Digital Signal Processing

Introduction

- Sequential logic
  - combinational logic + memory
  - output \( \sim f(\text{current inputs, current states}) \)
  - states: binary information stored in the memory elements at any given time

- Flip Flop
  - basic memory element made up of logic gates
    - one or more inputs
    - two outputs (\(Q, \bar{Q}\))
    - two output states
      - Set, Reset(Clear)
**Digital Signal Processing**

**R-S (S-C) F/F using NAND gates**

- Set/Clear(Reset)
- to keep previous values when $S = R = 1$ (memory)

<table>
<thead>
<tr>
<th>Set</th>
<th>Reset</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>No change</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>$Q=1$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$Q=0$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>$\overline{Q} = \overline{Q}$</td>
</tr>
</tbody>
</table>

**R-S (S-C) F/F using NOR gates**

- to keep previous values when $S = R = 0$ (memory)
- invalid state ($Q = \overline{Q}$) when $S = R = 1$

<table>
<thead>
<tr>
<th>Set</th>
<th>Reset</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>No change</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$Q=1$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>$Q=0$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$\overline{Q} = \overline{Q}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>$A + B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Debouncing circuit

- Phenomenon of contact bounce

Debouncing circuit

- debouncing circuit

<table>
<thead>
<tr>
<th>Set</th>
<th>Reset</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>No change</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Q=1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Q=0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>Q = \bar{Q} = 1</td>
</tr>
</tbody>
</table>

Clock signals

- Asynchronous system
  - the outputs of logic circuits can change state whenever one or more inputs change

- Synchronous system
  - the exact time at which any output can change states are determined by a signal, called CLOCK

- CLOCK
  - rectangular pulse train or square wave

Positive going edge

Negative-going edge

Duty cycle = \frac{T_1}{T}
Triggering of F/F

- To trigger F/F
  - to switch the state of F/F by a change in the input
- level trigger: to trigger when a signal is at level 0 or 1
  - to cause a racing (multiple transition) problem
    (two or more state change during triggering signal activated)
  - a group of F/Fs sensitive to pulse duration is called a **latch**
- edge trigger: states are changed during signal transition
  - positive edge triggered
  - negative edge triggered
  - a group of F/Fs sensitive to pulse transition is called a **register**

Implementation of edge triggered F/F

- Master-slave F/F
- using an edge detector

Clocked R-S(S-C) F/F

Asynchronous F/F \(\rightarrow\) synchronous F/F

- when CP=1, outputs change according to the inputs
- when CP=0, outputs remains their previous states
\(\rightarrow\) level-triggered F/F

\[\begin{array}{cccc|c}
\text{Set} & \text{Reset} & \text{CP} & \text{Q} \\
0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 \\
\end{array}\]
Digital Signal Processing

Master-Slave F/F

Act as if it is a negative edge triggered F/F
- master F/F: normal F/F behavior
  - triggered when CP=1
- slave F/F: copy the results of master F/F
  - triggered when CP=0

Digital Signal Processing

Edge detector for clocked F/F

- Edge detector
  - an RC circuit to generate a spike in response to a momentary change of input signal (HPF)
  - using propagation delay of logic gates
    - duration of CLK pulses is typically 2-5 nanoseconds.
Digital Signal Processing

Edge triggered S-C(R-S) F/F

- Internal circuitry

![Diagram of S-C(R-S) F/F]

- F/F responding to positive(negative) edge

<table>
<thead>
<tr>
<th>S</th>
<th>C</th>
<th>CLK</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>x</td>
<td>$Q_0$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>↑</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>↑</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>↑</td>
<td>Invalid</td>
</tr>
</tbody>
</table>

Digital Signal Processing

Clocked J-K F/F

To convert invalid state in R-S F/F to toggle mode

<table>
<thead>
<tr>
<th>S</th>
<th>C</th>
<th>CLK</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>x</td>
<td>$Q_0$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>↑</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>↑</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>↑</td>
<td>Invalid</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>CLK</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>x</td>
<td>$Q_0$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>↑</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>↑</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>↑</td>
<td>$\bar{Q}_0$</td>
</tr>
</tbody>
</table>
Clocked D F/F

- To avoid invalid state by connecting S and C
- only one control input D, which stands for data
- used for parallel transfer of binary data
  - Q takes on the value of D only at certain time instance
  - Q is then used for subsequent processing

\[
\begin{array}{c}
D \quad S \quad R \\
Q \quad \bar{Q} \quad \bar{Q}
\end{array}
\]

- D latch (transparent latch) : level-triggered D F/F
  - when CP=0, the outputs are latched to their current level
  - when CP=1, the outputs will look exactly like D (transparent)

\[
\begin{array}{c}
D \quad Q \\
CP \quad \bar{Q}
\end{array}
\]

Asynchronous inputs

- Synchronous inputs : control inputs such as D, J, K, S, C
  - the effect on the F/F output is synchronized with the CP input
- asynchronous inputs (direct inputs)
  - the output is directly affected, regardless of the conditions at other inputs including CP
  - PRESET and CLEAR

\[
\begin{array}{c}
\text{CLR} \quad \text{PRE} \quad \text{CLK} \quad J \quad K \quad Q \quad Q'
\end{array}
\]

<table>
<thead>
<tr>
<th>CLR</th>
<th>PRE</th>
<th>CLK</th>
<th>J</th>
<th>K</th>
<th>Q</th>
<th>Q'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>↑</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>↑</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>↑</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>non</td>
</tr>
</tbody>
</table>

Invalid

memory

toggle
Timing considerations

- **Setup time** $t_s$ (Hold time $t_H$)
  - the time interval immediately preceding (following) the active transition of the CLK signal during which the control input must be maintained at the proper level

- **propagation delay**
  - the delay from the time the signal is applied to the time when output makes its change

- **Maximum clocking frequency** $f_{\text{MAX}}$
  - the highest frequency that may be applied to the CLK
  - $f = 1/T$, where $T$ is the period

- **clock pulse HIGH and LOW times**
  - the minimum time duration that the CLK signal must remain its HIGH (or LOW) level

- **CLK transition time**
  - time duration during which the CLK goes from LOW to HIGH, or vice versa
  - $\leq 50$ nS for TTL devices
  - $\leq 200$ nS for CMOS devices
Clock skew

- Each FFs are triggered by the same CLK signal.

- Clock skew:
  - Because of propagation delay, a clock signal may arrive at individual FFs at different times.
  - Remedy: Equalizing the delays in various paths of clock signal.

Timing example in Data sheet
**F/F synchronization**

- To synchronize the behavior of an asynchronous input, actuated by human, with the clock input

  ![Debouncing switch diagram]

- preventing the appearance of partial pulse, X contains only complete pulses

  ![Debouncing switch diagram with clock pulses]

**Parallel data transfer**

- Data transfer:
  - the transfer of data from one F/F to another

- synchronous transfer
  - the data is transferred at the time when the CLK is asserted

- asynchronous transfer
  - the data transfer is performed independently of CLK input
  - level-triggered

- parallel data transfer
  - the contents of group of F/Fs are transferred simultaneously into other F/Fs
Serial data transfer

- Serial transfer
  - the contents of the register is transferred to another register one bit at a time
- Serial register
  - a group of F/Fs arranged so that the bits in the F/Fs are shifted from one F/F to the next for every clock pulse

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
</tr>
</thead>
<tbody>
<tr>
<td>before pulse applied</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>after first pulse</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>after second pulse</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>after third pulse</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Frequency divider
- division of a frequency by \(1/2^N\) using \(N\) flip-flops

Binary counter:
- a circuit that makes an increment by 1 at the occurrence of each clock pulse

Clock state transition table

<table>
<thead>
<tr>
<th>Q2</th>
<th>Q1</th>
<th>Q0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Counter(2/2)

- State transition diagram
  - to show how the states of the flip-flops change with each clock pulse
  - a circle represents the state
  - arrows indicate the state transition paths

- Mod number
  - # of states in the counting sequence
  - if $N$ flip-flops are connected, it would be an MOD-$2^N$ counter which counts from 0 to $2^N-1$ repeatedly
  - ex) MOD-6 counter
    - 1, 2, 3, 4, 5, 6, 1, 2, 3,……
    - 0, 1, 2, 3, 4, 5, 0, 1, 2,… …

Schmitt-trigger devices

- Schmitt-trigger devices
  - designed to accept slow-changing signals
  - produce an output that has oscillation-free transition
One-Shot

- **Multivibrator**
  - bistable multivibrator: two stable states (flip-flop)
  - monostable multivibrator (one shot)
    - only one stable states
    - once triggered, it remains in the quasi-stable state for a fixed period of time $t_p$
    - non-retriggerable one-shot
      - ignore triggering pulse during quasi-stable state
    - retriggerable one-shot
      - retriggered while it is in the quasi-stable state, and it will begin a new $t_p$ interval

Clock generator

- **Schmitt-trigger Oscillator**
- **555 Timer**
- **Crystal-controlled clock generators**
  - using CMOS 74HC04
    - $R = 100\, \Omega$
    - $f \leq 10\, \text{MHz}$
  - using 74LS04
    - $R = 300-1500\, \Omega$
    - $f \leq 20\, \text{MHz}$
Digital Signal Processing

HomeWork #4

- 5-3, 5-9, 5-12, 5-22, 5-25, 5-26
- 5-30, 5-33, 5-34, 5-58