Setup Time – Input must be stable pre-rising edge
Hold Time – Input must be stable post-rising edge
tCQ Delay – Time taken for output to change, from rising edge of clock
Flip Flop – 1 bit of state sampling w/ rising edge
Register – Bits of state sampling on CLK or LOAD

Lecture 15: Synchronous Digital Systems State

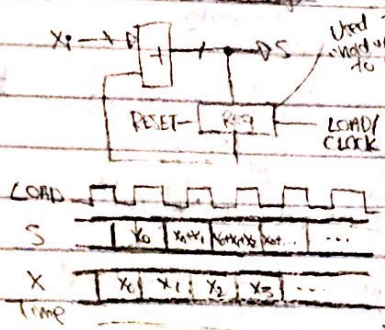
State Elements

① Build memory
② Control info flow

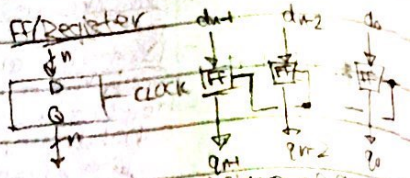
Accumulator (Faulty)

- Problem 1: no control over iterations in for loop
- Problem 2: no way to initialize the sum to 0 (or reset)

Accumulator (Working)

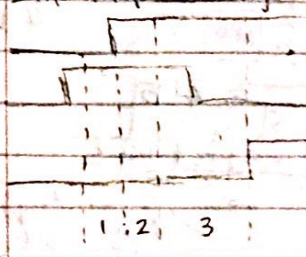


FF/Register



- Register has n instances of FF/flip-flops
- D = data, Q = output

Flip Flop Timing

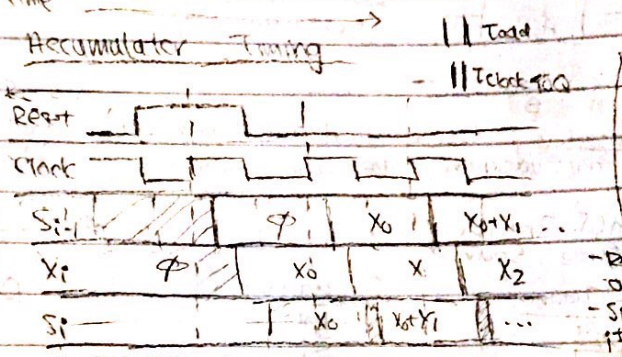


Note: D is only sampled & transferred to output on rising edge of clock

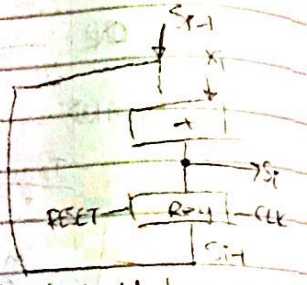
Input must be stable during this time
- Want to minimize setup/hold & clock to Q delay

① Setup Time: pre-rising edge
② Hold Time: post-T
2+3: Clock to Q Delay

Accumulator Timing

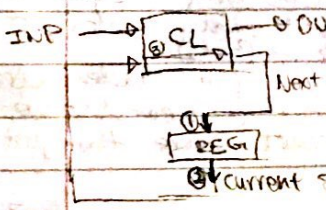


- Si is sampled, then shows up in Si-1
- Have to wait Tadd before Si-1+Xi is reflected by Si
- Xi does not arrive @ adder @ the same time as Si-1
- Rising edge of Reset ⇒ Si-1 reset after Tclock-to-Q. Then, adder has Si-1, Xi after Tadd ⇒ Si has Si-1+Xi
- After Tclock-to-Q, Si samples Si during which a window of time Si-1 and Xi = X0, so Si = X0+X0

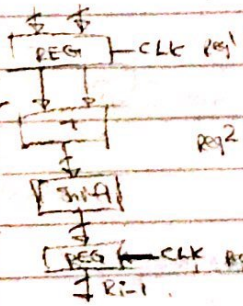


- Reset should have priority over new input
- Si-1 has result FROM ith iteration
- Timing analyzed @ the output
- Si is temp wrong but is stabilized from tsetup to t
- Good circuits should not be unstable around rising edge of clock
  - Instability caused by freq. of the clock

Pipelining



- Delay: combinational logic block, possibly from asynchronous inputs
- Max Delay = Clock to Q Delay (2) + CL Delay (3) + Setup time (1) which fits into 1 period
- Use of extra registers: speed up the clock rate

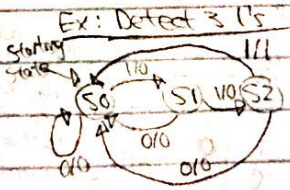


- Add register allows for a higher clock freq ⇒ ↑output/second

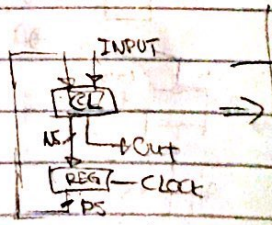
Finite State Machines

- FSM: Repr. a func w/ state transition diagram

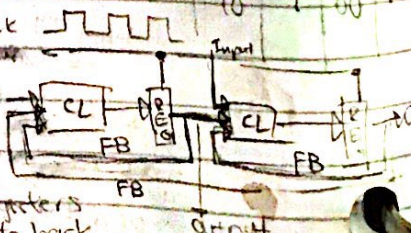
Ex: Detect 3 1's



Encoding: Input/Output
- Detect 3 consecutive 1's, then report & state 0 even if 3 found
  - Each state repr. # of 1's seen so far



Prev State	Input	Next State	Output
00	0	00	0
00	1	01	0
01	0	00	0
01	1	10	0
10	0	00	0
10	1	00	1



FB: Optional feedback

Synchronous Systems Model

- CL blocks separated w/ registers
  - CL/reg's can be back to back
- Output of reg can be feedback
- CLOCK connected only to reg. inputs