EENG 2410 – Digital Circuits

Lindsey Erps

Prepared for Dr. Ricky Castles

April 27, 2015

The *Legend* Continues

This project demonstrates the design and implementation of a simplified version of the classic Nintendo game *The Legend of Zelda*.

In the original, the protagonist, Link, bravely navigates through labyrinthine dungeons in the service of the wise and enchanting Princess Zelda. In the version developed in this project, there is only 1 dungeon which contains 4 rooms (see Figure 1).

Link begins his quest in room 1. Before Link can reach Princess Zelda, he must first obtain the Triforce of Power which is in room 2. After Link picks up the triforce, Zelda is regurgitated from the belly of a dim-witted, ravenous monster into room 4. When Link enters room 4 with the Triforce of Power in his possession, Link and Zelda are reunited, and the day is won.

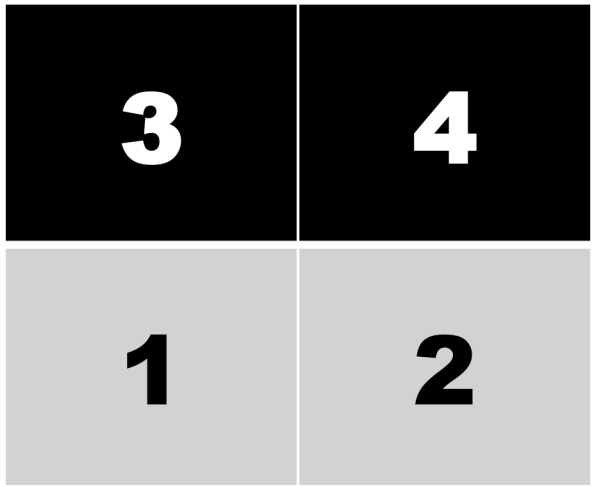


Figure 1 Room Layout

Process Overview

The game is designed and simulated in the form of a sequential logic circuit. JK flip flops are used to maintain each logical state and to control the output.

The process begins with a mapping of the game and its input logic (see Table 1) into sequential states whose output is a function of the current state only. This is a Moore circuit.

These states are then assigned binary representations and the combinational logic for the outputs and flip flop inputs is derived using Karnaugh maps.

Move Logic M_1M_2	Movement
00	Left
01	Right
10	Up
11	Down

Table 1 Player Movement Logic

Inputs

Binary switches are used to simulate each of these inputs.

- M_1 – most significant bit of move logic
- M_2 – least significant bit of move logic
- Clock – a common clock to all flip flops processes the input move logic
- Reset – resets the system to the initial state

Outputs

- P – 1 when in possession of the Triforce of Power, 0 when not
- V – 1 when the game is won (Victory), 0 when still playing

Truth Table

Present State	Next State for given input (M_1M_2) Combination				P	V
	00	01	10	11		
A	A	B	C	A	0	0
B	E	B	F	B	1	0
C	C	D	C	A	0	0
D	C	D	D	B	0	0
E	E	B	G	E	1	0
F	F	F	F	F	1	1
G	G	F	G	E	1	0

Truth Table

To create the truth table, states were assigned to Link's status when in each room according to whether or not he was in possession of the triforce or not.

For example, Link starts in room 1 which is represented as state A. If Link moves right (01), he picks up the Triforce of Power in room 2 which is represented as state B. Once Link has the triforce, if he goes back to room 1 his status is represented by state E. The state representations for all of the other rooms follow in the same fashion.

State F represents Link's arrival in room 4 whilst he is in possession of the triforce. Link and Zelda are reunited and V goes to a 1. The game must be reset for Link to get out.

Present State	Next State for given input (M_1M_2) Combination				P	V
	00	01	10	11		
A	A	B	C	A	0	0
B	E	B	F	B	1	0
C	C	D	C	A	0	0
D	C	D	D	B	0	0
E	E	B	G	E	1	0
F	F	F	F	F	1	1
G	G	F	G	E	1	0

State Interpretations

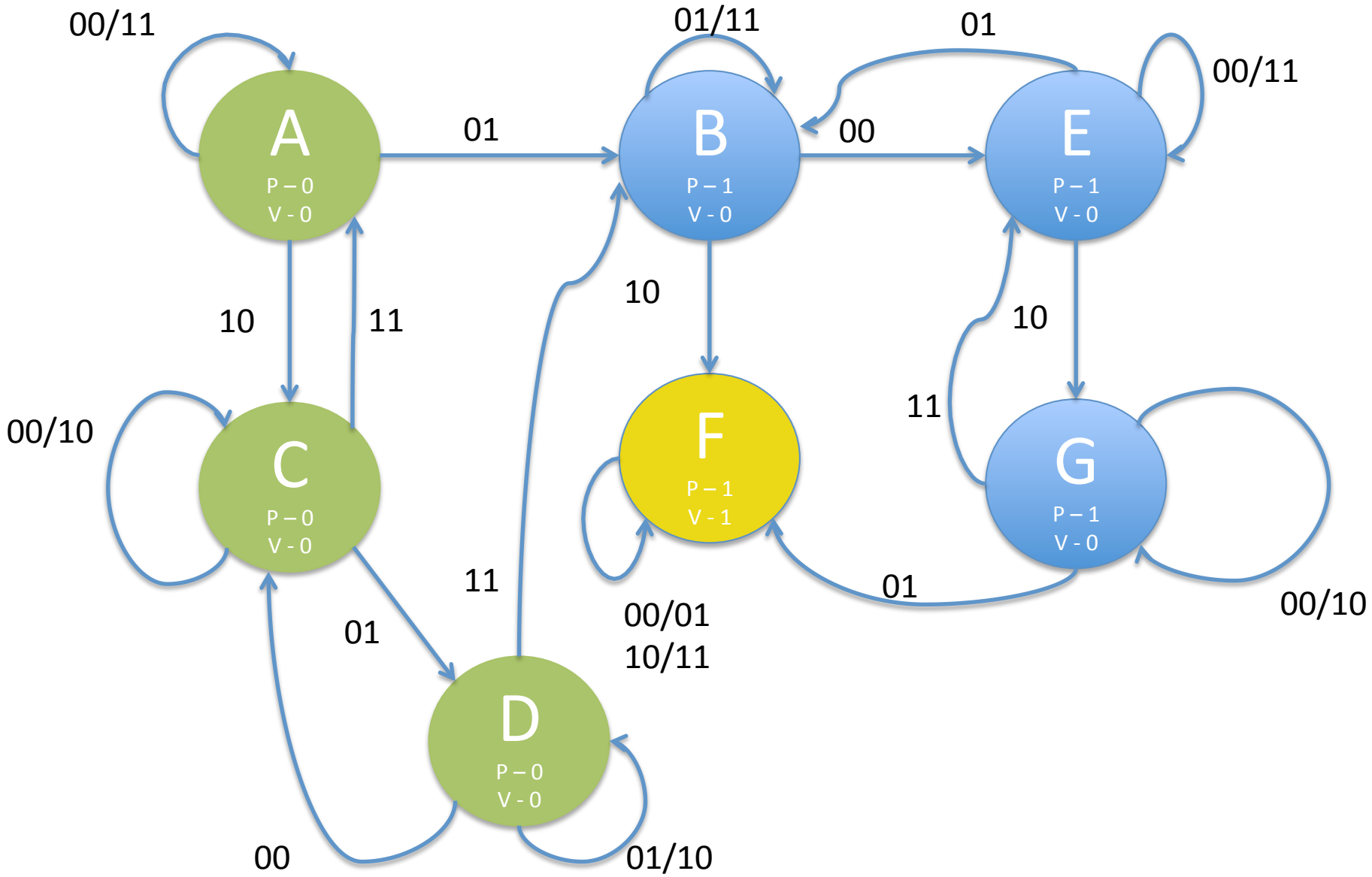
State	Interpretation
A	Initial state – Link is in room 1 w/o triforce
B	Room 2 – Link finds triforce
C	Room 3 – Link does not have triforce
D	Room 4 – Link does not have triforce
E	Room 1 – Link has triforce
F	Room 4 - Victory
G	Room 3 – Link has triforce

State Reduction

B	X				
C	B – D X	X			
D	A – C B – D X C – D A – B	X	C – D A – B X		
E	X	F – G B – E X	X	X		
F	X	X	X	X	X	
G	X	E – G B – F X F – G B – E	X	X	B – F X	X
	A	B	C	D	E	F

*The implication chart indicates that there are no equivalent states.

State Transition Diagram



Binary State Assignment

Guideline 1: States that go to the same next state for a given input should be adjacent

(B & E)** (C & D)*** (F & G)* (B, D & F) (A, B & E)

(A & C)*** (B & F)** (E &G)**

Guideline 2: States that are the next states of the same state should be adjacent

(B & C)* (A & B) (E & F) (A & D) (B & G)* (G & A)

*indicates that the adjacency satisfies multiple conditions or guidelines

/State adjacencies in red are satisfied by binary state assignment actually used to implement circuit

Binary State Assignment

BC \ A	0	1
00	A	C
01	F	B
11	G	E
10	X	D

Binary Truth Table

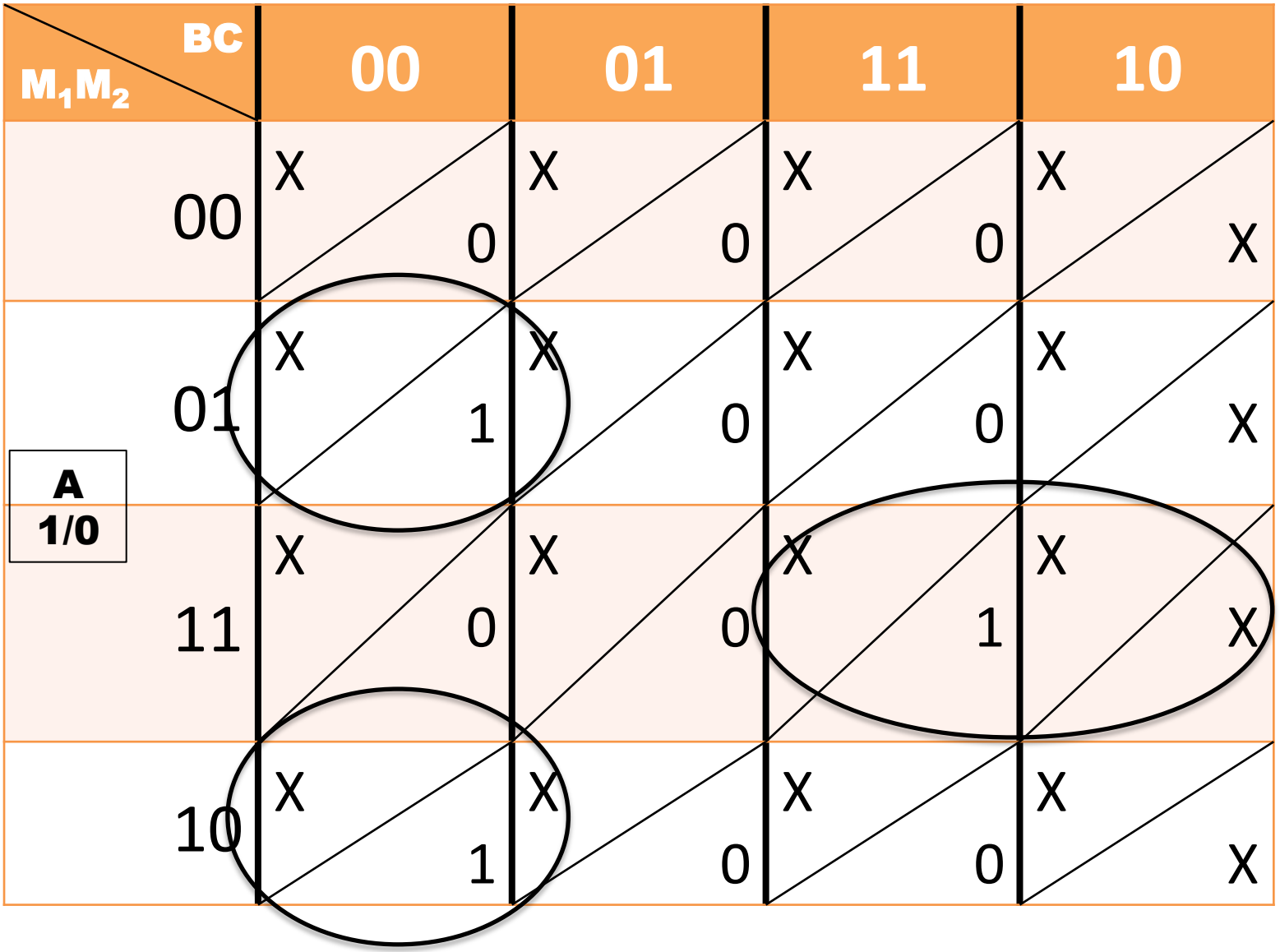
Present State	Next State for given input (M_1M_2) Combination				P	V
	00	01	10	11		
A 000	000	101	100	000	0	0
B 101	111	101	001	101	1	0
C 100	100	110	100	000	0	0
D 110	100	110	110	101	0	0
E 111	111	101	011	111	1	0
F 001	001	001	001	001	1	1
G 011	011	001	011	111	1	0

*010 is unassigned

JK Flip Flop Input Logic

A Bit

$$A^{+}_{J}=B'C'M_1'M_2 + B'C'M_1M_2' + BM_1M_2$$



$$A^{+}_{K} = CM_1M_2' + B'C'M_1M_2$$

M_1M_2 \ BC	00	01	11	10
00	0 / X	0 / X	0 / X	0 / X
01	0 / X	0 / X	0 / X	0 / X
11	1 / X	0 / X	0 / X	0 / X
10	0 / X	1 / X	1 / X	0 / X

A 1/0

JK Flip Flop Input Logic

B Bit

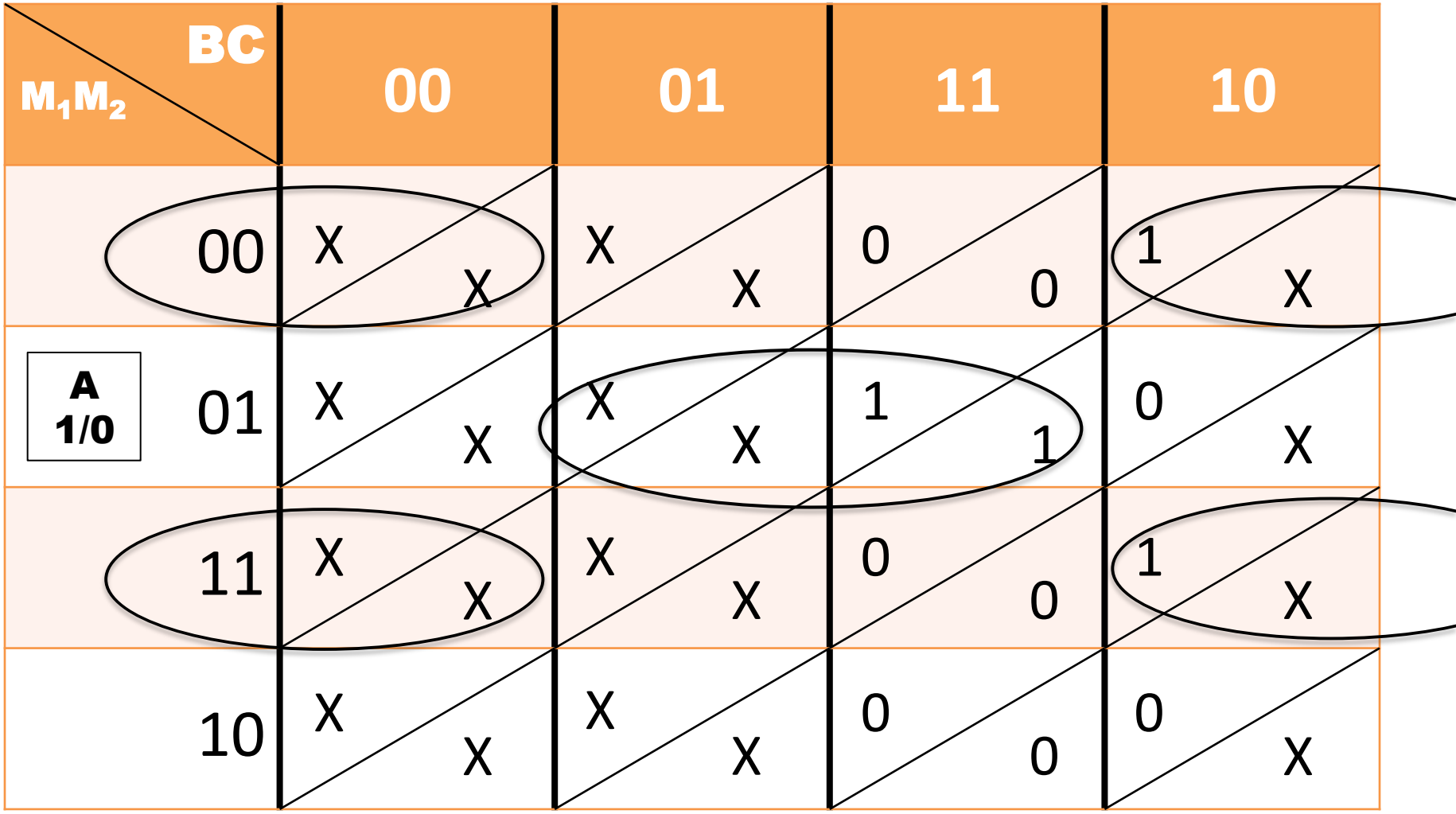
$$B^{+}_{J} = ACM_1'M_2' + AC'M_1'M_2$$

M_1M_2 \ BC	00	01	11	10
00	0	1	X	X
A 1/0				
01	1	0	X	X
11	0	0	X	X
10	0	0	X	X

$$B^{+}_{K} = C'M_1'M_2' + CM_1'M_2 + C'M_1M_2$$

M_1M_2 \ BC	00	01	11	10
00	X	X	0	1
01	X	X	1	0
11	X	X	0	1
10	X	X	0	0

A
1/0



JK Flip Flop Input Logic

C Bit

$$C^{+}_{J} = A'M_1'M_2 + BM_1M_2$$

M_1M_2 \ BC	00	01	11	10
00	0 / 0	X / X	X / X	0 / X
A 1/0 — 01	0 / 1	X / X	X / X	0 / X
11	0 / 0	X / X	X / X	1 / X
10	0 / 0	X / X	X / X	0 / X

$C^{+}_{K} = 0$

M_1M_2 \ BC	00	01	11	10
00	X / X	0 / 0	0 / 0	X / X
A 1/0 — 01	X / X	0 / 0	0 / 0	X / X
11	X / X	0 / 0	0 / 0	X / X
10	X / X	0 / 0	0 / 0	X / X

Output Logic

BC \ A	0	1
00	0	0
01	1	1
11	1	1
10	X	0

P = C

BC \ A	0	1
00	0	0
01	1	0
11	0	0
10	X	0

V = A'B'C