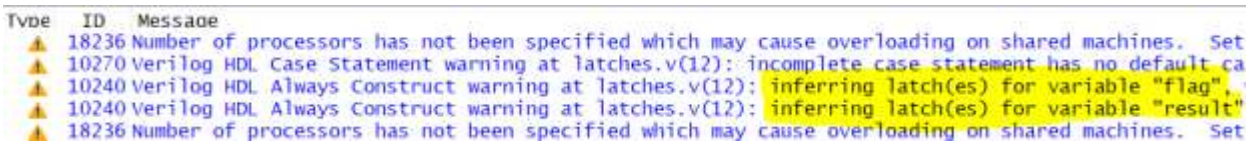
Home Work 11. LATCH ELIMINATION

Latches are prone to glitches and we try not to use them in digital circuits.

Latches are basic storage elements. The fundamental difference between flip flops and latches is that the flip flops represents synchronous storage elements controlled by clock and sensitive to clock transition, while latches are asynchronous storage elements and sensitive to levels, not transitions.

We usually try to write modules without latches. However, sometimes we write modules that cannot be implemented without latches and synthesis tool adds latches on its own. Such laches is called inferred latches.



INFERRED LATCHES ARE UNDESIRABLE!!!!!

We have seen couple of modules for ALU, BCE, etc. with INFERRED LATCHES. Such modules ARE NOT WELCOME by the processor.

Task for Assignment 10 is to review your modules for *Immediate Extension Unit, Arithmetic Logic Unit (ALU), Branch Condition Evaluation unit (BCE), Shifter, I Decoder (Lab 10), MEMORY*, and eliminate inferred latches.

HINTS

Common causes of inferred latches are the following:

1) Case statements without default case. We would have latch for flag and result.

```
always @(*) begin
case(condition)
4'd0:    begin
            flag = 1'd0;
            result = 4'd0;
         end
4'd1:    begin
            flag = 1'd1;
            result = 4'd10;
         end
endcase
end
```

2) IF-ELSE statements without else part. We would have latch for flag and result.

```
always @(*) begin
if(condition == 4'd0) begin
    flag = 1'd0;
    result = 4'd0;
end
else if(condition == 4'd1) begin
    flag = 1'd1;
    result = 4'd1;
end
```

3) Not Changing value of a signal in all cases or else if parts. We would have latch for flag because it is not updated in case 4'd1.

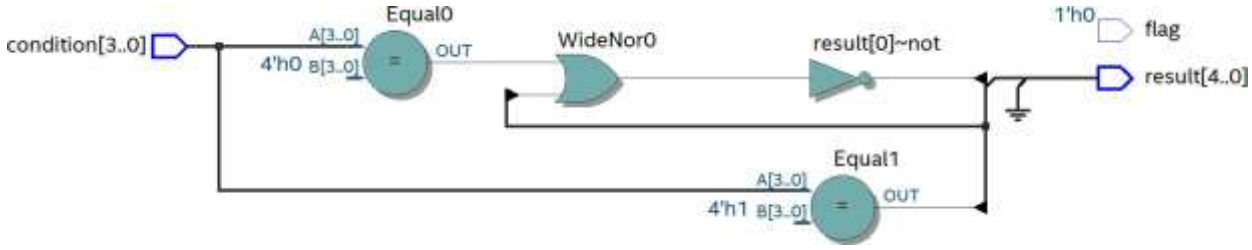
```
always @(*) begin
case(condition)
4'd0:    begin
            flag = 1'd0;
            result = 4'd0;
         end
4'd1:    begin
            result = 4'd10;
         end
default: begin
            flag = 1'd1;
            result = 4'd1;
         end
endcase
end
```

We would have latch for result because it is not updated in the first if part.

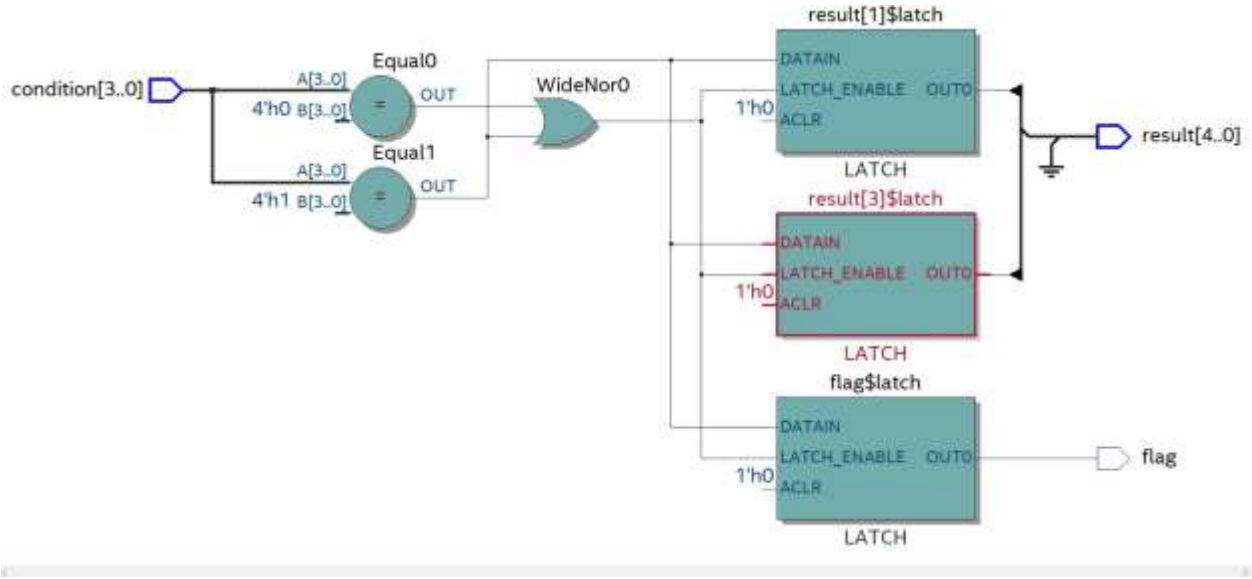
```
always @(*) begin
if(condition == 4'd0) begin
    flag = 1'd0;
end
else if(condition == 4'd1) begin
    flag = 1'd1;
    result = 4'd1;
end
else begin
    flag = 1'd1;
    result = 4'd1;
end
```

You MUST see Quartus warnings to check if you have inferred latches.

The same circuits with and without latches:



Circuit without Latches:



Circuit with Latches: