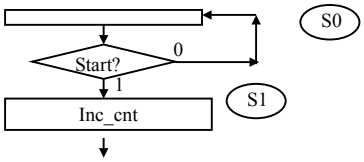
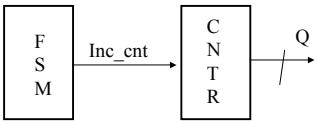
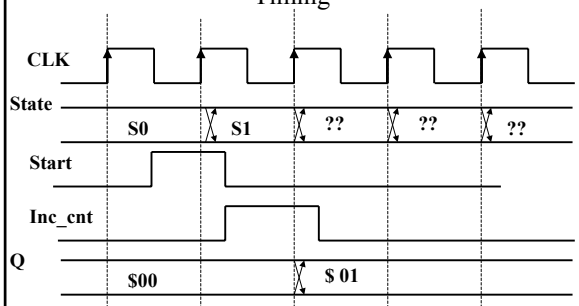
FSM Timing Examples



What does timing look like?

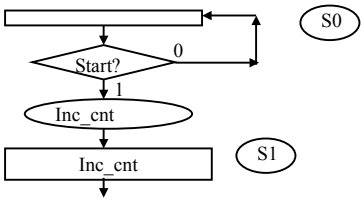


Timing

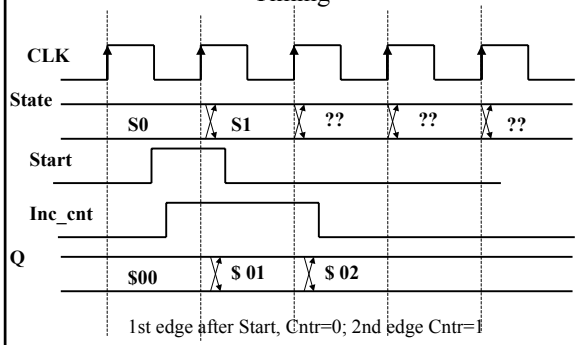


1st edge after Start, Cntr=0; 2nd edge Cntr=0; 3rd edge Cntr=1

How is this different?

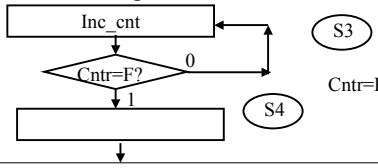


Timing



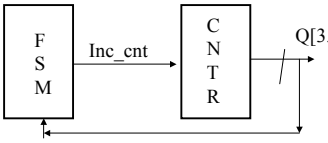
1st edge after Start, Cntr=0; 2nd edge Cntr=1

Checking for End of Count



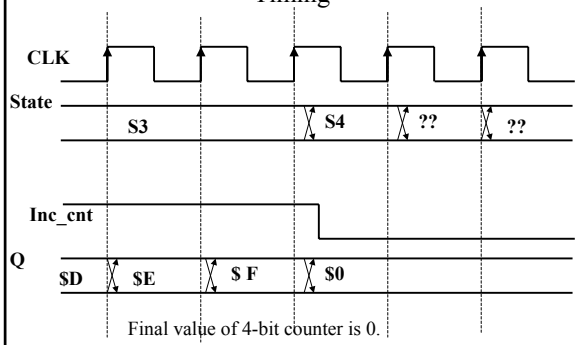
Cntr=F = 1111?

What does timing look like?



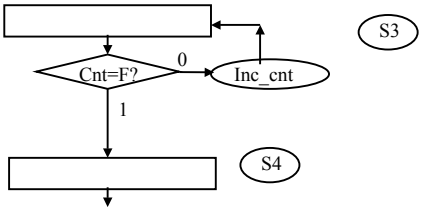
4 bit counter

Timing

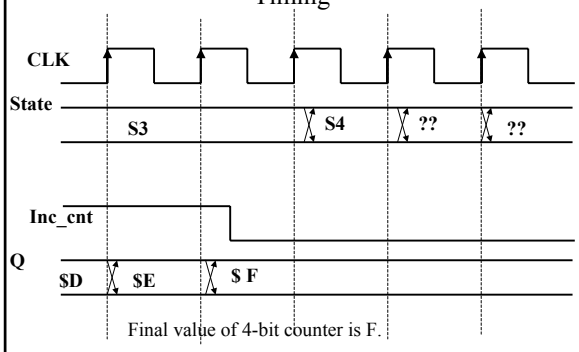


Final value of 4-bit counter is 0.

How is this different?



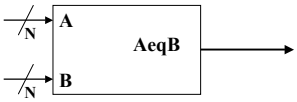
Timing



Final value of 4-bit counter is F.

A Comparator

Another common combinational building block is a comparator.

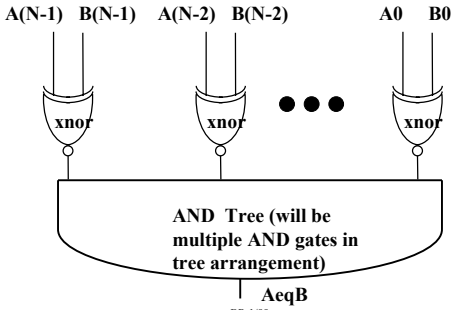


A=B if A(0) = B(0) and A(1) = B(1) … and A(n-1)=B(n-1)

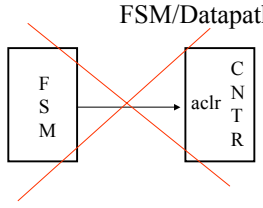
Recall that "xnor" function is '1' if A=0, B=0 or A=1, B=1!
So AeqB is:

AeqB = (A(0) xnor B(0)) and (A(1) xnor B(1)) and ….etc.

What is logic structure for equality comparator?

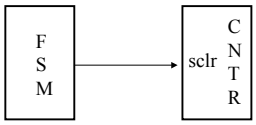


FSM/Datapaths No-Nos



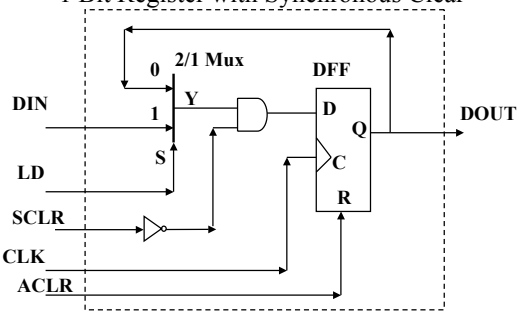
Control of asynchronous clear by FSM - bad!!

Glitch on aclr line by FSM logic can cause inadvertent clear operation! Also, generally want synchronous behavior of CNTR during FSM operation (all outputs changing on clock edges).



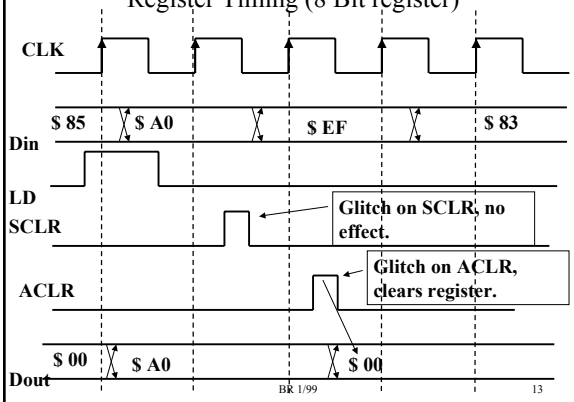
Control of synchronous clear by FSM. Good!

1 Bit Register with Synchronous Clear



Note that SCLR = 1 will set DFF=0 on next active clock edge.

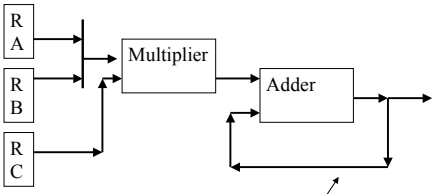
Register Timing (8 Bit register)



What Causes Glitches on outputs of FSMs?

- Many combinational paths in logic that defines next state.
- If these paths have unequal numbers of gates, or gates have different delays, then glitches can occur.
- We normally don't care about these glitches as long as the output lines are STABLE before the next clock edge (satisfy the setup time requirement)
- If output lines are connected to asynchronous control inputs, then glitches can be a BIG problem!
- Solution: Don't connect asynchronous controls lines to FSM outputs or guarantee FSM outputs are glitch free (come directly from a FF).

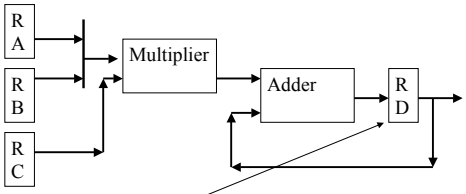
FSM/Datapaths No-Nos



This datapath is trying to accumulate a multiply/add from data values in registers RA,RB, RC.

The adder has a COMBINATIONAL LOOP!! Will oscillate!!! Bad!!!

FSM/Datapaths No-Nos



Must use a Register to hold value of multiply/add for next multiply/add operation. Good!!!