sensitive vs Edge-Triggered Storage Elements

- Ref to fig in the previous slide:
- The D input changes its values more than once during each half of the clock cycle.
- Observe that:
  - the gated D latch follows the D input as long as the clock is high.
  - The positive-edge-triggered flip-flop responds only to the value of D when the clock changes from 0 to 1
  - The negative-edge-triggered flip-flop responds only to the value of D when the clock changes from 1 to 0.

Expressing Sequential Circuits in Verilog-HDL

Sequential Logic in Verilog

- Define blocks that have memory:
  - *Flip-Flops, Latches, Finite State Machines*

- Sequential Logic state transition is triggered by a "CLOCK" signal:
  - Latches are sensitive to level of the signal
  - Flip-flops are sensitive to the transitioning of signal

- Combinational constructs are **not** sufficient:
  - We need **new constructs**:
    - **`always_ff`**
    - **`posedge/negedge`**

Recall: The “always” Block

```
always @ (sensitivity list)
     statement;
```

Whenever the event in the sensitivity list occurs,
the statement is executed

Describing a Positive-Edge D-Flip Flop

```
module flop(input clk,
            input [3:0] d,
            output [3:0] q);

  always_ff @ (posedge clk)
    q <= d;                  // pronounced “q gets d”

endmodule
```

- posedge defines a rising edge (transition from 0 to 1).
- Statement executed when the clk signal rises (posedge of clk)
- Once the clk signal rises: the value of d is copied to q

Describing a Positive-Edge D-Flip Flop

```
module flop(input logic clk,
            input logic d,
            output logic q);

  always_ff @ (posedge clk)
    q <= d;                // pronounced “q gets d”

endmodule
```

- **Also note:**
  - assign statement is **not** used within an always block
  - <= describes a non-blocking assignment
  - clk is the only input that can cause an event on the output
    - Therefore clk is the only signal in the sensitivity list
  - Special sensitivity list @(posedge Clock):
    - This event expression tells the Verilog compiler that any reg variable assigned a value in the always construct is the output of a D flip-flop

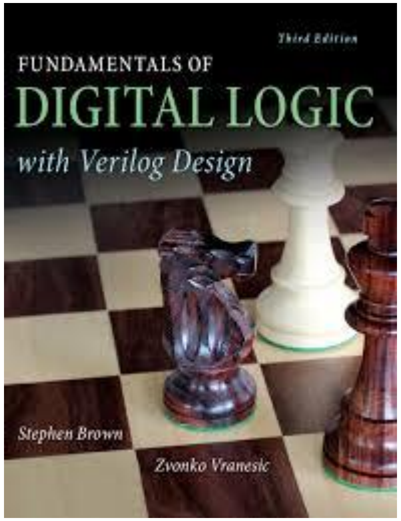
Describing a Positive-Edge D-Flip Flop

```
module flop(input logic clk,
            input logic d,
            output logic q);

  always_ff @ (posedge clk)
    q <= d;                // pronounced “q gets d”

endmodule
```

- Assigned variables need to be declared as reg
- The name reg does not necessarily mean that the value is a register (It could be, but it does not have to be)
  - We have seen examples before while doing combinational ccts!!!

Recommended Reading

- Digital System Design with Verilog HDL, 3/e, b **S**tephen Brown and **Z**vonko Vranesic. [**S&Z**]
  - S&Z,
    - Chapter-5



THANK YOU