CHAPTER 4

Sequential Logic design principles

Logic circuits are defined into two types, “combinational” and “sequential.” A combinational logic circuit is one whose outputs depend only on its current inputs. A sequential logic circuit is one whose outputs depend not only on its current inputs, but also on the past sequence of inputs, possibly arbitrarily far back in time. So it is impossible to describe the until the current time.

The state of a sequential circuit is a collection of state variables whose values at any one time contain all the information about the past necessary to account for the circuit’s future behavior. State variables need not have direct physical significance, and there are often many ways to choose them to describe a given sequential circuit. In a digital circuit, state variables, are binary values, corresponding to certain logic signals in the circuit. A circuit with \( n \) binary state variables has \( 2^n \) possible states. As large it as it might be, \( 2^n \) is always finite, so sequential circuits are sometimes called finite-state machines.

The state changes of most sequential circuits occur at times specified sequential circuit by means of table that lists output as a function of the input sequence that has been received up by a free-running clock signal. By convention a clock signal is active high if the state changes occur at the clock’s rising edge or when the clock is HIGH, and active low in the complementary case. The clock period is the time between successive transitions in the same direction, and clock frequency is the reciprocal of the period. The first edge or pulse in a clock period or sometimes the period itself is called a clock tick. The duty cycle is the percentage of time that the clock signal is at its asserted level. Typical digital systems, from digital watches to supercomputers, use a quartz-crystal oscillator to generate a free-running clock. Clock frequencies range form 32.768kHz to 400Mhz. Typical systems using TTL and CMOS parts have clock frequencies in the 5-66 MHz range.

Majorities of practical discrete digital designs are either feedback sequential circuit or a clocked synchronous state machine. A feedback sequential circuit uses ordinary gates and feedback loops to obtain memory in a logic circuit, thereby creating sequential-circuit building blocks such as flip-flops and latches that are used in higher-level designs. A clocked synchronous state machine uses these building blocks, in particular edge-triggered D flip-flops, to create circuits whose inputs are examined and whose outputs change with respect to a controlling clock signal.

4.1 Bistable Elements

The simplest sequential circuit consists of a pair of inverters forming a feedback loop, as shown in figure 4.1 below. It has no inputs and two outputs, \( Q \) and \( \overline{Q} \).
4.2 Digital Analysis

Strict digital analysis for the circuits in figure 4.1 shows that it has two stable states. If Q is high the bottom inverter has a HIGH input and a LOW input, which forces the top inverter’s output HIGH. But if Q is low, then the bottom inverter has a LOW input and a HIGH output, which forces Q LOW, another stable situation. Using a single state variable Q it is possible to describe the circuit to exhibit two possible states Q=0 and Q=1.

The bistable element has no inputs. So when power is applied to bistable circuit, it randomly comes in one state or the other and stays there forever.

4.3 Analog analysis.

Bistable exhibits three equilibrium points when an analysis of it is done from analog point of view. The two of the equilibrium points correspond to two stable digital states Q=0 (LOW) or Q=1 (HIGH). The third equilibrium point labeled metastable occurs in between the two stable states when the input voltage applied is not for sufficient duration. When bistable is in metastable state tiny noise at the input may force the circuit to either one of the stable digital states. All sequential circuits are subjected to metastable behavior.

4.4 Basic Sequential circuit building blocks

4.4.1 Latches and Flip-Flops.

Latches and flip-flops are basic building blocks of most sequential circuits. Typical digital system use latches and flip-flops prepackaged in functionally specific devices in a standard IC. In ASIC design environment, latches and flip-flops are typically
predefines cells specified by the ASIC vendor. However, within a standard IC or an ASIC, each latch or flip-flop is typically designed as a feedback sequential circuit using individual logic gates and feedback loops.

Flip-flop is a sequential device that samples its inputs and changes the outputs only at times determined by a clocking signal. On the other hand latch is a sequential device that watches all of its inputs continuously and changes its output at any time, independent of clocking signal.

4.4.2 S-R Latch

In S-R (set-reset) latch based on NOR gates is shown in Figure 4.2

![S-R Latch Circuit Diagram](image)

Figure 4.2 S-R latch circuit design using NOR gates

The circuit has two inputs, S and R, and two outputs, Q and /Q where /Q is the complement of Q.

The functional description of this latch for a typical sequence of inputs is given in table 4.1 as shown in the next page. The S input sets or presets the Q to 1; R resets or clears the Q output to 0. If S and R

<table>
<thead>
<tr>
<th>/S</th>
<th>/R</th>
<th>Q</th>
<th>/Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Last Q</td>
<td>Last /Q</td>
</tr>
</tbody>
</table>

Table 4.1 S-R Latch function table.

are both 0 the outputs Q and /Q retain their last state. When the inputs are both 1 the outputs Q and /Q are both 0.
4.4.3 Propagation Delay

The propagation delay is the time it takes for a transition on an input signal to produce a transition on an output signal. A given latch or flip-flop exhibits different propagation delay for each pair of input and output signals. With an S-R latch a LOW-to-High transition on S can cause a LOW-to-HIGH transition on Q which is denoted by the term $t_{pLH(SQ)}$. Similarly a LOW-to-HIGH transition on R can cause HIGH-to-LOW transition on Q with propagation delay $t_{pHL(RQ)}$.

Because of the propagation delay most latch will go into a metastable state if input pulse width less than or equal to the propagation delay is applied at its inputs.

4.4.4 S-R Latch with Enable

An S and R input is sensitive to its inputs all times. However, it maybe modifies to create a device that is sensitive to these inputs only when an enabling input C is asserted.
Figure 4.5 S-R latch logic symbol.

Such an S-R latch with an enabling input C is shown in figure 4.4. As shown by the function table 4.2, the circuit behaves like an S-R latch when C=1 and retains its previous state when C is 0. The logic symbol for this latch is shown in figure 4.5.

<table>
<thead>
<tr>
<th>S</th>
<th>R</th>
<th>C</th>
<th>Q</th>
<th>/Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Last Q</td>
<td>Last /Q</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>0</td>
<td>Last Q</td>
<td>Last /Q</td>
</tr>
</tbody>
</table>

Table 4.2 Function table for S-R latch with enable

4.4.5 D Latch

S-R latches are used in control applications where setting and resetting a flag is done to indicate a changed condition. However for applications that require the information to be stored as presented to the signal line, D-latch may be used.

As shown in figure 4.6 D latch is an S-R latch with an inverter added to generate S and R inputs from the single D (data) input. This eliminates the troublesome situation in S-R latches, where S and R may be asserted simultaneously. The control input of a D-latch labeled C in Figure 4.7 is sometimes named ENABLE, CLK, or G and is active low in some D-latch designs.

D latch’s functional behavior is given in table 4.3
Figure 4.6 D latch circuit design using NAND gates

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>Q</th>
<th>/Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>x</td>
<td>Last</td>
<td>Last</td>
</tr>
</tbody>
</table>

Table 4.3 D latch function table

Figure 4.7 D latch logic symbol

Although D-latch eliminates the $S = R = 1$ problem of the $S\_R$ latch, it does not eliminate metastability problem. If D input changes anytime during the setup and hold time window, the output of the latch is unpredictable and may become metastable.

4.4.6 T Flip-Flop

A T (toggle) flip-flop changes the state on every tick of the clock. Figure 4.8 shows the symbol of T flip-flop. A D flip-flop can be be modified to get a T flip-flop as shown in Figure 4.9.

In many applications of T flip-flops, the flip-flop need not be toggled on every clock tick. Such applications can use a flip-flop with enable.

4.8 Logic symbol of Positive edge triggered T flip-flop

Its symbol is shown in Figure 4.10.
Figure 4.9 Circuit design for a T flip-flop using a D flip-flop.

Figure 4.10 Logic symbol of Positive edge triggered T flip-flop with enable.

Figure 4.10 Circuit Diagram of Positive edge triggered T flip-flop with enable.