Chapter 5 Combinational Logic Design Practices

This chapter is concerned with examples of basic combinational circuits including decoders, comparators, xor gate and parity circuits, multiplexers, and adders. Those basic building circuits frequently appear in the combinational circuits.

Documentation Standards

Documentation is necessary for correct design and efficient debug and maintenance of digital systems. A documentation should consist of the following items:

- a specification describes the functionality (e.g., what it does) of a circuit and its inputs and outputs.
- a block diagram is an informal pictorial description of the circuit’s major functional modules and interconnections.
- a schematic diagram is a formal specification of the electrical components of the system, their interconnections, and details about each IC chips.
- a timing diagram shows the various logic signals as a function of time.

Block Diagrams

Block diagram shows the inputs, outputs, functional modules, internal data paths, and important control signals of a system. See Figure 5.1 for example.

Note a bus is a collection of two or more signal lines. It represents interconnection and data flow between two functional modules.
Gate Symbols

Different gate symbols will appear in schematic diagram of a circuit. Standard gate symbols have been introduced. Figures 5-3 and 5-4 show standard shapes for commonly used gates.

Note a buffer is a circuit that converts weak logic signals to strong ones.

Active Levels

Each signal has an active level associated with it. A signal is \textbf{active high} if it performs the named action when it is high. A signal is \textbf{active low} if it performs the named action when it is low. A signal is said to be \textbf{asserted} if it is at its active level. A signal is said to be \textbf{negated} if it is not activated.

Active Levels (cont’d)

Different naming conventions may be used to represent active levels. See table 5-1. Figure 5-5 shows how to represent the active levels of the input and output a logic circuit (as represented by a rectangular box). An inversion bubble to indicate active low while the absence of a bubble to indicate an active high. Alternatively, a logic symbol for a digital circuit may have its pins labeled as active high or active low.

Schematic Diagrams

Figure 5-14 and 5-15 show two different schematic structures: flat and hierarchical. Figure 5-17 shows a detailed schematic digram for a circuit using a 74HCT00. Figure 5-18 shows pinouts for standard 74-series ICs.
Timing Diagrams

A timing diagram illustrates the logical behaviors of signals in a digital circuit as a function of time. The most important information provided by a timing diagram is a specification of delay. See Fig. 5-19. Note a delay is subject to many factors and is often specified by a range between maximum and minimum delay or by a typical delay value.

Decoders

A decoder is a multiple input and multiple output logic circuit that decodes the coded input. It maps an input code into an output code. The number of inputs is usually fewer than that of output. A decoder usually contains a special input called enable. It must be asserted before the decoder can perform its normal function. It is not counted as an input. See figure 5-31.

Programmable Logic Arrays (PLA)

PLA is a combinational two level AND-OR device that can be programmed to realize any sum of products expression. A PLA often has a maximum number of inputs, outputs, and the product terms. See figs. 5.21 and 5.22. See Fig. 5-23 for programmed PLA. Note both the AND and OR gates are programmable. Another programmable logic device is Programmable Array Logic (PAL), which has fixed OR gates and only AND gates can be programmed.

Binary Decoders

The most common decoder is the binary decoder where the number of output is the power of 2 of the number of input. For example, we have $n$ to $2^n$ binary decoder, where $n$ is the number of input and $2^n$ is the number of output. A binary decoder is used when you need to activate exactly one of $2^n$ outputs based on $n$ inputs. Note the "don't care" notation in the truth table.
Binary Decoders (cont’d)

Table 5-4 gives the truth table of a 2-to-4 binary decoder. The logic symbol and the logic diagram of a 2-to-4 binary decoder are shown in Figure 5-32.

The input codes are not limited to decimal numbers but may represent any consecutive physical entities and not all outputs of a decoder may be used. For example, see table 5-6.

Logic Symbols for Decoders

The logic symbol for a decoder is drawn with input pins on the left and output pins on the right of the symbol. The active levels of the input and output pins need be specified. Each pin has two names: one internal and one external name. For active high, they are the same. For active low, the external name is suffixed with \( L \). See figures 5-33 and 5-34.

Standard Binary Decoder ICs

- 74 × 139
- 74 × 138

74 × 139 Dual 2-to-4 Decoder

The 74 × 139 dual 2-to-4 decoder consists of two independent but identical 2-to-4 decoders, contained in a single MSI (middle size integration) chip. It is designed with active-low at outputs. Figure 5-35 gives the logic diagram and logic symbol for the decoder. The truth table for half of the 74 × 139 dual 2-to-4 decoder is shown in table 5-6. Note for both input and output the top input and the top output bit represent the LSB while the bottom input bit and the bottom output bit represent the MSB.
**74 × 138 MSI 3-to-8 Decoder**

It has three enable inputs and its outputs are active low. All the three enable inputs must be asserted before the decoder can perform its function. Its logic diagram and logic symbol are shown in Figure 5-72. Truth table is shown in Table 5-7. An output is asserted if only the decoder is enabled and the output is selected (e.g., \( Y_{3L} \)).

**Cascading Binary Decoders**

Multiple binary decoders can be combined in cascade to decode larger code words. Figure 5-38 shows how to combine two 3-to-8 decoders into a 4-to-16 decoder. Figure 5-39 shows a 5-to-32 decoder constructed from 4 3-to-8 decoders and 1 2-to-4 decoder.

**Three State Buffers**

A three state buffer contains two inputs and one output. One of the inputs is *enable* input. It must be asserted before the buffer can function. The three state buffer can be inverting or non-inverting with either active high or low enable input. See figure 5-53 for example.

**Standard 3 State Buffers ICs**

Several independent three state buffers may be packaged in a single SSI IC or MSI IC. 74 × 125 and 74 × 126 each contains 4 three state buffers (see figure 5-56). 74 × 541 contains 8 three state buffers (see figure 5-57).
Use of 3 State Buffers
See figure 5-54, the use of three state buffers allow to 8 sources of data to drive a single line at a time.

Multiplexers
A multiplexer is a digital switch that determines which of its $n$ input data should be routed to output. It is a useful device in any application where data must be switched from multiple sources to a destination source. For example, the multiplexer between the processor's registers and its arithmetic logic unit (ALU). The data from one of the registers must be routed to ALU for execution.

Multiplexers (cont’d)
The input of a multiplexer consist of
- $n$ input data, each of which is $b$ bit wide. $n$ is usually 2,4,8,16, and $b$ is 1,2, and 4.
- $s$ selections inputs to determine which of the $n$ sources to select. $s = \log_2 n$.
- an enable input, which must be asserted for the multiplexer to function.

Output: one of the $n$ input data
see figure 5-62.

Standard Multiplexers ICs
$74 \times 151$ selects among 8 1-bit inputs. It provides both active high and low output as shown in Figure 5-63. Its truth table is shown in Figure table 5-34. $74 \times 157$ has two 4-bit inputs and 1 4-bit output. See Figure 5-64 for its logic diagram and logic symbol.
**Standard Multiplexers ICs**

See logic symbols and truth tables for $74 \times 157$ (2 inputs and 4 bit) and $74 \times 153$ (4 input 2-bit).

---

**Comparator**

A circuit that compares two binary inputs and indicates if they are equal are called comparator.

By definition, an XOR gate is a one-bit comparator. Multiple XOR gates can be used to perform multiple bit comparator. For example, 4 XOR gates for a 4-bit comparator is shown in Figure 5-78.

How to determine if two inputs are equal?

---

**XOR and XNOR gates**

XOR (exclusive-or) is a 2 input gate whose output is 1 if exactly one of its input is 1.

XNOR (exclusive-nor) is a 2 input gate whose output is 1 if its input are the same.

Refer to figure 5-37 for logic diagrams and symbols for XOR and XNOR gates.

$74 \times 86$ SSI IC contains 4 XOR gates.

---

**Adder**

An adder performs arithmetic addition of two operands using the addition table. An adder whose two operands are 1-bit is called half-adder. An adder whose two operands are more than 1 bit are called full adder.
**Half Adder**

The inputs and output of a half adder are:

Inputs: the two operands $X$ and $Y$

Outputs:
1) sum ($S$)
2) carry out ($CO$)

$$S = X \oplus Y$$
$$CO = X \cdot Y$$

---

**Full Adder (cont’d)**

$$S = X \oplus Y \oplus CIN$$
$$COUT = X \cdot Y + X \cdot CIN + Y \cdot CIN$$

See figure 5-86 for its logic diagram and symbol.

---

**Full Adder**

A full adder handles one bit of binary addition. While performing bitwise addition, each bitwise addition has the following inputs and outputs:

Inputs:
- input operands $X$ and $Y$
- carry in ($CIN$)

Outputs:
- sum $S$
- carry out ($CO$)

---

**Ripple Adder**

A ripple adder is a cascade of $n$ full-adders, each of which handles one bit. See Figure 5-87 for 4-bit ripple adder. It can perform 4-bit addition.
**Full Subtractor**

A full subtractor handles 1 bit binary subtraction.

Inputs:
- operands $X$ and $Y$
- a borrow in (BIN)

Outputs:
- difference ($D$)
- a borrow out (BOUT)

---

**Ripple Subtractor**

We can similarly build a $n$ bit ripple subtractor by cascading $n$ full subtractors as shown in figure 5-88.

Note a full subtractor can be implemented with a full adder circuit plus inverters.

---

**Full Subtractor (cont’d)**

\[
D = X \oplus Y \oplus BIN
\]

\[
BOUT = X' \cdot Y + X' \cdot BIN + Y \cdot BIN
\]

This is very similar to the equations for full adder.