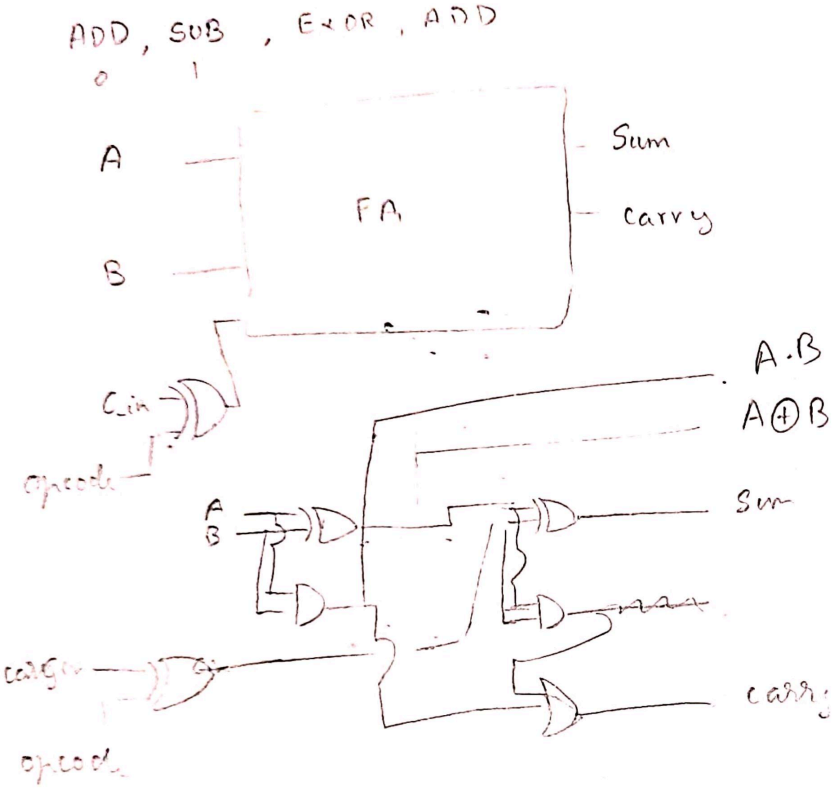
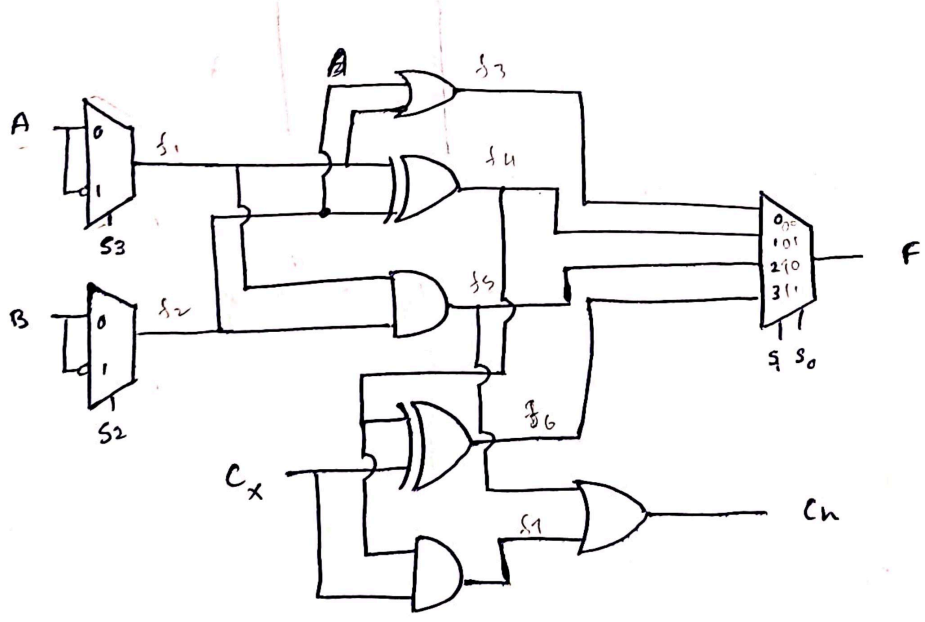
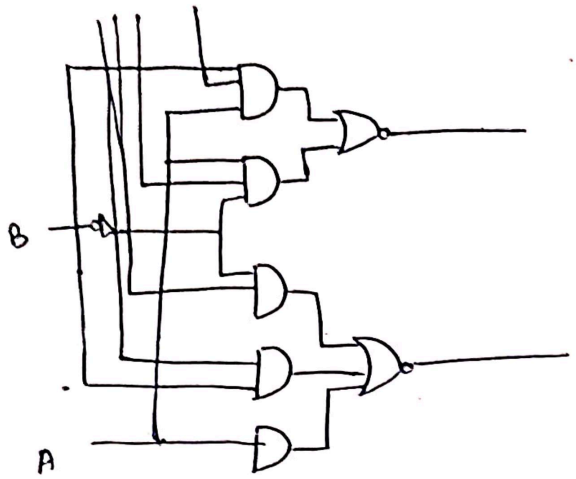
	Text editor	Tools	
	micro	Vivado	IEEE 1800:
open-xc7	VScode	Quartus	VHDL 93
	Sublime text	Yosys	

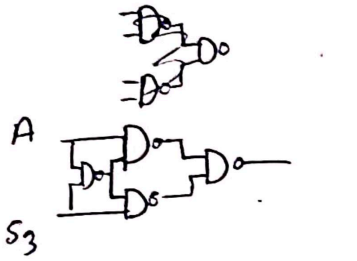
TH-181 ⇒ bit slicable ALU

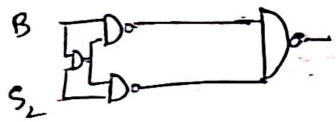


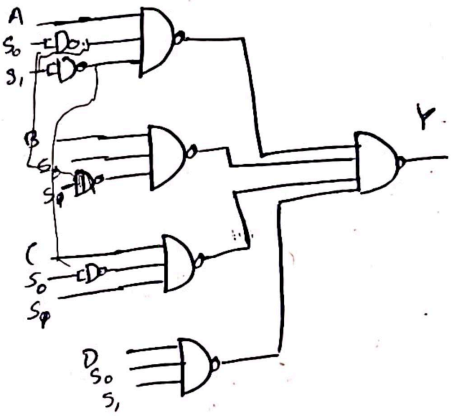




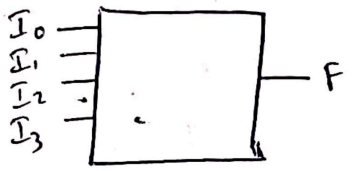
MOXIO
ICE40 → LE
Artix → LC







LE → Logic Element



XC7 → Artix 7

35t → 35k cells
- extension cell within 32k LUT

cgp → package type

238 SMD - 238 pin.

-1 → speed programmes.

→ 100 MHz

Carry look ahead adder.

$\frac{n}{2}(n+1)$ AND, n OR gate

ALU - 1 bit

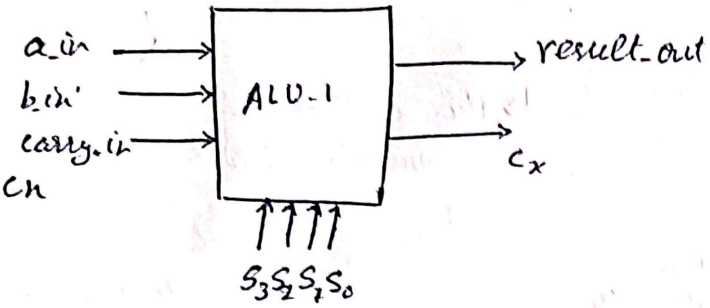
a_in - [+ −]

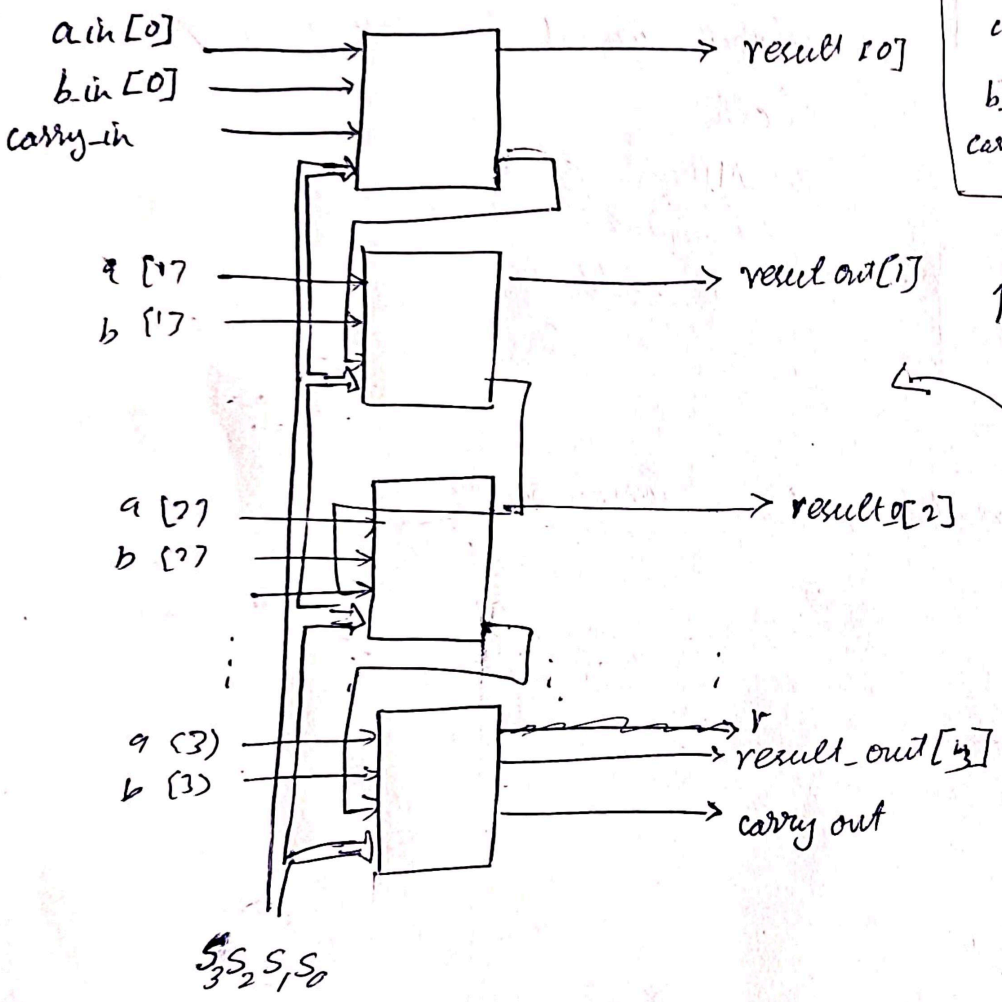
b_in - 1 & 1

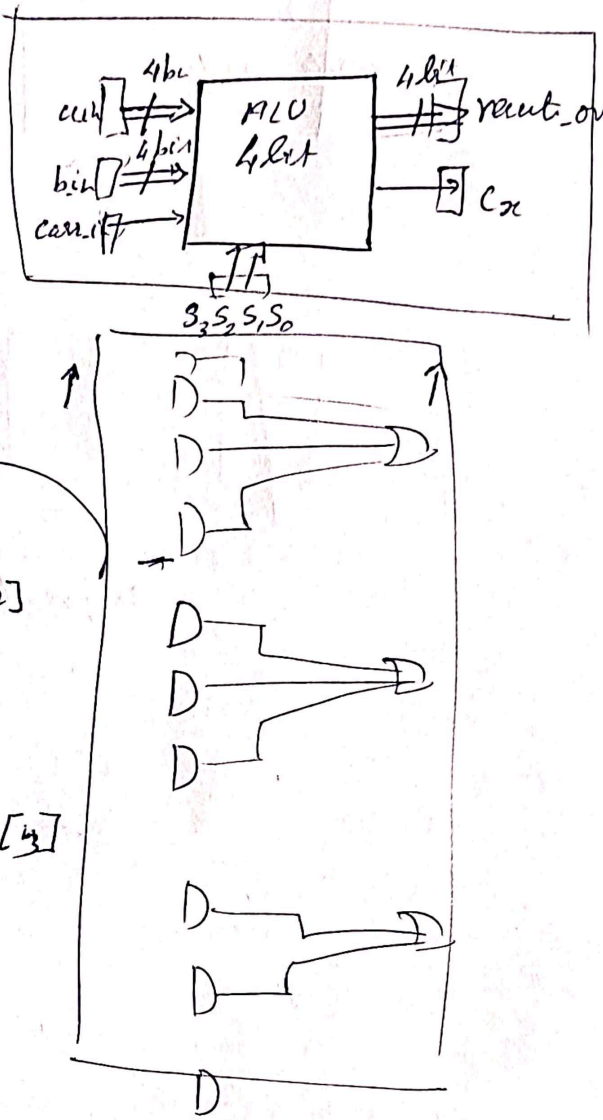
carry_in

result_out

carry_out







Advantage

→ less time

Disadvantage

→ increase complexity

→ adds more gates to design

→ n - bits

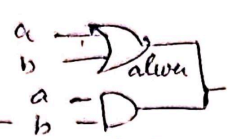
$\frac{n(n+1)}{2}$ AND n - OR

Run Linter

↳ avoid multidriver.

↳ no-connected nets

↳ latch infer



⇒ ✓ Synthesis

```
reg d_reg;
wire c;
```

Timing constrain file

⇒ Add source → clk.xdc

↳ write constrain file.

↳ generate.

↳ Clock wizard.

→ ☑ clk

→ 200 MHz

→ select signals

→ intial condition.

... ... Finish.

↳ Run synthesis.

Constrain wizard

↳ before finish

☑ view timing constrain

☑ Create timing summary report

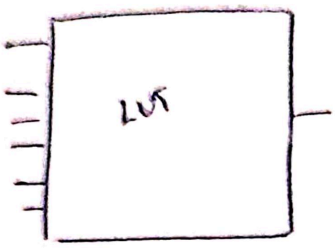
☐

☐

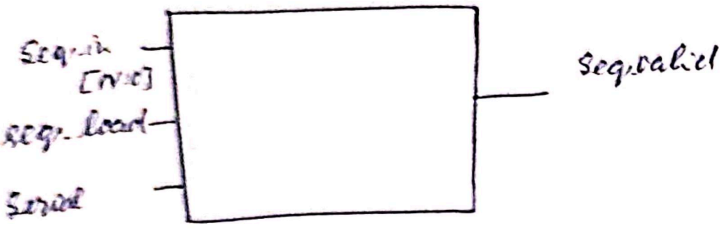
⇒ Generate report ⇒ Timing report

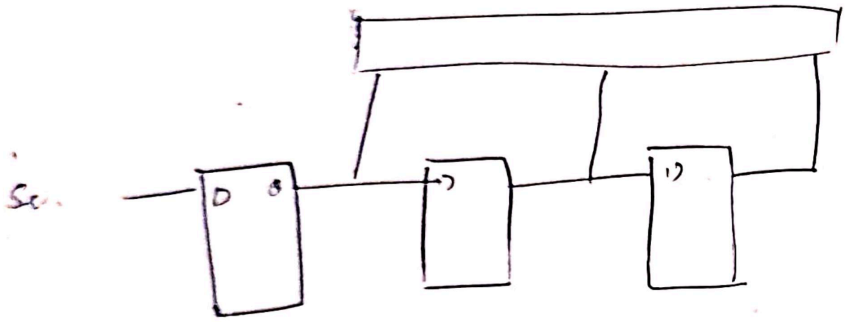
use of case: ⇒ optimise for ALU

LFFR - Linear feedback feedforward register.



Q: 8b





5A5A5A

6×4

0101 1010 ×3

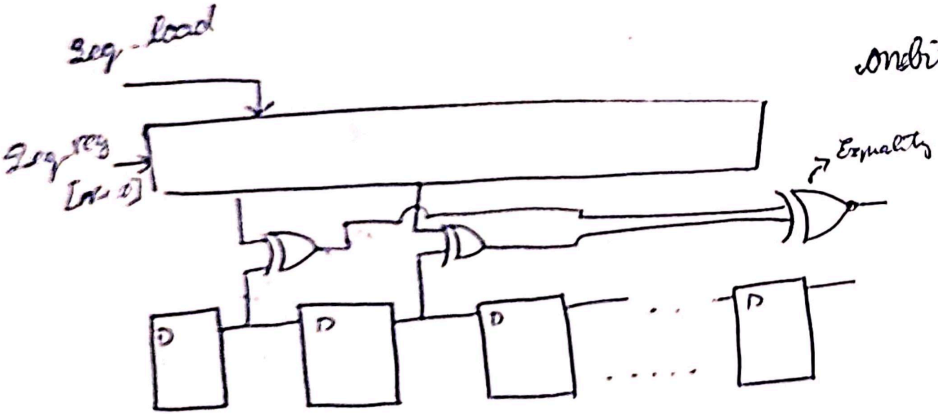
1-
0-

~~x d2~~

x d2 d2 d → 1
0

↓

→



onebit right shift.

~~[DW]~~

[DW-1] → 1

↓ glitch → cross coupled NAND inverter.

$I_b = 0$

BJT ⇒ oscillate

$I_g = 0$

MOSFET ⇒ stick
to a stat

9.19.1

(a) transitional state $\begin{cases} 0 \to 1 \\ 1 \to 0 \end{cases}$ => floating — NO internal PULLUP / PULL-DOWN } resistor in FPGA gates

MOSFET => can be used to implement

but, generates impulse response. ↦

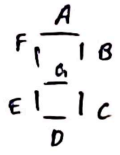
VHDL => 9 state variable.

Verilog => 4 state variable => That cannot account for floating state in a BJT

↳ commercial design the use MOSFET so does not have to account for.

Metastability avoided by registring o/p

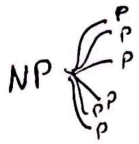
BCD => 7 segment.

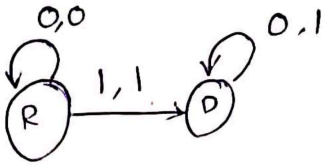


	B_3	B_2	B_1	B_0	A	B	C	D	E	F	G
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	1	1	0	0

Automata Theory -

- Non-probabilistic problem
- Probabilistic problem
  - → finite steps.





101 RESET → 1 DET_1

↳ | 0 RESET

DET_1 → 0 DET_10

| 1 DET_1

DET_10 → 1 DONE

| 0 RESET

- - - - - - - - - - - -

DONE → 1 DET_1

| 0 DET_10

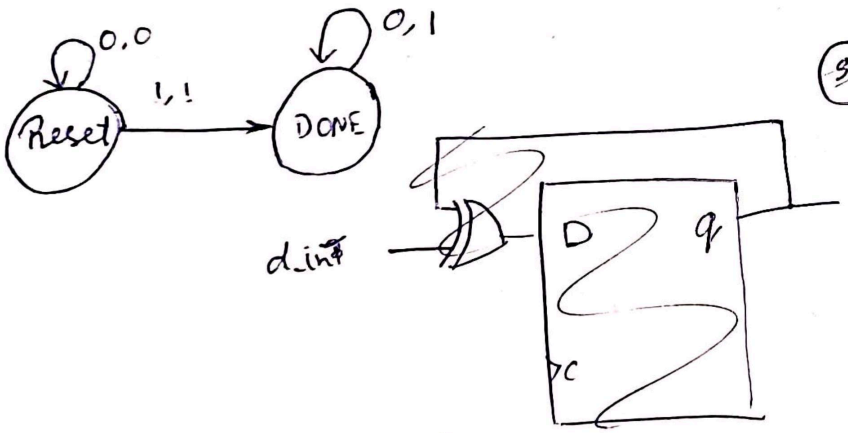
din: 1 0 1 1 0 1 1 0 1 1 1 . . .

q: 1 1 0 1 1 0 1

0 0 0 1 1

0 0 0 1 0

1 0



d - 0 1 1 0

S → q - 0 1 0 →

00	1
01	1
10	1
11	0

case(d_in)

0 : q_out = ~q_out

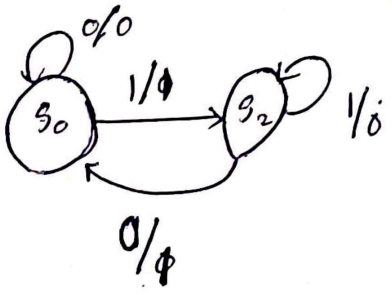
1 : q_out = q_out

if (din)

q0 = ~q_0

else

q0 = q_0



RESET → 1 DONE 1

| 0 RESET 0

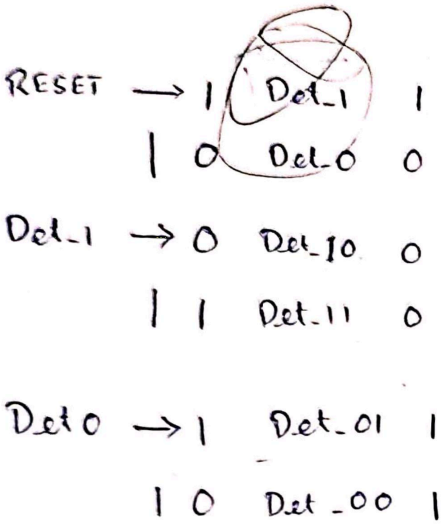
DONE → 1 DONE 0

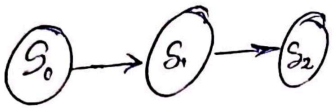
| 0 RESET 1

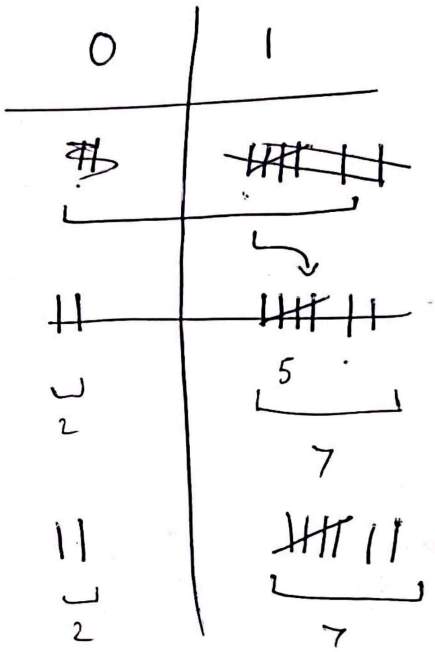
odd 1's & even 0's

1 ↓ 01

100
010
001







RESET → 1 DONE , 1
1 0 E1_00 , 0

Ee1_Qd0 → 1 01_00 , 0
1 0 RESET , 0

O1_00 → 1 En1_Odd0 , 0
1 0 DONE , 1

DONE → 1 RESET , 0
1 0 Odd1_Odd0 , 0

→ Odd & Even.

⇒ detect_odd_even.SV
- seq_in - i/p
- clk
- reset_n
- Y_out - o/p

TB ✓

→ Rom to store & release seq.

⇒ input_rom.SV
- [15:0] seq_in - i/p → from switches in Basys3
- load_seq
- release_rom
- clk
- rest
- seq_out - o/p

→ counter & display.

⇒ ~~bcd~~ counter_display.SV
- seq_in → i/p
- clk
- rest
- [15:0] disp_seq → o/p.

06-11-2024

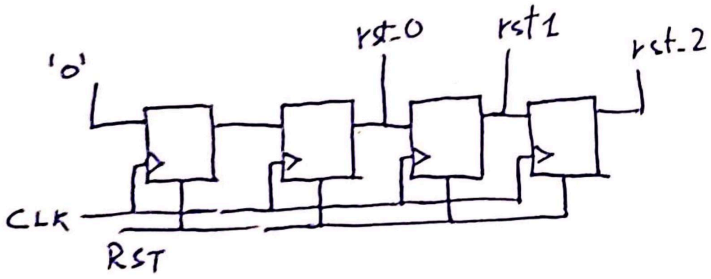
- Design Patterns in System Verilog HDL

- Clear SRAM

- reset

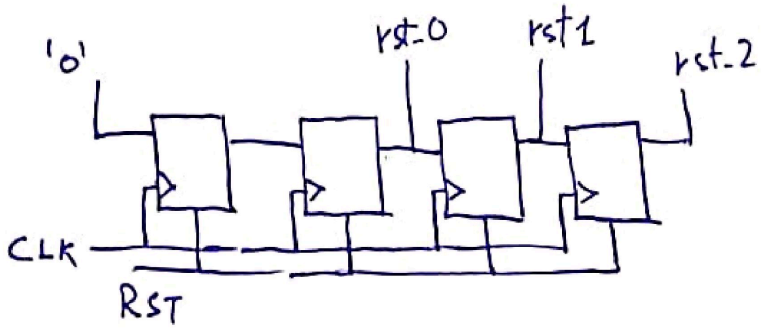
Note 7
↳ done by a RESET_synchronous of time period 100ms

↓
Test Bench reset



- clear SRAM

- reset



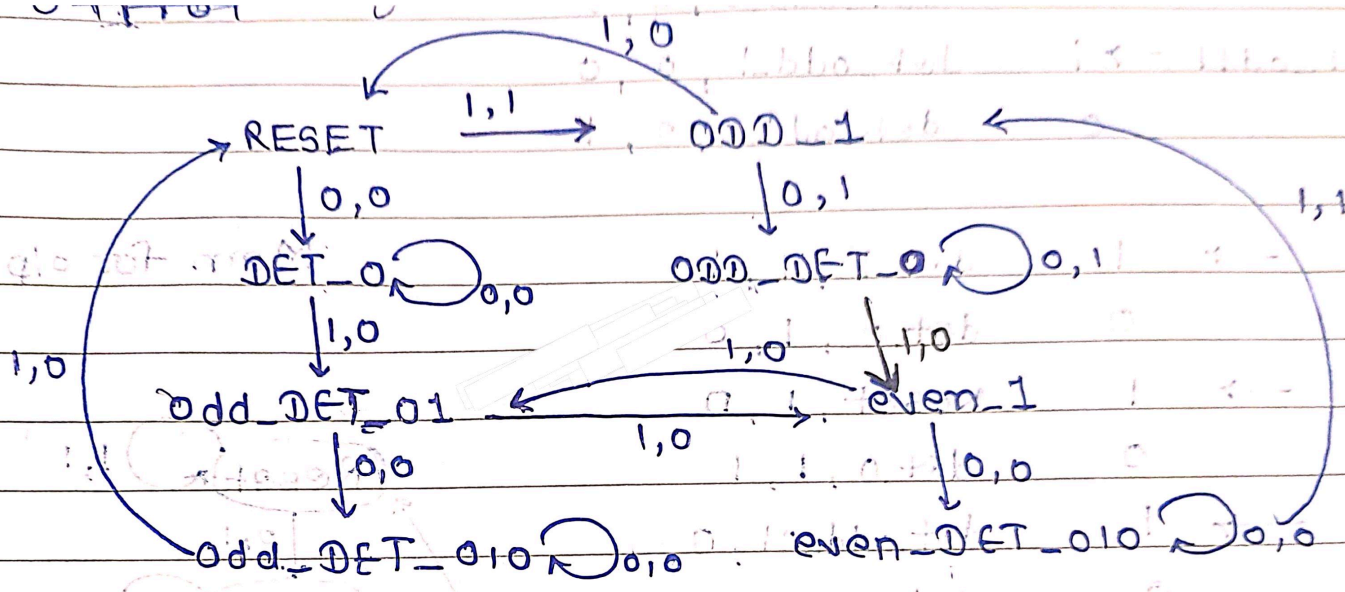
Xilinx 7

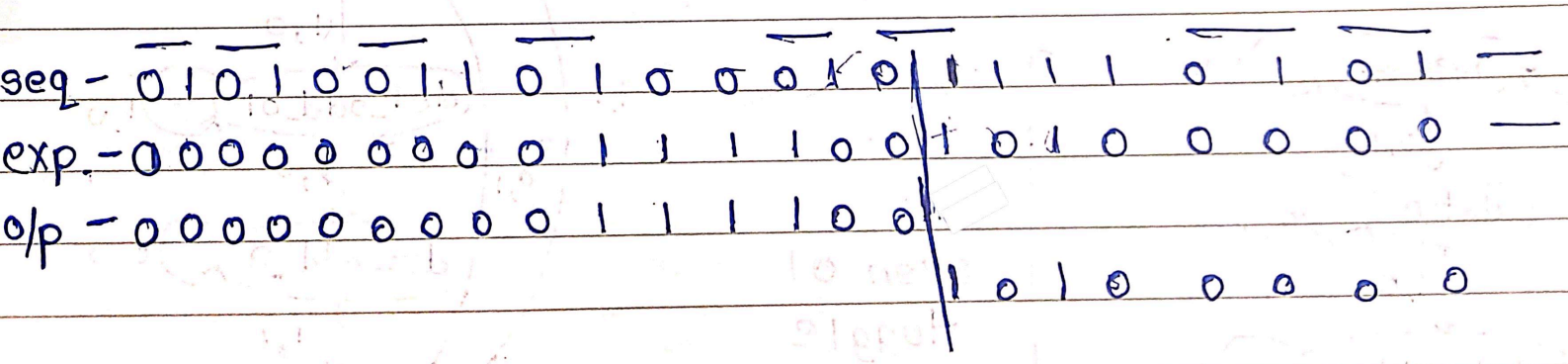
↳ done by a RESET_synchronous of time period 100ms

↓

Test Bench reset

Sequence detector odd 1's and even 01 toggles





Mute → Toggle

Volume UP → press & hold

16ms

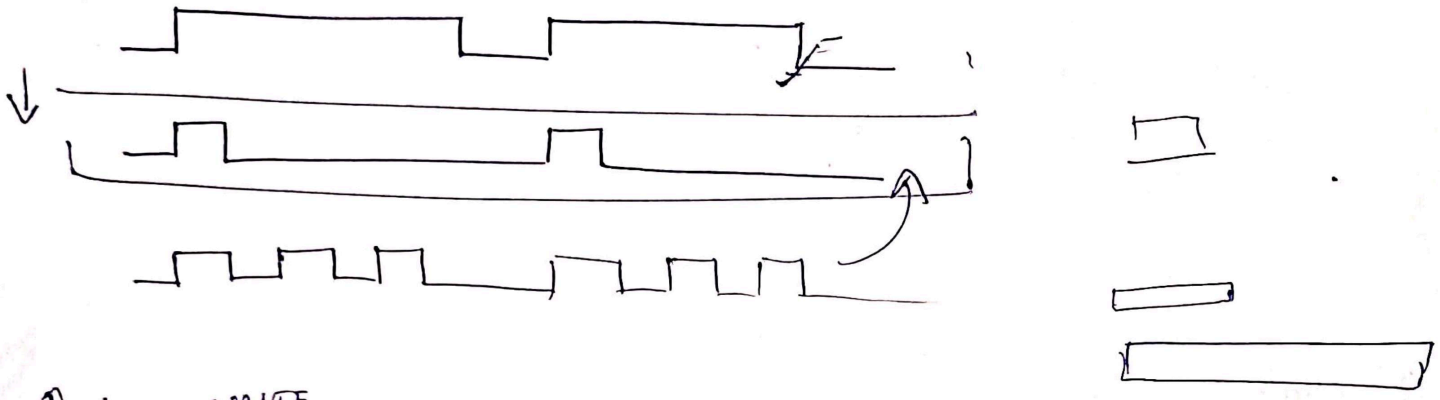
20ms

$\frac{1}{20} \times 10^{-3}$

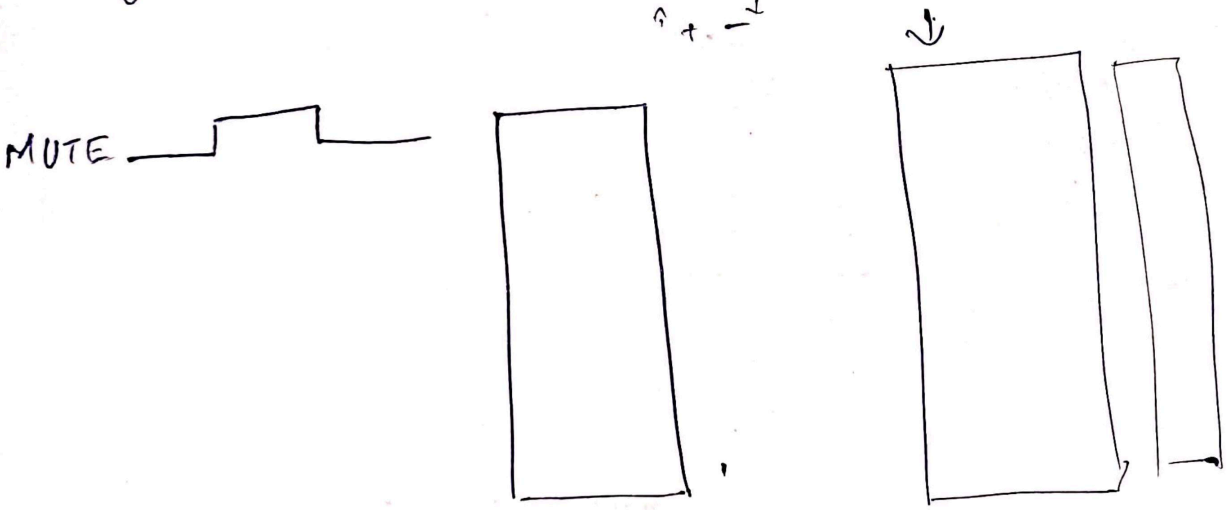
50Hz

62.5kHz

Phy



↑ ↓ ×MUTE



PS/2 → OS/2
PS/2 → DB/2

accept i/p from synchronous system & display over 7-segment displ.

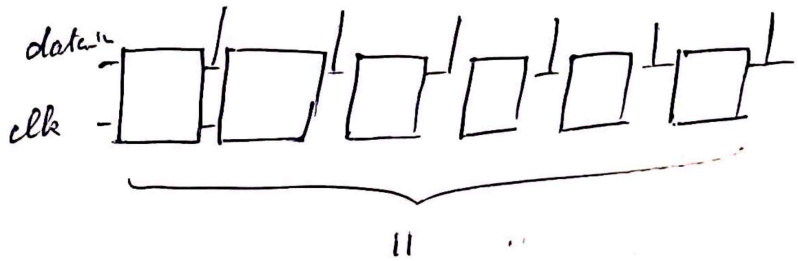
→ Vcc gnd

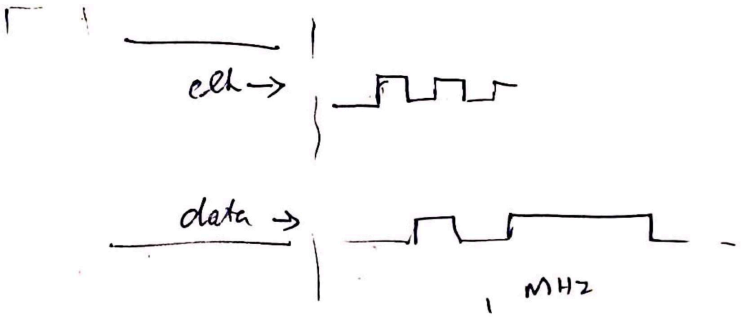
5 - clk 1- data

Data - width → 11 bits.

↓

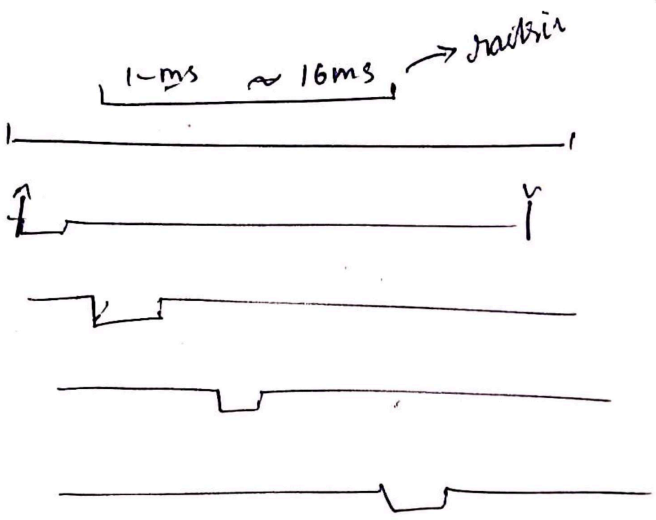
shift - register & latch

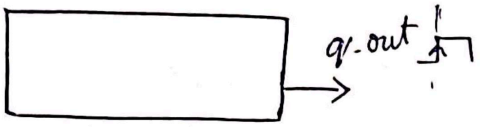


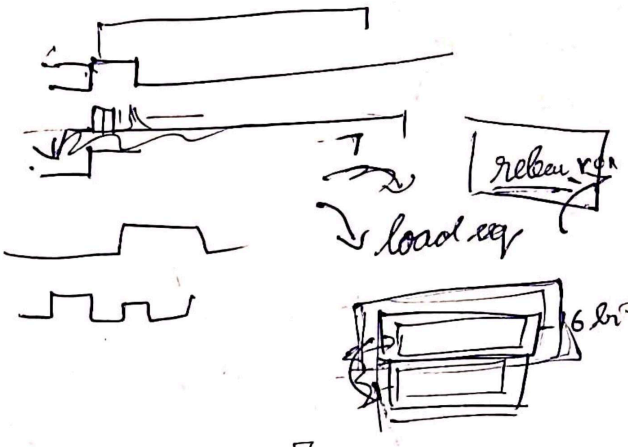


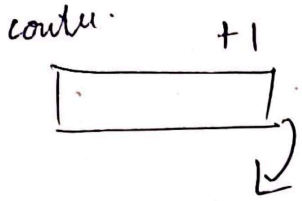
1kHz → 4kHz

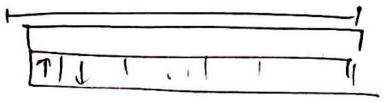
refreshing frequen-







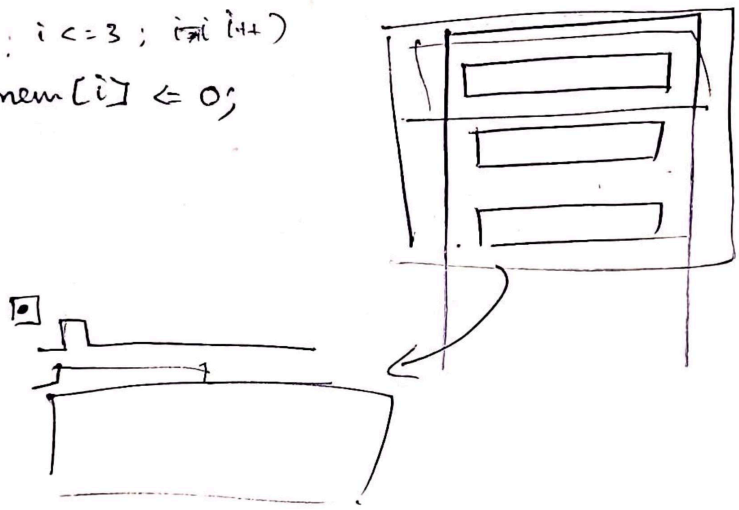




→ store sequence in ROM

→ on run, flush the sequence & detect sequence.

```
for (i=0 ; i<=3 ; i=i+1)
  rom mem[i] <= 0;
```



[15:0] ~~mem~~ ROM [3:0]

[3:0]

[0:3]

ptr=0

load seq

ilp ptr ≤64

ROM [ptr] <=

ptr = +1

+branch

Rajan Chitrapa

ARM → Bus ← 1.
Memory ← 2.
UVM ← 3

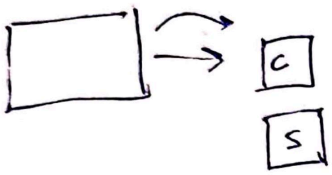
→ Master , Slave , ...

AXI, AXB, ...

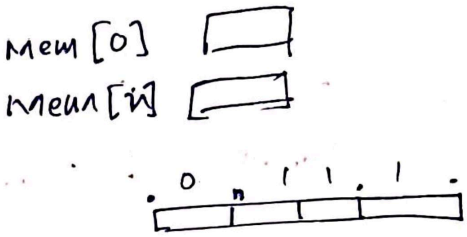
Out of order

~~Store~~ Store loading.

UVM based coverage → func, fsm




```
if (sig[i]) begin
    mem[i] <= in-data;
end
```



Read Master

IDLE —ARREADY→ 1 READ_DATA

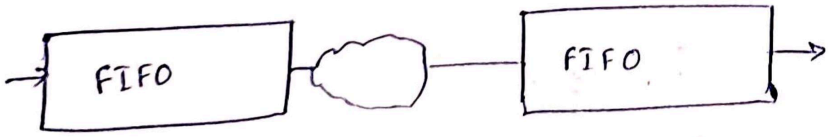
1 0 IDLE

READ_DATA —RRREADY→ 1

1 0

13-11-2024

Single clk pipelining



✓ works without metastability

✓ wider frequency range

2

AMBA - AXI-4 Lite

READY, VALID ⇒ Common for different modes.

↓

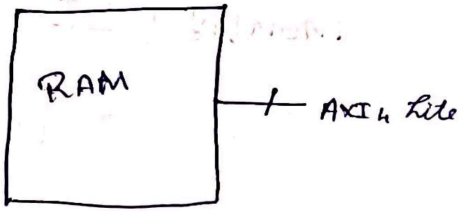
High → channel should have valid data

Handshake

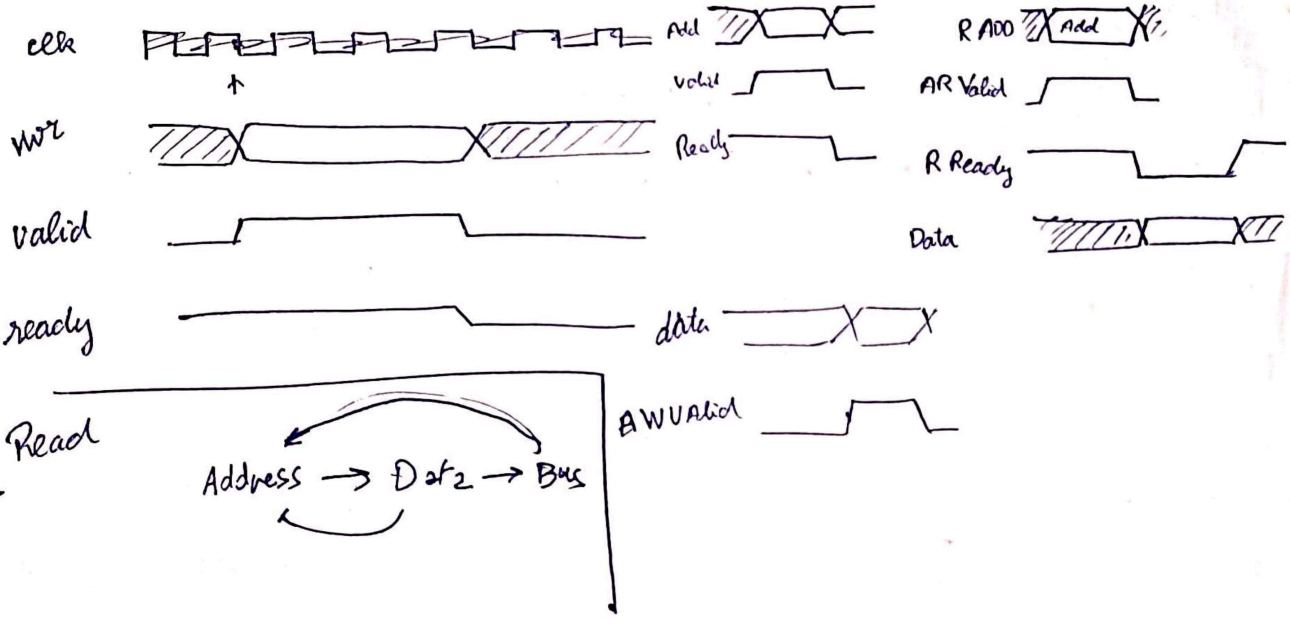
i) R, V ⇒ handshake completed

ii) R, V

- Variable width verification RAM, with AXI 4 subordinate

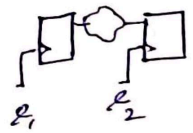


~~address~~ Channel in AXI4



Address Handshake → Data Handshake

$t_1 - t_2 \leq t_{hold_2}$ → Implementation of overclocking. AMD alternatively



Valid multiprocessor clk system ?

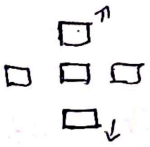
used to scale up scalar processor.

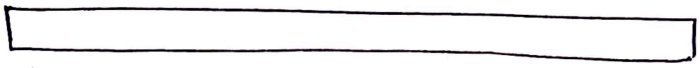
MOS6502

→ single clk pipeline

$\Rightarrow \frac{t_{p_2}}{2} - \frac{t_{p_1}}{2} \geq t_{setup1}$

↳ higher throughput at same clock frequency





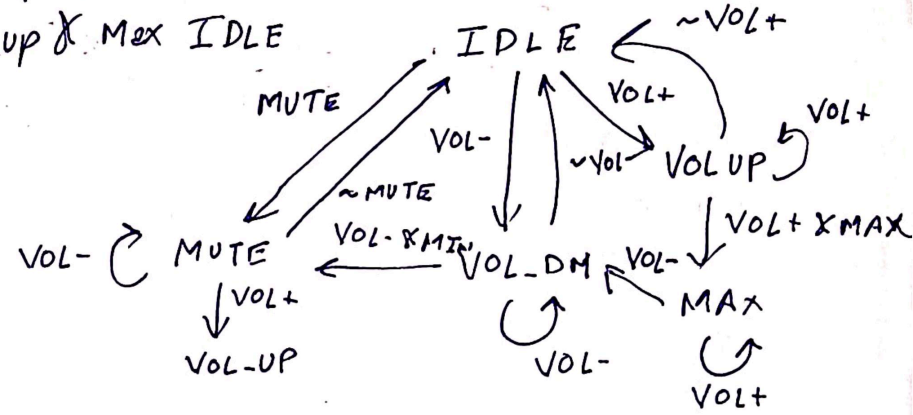
~~IDLE~~

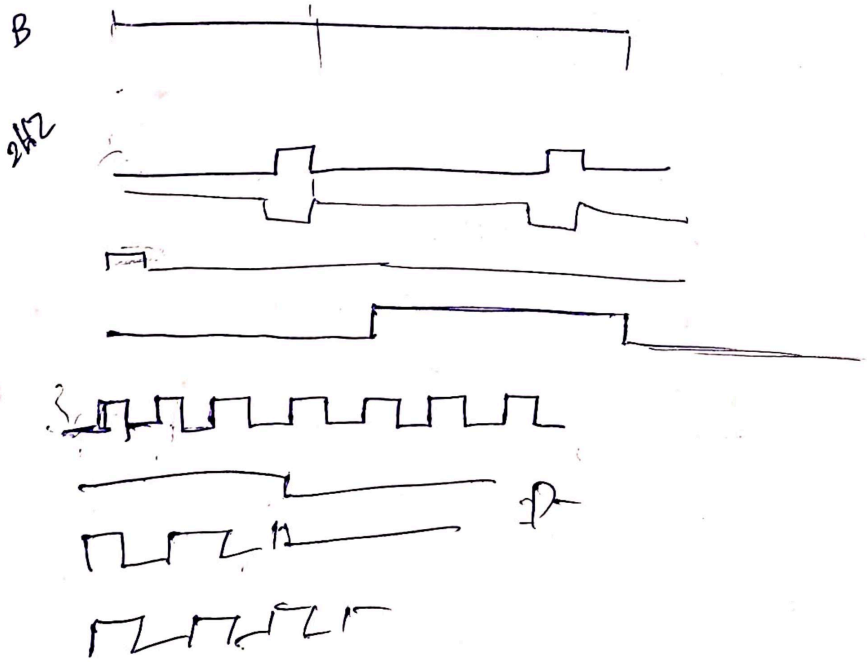
MUTE → mute MUTE
| ~mute IDLE
| up VOL_UP
| down MUTE

VOL_UP → ~up IDLE
| up VOL_UP
| up & Max IDLE

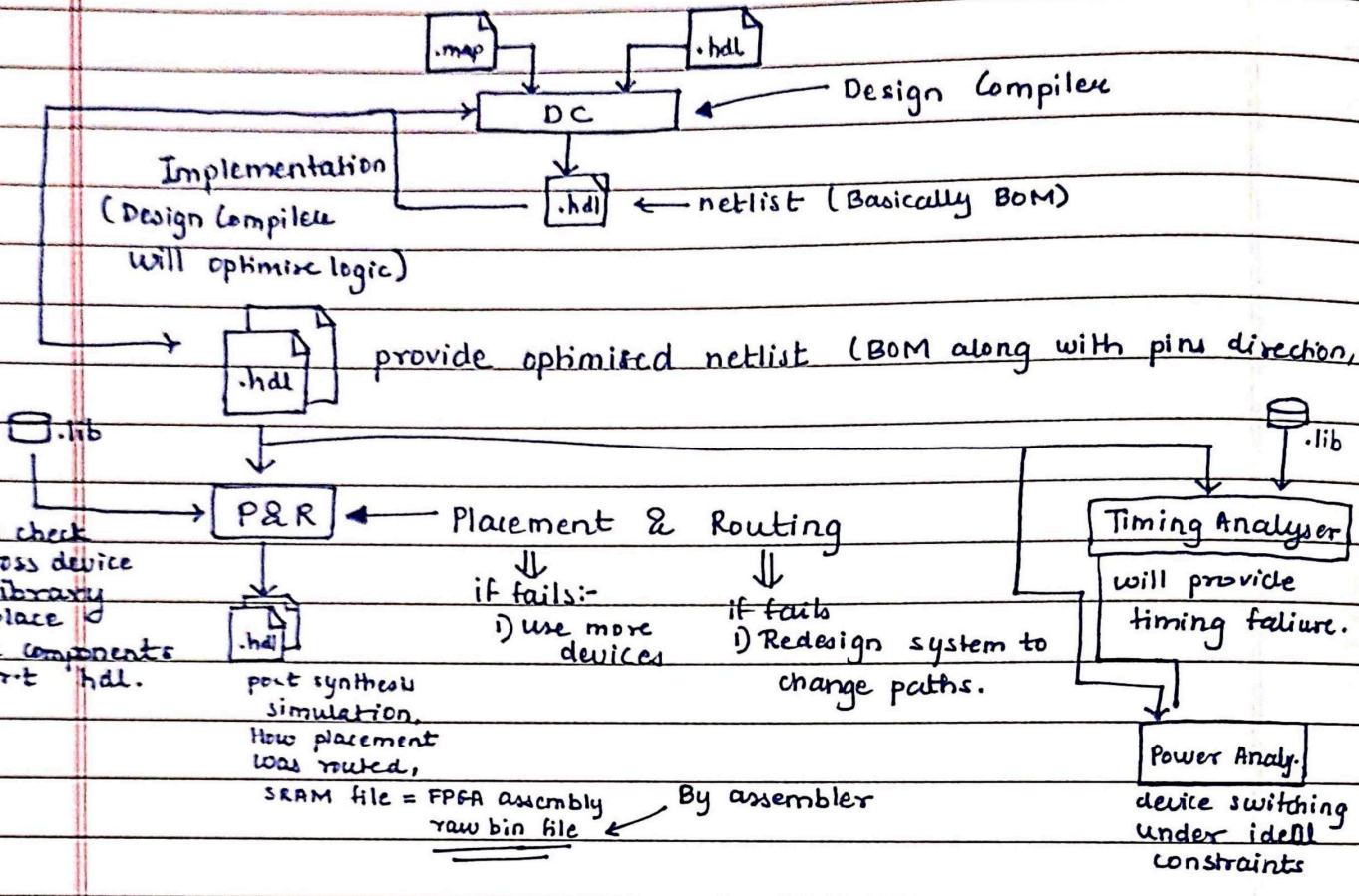
IDLE → mute MUTE
| up VOL_UP vol+1
| down VOL_DOWN vol-1
check max

MUTE → ~mute & mute clr IDLE
| mute MUTE





*) Reporting (Vivado) :- Vivado generates report at every stage except Linting & Elaboration.



6 types of reporting. & 3 more :- 1) utilization - No. of reg, buf, nets and all other used components 2) DRC :- If there exists any violations in designs. 3) Methodology :- parameters or how design was generated.

Floorplanning & IO planning.

TOP right ~~left~~ => SRAM.

XOV1

XOVD => UK regions

*****

very good!!!.

bitstream = from SRAM file to FPGA SRAM (Raw bin to Flash mem)

~~Mstream~~ MCS = more reliable than raw bin.

memory calibration system.

Low Power VLSI design

DFT, xvlog, xsim.

<u>Design constraint.</u> → Timing analyser, FPGA analyser, Simulator.

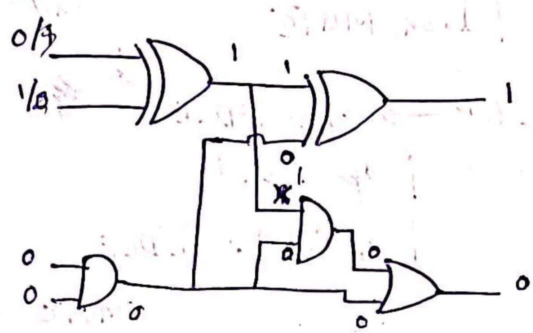
generatedclk, propagatedclk

Method
① ↳ out of context synthesis.
② → Block design.

<u>Design for testability</u>

Scan flop

1 0
0 1



[00 01 10]

A B	C D
0 1	0 0
1 0	0 0
0 1	0 1
1 0	0 1
0 1	1 0
1 0	1 0

<u>Block Designer.</u>

AXI BRAM controller.

Block mem generator

Altera (Lookup table) ✓ · LE vs Xilinx LUT* → config time

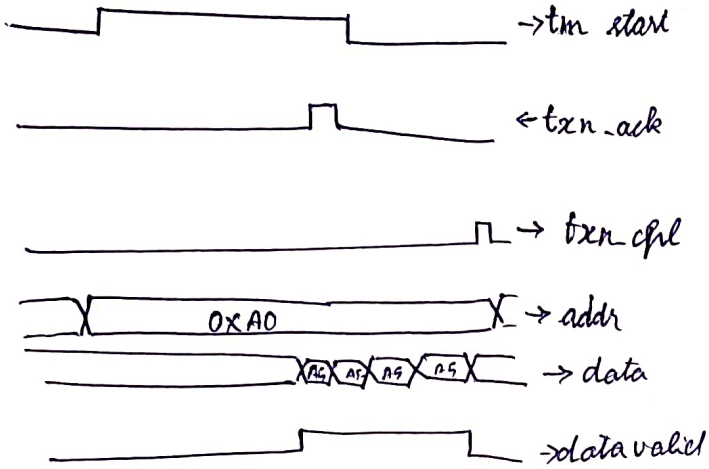
iCE40 ↓ Small form, Small power

· memory
~~Bulk~~ memory vs distributed v.s area
Through silicate — DRAM, between region →
faster — high density

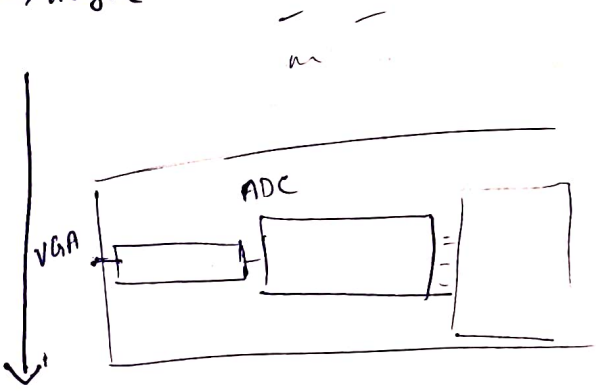
faster

ECP5
· Small factor
High perfor

Pipelining → Advantages
- long combinational
- Multi-cycle
- Failed / Failing.



VGA → Vsync, Hsync



Implementation in Basys 3

→ AXI4S to video out
video timing controller

IDLE txn_ack ⇐ 0

START txn_ack ⇐ 1 | →

if (data_valid == 1)
mem [addr] ⇐ data

TPGS → R, G, B

4563
1011|0011

1 Sec → 100×10^6 Hz → 99_992_999

30 Sec → x30 → 99_999_9999

12'b 1111 1111 1111

d'4095

30 Sec

0 0

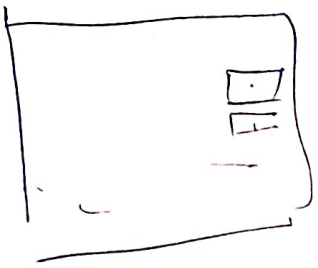
6x4
24

Sec

136.5

136.5 Hz ← 1
0 ← 0

$f(\Sigma, Q_1)$



$\delta_0 = f(IDLE, ask\_order)$

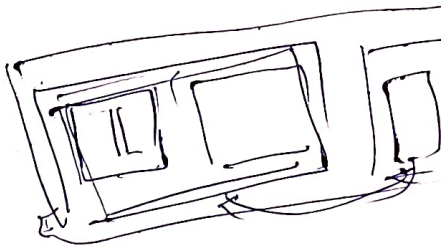
~~$\delta_1 = f(ASK\_ORDER, ip\_token)$~~

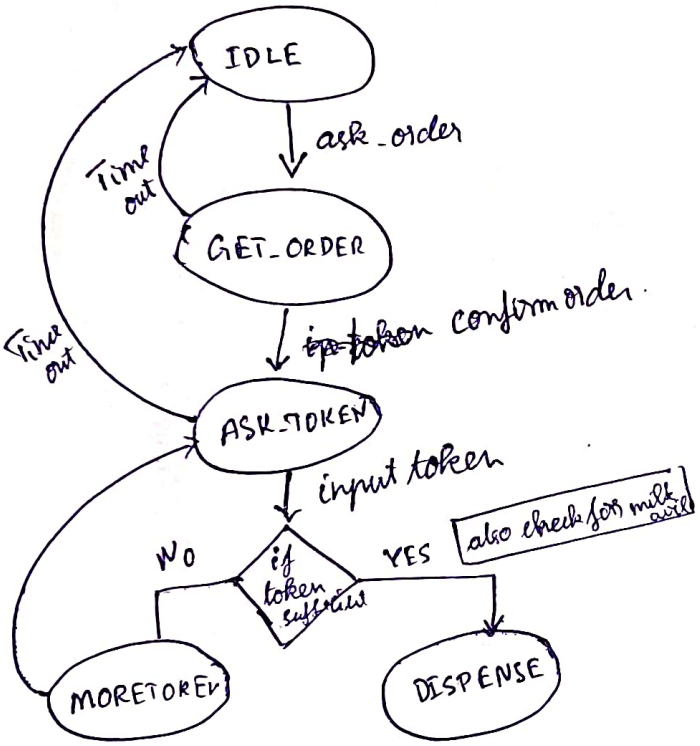
$\delta_1 = f(GET\_ORDER, in\_token)$

$\delta_2 = f(ASK\_TOKEN, dispense)$

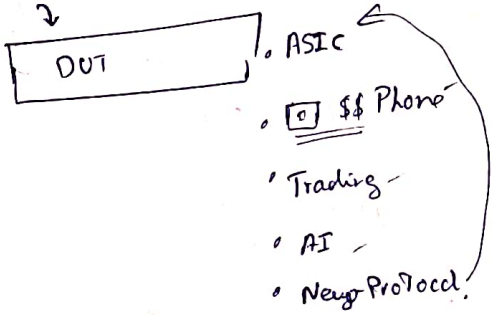
$\delta_3 = f(MORE\_TOKEN, dispense)$

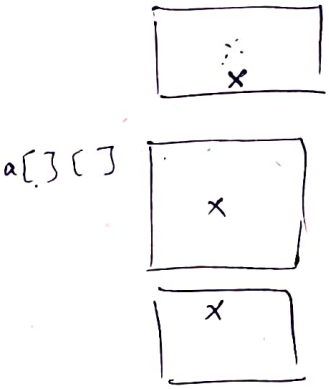
$\delta_4 = f(DISPENSE, ack\ order)$





→ FPGA





wait

parallel & full case.

disable

def param , # operator

full & parallel connection.

race condition

determitic non determinitic

PLI , VCD

ASMD chart.

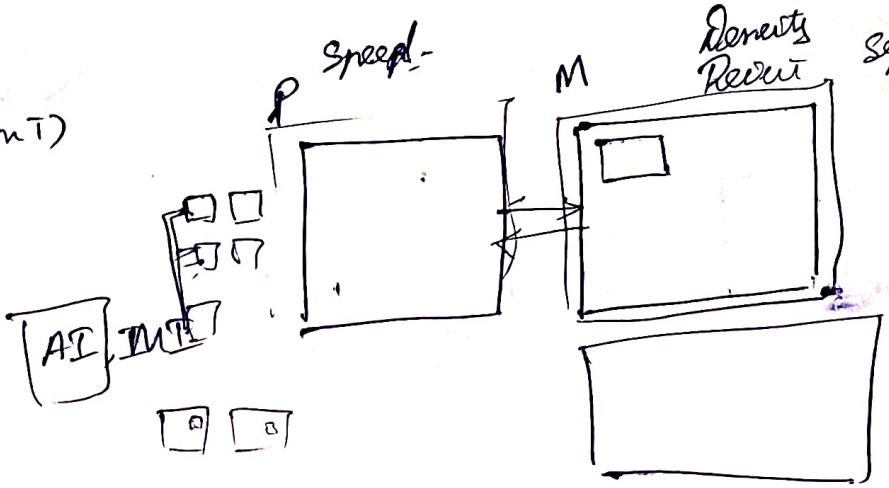
switch level modeling

Execute_data (transaction T)

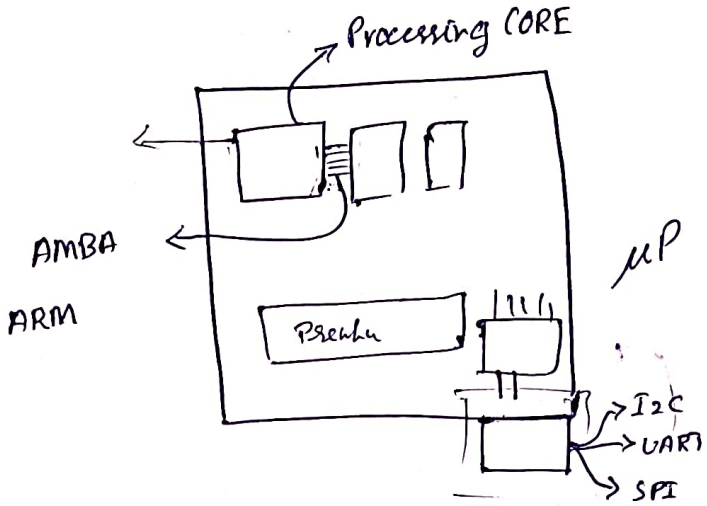
do_write

do_read

exite_addr.



do_write
(TW.sine)
WORD:



ERROR ; detection

× AMBA

× I2C

Ethenet ✓

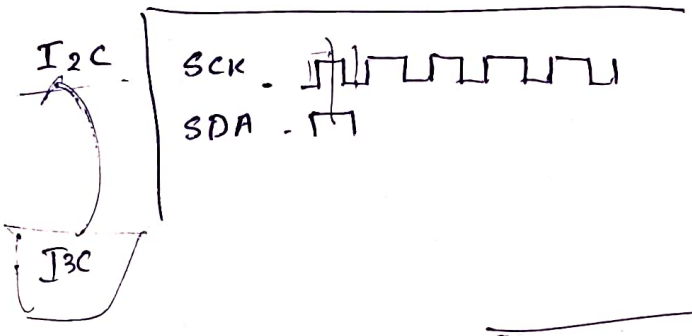
PCIe ✓

USB ✓

136KB/s

1Mb/s

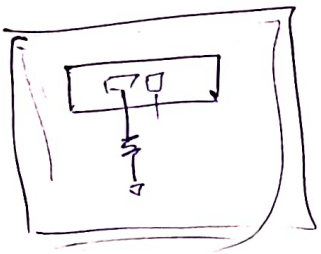
I2C



Multi

754,→

USB



I1 ✓

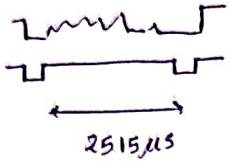
×

×,

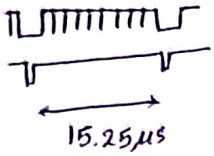
)

VGA - Video Graphic Array

- Resolution → 1080 x 1920 (Full HD)
- Frame Rate → 60 Hz
- Synchronization → hsync



vsync



- Xilinx AXI4-Stream Video Out
  - valid, ready, data
  - tlast, tready, tstart
    - tlast → EOL (End of Line)
    - tstart → SOF (Start of frame)
- Video Timing Control
  - hsync, vsync

Clocking Wizard

- clk_out1 → 148.5 (remove reset, locked for simplicity)

- Video Timing Controller
  - Detection / Generation
    - (uncheck AXI4-Lite interface) for test pattern
    - (uncheck enable detection)
  - Default / constant
    - video mode 1080p

-- Connecting all blocks --

- Slice
  - Din width - 24
    - from -
    - downto - } depending on colour
  - Dout width - 4

- Video test pattern generator.

clk

reset_n

data_ready

data_valid

[7:0] data

sof

eol