CME 341: Introduction to Synchronous Logic in Verilog

Brian Berscheid

Department of Electrical and Computer Engineering
University of Saskatchewan

Today's agenda

1. Introduction to Synchronous Logic in Verilog

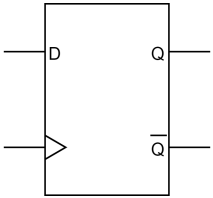
Introduction to Synchronous Logic in Verilog

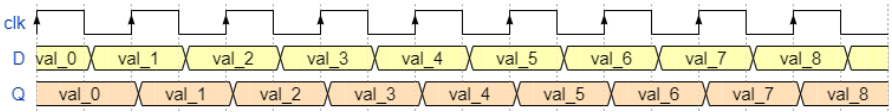
Synchronous logic[1]

- Refers to logic that operates based on a clock signal (typically at the positive edge)
- Key element is a "flip-flop" or "register": passes signal from input to output on clock
- Generates a sequence of values which change on positive clock edges
- Note that synchronous logic generally combines flip-flops with combinational logic gates
- Quartus can infer flip-flops (FFs) using behavioral code
  - ▷ Procedures beginning with `always @ (posedge clk)`
  - ▷ The behavior that we describe will determine what FF primitive will be instantiated and what logic is built around it

[1] Also known as sequential logic

Review: D flip-flop

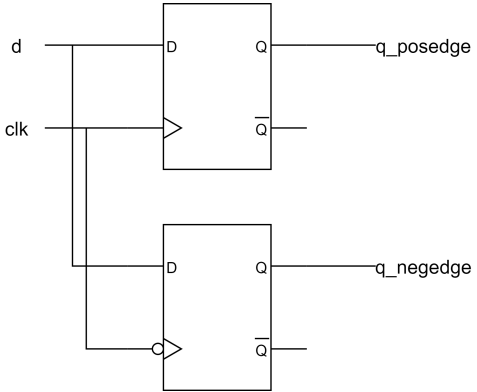




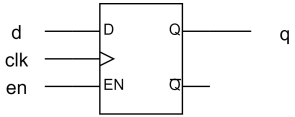
"When the positive edge of the clock arrives, copy D to Q."

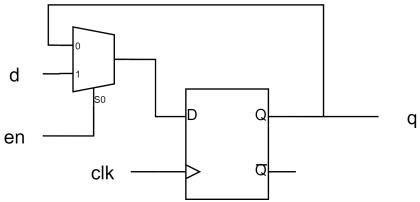
Verilog code for basic D-FF

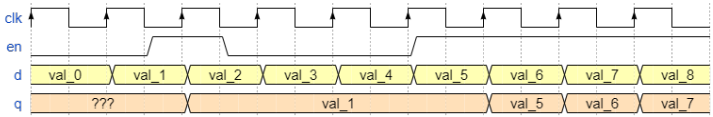
```
reg q_posedge;
reg q_negedge;

// positive edge triggered
always @ (posedge clk)
  q_posedge = d;

// negative edge triggered
always @ (negedge clk)
  q_negedge = d;
```



D flip-flop with enable







"When the positive edge of the clock arrives, copy D to Q only if enable is active. Otherwise, do not update Q."

Based on this description, try to write the Verilog code for an enabled D-FF.

Verilog for D-FF with enable

```
reg q;

always @ (posedge clk)
  if (enable == 1'b1)
    q = d;
  else
    q = q;  // optional, but recommended for clarity
            // (compiler will insert for us if we forget)
```

Note: Feedback in synchronous logic is OK! The guideline to avoid feedback applies only to purely combinational logic. Feedback loops which include registers are synchronized to a clock signal and are generally safe from instability (providing setup and hold requirements are met).

Thank you!
Have a great day!