CHAPTER 9: ASYNCHRONOUS SEQUENTIAL CIRCUITS
Chapter Objectives

- Sequential circuits that are not synchronized by a clock – Asynchronous circuits
- Analysis of Asynchronous circuits
- Synthesis of Asynchronous circuits
- Hazards that cause incorrect behavior of a circuit
Asynchronous sequential circuits

- Synchronous sequential circuits
  - state variables: F/Fs
  - controlled by a clock
  - operate in pulse mode

- Asynchronous sequential circuits
  - do not operate in pulse mode
  - do not use F/Fs to represent state variables
  - Changes in state are dependent on whether each of inputs to the circuit has the logic level 0 or 1 at any given time

- To achieve reliable operation (focus on the simplest case)
  - the inputs to the circuit must change one at a time
  - there must be sufficient time between the changes in input signals to allow the circuit to reach a stable state

- A circuit that adheres to these constraints is said to operate in the fundamental mode
Asynchronous behavior

(a) Circuit with modeled gate delay

\[ Y = (y + S) + R \]

(b) State-assigned table

<table>
<thead>
<tr>
<th>Present state</th>
<th>Next state</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y )</td>
<td>( y ) ( y ) ( y ) ( y )</td>
</tr>
<tr>
<td>( SR = 00 )</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>( 0 )</td>
<td>0 0 1 0 0</td>
</tr>
<tr>
<td>( 1 )</td>
<td>1 0 1 0 0</td>
</tr>
</tbody>
</table>

stable state

two NOR gate delay
FSM model for the SR latch

(a) State table

<table>
<thead>
<tr>
<th>Present state</th>
<th>Next state</th>
<th>Output Q</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$SR = 00$</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>B A A A</td>
<td></td>
</tr>
</tbody>
</table>

(b) State diagram

Moore-type FSM
Synthesis of an asynchronous circuit

(a) State table

<table>
<thead>
<tr>
<th>Present state</th>
<th>Next state $SR = 00$</th>
<th>$01$</th>
<th>$10$</th>
<th>$11$</th>
<th>Output $Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$A$</td>
<td>$A$</td>
<td>$B$</td>
<td>$A$</td>
<td>$0$</td>
</tr>
<tr>
<td>B</td>
<td>$B$</td>
<td>$A$</td>
<td>$B$</td>
<td>$A$</td>
<td>$1$</td>
</tr>
</tbody>
</table>

(b) State-assigned table

\[
Y = \overline{R} \cdot (S + y) \\
= \overline{R} \cdot (S + y) \\
= (\overline{R} + (S + y)) \\
= (R + (S + y)) \\
z = y
\]
Mealy representation of the SR latch

(a) State table

<table>
<thead>
<tr>
<th>Present state</th>
<th>Next state</th>
<th>Output, Q</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SR = 00</td>
<td>00 01 10 11</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>0 0 – 0</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>1 – 1 –</td>
</tr>
</tbody>
</table>

(b) State diagram

\[ Y = \overline{R} \cdot (S + y) \]
\[ = \overline{R} \cdot (S + y) \]
\[ = (\overline{R} + (S + y)) \]
\[ = (R + (S + y)) \]
\[ Q = y \]
Terminology

- Asynchronous circuits
  - state table $\rightarrow$ flow table
  - state-assigned table $\rightarrow$ transition table or excitation table

- We will use the term *flow table and excitation table*
Analysis of Asynchronous Circuits
Analysis of Asynchronous circuits

(a) gated D latch

\[
Y = (C \cdot D) \cdot ((C \cdot \overline{D}) \cdot y)
\]
\[
= (C \cdot D) + ((C \cdot \overline{D}) \cdot y)
\]
\[
= CD + (\overline{C} + D) \cdot y
\]
\[
= CD + \overline{C}y + Dy
\]
\[
= CD + \overline{C}y
\]

(b) Excitation table

<table>
<thead>
<tr>
<th>Present state</th>
<th>Next state</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y )</td>
<td>( y )</td>
</tr>
<tr>
<td>( y )</td>
<td>( y )</td>
</tr>
<tr>
<td>( y )</td>
<td>( y )</td>
</tr>
<tr>
<td>( y )</td>
<td>( y ), 1</td>
</tr>
<tr>
<td>( y )</td>
<td>( y ), 0</td>
</tr>
</tbody>
</table>

(c) Flow table

<table>
<thead>
<tr>
<th>Present state</th>
<th>Next state</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A )</td>
<td>( A ), ( A ), ( A ), ( B )</td>
</tr>
<tr>
<td>( B )</td>
<td>( B ), ( B ), ( A ), ( B )</td>
</tr>
</tbody>
</table>

(d) State diagram

\( CD = 00 \), 01, 10, 11

\( 10 \), 11

\( 0x \), \( x0 \)

\( 0x \), \( x1 \)
Analysis of the circuit in example 9.3

\[ Y_1 = y_1 \bar{y}_2 + w_1 \bar{y}_2 + \bar{w}_1 \bar{w}_2 y_1 \]
\[ Y_2 = y_1 y_2 + w_1 y_2 + w_2 + \bar{w}_1 \bar{w}_2 y_1 \]
\[ z = \bar{y}_1 y_2 \]
Excitation and flow tables for the circuit in example 9.3

(a) Excitation table

<table>
<thead>
<tr>
<th>Present state $y_2y_1$</th>
<th>Nextstate $w_2w_1 = 00 01 10 11$</th>
<th>Output $z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_2y_1$</td>
<td>$Y_2Y_1$</td>
<td>$Y_2Y_1$</td>
</tr>
<tr>
<td>00</td>
<td>(00)</td>
<td>01</td>
</tr>
<tr>
<td>01</td>
<td>11</td>
<td>(01)</td>
</tr>
<tr>
<td>10</td>
<td>00</td>
<td>(10)</td>
</tr>
<tr>
<td>11</td>
<td>(11)</td>
<td>10</td>
</tr>
</tbody>
</table>

(b) Flow table

<table>
<thead>
<tr>
<th>Present state</th>
<th>Nextstate $w_2w_1 = 00 01 10 11$</th>
<th>Output $z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_2w_1$</td>
<td>$A$</td>
<td>$B$</td>
</tr>
<tr>
<td>$A$</td>
<td>$B$</td>
<td>$C$</td>
</tr>
<tr>
<td>$B$</td>
<td>$A$</td>
<td>$C$</td>
</tr>
<tr>
<td>$C$</td>
<td>$D$</td>
<td>$C$</td>
</tr>
</tbody>
</table>
Modified flow table for Example 9.3.

<table>
<thead>
<tr>
<th>Present state</th>
<th>Next state</th>
<th>Output z</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$w_2 \ w_1$ = 00 01 10 11</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>A B C -</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>D B - D</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>A C C C</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>D C C C</td>
<td>0</td>
</tr>
</tbody>
</table>
State table for Example 9.3
Flow table for a simple vending machine

<table>
<thead>
<tr>
<th>Present state</th>
<th>Next state</th>
<th>Output z</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$w_2 w_1 = 00$</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>D</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
<td>0</td>
</tr>
</tbody>
</table>

$w_2 \equiv$ dime $w_1 \equiv$ nickel
Steps in the Analysis Process

- Each feedback path is cut
  - A delay element is inserted at the point where the cut is made
  - A cut can be made anywhere in a particular loop formed by feedback connection, as long as there is only one cut per (state variable) loop
- Next-state and output expressions are derived from the circuit
- The excitation table is derived
- A flow table is obtained
- A corresponding state diagram is derived from the flow table if desired
Synthesis of Asynchronous Circuits
Synthesis of Asynchronous Circuits

- the same basic steps used to synthesize the synchronous circuits

- Devise a state diagram for an FSM
- Derive the flow table and reduce the number of states if possible
- Perform the state assignment and derive the excitation table
- Obtain the next-state and output expressions
- Construct a circuit that implements these expressions
Example: serial parity generator

- Serial parity generator
  - input \( w \): pulses are applied to \( w \)
  - output \( z \)
  - \( z=1 \) if the number of previously applied pulses is odd

(a) State diagram
Parity–generating asynchronous FSM

(a) State diagram

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next state $w = 0$</th>
<th>Next state $w = 1$</th>
<th>Output $z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>B</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>B</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>D</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>A</td>
<td>D</td>
<td>0</td>
</tr>
</tbody>
</table>

(b) Flow table
State assignment

- State assignment (a) has a major flaw
  - state D = 11: \( w=0 \rightarrow \) state A
  - \( y_2y_1=11 \rightarrow y_2y_1=00 \)
  - the values of the next-state variables determined by the networks of logic gates with varying delays
    - suppose \( y_1 \) changes first
      - \( y_2y_1=10 \rightarrow \) state C (10)
      - state C is stable when \( w=0 \)
    - suppose \( y_2 \) changes first
      - \( y_2y_1=01 \rightarrow \) state B (01)
      - try to change to \( y_2y_1=10 \) when \( w=0 \)
      - if \( y_1 \) changes first, \( y_2y_1=00 \)
    - race condition occurs

<table>
<thead>
<tr>
<th>Present state ( y_2y_1 )</th>
<th>Next state ( w=0 )</th>
<th>Next state ( w=1 )</th>
<th>Output ( z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>(00) 01</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>10 01</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10 11</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>00 11</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

(a) Poor state assignment

<table>
<thead>
<tr>
<th>Present state ( y_2y_1 )</th>
<th>Next state ( w=0 )</th>
<th>Next state ( w=1 )</th>
<th>Output ( z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>(00) 01</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>11 01</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>11 10</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>00 10</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

(b) Good state assignment
Circuit that implements the FSM

\[ Y_1 = w \overline{y}_2 + \overline{w} y_1 + y_1 \overline{y}_2 \]

\[ Y_2 = wy_2 + \overline{w} y_1 + y_1 y_2 \]

\[ z = y_1 \]

Synchronous solution

Asynchronous solution
Circuit that implements a parity-generating asynchronous FSM

- The asynchronous implementation is *more complex* than the synchronous one?

- It’s a negative-edge-triggered master/slave F/F
  - With the complement of its output connected to its D input
Master–slave D F/F (example 9.2)

- Analyze synchronous circuit as if it were an asynchronous circuit.
  - Actually all circuits are asynchronous

\[
Y = CD + \overline{C}y + Dy
\]

in the previous example of gated D - Latch

\[
Y_m = CD + \overline{C}y_m + Dy_m
\]

\[
Y_s = \overline{C}y_m + Cy_s + y_m y_s
\]

Circuit for the master-slave D flip-flop.
Excitation table for example 9.2

<table>
<thead>
<tr>
<th>Present state</th>
<th>Next state</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_m y_s$</td>
<td>$Y_m Y_s$</td>
<td>$CD$</td>
</tr>
<tr>
<td></td>
<td>00 01 10 11</td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>00 00 00 10</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>00 00 01 11</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>11 11 00 10</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>11 11 01 11</td>
<td>1</td>
</tr>
</tbody>
</table>

(a) Excitation table
Flow tables for Example 9.2

(b) Flow table

<table>
<thead>
<tr>
<th>Present state</th>
<th>Next state</th>
<th>Output Q</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CD = 00</td>
<td>0</td>
</tr>
<tr>
<td>S1</td>
<td>S1</td>
<td>0</td>
</tr>
<tr>
<td>S2</td>
<td>S1</td>
<td>1</td>
</tr>
<tr>
<td>S3</td>
<td>S1</td>
<td>0</td>
</tr>
<tr>
<td>S4</td>
<td>S1</td>
<td>1</td>
</tr>
</tbody>
</table>

(c) Flow Table with unspecified entries
State diagram for the master–slave D Flip/Flop
Parity generating FSM and Master–slave D F/F

\[ Y_1 = w\bar{y}_2 + \bar{w}y_1 + y_1\bar{y}_2 \]
\[ Y_2 = w y_2 + \bar{w}y_1 + y_1y_2 \]
\[ z = y_1 \]

\[ Y = CD + \bar{C}y + Dy \]
in the previous example of gated D - Latch

\[ Y_m = CD + \bar{C}y_m + Dy_m \]
\[ Y_s = \bar{C}y_m + Cy_s + y_m y_s \]

\[ y_1 = y_m, y_2 = y_s \]
\[ w = C, \bar{y}_2 = D, \]
\[ z = y_1 = y_m \]
Hazard and Glitches
Hazards and glitches

In asynchronous circuits
- undesirable glitches on signals should not occur
- hazards
  - the glitches cause by the structure of a given circuit and propagation delays in the circuit
- two types of hazards
  - static
    - the signal undergoes a momentary change in its required value
  - dynamic
    - when a signal is supposed to change from 1 to 0 or from 0 to 1
    - a change involves a short oscillation before the signal settles into its new level
Definition of hazards

(a) Static hazard

(b) Dynamic hazard
Hazards and glitches

- **Usual solutions**
  - wait until signals are stable by using a clock
    - preferable
    - easiest to design when there is a clock
    - synchronous circuits
  - design hazard-free circuits
    - sometimes necessary
    - asynchronous design
Static hazards

\( f = x_1 x_2 + \overline{x}_1 x_3 \)

\( f = x_1 x_2 + \overline{x}_1 x_3 + x_2 x_3 \)

(a) Circuit with a hazard

(b) Karnaugh map

(c) Hazard-free circuit

hazard-free if more than one bit of inputs change simultaneously?
Two-level implementation of master–slave D flip–flop

<table>
<thead>
<tr>
<th>Present state $y_m\ y_s$</th>
<th>Next state $CD$ = 00 01 10 11</th>
<th>$Y_m\ Y_s$</th>
<th>Output $Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00 00 00 10</td>
<td>00 10</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>00 00 01 11</td>
<td>00 11</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>11 11 00 10</td>
<td>11 01</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>11 11 11 11</td>
<td>11 11</td>
<td>1</td>
</tr>
</tbody>
</table>

(a) Excitation table

(b) Karnaugh maps for $Y_m$ and $Y_s$ in Figure 9.6a

$Y_m = CD + \overline{C}y_m + Dy_m$

$Y_s = \overline{C}y_m + Cy_s + y_m y_s$
Two-level implementation of master–slave D flip-flop (2)

(a) Minimum-cost circuit

\[ Y_m = CD + \overline{C}y_m + Dy_m \]
\[ Y_s = \overline{C}y_m + Cy_s + y_my_s \]

(c) Hazard-free circuit
Static hazard in a POS circuit (0-hazard)

(a) Circuit with a hazard

(b) Karnaugh map

(c) Hazard-free circuit
dynamic hazards

- there exist multiple paths for a given signal change to propagate along
- neither easy to detect nor easy to deal with
- using two-level hazard-free circuits

(a) Circuit

(b) Timing diagram

One gate delay
CLOCK SYNCHRONIZATION (CHAPTER 10.3)
Clock skew

- the clock signal arrives at different times at different F/Fs
  - with or without clock enable circuits
  - wires whose lengths vary appreciably
An H-tree clock distribution network
F/F timing parameters

- setup time $t_{su}$
- hold time $t_h$
- register delay or propagation delay $t_{rd}$
- output delay time $t_{od}$
  - required for the change in $Q$ to propagate to an output pin on the chip

A flip-flop in an integrated circuit
F/F timing parameters, cont’d

- $t_{co}$ delay: active clock edge $\rightarrow$ output change at an output pin
  - $t_{\text{Clock}} + t_{rd} + t_{od}$
  - Example
    - $t_{\text{Clock}} = 1.5\,\text{ns}$, $t_{rd}=1\,\text{ns}$, $t_{od}=2\,\text{ns} \rightarrow t_{co} = 4.5\,\text{ns}$

- F/F timing in a chip
  - $t_{\text{Clock}}=1.5\,\text{ns}$, $t_{\text{Data}}=4.5\,\text{ns}$, $t_{su}=3\,\text{ns}$

![Diagram showing F/F timing parameters with an example of setup time violation]
Metastability and Asynchronous Inputs

Asynchronous Inputs Are Dangerous!

Since they take effect immediately, glitches can be disastrous.

Synchronous inputs are greatly preferred!

But sometimes, asynchronous inputs cannot be avoided e.g., reset signal, memory wait signal
Never allow asynchronous inputs to be fanned out to more than one FF within the synchronous system.
Metastability and Asynchronous Inputs

What Can Go Wrong

In is asynchronous Fans out to D0 and D1 One FF catches the signal, one does not

impossible state might be reached!

Single FF that receives the asynchronous signal is a synchronizer
Metastability and Asynchronous Inputs

When FF input changes close to clock edge, the FF may enter the *metastable* state: neither a logic 0 nor a logic 1

It may stay in this state an indefinite amount of time, although this is not likely in real circuits

Small, but non-zero probability that the FF output will get stuck in an in-between state

Oscilloscope Traces Demonstrating Synchronizer Failure and Eventual Decay to Steady State
Metastability and Asynchronous Inputs

Solutions to Synchronizer Failure

- the probability of failure can never be reduced to 0, but it can be reduced
- slow down the system clock
  this gives the synchronizer more time to decay into a steady state
  synchronizer failure becomes a big problem for very high speed systems
- use fastest possible logic in the synchronizer
  this makes for a very sharp "peak" upon which to balance
  S or AS TTL D-FFs are recommended
- cascade two synchronizers