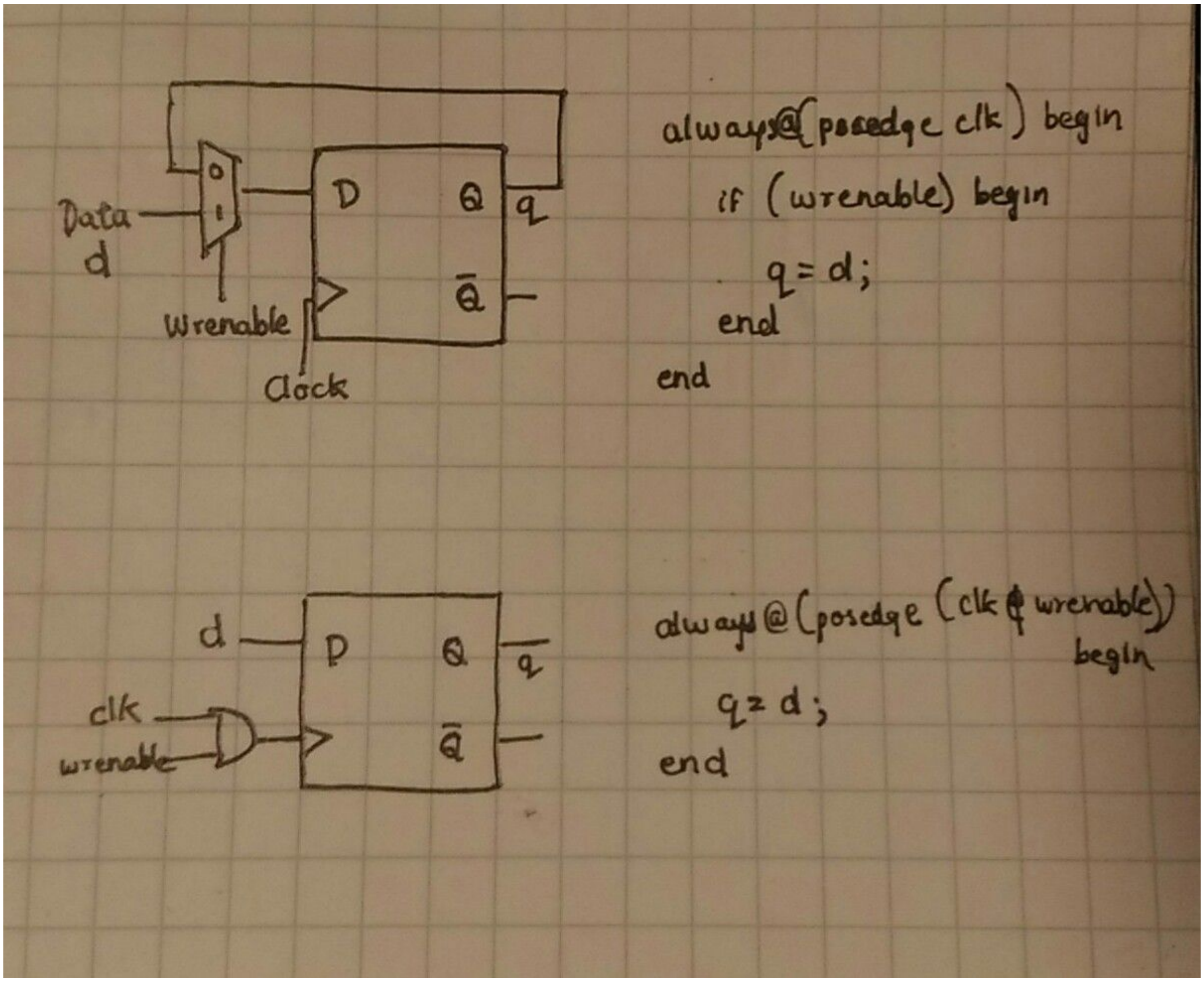
Deliverable 1:



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

```
always@(posedge (clk & wrenable))
                                begin
   q = d;
end
```

Deliverable 2:

```
module register32
(
output reg[31:0]  q,
input[31:0]       d,
input             wrenable,
input             clk
);
    always @(posedge clk) begin
        if(wrenable) begin
            q = d;
        end
    end
endmodule
```

Deliverable 3:

```
module register32zero
(
output reg[31:0]  q,
input[31:0]       d,
input             wrenable,
input             clk
);
    assign q = 32’b0;
endmodule
```

Deliverable 4:

```
module mux32to1by1
(
output      out,
input[4:0]  address,
input[31:0] inputs
);
    assign out = inputs[address];
endmodule
```

Deliverable 5:

```
module mux32to1by32
(
output[31:0]    out,
input[4:0]      address,
input[31:0]     input0, input1, input2, input3, input4, input5, input6, input7, input8,
input9, input10, input11, input12, input13, input14, input15, input16, input17, input18,
input19, input20, input21, input22, input23, input24, input25, input26, input27, input28,
input29, input30, input31
);

  wire[31:0] mux[31:0];         // Create a 2D array of wires
  assign mux[0] = input0;       // Connect the sources of the array
  assign mux[1] = input1;
  assign mux[2] = input2;
  assign mux[3] = input3;
  assign mux[4] = input4;
  assign mux[5] = input5;
  assign mux[6] = input6;
  assign mux[7] = input7;
  assign mux[8] = input8;
  assign mux[9] = input9;
  assign mux[10] = input10;
  assign mux[11] = input11;
  assign mux[12] = input12;
  assign mux[13] = input13;
  assign mux[14] = input14;
  assign mux[15] = input15;
  assign mux[16] = input16;
  assign mux[17] = input17;
  assign mux[18] = input18;
  assign mux[19] = input19;
  assign mux[20] = input20;
  assign mux[21] = input21;
  assign mux[22] = input22;
  assign mux[23] = input23;
  assign mux[24] = input24;
  assign mux[25] = input25;
  assign mux[26] = input26;
  assign mux[27] = input27;
  assign mux[28] = input28;
  assign mux[29] = input29;
  assign mux[30] = input30;
  assign mux[31] = input31;
  assign out = mux[address];    // Connect the output of the array
endmodule
```

Deliverable 6:

```
module decoder1to32
(
output[31:0]    out,
input           enable,
input[4:0]      address
);
    assign out = enable<<address;
endmodule
```

out is a 32 bit number that is selected by the decoder
enable is a 1 bit number that is the most significant bit in out
address is a 5 bit number with a integer value in the range 0 to 31 (both included)

The value of out is determined by enable and address. The j^th bit in out is set to enable (where j is determined by the value of address) by left-shifting.

Example:
Say enable = 1 and address = 00101
out = 00000000000000000000000000100000