CS1026 – Digital Logic Design

Feedback Circuits

Alistair Morris [1]

[1]Distributed Systems Group
Trinity College Dublin

March 7, 2016

Today's Overview

A quick recap.. I

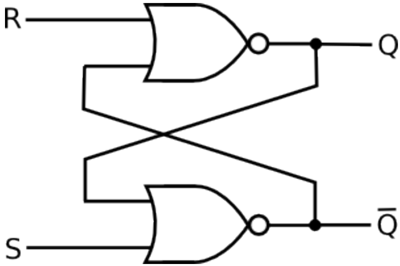
Last time we looked at a *SR Latch*:

- A multivibrator
- Used to store an input value

Real World Use

Often we want to react to what *has* happened

The SR Flip Flop [Kojima, 2013] I



Set and **R**eset inputs

The SR Flip Flop [Kojima, 2013] II

Q^+ defines the next output state (Q_{next})

- Q^+ dependent on:
  - Inputs S and R
  - Current state: Q

The SR Flip Flop [Kojima, 2013] III

Characteristic Table				Excitation Table			
S	R	Q_{next}	Action	Q	Q_{next}	S	R
0	0	Q	Hold State	0	0	0	X
0	1	1	Reset	0	1	1	0
1	0	1	Set	1	0	0	1
1	1	X	Now allowed	1	1	X	0

Table: Characteristic and Exciation Table

You can think of this as the sequential circuit equivalent to a combination circuit's truth table

The SR Flip Flop [Kojima, 2013] IV

S	R	Q^+
0	0	Q
1	0	1
1	1	X

If we feel lazy:

- Write down the *Compressed Characteristic Table*
- We almost have a truth table

The SR Flip Flop [Kojima, 2013] V

SR \ Q	0	1
00	0	1
01	0	0
11	0	0
10	1	1

We can also do a K-Map

- But remember states (mid-terms) 3, 4 and 7
  - These define unstable states (i.e. don't go there)
  - Δ time needed for flip flop to respond

The SR Flip Flop [Kojima, 2013] VI

SR \ Q	0	1
00	0	1
01	0	0
11	0	0
10	1	1

The K-Map works too!

- $Q^+(S, R, Q) = QR' + SR'$

SR Flip-Flop Issues I

SR Flip Flop denotes a *Latch*:

- No Clock to sync signals
  - $\therefore$ We have an async circuit

SR Flip-Flop Issues II

This becomes problematic in circuit design:

- A *race condition* exists:
  - $\{S, R, Q\} = 0$
  - $S \implies 1 \implies \{S, R, Q\} = 100$ (Unstable state!!) $\implies Q^{+} = 1 \implies \{S, R, Q\} = 101$
  - After ΔT

SR Flip-Flop Issues III

If the S and R inputs change quickly before the output settles into a new stable state:

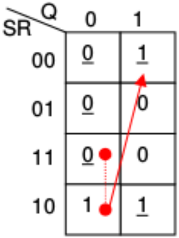
- The input provides a race condition
- Each trying to change the output first). If the output becomes a predictable stable state
  - then we have a non-critical race

SR Flip-Flop Issues IV

A critical race occurs if the circuit output ends in an *unpredictable* stable state.

- For Example
  - $S, R, Q = 110$
  - SR becomes 00
    - S changes first – $\{S, R, Q\} = 010 \implies Q^+ = 0$
    - R changes first – $\{S, R, Q\} = 100 \implies Q^+ = 1$

SR Flip-Flop Issues V



Note that depending on if S changed first or R changed first:

- Final state will be different
- This means we have a critical race.

Finite State Machines I

Step 1. Write the Compressed Characteristic Table ⇨

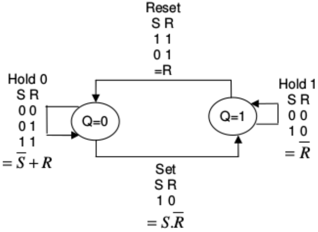
S	R	Q+
0	0	Q
0	1	0
1	0	1
1	1	0

(reset dominant)

Step 2. Present State /Next State (PS/NS) Table ⇨

S	R	Q	Q+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Step 3. Draw a circle for each state (one output, Q, so 2 –state)
Step 4. Draw arrow showing change for each possible input in each state.



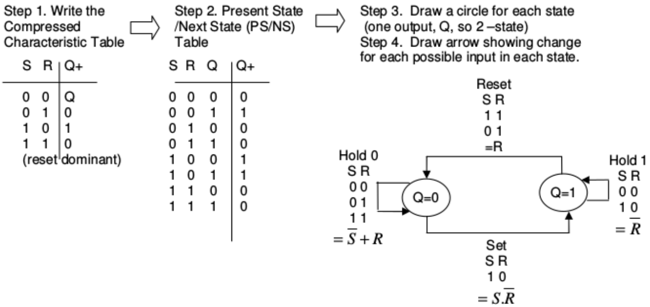
We can draw the relation between states in the SR Flip FLop

Finite State Machines II

A completely specified state machine is one for which all input conditions specify each next state condition:

- $\sum = 1$ Rule
- For state 0 in SR Flip Flop:
  - $(S' + R) + SR' = 1$

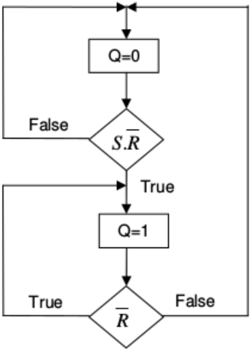
Finite State Machines III



In the case of incomplete state machines:

- We can consider those states as *don't cares*

Finite State Machines IV



Algorithmic State Machine (ASM) help us out even more!

- Basically a *Flowchart*

Finite State Machines V

Chart uses three symbols:

- Rectangle – State Box
- Diamond – Decision Box
  - If true take "1" path
  - Else take "0" path
- Oval – Output box

References (Homework) I

Kojima, S. (2013).
Sr flip-flop.
US Patent 8,497,722.