Week 13

CS2100 - L21b - Sequential Logic (Circuit Construction)

21b.1 - Analysis

- State tables
- State diagrams

21b.2 - Design

- Excitation tables
- Design procedure
- Unused states

21b.1 - Analysis

Sequential Circuits Analysis: Overview

- To analyze a sequential circuit:
  - Need to derive its ***state table*** (can be summarized as a ***state diagram***)
- General Approach:
  - Derive ***state equations*** for the flip-flop inputs AND
  - Derive ***output functions*** for the circuit outputs (if any)
- Terminology:
  - For a flip-flop **A**, ***A(t)*** and ***A(t+1)*** (or simply ***A*** and ***A$^+$***) represent the its **present state** and **next state**

State tables

State Table: Overview

- From the ***state equations*** and ***output function***, we derive the ***state table***, consisting of all possible binary combinations of present states and inputs.
- State table:
  - Similar to truth table
  - **Inputs** and **Present State** on the left side
  - **Outputs** and **Next State** on the right side
- m flip-flops and n inputs $\rightarrow 2^{m+n}$ rows.

Present State	Next State		Output	
	x=0	x=1	x=0	x=1
AB	A^+B^+	A^+B^+	y	y
00	00	01	0	0
01	00	11	1	0
10	00	10	1	0
11	00	10	1	0

Compact table

Alternative

Analysis Example: State Table

- ***State table*** for circuit of Figure 1:

n=1

m=2

2^{m+n} permutat-ions

Present State		Input	Next State		Output
A	B	x	A^+	B^+	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

State equations:

$A^+ = A \cdot x + B \cdot x$

$B^+ = A' \cdot x$

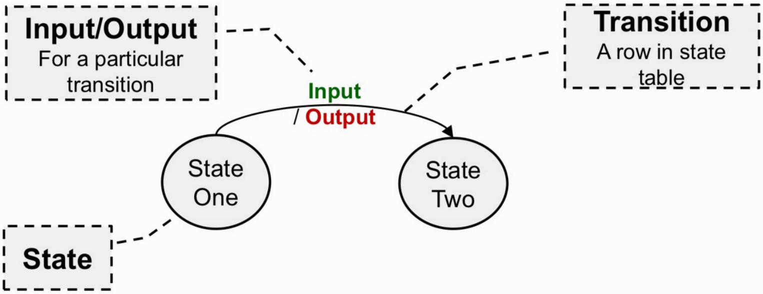
Output function:

$y = (A + B) \cdot x'$

State diagrams

State Diagram: Revisit

- From the ***state table***, we can draw the ***state diagram***:

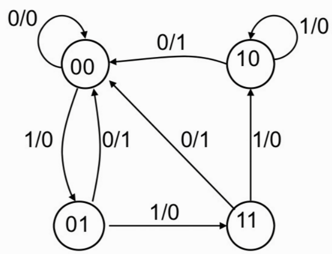


- Each combination of the flip-flop values represents a state. Hence, ***m* flip-flops → up to 2^m states**.

Analysis Example: State Diagram

- **State diagram** of the circuit of Figure 1:

Present State	Next State		Output	
	x=0	x=1	x=0	x=1
AB	A^+B^+	A^+B^+	y	y
00	00	01	0	0
01	00	11	1	0
10	00	10	1	0
11	00	10	1	0



21b.2 - Design

Analysis: vs Design:

Analysis:

- **Derive the behaviour** from a given circuit diagram
- Behaviour can be summarized by state table or state diagram
- Use ***characteristic tables***

Design:

- **Derive the logic circuit** from a given set of specifications
- Specification usually in the form of state equations, state table, or state diagram
- Use ***excitation tables***

Excitation tables

Flip-Flop Excitation Table

- **Excitation tables**:
  - Given the required transition from present state to next state, determine the flip-flop input(s)

J	*K*	*Q(t+1)*	Comments
0	0	*Q(t)*	No change
0	1	0	Reset
1	0	1	Set
1	1	*Q(t)'*	Toggle

Characteristic Table

Q	Q^+	*J*	*K*
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Excitation Table

- Can you deduce the excitation table from characteristic table?

Flip-Flop Excitation Tables

Q	Q^+	*J*	*K*
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

JK Flip-flop

Q	Q^+	*S*	*R*
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

SR Flip-flop

Q	Q^+	*D*
0	0	0
0	1	1
1	0	0
1	1	1

D Flip-flop

Q	Q^+	*T*
0	0	0
0	1	1
1	0	1
1	1	0

T Flip-flop

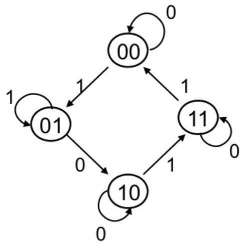
Design procedure

Sequential Circuit: Design Procedure

- **Steps:**
  - Start with **circuit specifications** – description of circuit behaviour, usually a state diagram or state table
  1. Derive the **state table**:
     - Perform state reduction if necessary
     - Perform state assignment if necessary
  2. Determine number of flip-flops and label them
  3. Choose the type of flip-flop to be used
  4. Derive circuit excitation and output tables from the state table
  5. Derive circuit output functions and flip-flop input functions
  6. Draw the logic diagram

Design Example #1: State Table

- Circuit state/excitation table, using *JK* flip-flops.



Present state		Input	Next state		Flip-flop inputs			
A	*B*	*x*	A^+	B^+	*JA*	*KA*	*JB*	*KB*
0	0	0	0	0	0	x	0	x
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

Q	Q^+	*J*	*K*
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

JK Flip-flop's excitation table.

Unused states

Design Example #3: Unused States

- Design involving **unused states**:

Present state			Input	Next state			Flip-flop inputs						Output
A	*B*	*C*	*x*	A^+	B^+	C^+	*SA*	*RA*	*SB*	*RB*	*SC*	*RC*	*y*
0	0	1	0	0	0	1	0	X	0	X	X	0	0
0	0	1	1	0	1	0	0	X	1	0	0	1	0
0	1	0	0	0	1	1	0	X	X	0	1	0	0
0	1	0	1	1	0	0	1	0	0	1	0	X	0
0	1	1	0	0	0	1	0	X	0	1	X	0	0
0	1	1	1	1	0	0	1	0	0	1	0	1	0
1	0	0	0	1	0	1	X	0	0	X	1	0	0
1	0	0	1	1	0	0	X	0	0	X	0	X	1
1	0	1	0	0	0	1	0	1	0	X	X	0	0
1	0	1	1	1	0	0	X	0	0	X	0	1	1

Given these

Derive these

Are there other unused states?

110, 111

Unused state 000:

0	0	0	0	X	X	X	X	X	X	X	X	X	X
0	0	0	1	X	X	X	X	X	X	X	X	X	X