Digital Embedded Systems

Lecture Notes

Chen, Henry 22703907

August 14, 2020

Contents

1 Combinatorial and Sequential Circuits **2**
1.1 Combinatorial Circuits 2
1.1.1 De Morgan's Law 2
1.1.2 Decoder 2
1.1.3 Encoders 2
1.1.4 Combining Encoders and Decoders 2
1.1.5 Multiplexer 3
1.1.6 Demultiplexer 3
1.1.7 Half Adder 3
1.1.8 Full Adder 3
1.1.9 Tri-State 3
1.2 Sequential Circuits 3
1.2.1 Memory Cell & Level Triggered Latches 3

Chapter 1

Combinatorial and Sequential Circuits

Combinatorial Circuits only use logic gates with no feedback or memory - it has no state of operation. In contrast, sequential circuits require logic gates with feedback and memory - allowing it to have a state.

1.1 Combinatorial Circuits

These are the lowest level of circuits - the building blocks of higher complexity circuits (and gates, or gates and not gates can make other types of gates). Much like a function - data goes into the gate input and out in the other side. To build more ocmplex circuits using AND and NOT gates, we will also require encoders and decoders.

1.1.1 De Morgan's Law

It is used to identify if circuit combinations are equivalent. This law states:

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$

1.1.2 Decoder

A decoder is equivalent to the *switch function*. It takes a binary input and decodes it into a binary output. To build this, you can use different logic that map a set number of inputs into a set number of outputs.

1.1.3 Encoders

Encoders take in single value inputs (e.g. 3 = only pin 3 high out of n pins) inputs and outputs a binary representation of the value (e.g. pins 1 and 2 will be high whereas everything else remains low thus encoding it into a binary representation). This is considered a decoder as there is no set state that determines what comes out of the encoder.

1.1.4 Combining Encoders and Decoders

Encoder Decoder configurations can be used to compress information into two wires. As long as the systme encodings are the same, the output that is decoded will be transmitted properly.

1.1.5 Multiplexer

A multiplexer takes in two inputs and returns a single output. The multiplexer needs to select between the different inputs - outputting it into a single output. Multiplexing requries a switch that changes which input the data is being read in form. Assuming we are reading inputs from I_1 & I_2, choosing which input we are reading from with S and outputting to Z, the a multiplexer is equivalent to the following formula:

$$Z = (A \cap \overline{S}) \cup (B \cap S)$$

Furthermore, we can encode multiple selects with more select pins. This is achieved by using a decoder which takes in the binary encoding of the select pin and outputs it into a mapped decimal pin. Furthermore, we can combine multiple multiplexers switched by a master select - taking m^n amount of multiplexers and outputting n outputs.

1.1.6 Demultiplexer

A demultiplexer takes in a single input and outputs it to different channels with a select input. To scale up the multiplexer, more select lines can be used, we use a decoder that maps to any number of outputs - connecting each output to an OR gate and generating a signal from there.

1.1.7 Half Adder

A half adder takes in two inputs and two outputs - showing whether or not two inputs require a carry or not. It is made up of an XOR gate for the sum and an OR gate for the carry.

1.1.8 Full Adder

A full adder is used to properly add binary numbers. They are made up of two half adders - one which computes the first bit and another which takes the carry bit and adds that to compute another sum. To achieve an n sized adder, we chain full adders that move the carry to the next subsequent adder.

1.1.9 Tri-State

This is required when talking about memory. It contains an enable pin that connects the input from the output. As a result, the enable pin creates another state that we can term *floating* - a state where the state is neither *high* or *low*.

1.2 Sequential Circuits

These are circuits that have a feedback loop that allow the circuit to store a single bit of information. Typically, we achieve feedback by bringing the output of the circuit back into the circuit's input pins.

1.2.1 Memory Cell & Level Triggered Latches

A memory cell is made up of two NOT gates in series - with the output fed back into the input of the first NOT gate. State is held as the system will hold it when the system is stable - there is no way to reset this system.

Alternatively, we can use a NOR or NAND gates to to create volatile memory - memory that can store states store state S - resetable by reset input R.