Sequential Networks

- Have memory (state)
  - Present state depends not only on the current input, but also on all previous inputs (history)
  - Future state depends on the current input and state

\[
X = x_1, x_2, \ldots, x_n
\]
\[
Q = Q_1, Q_2, \ldots, Q_k
\]
\[
Z(t) = F(X(t), Q(t))
\]
\[
Q(t+1) = G(X(t), Q(t))
\]

Flip-flops are commonly used as storage devices: D-FF, J-K FF, T-FF

Review: Clocked D Flip-Flop with Rising-edge Trigger

Next state

The next state in response to the rising edge of the clock is equal to the D input before the rising edge

Review: Clocked J-K Flip-Flop

Next state

\[ Q^* = JQ' + K'Q \]

J-K = 00 -> no state change occurs
J-K = 10 -> the flip-flop is set to 1, independent of the current state
J-K = 01 -> the flip-flop is always reset to 0
J-K = 11 -> the flip-flop changes the state \[ Q^* = Q \]

Review: Clocked T Flip-Flop

Next state

\[ Q^* = QT + Q' = Q \]

T = 1 -> the flip-flop changes the state \[ Q^* = Q \]
T = 0 -> no state change
Review: S-R Latch, Transparent D-Latch

Mealy Sequential Networks

General model of Mealy Sequential Network

(1) X inputs are changed to a new value
(2) After a delay, the Z outputs and next state appear at the output of CM
(3) The next state is clocked into the state register and the state changes

An Example: 8421 BCD to Excess3 BCD Code Converter

State Graph and Table for Code Converter

State Assignment Rules

Transition Table
Sequential Network Timing

- Code converter
  - \( X = 0010_{1001} \Rightarrow Z = 1110_{0011} \)
  - Changes in \( X \) are not synchronized with active clock edge \( \Rightarrow \) glitches (false output), e.g. at \( t_b \)

Sequential Network Timing (cont’d)

Timing diagram assuming a propagation delay of 10 ns for each flip-flop and gate
(State has been replaced with the state of three flip-flops)

Setup and Hold Times

- For a real D-FF
  - \( D \) input must be stable for a certain amount of time before the active edge of clock cycle \( \Rightarrow \) Setup time
  - \( D \) input must be stable for a certain amount of time after the active edge of the clock \( \Rightarrow \) Hold time

- Propagation time: from the time the clock changes to the time the output changes

Maximum Clock Frequency

- \( t_c \) max - Max propagation delay through the combinational network
- \( t_p \) max - Max propagation delay from the time the clock changes to the flip-flop output changes (\( = \) max(\( t_{plh}, t_{phl} \))
- \( t_{ck} \) - Clock period

\[
\frac{t_c}{t_{ck}} + \frac{t_p}{t_{ck}} + \frac{t_{su}}{t_{ck}} \leq 1
\]

Example:

- \( t_{ck} \) = 2 \( \times \) 15 + 15 + 5 = 50 ns
- \( t_c \) max = \( \frac{1}{50} \) = 20 MHz
Hold Time Violation

- Occur if the change in Q fed back through the combinational network and cause D to change too soon after the clock edge.

Hold time is satisfied if:
\[ t_{H} \text{ min} + t_{Z} \text{ min} \geq t_{H} \]

What about X?

Make sure that input changes propagate to the flip-flops inputs such that setup time is satisfied.

Make sure that X does not change too soon after the clock.

If X changes at time \( t_y \) after the active edge, hold time is satisfied if
\[ t_{H} \geq t_{H} - t_{EX} \text{ min} \]

Moore Sequential Networks

Outputs depend only on present state!

\[ X = x_1, x_2, \ldots, x_n \]
\[ Q = Q_1, Q_2, \ldots, Q_k \]
\[ Z(t) = F(Q(t)) \]
\[ Q(t^+) = G(X(t), Q(t)) \]

General Model of Moore Sequential Machine

Outputs depend only on present state!

\[ X = x_1, x_2, \ldots, x_n \]
\[ Q = Q_1, Q_2, \ldots, Q_k \]
\[ Z = z_1, z_2, \ldots, z_m \]

\[ Z(t) = F(Q(t)) \]
\[ Q(t^+) = G(X(t), Q(t)) \]

Code Converter: Moore Machine

Do we need state S0?

How many states does Moore machine have?

How many states does Mealy machine have?
Moore Machine Timing

- X = 0010_1001 = Z = 1110_0011

Moore

Mealy

Moore Machine: Another Example

A Converter for Serial Data Transmission: NRZ-to-Manchester

- Coding schemes for serial data transmission
  - NRZ: nonreturn-to-zero
  - NRZI: nonreturn-to-zero-inverted
    - 0 in input sequence – the bit transmitted is the same as the previous bit;
    - 1 in input sequence – transmit the complement of the previous bit
  - RZ: return-to-zero
    - 0 – 0 for full bit time; 1 – 1 for the first half, 0 for the second half
  - Manchester

Moore Network for NRZ-to-Manchester

State Assignments

Guidelines to reduce the amount of combinational logic:
1. States which have the same next state and the same input should be given adjacent assignments (most common in Moore machines).
2. States which are the next states of the same state should be given adjacent assignments (next state).
3. States which have the same input for a given output should be given adjacent assignments.

Rule I: (S0, S1, S3, S4, S5, S6, S7, S6, S9, S10)
Rule II: (S0, S1, S3, S4, S5, S6, S7, S8, S9, S10)
Rule III: (S0, S1, S3, S4, S5, S6, S8, S9)
(S1, S3, S5, S7, S10)

S0 – 0010
S1 – 0111
...
S10 - 0100

Synchronous Design

- Use a clock to synchronize the operation of all flip-flops, registers, and counters in the system
  - all changes occur immediately following the active clock edge
  - clock period must be long enough so that all changes flip-flops, registers, counters will have time to stabilize before the next active clock edge
- Typical design: Control section + Data Section

Sequential machine to generate control signals
timed by operation of data section
An Example

- Data section // s = n*(n+a) //
  R1=n, R2=a // R1=s
- Design flowchart for SMUL operation
- Design Control section
  S0 S1 F
  0  0  B
  0  1  B+C0
  1  0  B+C0
  1  1  A+B

Timing Chart for System with Falling-edge Devices

Timing Chart for System with Rising-edge Devices

Principles of Synchronous Design

- Method
  - All clock inputs to flip-flops, registers, counters, etc., are driven directly from the system clock or from the clock ANDed with a control signal
- Result
  - All state changes occur immediately following the active edge of the clock signal
- Advantage
  - All switching transients, switching noise, etc., occur between the clock pulses and have no effect on system performance

Asynchronous Design

- Disadvantage - More difficult
  - Problems
    - Race conditions: final state depends on the order in which variables change
    - Hazards
  - Special design techniques are needed to cope with races and hazards
- Advantages = Disadvantages of Synchronous Design
  - In high-speed synchronous design propagation delay in wiring is significant => clock signal must be carefully routed so that it reaches all devices at essentially same time
  - Inputs are not synchronous with the clock => need for synchronizers
  - Clock cycle is determined by the worst-case delay

To Do

- Read
  - Textbook chapters 1.6, 1.7, 1.8, 1.10, 1.11, 1.12