CE653 – Asynchronous Circuit Design

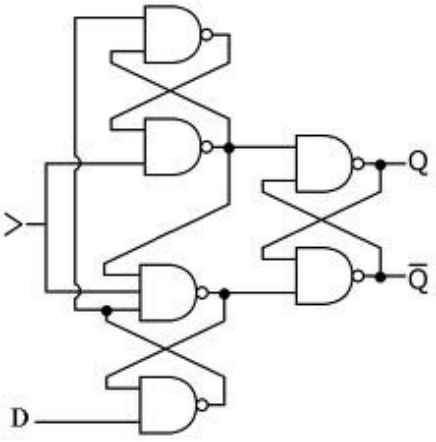
Instructor: C. Sotiriou

http://inf-server.inf.uth.gr/courses/CE653/

This Presentation

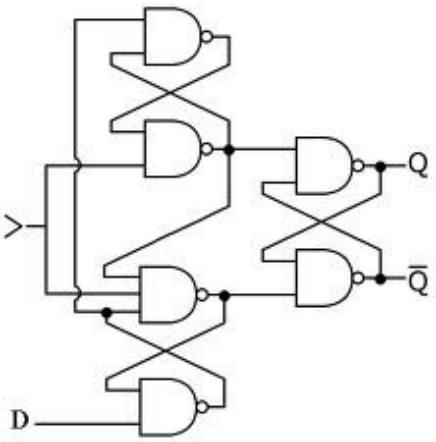
- MSFSM Intuitive Example
  - Edge-triggered FF Example
- D-type FF presented in Mano's Book
  - How do we formally verify correct operation?
  - What models are needed?
  - FSM product (composition of FSMs)
  - MSFSMs (Multiple Synchronised FSMs) – analysis without product using asynchronous FSMs

D-type FF from Mano's Book



- consists of 3 set-reset latches
- RHS latch produces Q, Q'
- Verbal functional explanation quite complex and non-intuitive
  - when **input D is high**, lower LHS latch is set whenever the clock is low
  - thus, the set input of the upper LHS latch is triggered, which sets the output latch (RHS) whenever the clock is high
  - when **input is D low**, lower LHS latch is reset, thus resetting the output latch (RHS), whenever the clock is high
- As a result, Q may only change state when clock makes a low→high transition

D-type FF from Mano's Book

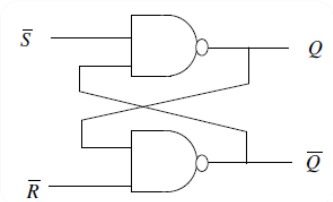


- How can we formally verify that output latch's inputs can never cause race?
  - are never 00, OR
  - can never transition from 00→11

Active-Low SR latch

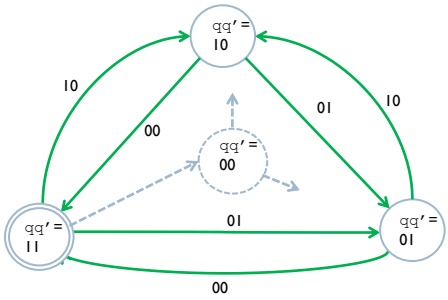
S	R	Q	Q_{nxt}	$\overline{Q_{nxt}}$
0	0	x	1	1
0	1	x	1	0
1	0	x	0	1
1	1	0	0	1
1	1	1	1	0

- QUESTION:
  How do we model the sequential SR latch behaviour?



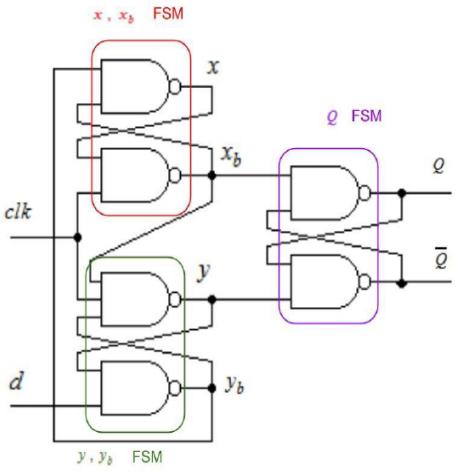
Single (Active-Low) SR latch FSM Model

- Choice between 2 states or 3 states
  - 2 states model hides possibility of SR = 00
- 00→11 transition result is uncertain, thus the fourth state QQ' = 11 does not have deterministic next states
- We assume initial state to be 11
  - May be different, e.g. 01
  - Transitions correspond to SR inputs

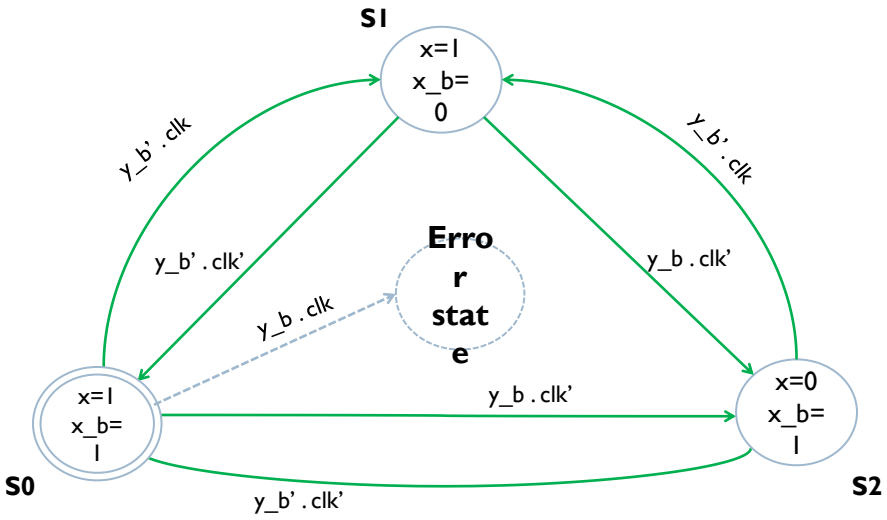


D-Type FF – inferring MSFSM model

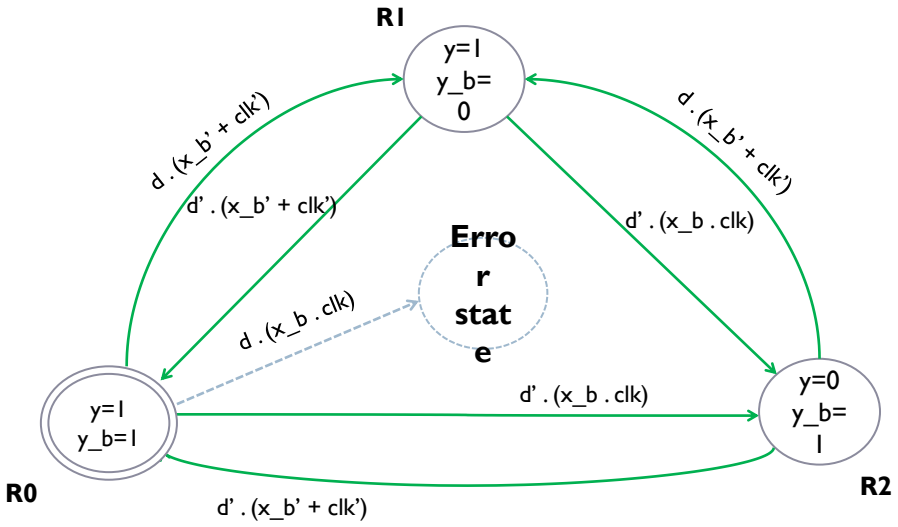
- We split the D-type FF circuit into 3 separate sub-circuits *(SR latches)*
- *Infer an FSM of each sub-circuit*
- Signals and states will transition asynchronously
- Goal:
  - Signals x_b, y may never assume 00, OR
  - may never transition from 00→11



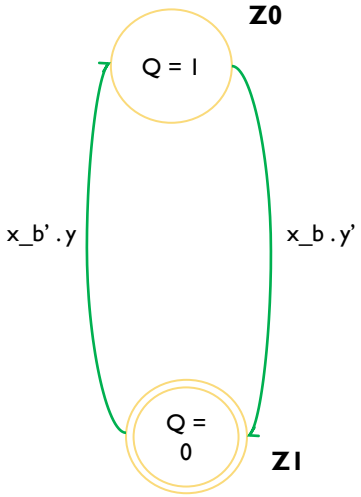
FSM #1 - x, x_b signal generation



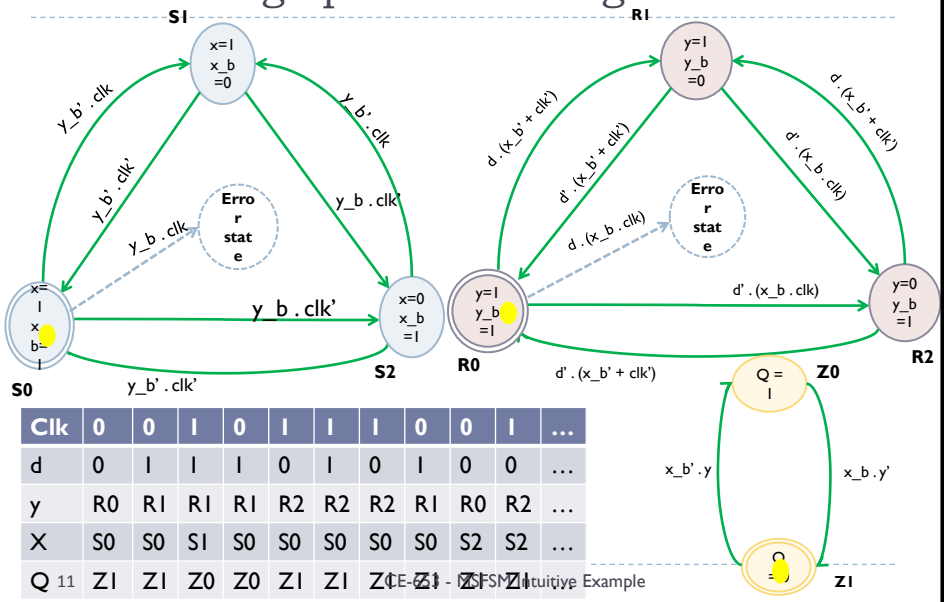
FSM #2 - y, y_b signal generation



FSM Q – RHS latch output Q



Simulating operation using MSFSMs



Clk	0	0	1	0	1	1	1	0	0	1	…
d	0	1	1	1	0	1	0	1	0	0	…
y	R0	R1	R1	R1	R2	R2	R2	R1	R0	R2	…
X	S0	S0	S1	S0	S0	S0	S0	S0	S2	S2	…
Q	Z1	Z1	Z0	Z0	Z1	Z1	Z0	Z1	Z1	Z1	…

Conclusions

- The traditional way of verifying a model demands multiplication of FSMs, which leads to bigger state spaces
- Using interactive FSMs and based on an asynchronous change of signals, we escape the boundaries of a monolithic FSM and avoid state explosion.
- The state space of the formal analysis in our example was reduced by almost 55%