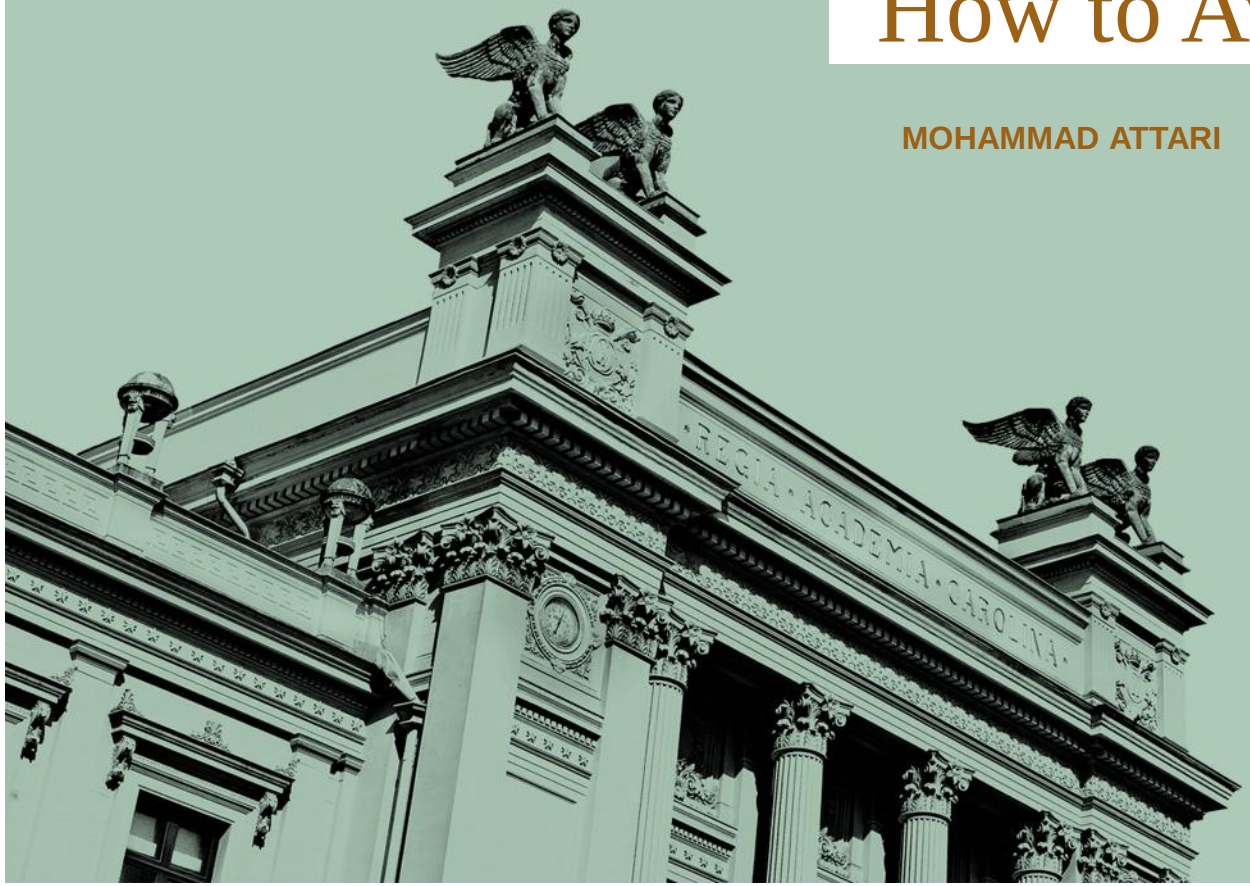
Introduction to Structured VLSI Design

How to Avoid Latches

MOHAMMAD ATTARI



What is a latch?

- It is a kind of memory
  - An asynchronous storage element
  - Not controlled by the clock
- Why is it bad?
  - Timing issues
  - Race conditions
  - Combinatorial feedback

Example – 1-Bit Comparator

- Let's design a simple 1-bit comparator
- The following code looks innocent enough!

```
entity comp is
    port(A, B : in std_logic;
         AeqB : out std_logic);
end comp:

architecture behavior of comp is
begin
    process (A, B)
    begin
        if A = B then
            AeqB <= '1';
        end if;
    end process;
end behavior;
```

Example – 1-Bit Comparator (Latch)

- Let's design a simple 1-bit comparator
- The following code looks innocent enough!

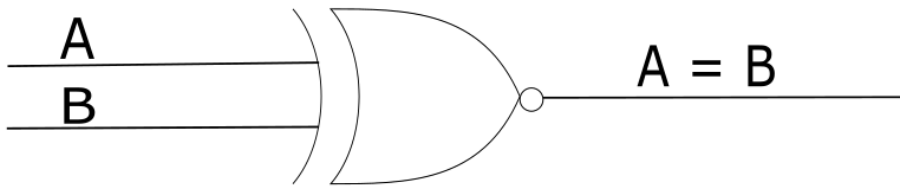
```
entity comp is
    port(A, B : in std_logic;
         AeqB : out std_logic);
end comp:

architecture behavior of comp is
begin
    process (A, B)
    begin
        if A = B then
            AeqB <= '1';
        end if;
    end process;
end behavior;
```

- But is it?

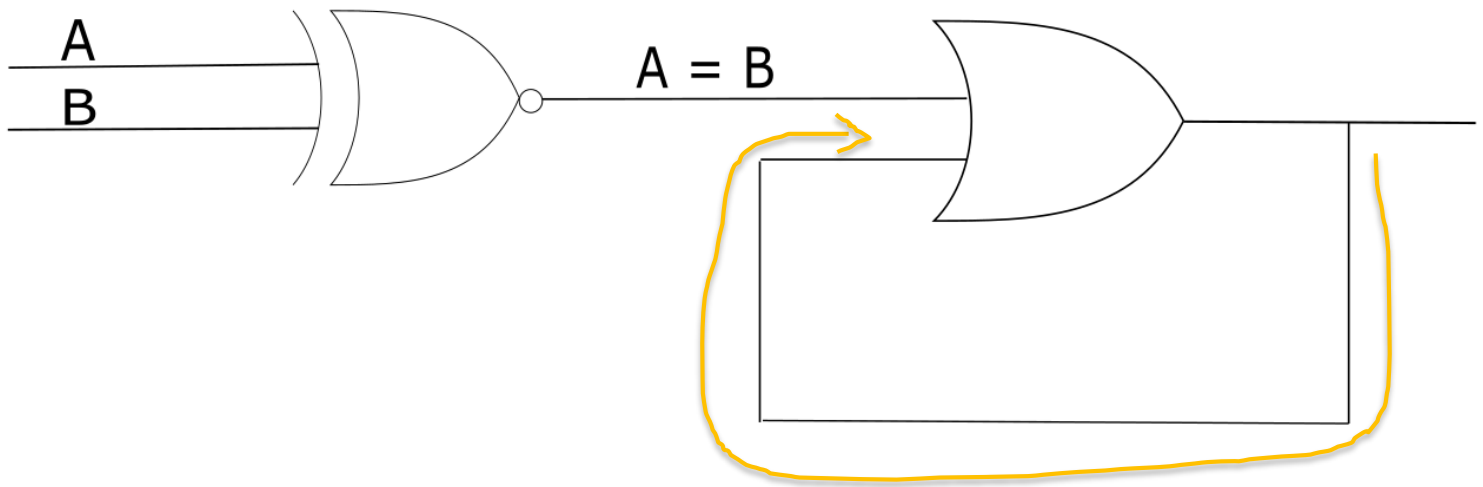
Example – 1-Bit Comparator

- This is what we wanted:
  - A simple combinational logic comparator



Example – 1-Bit Comparator (Latch)

- This is what we get:
  - A nasty feedback
  - Once the inputs are equal, the output will remain high forever!
  - Can you spot the memory?



Example – 1-Bit Comparator (Latch)

- But why?
  - We forgot to complete the if statement!
  - It implies **AeqB** must keep its old value if **A != B**

```
entity comp is
    port(A, B : in std_logic;
         AeqB : out std_logic);
end comp:

architecture behavior of comp is
begin
    process (A, B)
    begin
        if A = B then
            AeqB <= '1';
        end if;
    end process;
end behavior;
```

Where is my else?

Example – 1-Bit Comparator (Latch Free)

- How to rectify this?
  - Avoid incomplete conditionals in combinational logic (if, switch, …)
- This is the correct design:

```
entity comp is
    port(A, B : in std_logic;
         AeqB : out std_logic);
end comp:

architecture behavior of comp is
begin
    process (A, B)
    begin
        if A = B then
            AeqB <= '1';
        else
            AeqB <= '0';
        end if;
    end process;
end behavior;
```

Here is my else clause!

- Yes!