P8_31-08-21

31 August 2021 08:59

⊛ SPPU Assignment :- A-2

★ Universal Shift Register (USR) (4-bit)

⇒ SR (Shift Register) is a system which shifts bits from L→R or R→L

⇒ SR can accept i/p bits in a Serial / Parallel Fashion

⊛ Hence we have 2 options → Serial-In (SI)
→ Parallel-In (PI)

⇒ SR can output the stored bits in a Serial/Parallel Fashion

⊛ Hence we have 2 options → Serial-out (SO)
→ Parallel-out (PO)

⇒ A SR which allows all combinations of above 4 options is called as USR

⇒ Thus USR has 4 operating modes :-

i) SISO

ii) SIPO

iii) PISO

iv) PIPO

⇒ USR is basically a Register ie Memory

⇒ Being a Memory it has the following aspects :-

⇒ It has a Master-Reset (Asynchronous) (Independant of clock)

⇒ It is Clock-Driven → ↓ (Negative Edge Triggered)
→ ↑ (Positive Edge Triggered)

⇒ It has Mode-select → 2-bit

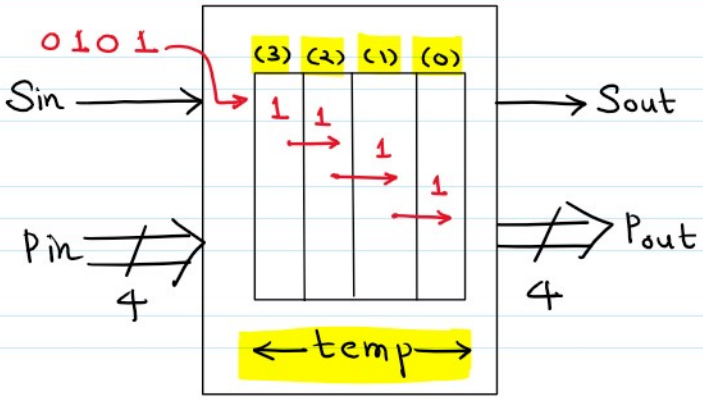
⇒ Based on All of the above facts, BD, FT of USR can be :-

Power Hierarchy :-

I/p signals	Power-Level
rst	①
clk	②

rst	①
clk	②
mode	③
Sin, Pin	④

Tells us the order in which to check status of i/p signals



✱ The above 4 operations will happen subject to :-

① rst = 0

② At the arrival of ↓ of clk.

✱ For rst = 1 :

Sout = 0

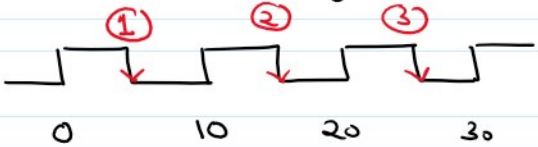
Pout = 0000

✱ Mode Latency = No. of clock cycles needed to execute the Mode

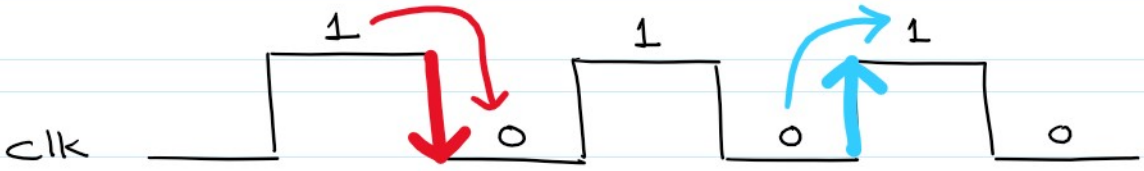
✱ Mode-Latency of Each Mode:-

	MODE	Latency (No. of clock cycles needed)
①	SISO	4(SI) + 4(SO) = 8 ↓
②	SIPO	4(SI) + 1(PO) = 5 ↓
③	PISO	1(PI) + 4(SO) = 5 ↓
④	PIPO	1(PI) + 1(PO) = 2 ↓

We define clock-period = 10 ns



✱ How to check status of clock ?



✱ Checking the Arrival of Falling-edge :-

① IF (clk'EVENT AND clk = '0') THEN

② IF falling_edge(clk) THEN

✱ Checking the Arrival of Rising-edge :-

① IF (clk'EVENT AND clk = '1') THEN

② IF rising_edge(clk) THEN

✱ Writing PROCESS for mode in TxM :-

```
mode : PROCESS
       BEGIN

          mode <= "00" ;
          WAIT FOR 80 ns;

          mode <= "01" ;
          WAIT FOR 50 ns;

          mode <= "10"
          WAIT FOR 50 ns;

          mode <= "11" ;
          WAIT FOR 20 ns;

       END PROCESS;
```