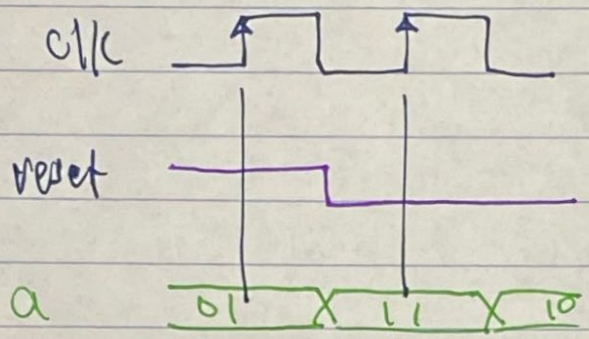
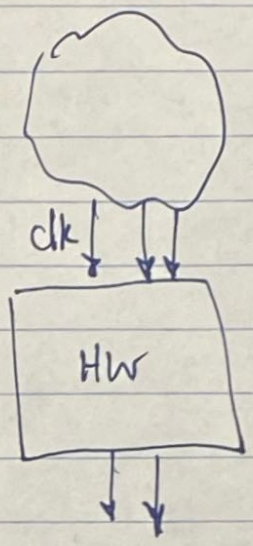
EEC 281

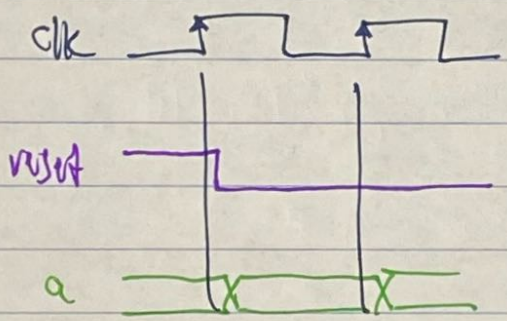
Jan 25, 2024

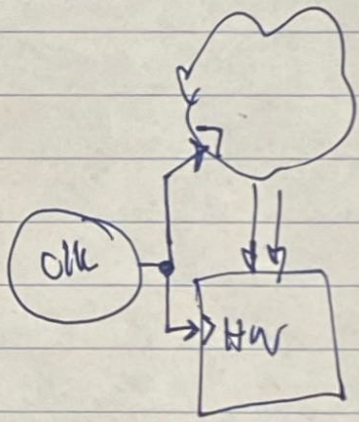
Test bench approach #1



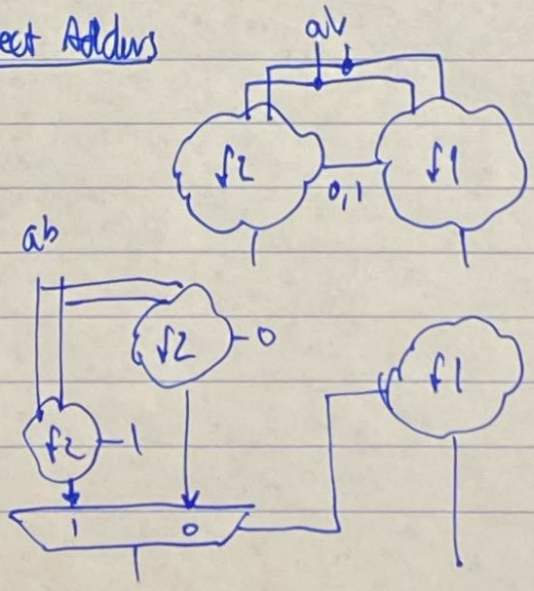


Approach #2

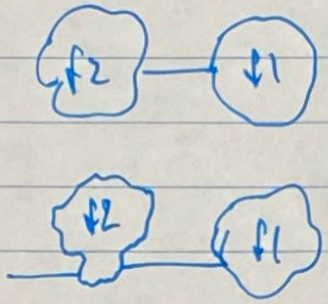




Carry-Select Adders



Carry- look ahead



Generate

$$\begin{array}{r} C_i \\ 1XXXXX \\ +\ 1XXXXX \\ \hline C_o = (1) \end{array}$$

regardless of C_{in} ⇒ Generate

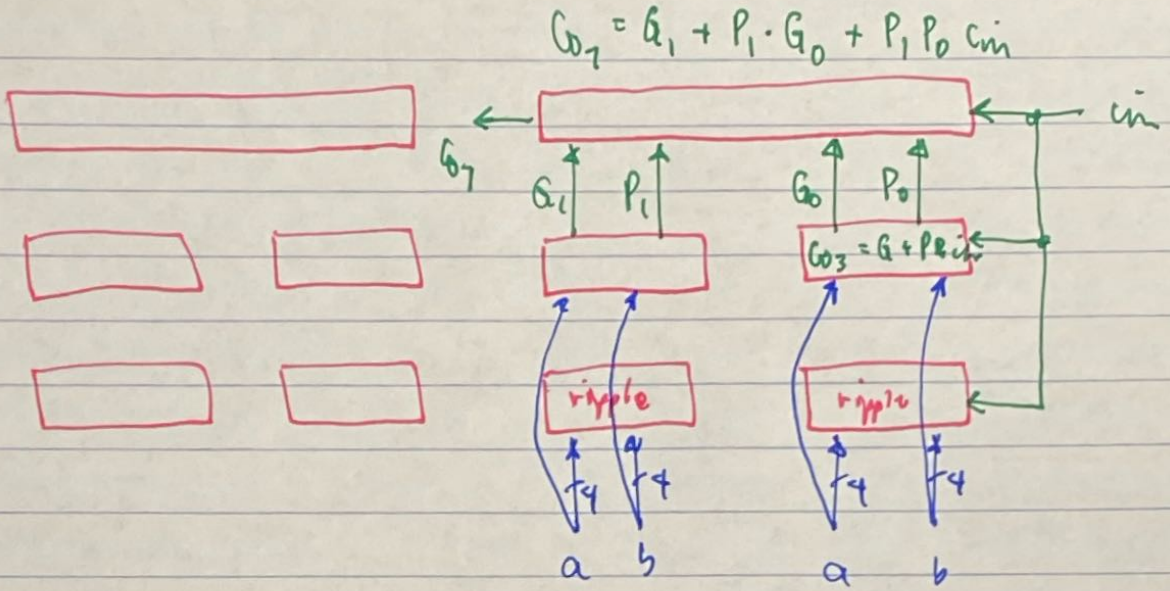
Propagate

$$\begin{array}{r} C_{in} \\ 111111 \\ +\ 000000 \\ \hline C_o\ ?\,?\,?\,?\,?\,? \end{array}$$

$C_o = 0$ if $C_{in} = 0$

$C_o = 1$ if $C_{in} = 1$ ⇒ Propagate

C_o = Gen OR (Prop & C_{in})



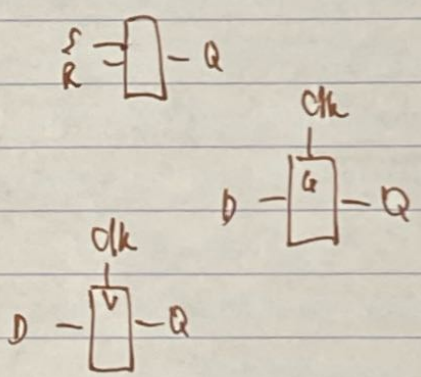
Memories

I. Single-Bit

a) Clock-less latches

b) Transparent level-sensitive latch

* c) Edge-Triggered FF



II. Array

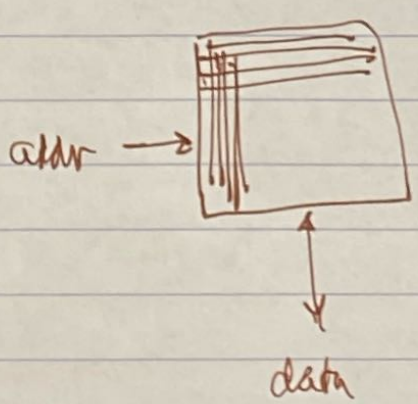
- address
- rd/wr command

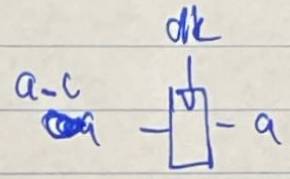
* 1) SRAM

2) DRAM

3) Flash

4) Multi-ported SRAM

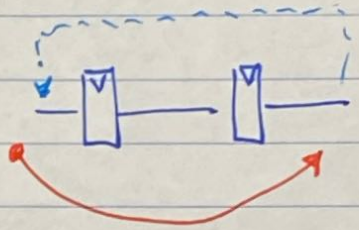





```
reg a;
always @ (posedge clk) begin
    a = a-c;
end
```

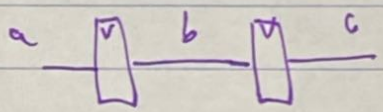
don't ✗

blocking assignment



Solve with a non-blocking assignment

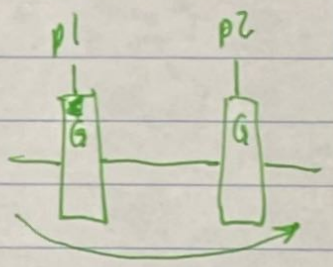
```
always @ (posedge clk) begin
    b <= a;
    c <= b;
end
```



Rule #1

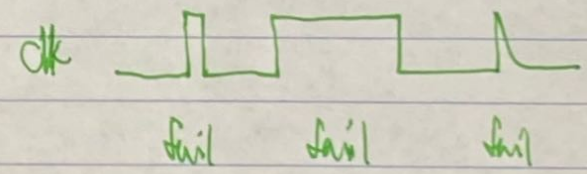
Use only FF

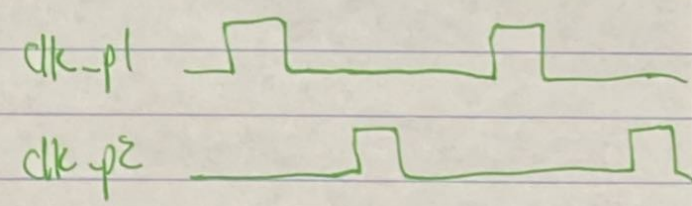
Transplatch



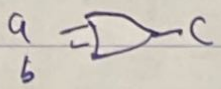
Rule #2

For comb. logic instantiated in reg's, always use blocking assignments






```
reg c;
always @ (a or b) begin
    c = a | b;
end
```



Rule #3

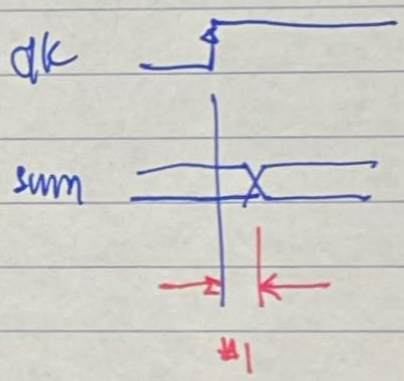
For FF's, always use non-blocking assignments

```
always @ (posedge clk) begin
    sum <= #1 sum_c;
    ...
end
```

Rule #4

Add a #1 clock-to-Q delay to increase readability

DC does give a warning
- ignore it



Rule #5

Normally do not include logic in FF declarations

```
always @ (posedge clk) begin
    ...
    ~~logic~~
    reset ok
end
```

Suggestion #6

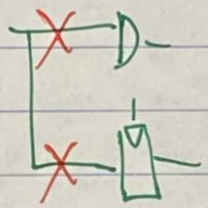
// Block 1
logic
FFs

// Block 2
logic
FFs
⋮

Rule #7 - Preview

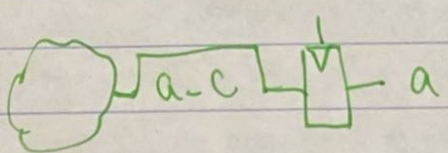
a) clock only

b) clock

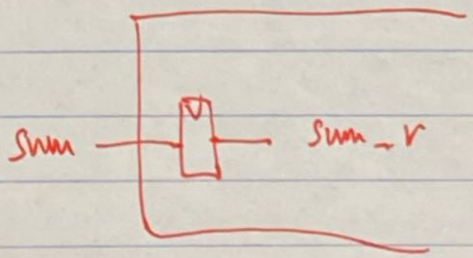


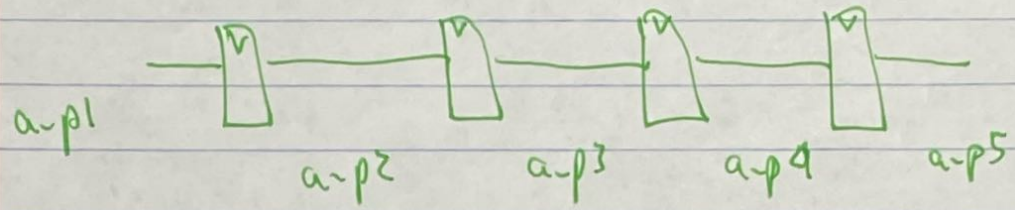
c) There are exceptions

Rule #8



comb. logic version





Clocked Timing Relationships