Advanced Digital System Design (EC6L033)

Verilog - Synthesis

Lecture 24: Logic Synthesis with Verilog HDL – Part7

Objectives

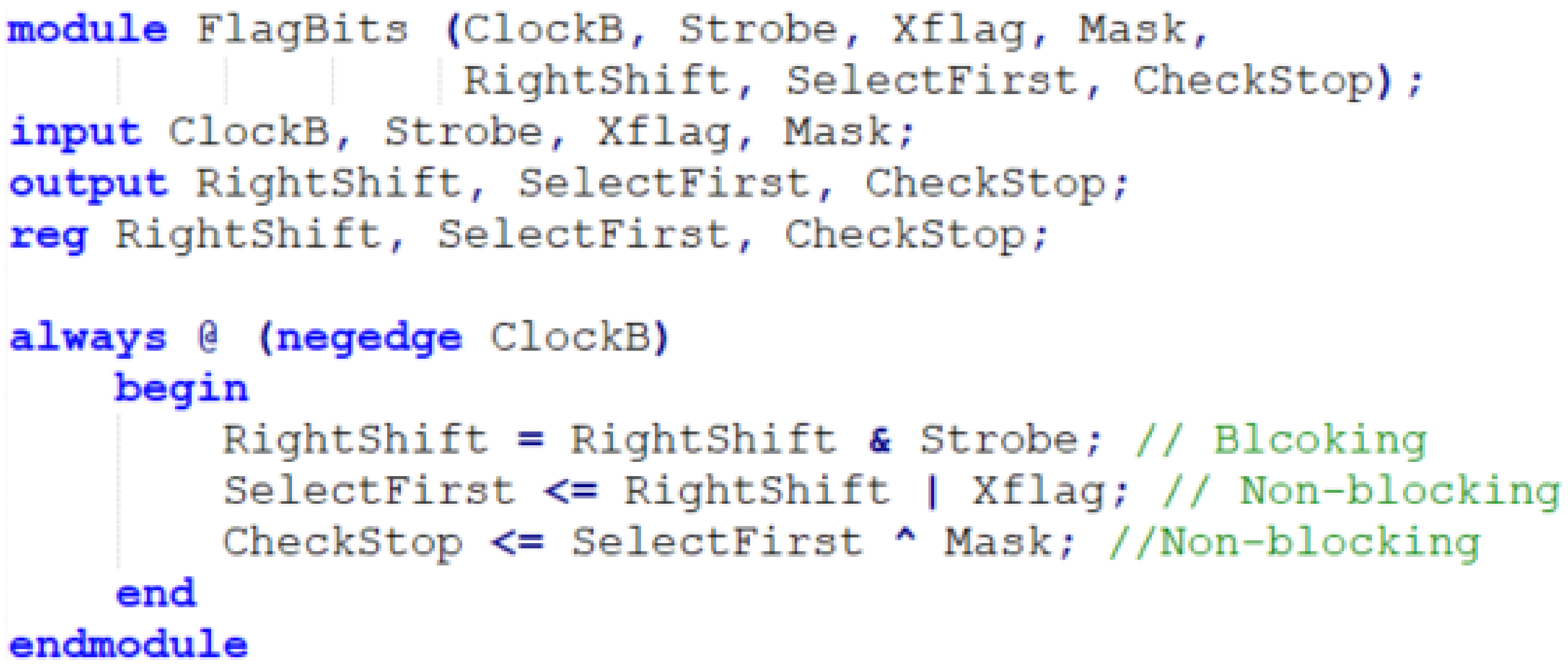
- Blocking Vs Non-blocking

Blocking Vs Non-blocking

- A single Non-blocking or Blocking assignment only inside an always block does not make a difference in the synthesis of the always block.

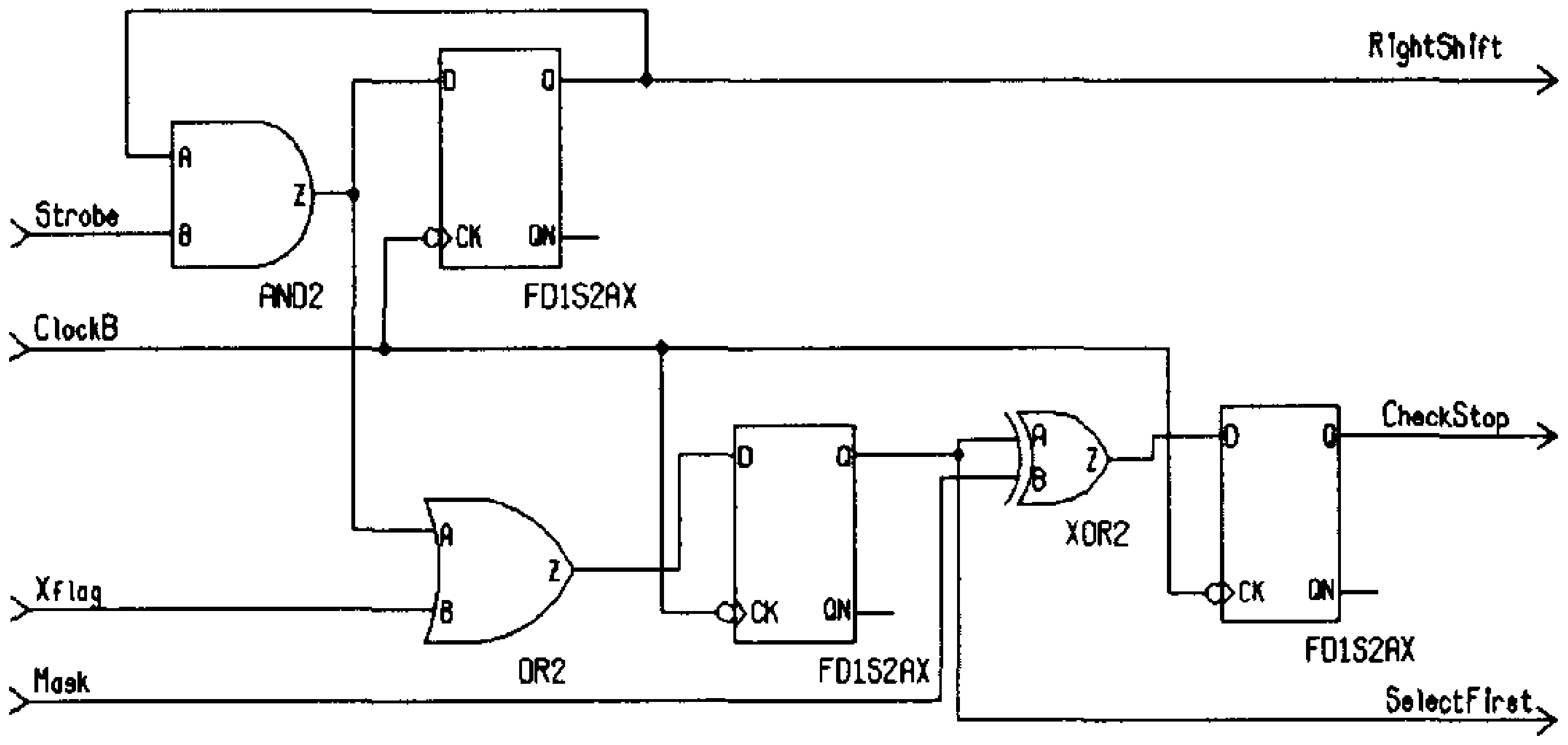
- If we mix both, then, there is a difference. Note the inputs to Flip-flop 2 and Flip-flop 3 (See the example in the next slides).

Blocking Vs Non-blocking



```verilog
module FlagBits (ClockB, Strobe, Xflag, Mask,
                 RightShift, SelectFirst, CheckStop);
input ClockB, Strobe, Xflag, Mask;
output RightShift, SelectFirst, CheckStop;
reg RightShift, SelectFirst, CheckStop;

always @ (negedge ClockB)
    begin
        RightShift = RightShift & Strobe; // Blcoking
        SelectFirst <= RightShift | Xflag; // Non-blocking
        CheckStop <= SelectFirst ^ Mask; //Non-blocking
    end
endmodule
```

Blocking Vs Non-blocking

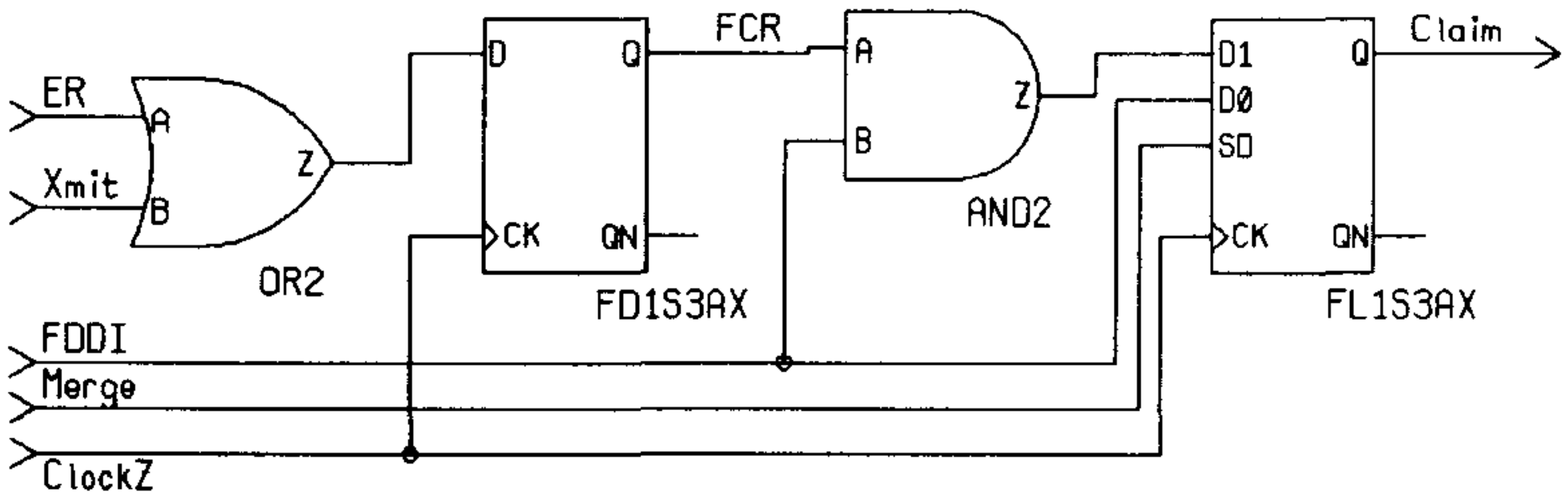


Non-blocking vs blocking procedural assignment.

Blocking Vs Non-blocking

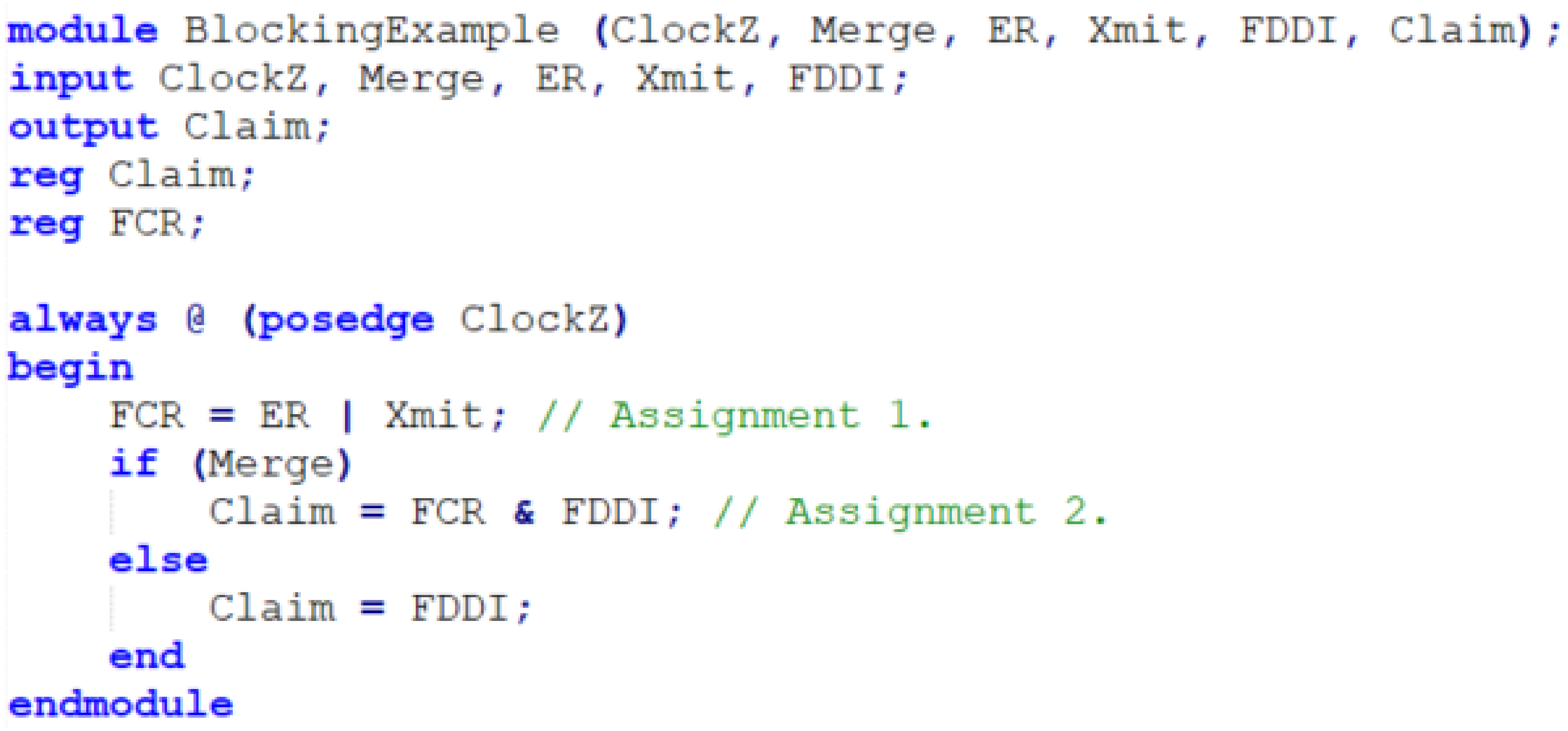
```
module NonBlockingExample (ClockZ, Merge, ER, Xmit,
                                         FDDI, Claim);
input ClockZ, Merge, ER, Xmit, FDDI;
output Claim;
reg Claim;
reg FCR;

always @ (posedge ClockZ)
    begin
        FCR <= ER | Xmit; // Assignment 1.
        if (Merge)
            Claim <= FCR & FDDI;
        else
            Claim <= FDDI; // Assignment 2.
end
endmodule
```

Blocking Vs Non-blocking



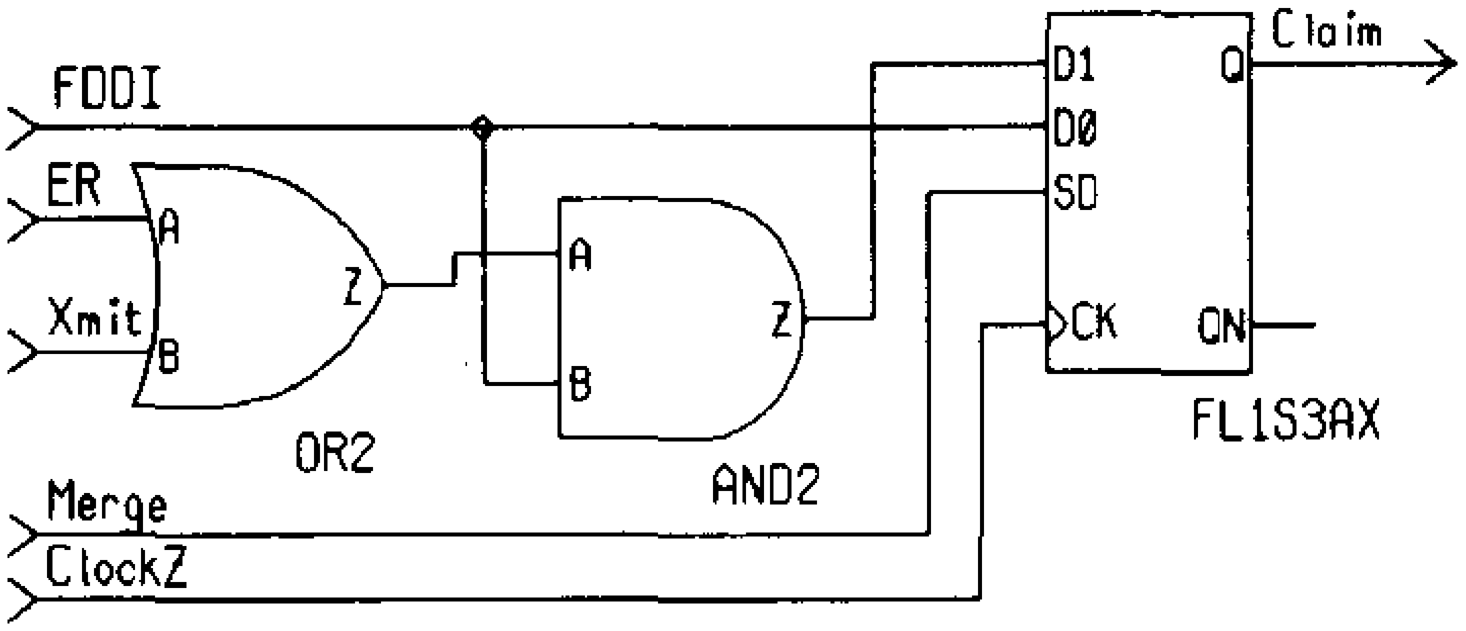
Non-blocking assignments.

The Blocking Counterpart



```
module BlockingExample (ClockZ, Merge, ER, Xmit, FDDI, Claim);
input ClockZ, Merge, ER, Xmit, FDDI;
output Claim;
reg Claim;
reg FCR;

always @ (posedge ClockZ)
begin
    FCR = ER | Xmit; // Assignment 1.
    if (Merge)
        Claim = FCR & FDDI; // Assignment 2.
    else
        Claim = FDDI;
    end
endmodule
```

The Blocking Counterpart



Blocking assignments.

Thank you