I. Motivation & Examples

- Output depends on current input and past history of inputs.
- “State” embodies all the information about the past needed to predict current output based on current input.
  - State variables, one or more bits of information.
- If the current State of the circuit is known at time t, what is the state of the circuit at time (t+1)?
  Answer: the next state depends on current state and input
Sequential logic networks

I. Motivation & Examples

Describing sequential circuit

- State table
  - For each current-state, specify next-states as function of inputs
  - For each current-state, specify outputs as function of inputs

- State diagram
  - Graphical version of state table

Example 1: TV channel control
- Let the channel # represent the state of the circuit
- Input are up/down on the channel control

Example diagram:

- States 1, 2, 3, 4, ..., 99
- Transitions:
  - u: up
  - d: down

on

1 ——> u ——> 2 ——> u ——> 3 ——> u ——> 4 ——> u ——> ...

1 ——> d ——> 2 ——> d ——> 3 ——> d ——> 4 ——> d ——> 99 ——> d
Sequential logic networks

I. Motivation & Examples

Example 2: A sequential process that inputs an n-bit binary string and outputs 1 if the string contains an even number of 1’s

What represents the state of the circuit?

- Case 1:
  State as the number of 1’s read so far (possibly infinite # of states)

- Case 2: Two states E and O
  - E (even): if the # of 1’s read so far is even
  - O (odd) if the # of 1’s read so far is odd
I. Motivation & Examples

Example 2: State Diagram for Case 1

Example 2: State Diagram for Case 2
Example 2: State Diagram for Case 2

- Better design
- Has less states
Example 3: Discuss sequential n-bits comparator

- Compare two n-bits numbers $X=[X_{n-1}, \ldots, X_0]$, $Y=[Y_{n-1}, \ldots, Y_0]$
- Output 1 if $X > Y$
- Use the basic 1-bit comparator designed in class

Operation controlled by a clock to decide:
- when to shift input data
- when output $F_i$ is stable
Example 4: Discuss sequential n-bits adder

- Add two n-bits numbers $X=[X_{n-1}, \ldots, X_0]$, $Y=[Y_{n-1}, \ldots, Y_0]$
- Output $S=X+Y$ where $[S_{n}, S_{n-1}, \ldots, S_0]$
- Use the basic 1-bit adder with carry in and carry out

Operation controlled by a clock to decide:
- when to shift input data
- when output are ready
II. General Representation

Clock signals

- Sequential circuit are controlled by a clock signal
- Very important with most sequential circuits
  - State variables change state at clock edge.

![Clock Signal Diagram](image)
II. General Representation

General diagram of sequential circuit

- Sequential circuit are controlled by a clock signal
- Very important with most sequential circuits
  - State variables change state at clock edge.

Input

- i0
- i1
- ... in

Feedback

Current states

SLN

Memory components

Next states

State variables: s0, s1, ... sk

Output

- o0
- o1
- ... om
II. General Representation

Some important questions

- How to represent the states of a sequential circuit?
- How to memorize the (current and next) states?
- How to determine the next of the circuit?
- How to determine the outputs
  - as a function $F(\text{state})$ of current state only?
  - as a function $F(\text{input, state})$ of both input and current state?

☛ The concept of STATE is very important
II. General Representation

Memory component

- How do we represent the states?
- Memory component are used as state variables
  - Goal: Memorize the current state of the circuit
  - How are memory components implemented?
    - Latch, Flip-flop are 1-bit memory component
III. Basic memory component

Bistable element

- The simplest sequential circuit
- Two states
  - One state variable, say, Q (QN or Q_L the complement of Q)
III. Basic memory component

Bistable element

- The simplest sequential circuit
- Two states
  - One state variable, say, Q

![Bistable Element Diagram]
Sequential logic networks

III. Basic memory component

Bistable element: Analog analysis

- Assume pure CMOS thresholds, 5V rail
- Theoretical threshold center is 2.5 V
III. Basic memory component

Bistable element: Analog analysis

- Assume pure CMOS thresholds, 5V rail
- Theoretical threshold center is 2.5 V
Sequential logic networks

III. Basic memory component

Bistable element: Analog analysis

- Assume pure CMOS thresholds, 5V rail
- Theoretical threshold center is 2.5 V
II. General Representation

Bistable element: summary

- If \(Q=0\), then input to Not gate 2 is 0
  
  \[\Rightarrow\] Output of Not gate 2 is 1 \((Q_L = 1)\)
  
  \[\Rightarrow\] The input of Not gate 1 is 1, so output of Not gate 1 is 0
  
  \[\Rightarrow\] Stable output \((Q=0)\) and \((Q_L = 1)\)

- If \(Q=1\), then input to Not gate 2 is 1
  
  \[\Rightarrow\] Output of Not gate 2 is 0 \((Q_L = 0)\)
  
  \[\Rightarrow\] The input of Not gate 1 is 0, so output Not gate 1 is 0
  
  \[\Rightarrow\] Stable output \((Q=1)\) and \((Q_L = 0)\)
Sequential logic networks

III. Basic memory component

S-R Latch....

- How to control it?
  - Screwdriver
  - Control inputs
- S-R latch

Contradiction!!!!

<table>
<thead>
<tr>
<th>S</th>
<th>R</th>
<th>Q</th>
<th>QN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>last Q</td>
<td>last QN</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
III. Basic memory component

S-R Latch

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

Set operation: SR 00 ----> 10, set the device output to Q=1 regardless of current value of Q
Reset operation: SR 00 ----> 01, set the device output to Q=0 regardless of current value of Q
Hold operation: SR 10 ----> 00 or 01 ----> 00,
Device output are the same as last output values

- Only one input value changes
- Possible input changes:
  - SR: 00 ---> 01 ---> 00 ---> 10 ---> 00 ....
  - Input SR = 11 is not allowed (Both NOR gates output 0, i.e. Q=Q’=0)
Sequential logic networks

III. Basic memory component

S-R latch operation
III. Basic memory component

S-R latch timing parameters

- Propagation delay
- Minimum pulse width

![Diagram of S-R latch timing parameters with labels for propagation delay and minimum time to maintain signal at 1.]
Sequential logic networks

S-R latch symbols
III. Basic memory component

S-R latch with enable

<table>
<thead>
<tr>
<th>S</th>
<th>R</th>
<th>C</th>
<th>Q</th>
<th>QN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>last Q</td>
<td>last QN</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 QN</td>
</tr>
</tbody>
</table>
III. Basic memory component

Sequential network architecture (revisited)

Operation rules:
• Memory components $M_i$ must be in stable state before input changes
• Only one input of the component $M_i$ can change at a time
### Characteristic equation of S-R latch

**Definition:** The characteristic equation specifies a flip-flop next state as a function of its current state and inputs.

**Notation:** Let $q$ represent the current state of the flip-flop and $Q$ its next state.

#### Characteristics table

<table>
<thead>
<tr>
<th>S R q</th>
<th>Q</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>0</td>
<td>$\rightarrow$ Q=q Hold</td>
</tr>
<tr>
<td>0 0 1</td>
<td>1</td>
<td>$\rightarrow$ Q=0 Reset</td>
</tr>
<tr>
<td>0 1 0</td>
<td>0</td>
<td>$\rightarrow$ Q=1 Set</td>
</tr>
<tr>
<td>0 1 1</td>
<td>0</td>
<td>$\rightarrow$ Not allowed</td>
</tr>
<tr>
<td>1 0 0</td>
<td>1</td>
<td>$\rightarrow$ Q=1 Set</td>
</tr>
<tr>
<td>1 0 1</td>
<td>1</td>
<td>$\rightarrow$ Not allowed</td>
</tr>
<tr>
<td>1 1 0</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>1 1 1</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Sequential logic networks

III. Basic memory component

Characteristics equation of S-R latch

Use the characteristics table to get an excitation map of the flip flop

**Characteristics table**

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

Use K-map method to derive the characteristics equation:

\[ Q = S + R'q \]
III. Basic memory component

Excitation table of SR flip flop

The excitation table describes the input values of S and R that cause the corresponding transitions (q --> Q) from current to next state.

Types of transitions: q --> Q

0 --> 0
0 --> 1
1 --> 0
1 --> 1

Excitation table of S R latch

<table>
<thead>
<tr>
<th>q --&gt; Q</th>
<th>S</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 --&gt; 0</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>0 --&gt; 1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1 --&gt; 0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1 --&gt; 1</td>
<td>X</td>
<td>0</td>
</tr>
</tbody>
</table>

0 0 to hold current value
OR
0 1 to set Q=1
1 0 to reset
Q=0
III. Basic memory component

D latch

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>D</td>
<td>Q</td>
<td>QN</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>x</td>
<td>last Q</td>
<td>last QN</td>
<td></td>
</tr>
</tbody>
</table>

Sequential logic networks
Sequential logic networks

III. Basic memory component

D-latch operation
D-latch timing parameters

- Propagation delay (from C or D)
- Setup time (D before C edge)
- Hold time (D after C edge)
Sequential logic networks

Edge-triggered D flip-flop behavior

<table>
<thead>
<tr>
<th>D</th>
<th>CLK</th>
<th>Q</th>
<th>QN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</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>x</td>
<td>0</td>
<td>last Q</td>
<td>last QN</td>
</tr>
<tr>
<td>x</td>
<td>1</td>
<td>last Q</td>
<td>last QN</td>
</tr>
</tbody>
</table>
Sequential logic networks

III. Basic memory component

Edge-triggered D flip-flop behavior

![D Flip-flop Diagram]

<table>
<thead>
<tr>
<th>D</th>
<th>CLK</th>
<th>Q</th>
<th>QN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</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>x</td>
<td>0</td>
<td>last Q</td>
<td>last QN</td>
</tr>
<tr>
<td>x</td>
<td>1</td>
<td>last Q</td>
<td>last QN</td>
</tr>
</tbody>
</table>
### III. Basic memory component

**D flip-flop timing parameters**

- Propagation delay (from CLK)
- Setup time (D before CLK)
- Hold time (D after CLK)

![D flip-flop timing parameters diagram]