Verilog Tutorial II

310H:

Honors Computer Organization and Programming

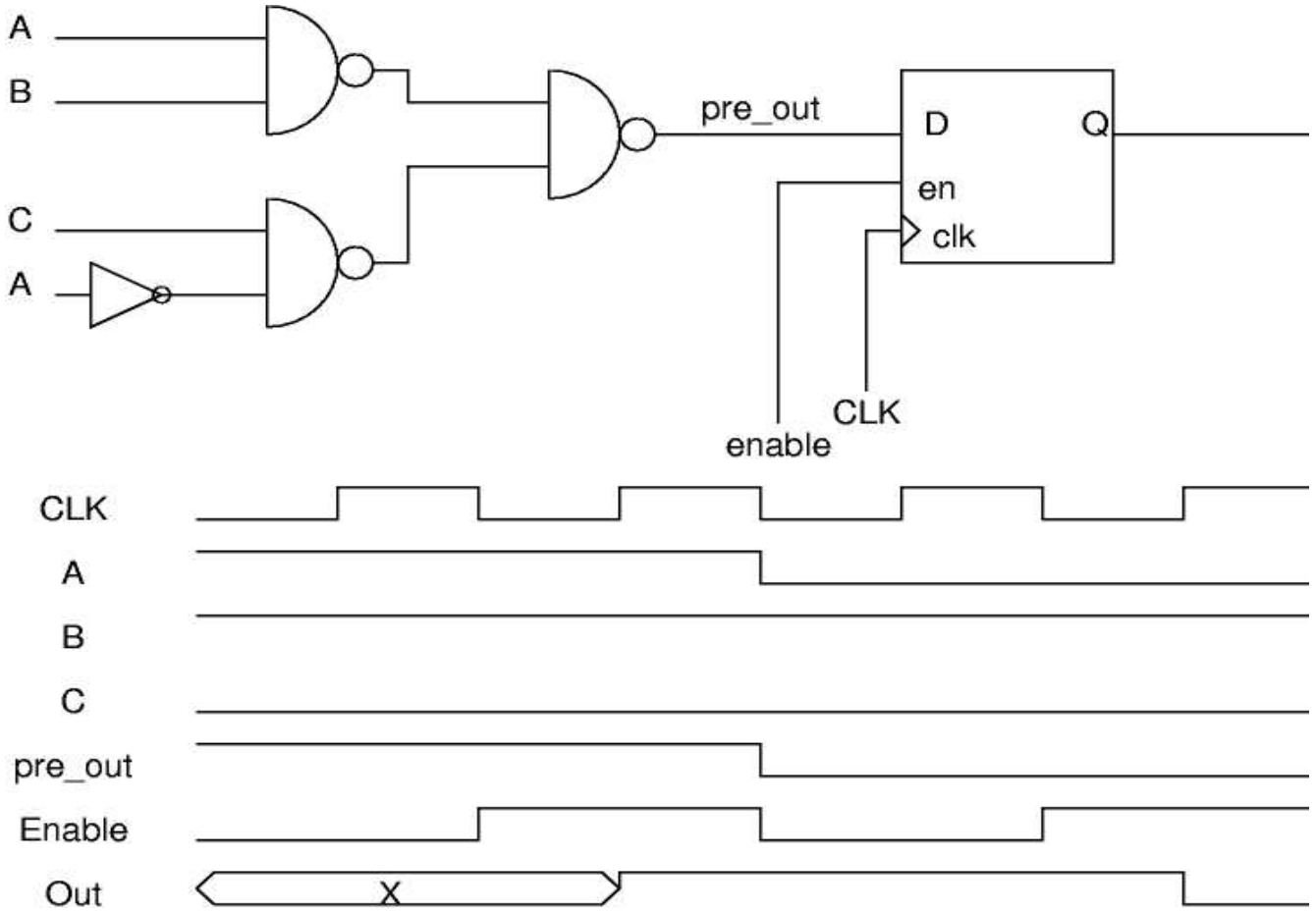
Outline

- Sequential Logic in Verilog
  - Schematic Example
  - Verilog
- Delay Modeling in Verilog
  - Assignment Delay
  - Hold and setup time checks
- Tristate Buffers and Busses in Verilog
  - Schematic Example
  - Verilog

Sequential Logic

- Uses a clock to synchronize logic
- Allows for preservation of state over time
  - “Storage” of data
  - Feed back of current state to determine next state
- Allows problems to be broken into discrete steps which can be pipelined

Schematic of Sequential logic



Verilog for sequential logic

```
seq1 ( Out, A, B, C, enable, CLK ) ;
   input  A, B, C, enable, CLK;
   output Out;
   wire   pre_out, net0, net1, A_bar;

   invt iv0 (.out (A_bar), .in(A));

   nand2 na0 (.out (net0), .in0 (A), .in1 (B));

   nand2 na1 (.out (net1), .in0 (A_bar), .in1 (C));

   nand2 na3 (.out (pre_out), .in0 (A), .in1 (B));

   dff mydff (.q (Out), .d (pre_out), .en(enable), .clk(CLK));

endmodule
```

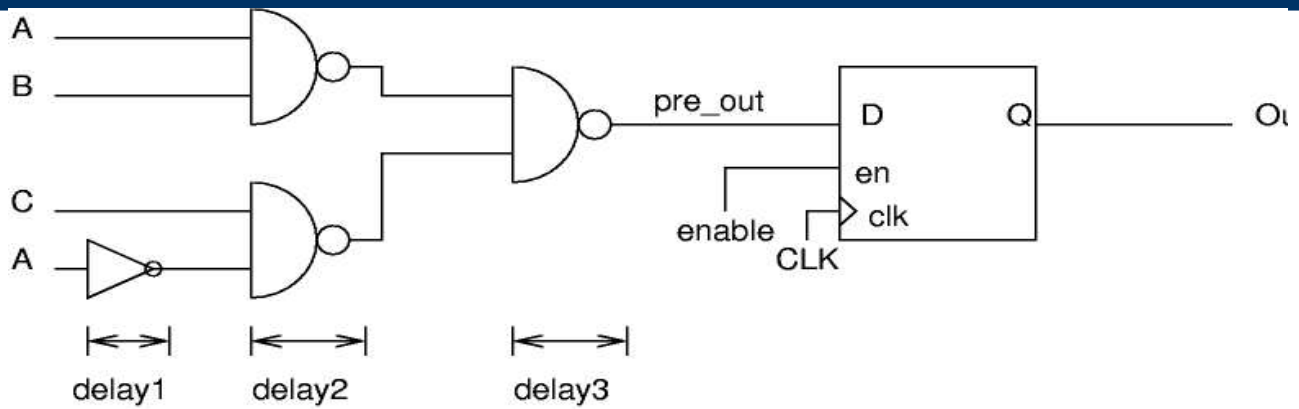
D-Flip Flop (Behavioral)

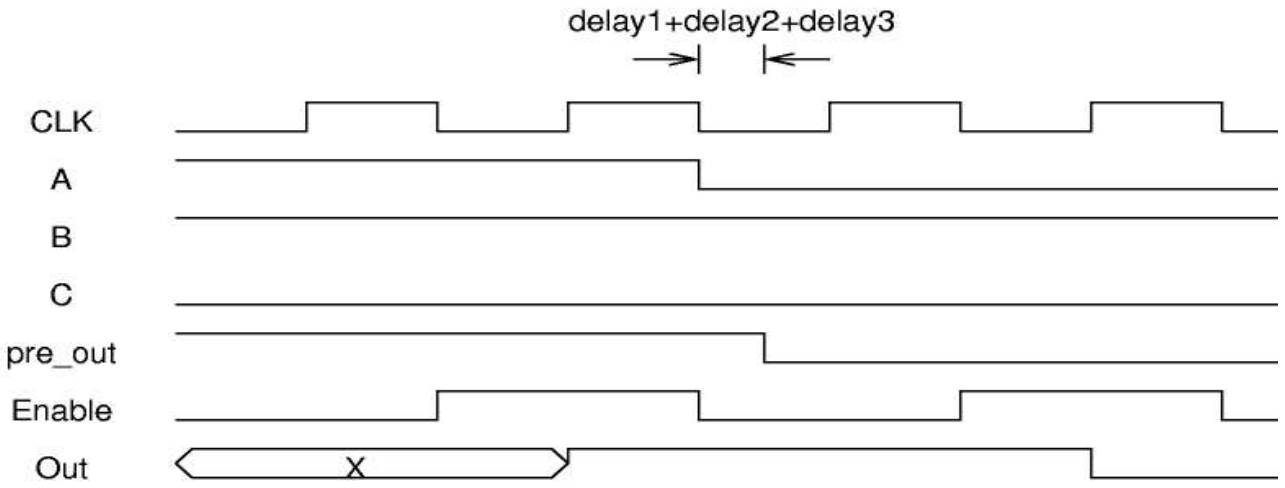
```
module dff (  q, d, clk, en );
   input  d, clk, en;
   output q;
   reg    q;

   always @ ( posedge clk ) begin
      if (en) begin
        q = d;
      end
   end

endmodule // dff
```

Delay Modeling in Verilog

- Verilog allows for path based delay modeling
- Verilog has constructs to check setup and hold time.
- Can be used to ensure signals are available when the clock rises.
- Reasonable results for “first-pass” timing.

Timing example





Adding delay to Combinatorial logic

```
module nand2 (out, in0, in1) ;
   input  in0, in1;
   output out;
   assign #1 out = ~(in0 & in1);
endmodule // and2


module invt (out, in) ;
   input  in;
   output out;
   assign #1 out = ~(in);
endmodule
```

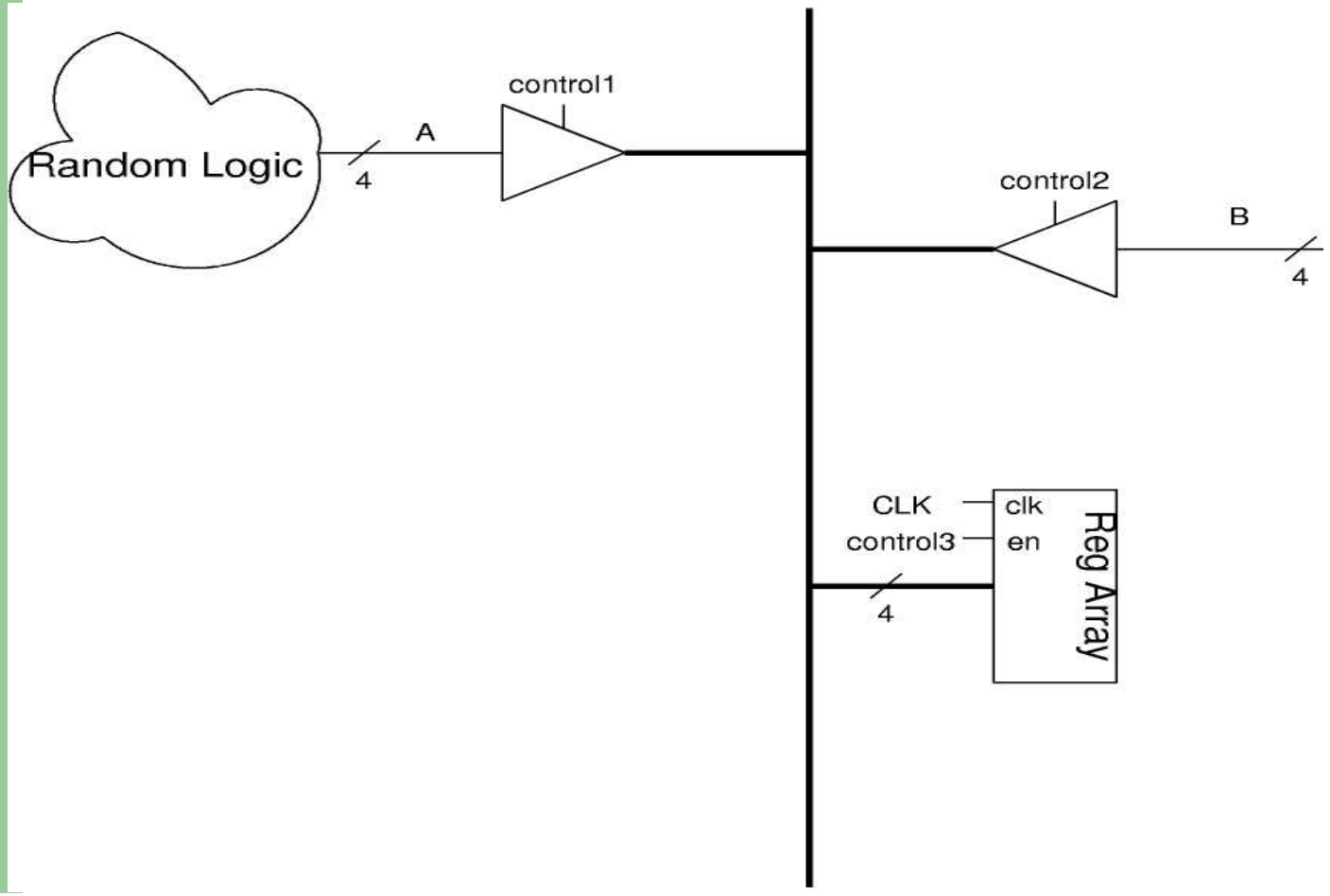
Evaluating setup and hold time for Sequential Logic

```
module dff (q, d, clk, en ) ;
   input  d, clk, en;
   output q;
   reg    q;

   always @ ( posedge clk ) begin
      if (en) begin
        q = d;
      end
   end

   specify
      $setup( d, posedge clk, 1);
      $hold( d, posedge clk, 1);
   endspecify
endmodule // dff
```

Tristate drivers and Busses

- Allow multiple drivers onto one bus as long as the control lines are mutex (mutually exclusive)
- Can act like an implied mux
- When not driven bus will “float” to a Z value.

Tristate Bus w/ drivers and receiver



Four bit tristate bus verilog

```
module tri_state ( q, A, B, control1, control2, enable, clk ) ;
   input  [3:0] A, B;
   input  control1, control2, enable, clk;
   output [3:0] q;
   tri [3:0]     tri_bus;

   tribuf driverA[3:0] (.out(tri_bus), .in(A), .control(control1));

   tribuf driverB[3:0] (.out(tri_bus), .in(B), .control(control2));

   dff mydff[3:0] (.q (q), .d (tri_bus), .en(enable), .clk(clk));

endmodule // tri_state
```

Behavioral Verilog for tristate buffer

```
module tribuf ( out, in, control ) ;
   input  in, control;
   output out;

   assign #2 out = control ? in : 1'bz;

endmodule // tribuf
```