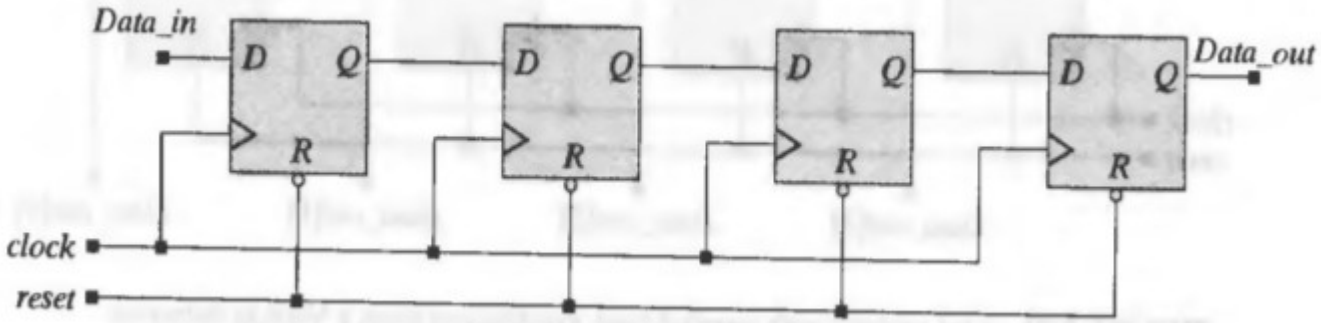
EEL 4783: HDL in Digital System Design

Lecture 6: Verilog Examples

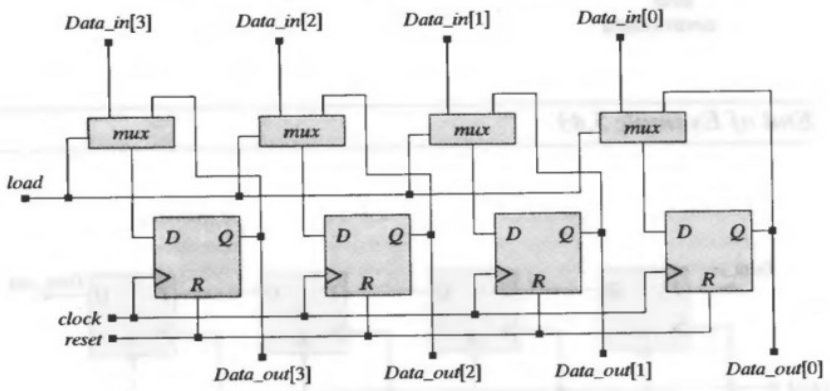
Prof. Mingjie Lin

Shift Register



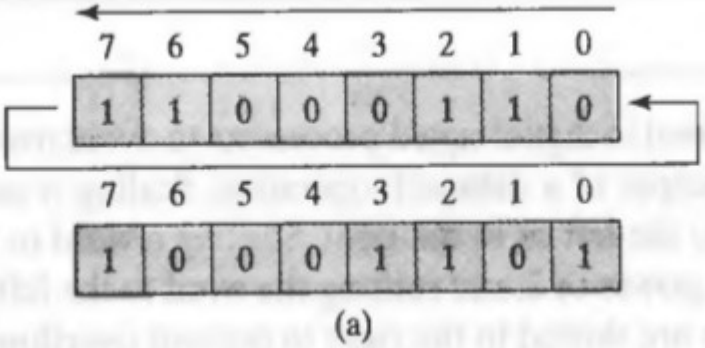

```
module Shift_reg4 (Data_out, Data_in, clock, reset);
  output                  Data_out;
  input                   Data_in, clock, reset;
  reg   [3: 0]            Data_reg;

  assign                  Data_out = Data_reg[0];
  always @  (negedge reset or posedge clock)
    begin
      if (reset == 1'b0)  Data_reg <= 4'b0;
     else                 Data_reg <= {Data_in, Data_reg[3:1]};
    end
endmodule
```

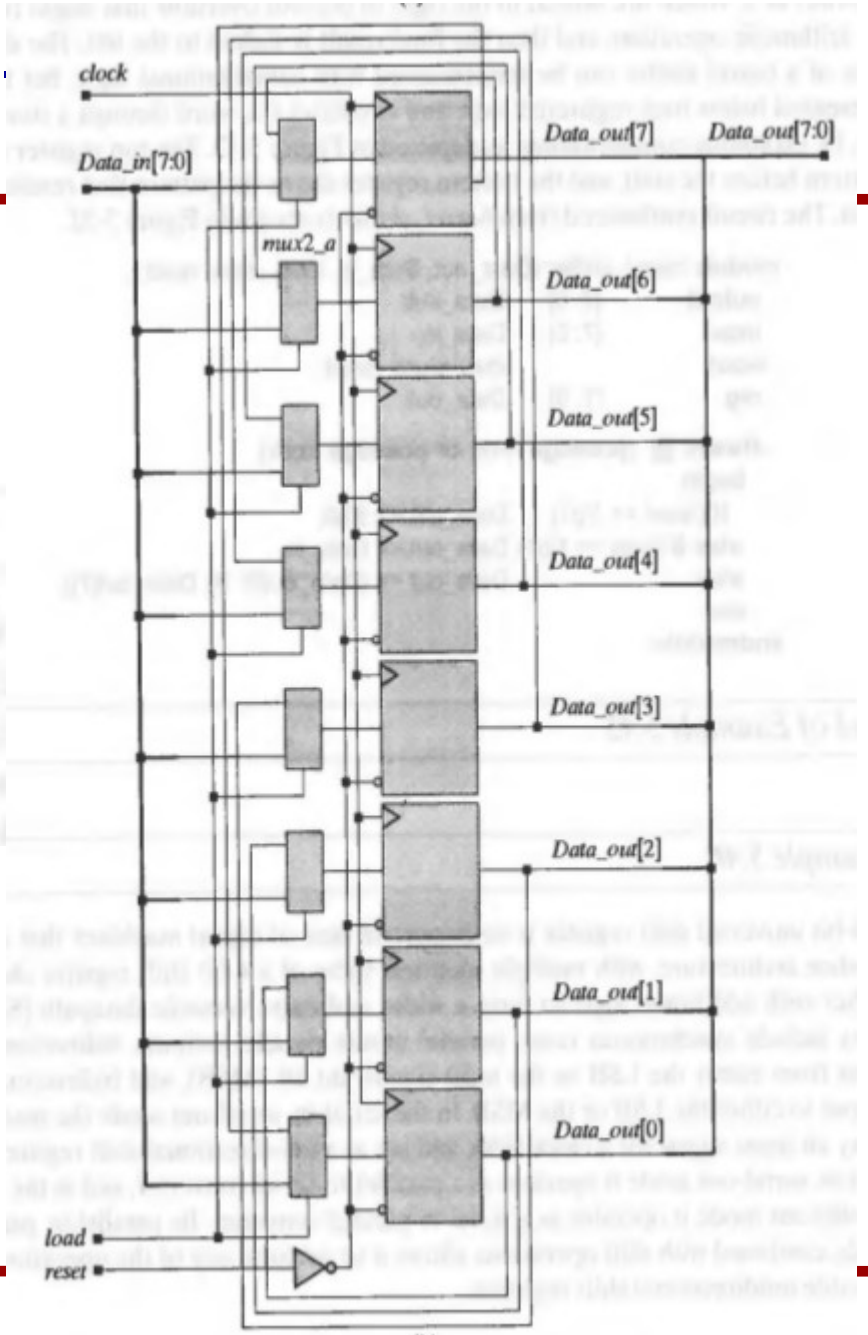
SR with Parallel Loads




```
module Par_load_reg4 (Data_out, Data_in, load, clock, reset);
 input        [3: 0]   Data_in;
 input                 load, clock, reset;
 output       [3: 0]   Data_out;                  // Port size
 reg                   Data_out;                  // Data type

 always @  (posedge reset or posedge clock)
  begin
    if (reset == 1'b1)              Data_out <= 4'b0;
    else if (load == 1'b1)          Data_out <= Data_in;
  end
endmodule
```

Barrel SR


7 6 5 4 3 2 1 0
1 1 0 0 0 1 1 0
7 6 5 4 3 2 1 0
1 0 0 0 1 1 0 1
(a)


clock
Data_in[7:0]
mux2_a
Data_out[7]
Data_out[7:0]
Data_out[6]
Data_out[5]
Data_out[4]
Data_out[3]
Data_out[2]
Data_out[1]
Data_out[0]
load
reset
(b)

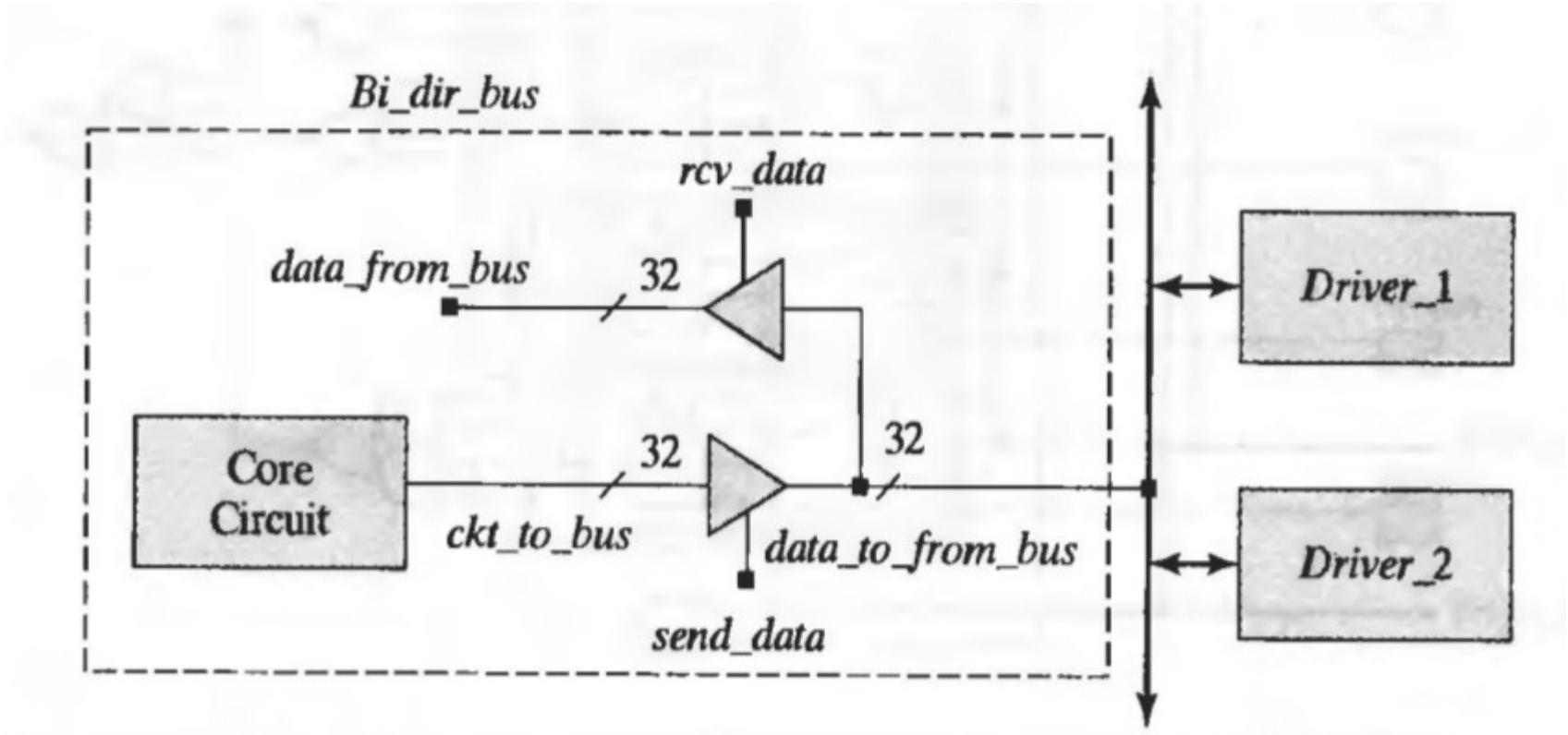
Barrel SR

```
module barrel_shifter (Data_out, Data_in, load, clock, reset);
  output        [7: 0]    Data_out;
  input         [7: 0]    Data_in;
  input                   load, clock, reset;
  reg           [7: 0]    Data_out;

  always @  (posedge reset or posedge clock)
    begin
      if (reset == 1'b1)      Data_out <= 8'b0;
    else if (load == 1'b1) Data_out <= Data_in;
    else                      Data_out <= {Data_out[6: 0], Data_out[7]};
    end
endmodule
```

Universal RF

```
module Universal_Shift_Reg  (Data_Out, MSB_Out, LSB_Out, Data_In,
  MSB_In, LSB_In, s1, s0, clk, rst);
  output  [3: 0]  Data_Out;
  output          MSB_Out, LSB_Out;
  input   [3: 0]  Data_In;
  input           MSB_In, LSB_In;
  input           s1, s0, clk, rst;
  reg             Data_Out;

  assign MSB_Out = Data_Out[3];
  assign LSB_Out = Data_Out[0];

  always @ (posedge clk) begin
   if (rst) Data_Out <= 0;
   else  case ({s1, s0})
     0:   Data_Out <= Data_Out;                  // Hold
     1:   Data_Out <= {MSB_In, Data_Out[3:1]};   // Serial shift from MSB
     2:   Data_Out <= {Data_Out[2: 0], LSB_In};  // Serial shift from LSB
     3:   Data_Out <= Data_In;                   // Parallel Load
   endcase
  end
endmodule
```

Bi-Directional Bus



Bi-Directional Bus

```
module Bi_dir_bus (data_to_from_bus, send_data, rcv_data);
  inout  [31: 0]   data_to_from_bus;
  input            send_data, rcv_data;
  wire   [31: 0]   ckt_to_bus;
  wire   [31: 0]   data_to_from_bus, data_from_bus;

  assign data_from_bus = (rcv_data) ? data_to_from_bus : 32'bz;
  assign data_to_from_bus = (send_data) ? reg_to_bus : data_to_from_bus;

  // Behavior using data_from_bus and generating
  // ckt_to_bus goes here

endmodule
```

When a FF is created?

Synthesis Tip

A variable that is referenced within an edge-sensitive behavior before it is assigned value in the behavior will be synthesized as the output of a flip-flop.

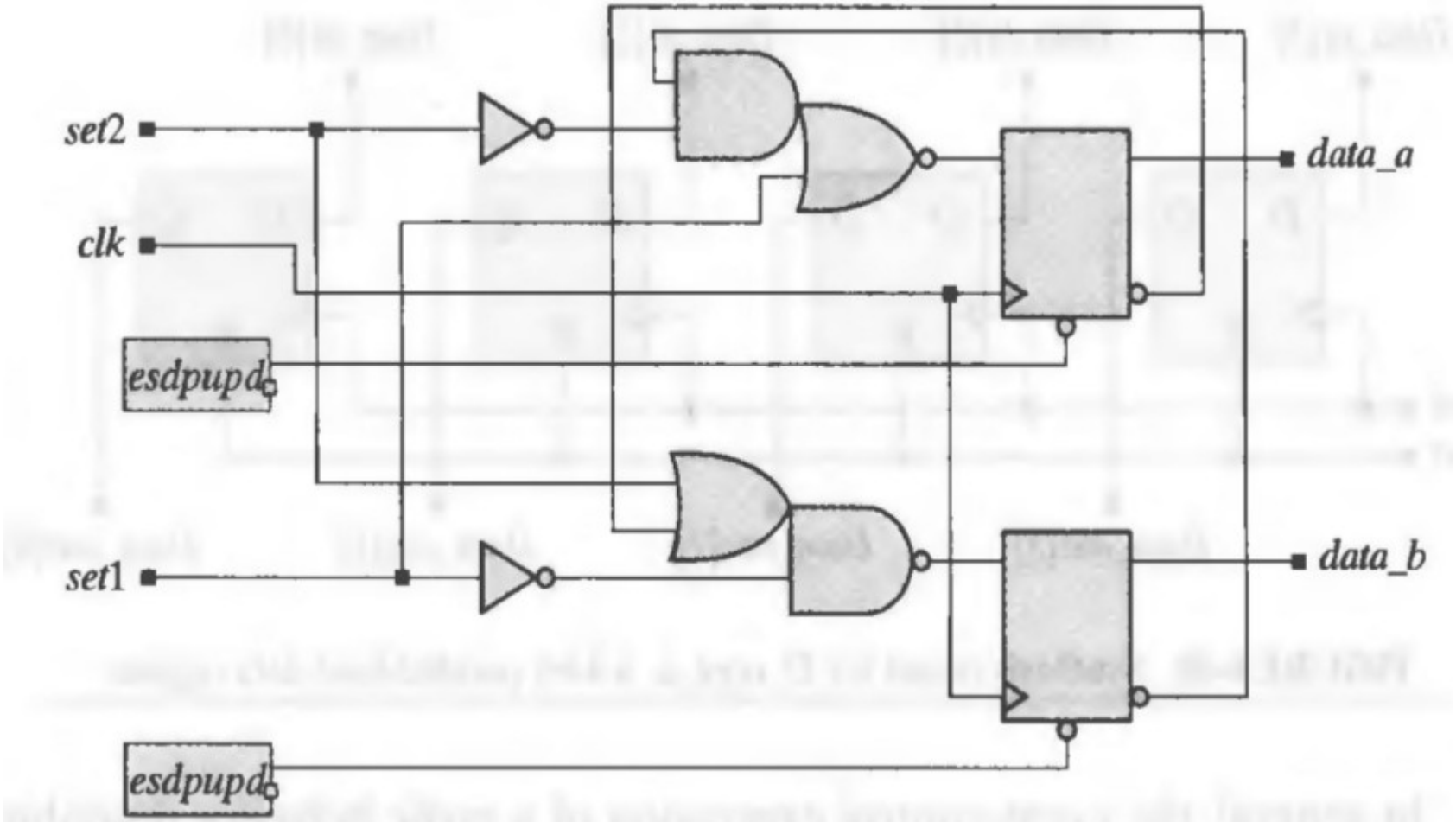
Sync. Data Swapping

```
module swap_synch (data_a, data_b set1, set2, clk,);
 output        data_a, data_b;
 input         clk, set1, set2, swap;
 reg           data_a, data_b;

 always @ (posedge clk)
  begin
   if (set1) begin data_a <= 1; data_b <= 0; end else
    if (set2) begin data_a <= 0; data_b < = 1; end
     else
      begin
       data_b <= data_a;
       data_a <= data_b;
      end
  end
endmodule
```

Sync. Data Swapping

```
module swap_synch (data_a, data_b set1, set2, clk,);
 output        data_a, data_b;
 input         clk, set1, set2, swap;
 reg           data_a, data_b;

 always @ (posedge clk)
  begin
   if (set1) begin data_a <= 1; data_b <= 0; end else
    if (set2) begin data_a <= 0; data_b < = 1; end
     else
      begin
       data_b <= data_a;
       data_a <= data_b;
      end
  end
endmodule
```

Sync. Data Swapping



S

S

S

S

S

Final issues

- Please fill out the student info sheet before leaving
- Come by my office hours (right after class)
- Any questions or concerns?