Arnav Goyal

Digital Design

Self Study

Preface

The purpose of this document is to act as a comprehensive note for my understanding on the subject matter. I may also use references aside from the lecture material to further develop my understanding, and these references will be listed here.

This document should eventually serve as a standalone reference for learning or review of the subject matter. There is also a lot of organization within these documents, please refer to the table of contents within your PDF viewer for ease of navigation.

References

- Provided Course Materials from ECE 2277, ECE 3375
- Digital Design with an Introduction to the Verilog HDL - 5e - M. Mano, D. Ciletti
- Verilog Complete Tutorial - VLSI Point (YouTube Link)

Finite State Machines

A **finite state machine** or[1] (**FSM**) is a system that can be in a finite number of **states**, accept a finite set of **inputs**, producing a finite state of **outputs**.

- Listing these states, the possible transitions between states based on the input, and the conditions required for each possible output, provides a complete function description of an FSM.

A FSM is a general concept, a mathematical concept. However, when dealing with a specific subset of then called **deterministic FSMs**, we can actually implement them with sequential circuits.

- An FSM is **deterministic** if every combination of current state and input results in only one transition (no probabilistic transitions).

Mathematical Formalism

There are two types of FSMs. A **Mealy** and a **Moore** FSM.

- The output of a Moore FSM depends on only the state it is currently in.
- The output of a Mealy FSM depends on transitions between states.

[1] Sometimes an FSM is called a Finite State Automaton

Lets make things more organized by introducing some basic mathematical notation to this.

- Let the set of all possible states in an FSM be denoted by Q, lets also call the starting state Q_0
- Let the set of all possible inputs be called I
- Let the transition function be denoted $\delta(Q,\ I)$, it determines the next state as a function of the current state and next input, we say: $\delta : Q \times I \mapsto Q$
- Let the output function be given by $f(\cdot)$, it determines the output as a function of something.

This means that any FSM M can be described by the following basic description[2]

$$\mathrm{M} = (Q, I, Q_0, \delta, f)$$

We can differentiate mathematically between Mealy and Moore FSMs through the description of their output functions. Let the set O be the set of all possible outputs.

- A Moore FSMs output function is defined[3] as $f : Q \mapsto O$.
- A Mealy FSMs output function is defined[4] as $f : \delta \mapsto O$

[2] Here the functions are written as variables for clarity, we also probably cant define these functions in any way other than a truth/state table. For now just assume that they can.

[3] A Moore FSMs output depends ONLY on its current state

[4] A Mealy FSMs current output depends on the SPECIFIC transition it is experiencing

Clock Domain Crossing

Clock Domains

So far we've only discussed **synchronous** circuits[5]. In the real world, it is extremely common to have **asynchronous** circuits[6]. Consider some circuit C with two design blocks $C_1, C_2 \subseteq C$

- we say that the circuit is synchronous if the only clock within C is f_1 or derivatives[7] of f_1
- we say that the circuit is asynchronous if there are multiple clocks - if C_1 is clocked by f_1, and C_2 clocked by f_2

A problem occurs[8] when trying to send data from C_1 to C_2, we are trying to send data at some frequency f_1, and trying to sample it on some frequency f_2. Whenever this occurs, the data we are trying to send has the possibility of becoming **metastable** .

[5] circuits that share the same clock.

[6] circuits with multiple clocks

[7] $f_1/2, f_1/4$, etc.

[8] This is the problem of Clock Domain Crossing (CDC)