Pakorn Watanachaturaporn
pakorn.wa@KMITL.ac.th

01076244 Advanced Digital System Design

Bachelor Program in Computer Engineering (B.Eng.)
Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang

Clocked Synchronous State-Machine Design

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

1. Construct a state / output table corresponding to word description or specification, using mnemonic names for the states.
2. (Optional) Minimize the number of state in the state/output table.
3. Choose a set of state - variable combination to the named state.
4. Substitute the state-variable combination into the state/output table to create a transition/output table that shows the desired next state-variable combination and output for each state/input combination

State / output table

State minimization

State assignment

Transition / output table

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

5. Choose a flip-flop (e.g. D or JK) for the state memory.
6. Construct an excitation table that shows the excitation values required to obtain the desired next state for each/input combination.
7. Derive excitation from the excitation table.
8. Derive output equations from the transition/output table.
9. Draw a logic diagram that shows the state-variable storage elements and realized the required excitation and output equations.

Excitation table

Excitation equations

Output equations

Logic diagram

State-Table Design Example

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

มี input เข้า A มาเป็น 2

Design a clocked synchronous state machine with two inputs, A and B, and a single output Z that is 1 if :

A same กัน 2 ตัว ก่อน clk ที่ขึ้น

- A had the same value at each of the two previous clock ticks, *or*

B เป็น 1 จะ ควรทำไว้

- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.

มิฉะนั้น O/P = 0

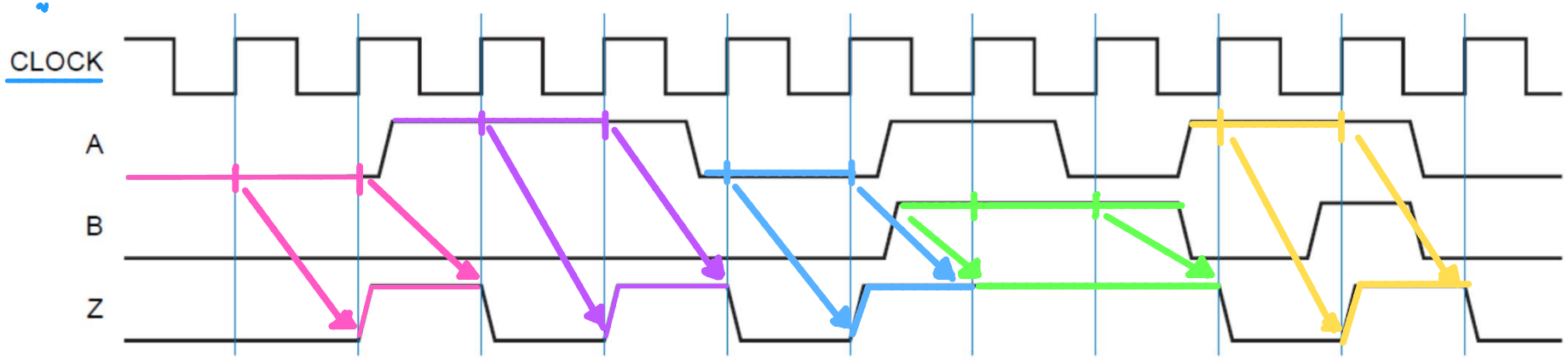
KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

The output Z is 1 if :

- A had the same value at each of the two previous clock ticks, *or*
- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.

ดูที่ clk ที่ขึ้น A ของ 2 clk เหมือน ⇒ Z=1



ทำให้เข้าใจ โจทย์มากขึ้น

Timing diagram for example state machine

Is it a Mealy or Moore machine?

Ans: Moore machine – its o/p depends only on the current State

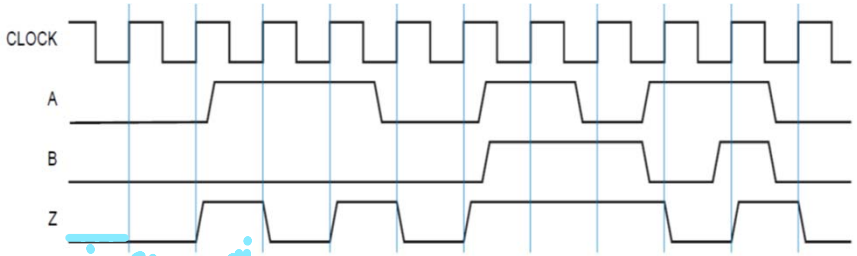
KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

The output Z is 1 if :

input 2 ตัวต้อง เหมือนกัน

- A had the same value at each of the two previous clock ticks, *or*
- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.



เพิ่งเปิดเครื่อง

Z is 1 as long as B = 1

Meaning	S	A B: 00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK	OK	A1	A1	0
Got a 1 on A	A1	A0	A0	OK	OK	0
Got two equal A input OK		?	OK	OK	?	1
what state would be ?		S*				

2 CLK ไม่ same กัน

We do NOT know what was the previous input !!!

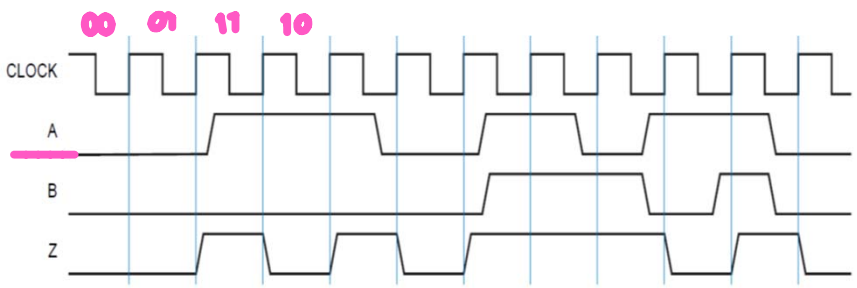
Our defined state does NOT "remember" enough to tell which way to go

The output Z is 1 if :

- A had the same value at each of the two previous clock ticks, *or*
- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.

Meaning	S	A B: 00	01	11	10	Z
Initial state	INIT					0
	...					
	...					
	...					
		S*				



KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

OK = CLK เท่ากัน 2 ตัว แล้ว same กัน ⇒ Z=1

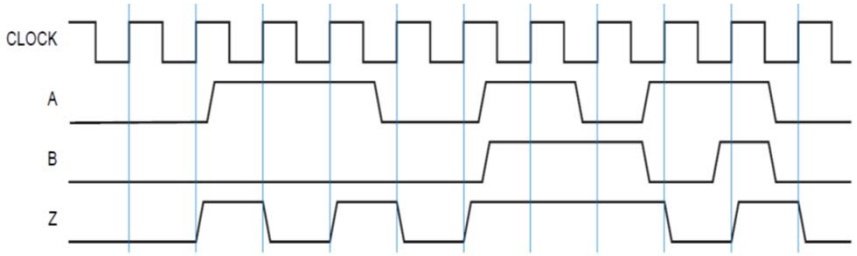
Meaning	S	A B: 00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK	OK	A1	A1	0
Got a 1 on A	A1	A0	A0	OK	OK	0
		S*				

2 CLK ใดๆ same กัน

The output Z is 1 if :

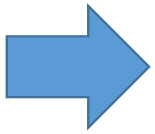
- A had the same value at each of the two previous clock ticks, *or*
- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.



KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

Meaning	S	A B 00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0					0
Got a 1 on A	A1					0
		S*				

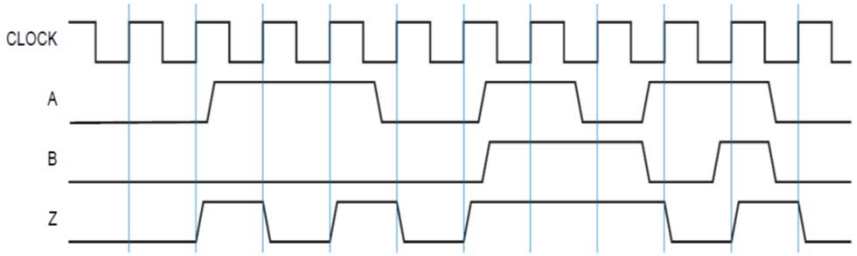


Meaning	S	A B 00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0					0
Got a 1 on A	A1					0
Got two equal A inputs	OK	?	OK	OK	?	1
		S*				

The output Z is 1 if :

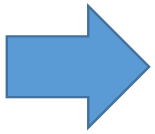
- A had the same value at each of the two previous clock ticks, *or*
- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.



KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

		A B				
Meaning	S	00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK	OK	A1	A1	0
Got a 1 on A	A1					0
Got two equal A inputs	OK					1
		S*				

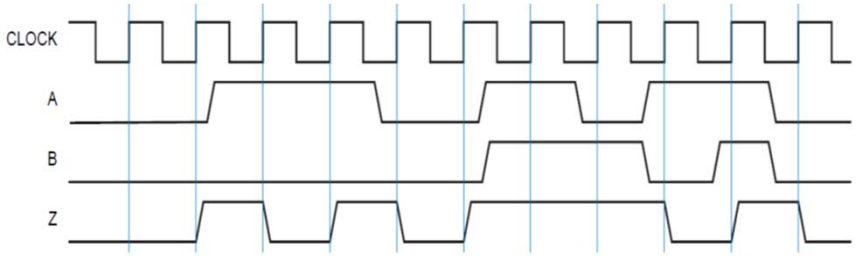


		A B				
Meaning	S	00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK	OK	A1	A1	0
Got a 1 on A	A1	A0	A0	OK	OK	0
Got two equal A inputs	OK					1
		S*				

The output Z is 1 if :

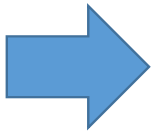
- A had the same value at each of the two previous clock ticks, *or*
- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.



KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

Meaning	S	A B: 00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK	OK	A1	A1	0
Got a 1 on A	A1	A0	A0	OK	OK	0
Got two equal A inputs	OK					1
		S*				

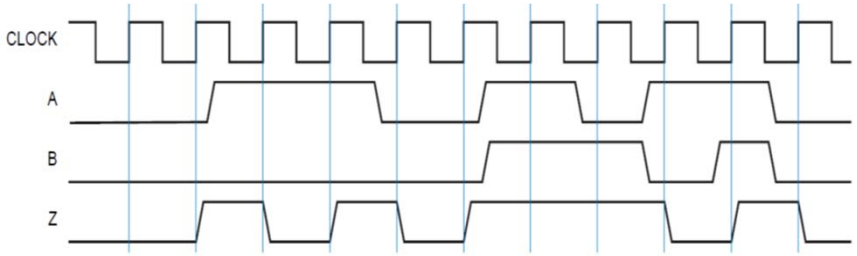


Meaning	S	A B: 00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK	OK	A1	A1	0
Got a 1 on A	A1	A0	A0	OK	OK	0
Got two equal A inputs	OK	?	OK	OK	?	1
		S*				

The output Z is 1 if :

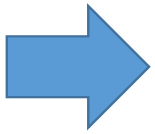
- A had the same value at each of the two previous clock ticks, *or*
- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.



KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

		A B				
Meaning	S	00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK	OK	A1	A1	0
Got a 1 on A	A1	A0	A0	OK	OK	0
Got two equal A inputs	OK	?	OK	OK	?	1
		S*				

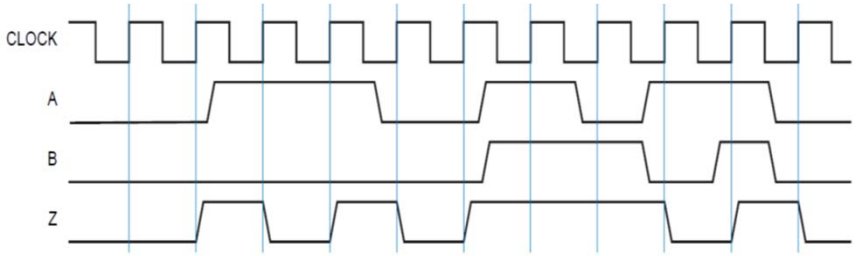


		A B				
Meaning	S	00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0			A1	A1	0
Got a 1 on A	A1	A0	A0			0
		S*				

The output Z is 1 if :

- A had the same value at each of the two previous clock ticks, *or*
- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.



KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

Meaning	S	A B 00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK0	OK0	A1	A1	0
Got a 1 on A	A1	A0	A0	OK1	OK1	0
Two equal, A=0 last	OK0					1
Two equal, A=1 last	OK1					1
		S*				

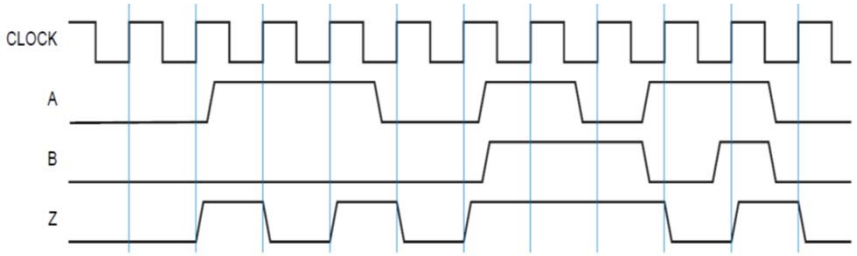
OK เหมือนกันกับตัวก่อนหน้า

Meaning	S	A B 00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK0	OK0	A1	A1	0
Got a 1 on A	A1	A0	A0	OK1	OK1	0
Two equal, A=0 last	OK0					1
Two equal, A=1 last	OK1					1
		S*				

The output Z is 1 if :

- A had the same value at each of the two previous clock ticks, *or*
- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.

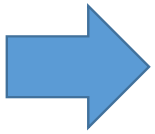


KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

5 state ใช้ FF ≥ 3 ตัว

จริงเหรอ?

Meaning	S	A B = 00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK0	OK0	A1	A1	0
Got a 1 on A	A1	A0	A0	OK1	OK1	0
Two equal, A=0 last	OK0	OK0	OK0	OK1	A1	1
Two equal, A=1 last	OK1					1
			S*			



Meaning	S	A B = 00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK0	OK0	A1	A1	0
Got a 1 on A	A1	A0	A0	OK1	OK1	0
Two equal, A=0 last	OK0	OK0	OK0	OK1	A1	1
Two equal, A=1 last	OK1	A0	OK0	OK1	OK1	1
				S*		

ค่าควรเป็น

ค่าควรเป็น

State Minimization

ยิ่ง state เยอะยิ่งยาก

Meaning	S	A B = 00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK0	OK0	A1	A1	0
Got a 1 on A	A1	A0	A0	OK1	OK1	0
Two equal, A=0 last	OK0	OK0	OK0	OK1	A1	1
Two equal, A=1 last	OK1	A0	OK0	OK1	OK1	1
		S*				

Minimal state table

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

Meaning	S	A B = 00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK00	OK00	A1	A1	0
Got a 1 on A	A1	A0	A0	OK11	OK11	0
Got 00 on A	OK00	OK00	OK00	OKA1	A1	1
Got 11 on A	OK11	A0	OKA0	OK11	OK11	1
OK, got a 0 on A	OKA0	OK00	OK00	OKA1	A1	1
OK, got a 1 on A	OKA1	A0	OKA0	OK11	OK11	1
		S*				

Meaning	S	A B = 00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK00	OK00	A1	A1	0
Got a 1 on A	A1	A0	A0	OK11	OK11	0
Got 00 on A	OK00	OK00	OK00	A001	A1	1
Got 11 on A	OK11	A0	A110	OK11	OK11	1
Got 001 on A, B=1	A001	A0	AE10	OK11	OK11	1
Got 110 on A, B=1	A110	OK00	OK00	AE01	A1	1
Got bb...10 on A, B=1	AE10	OK00	OK00	AE01	A1	1
Got bb...01 on A, B=1	AE01	A0	AE10	OK11	OK11	1
		S*				

Two Non-minimal state table

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

Two state S1 and S2 are equivalent if:

1. S1 and S2 must produce the same values at the state-machine output(s).
2. For each input combination, S1 and S2 must have either the same next state or equivalent next state.

เท่ากัน เท่ากัน

Meaning	S	A B: 00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK00	OK00	A1	A1	0
Got a 1 on A	A1	A0	A0	OK11	OK11	0
Got 00 on A	OK00	OK00	OK00	OKA1	A1	1
Got 11 on A	OK11	A0	OKA0	OK11	OK11	1
OK, got a 0 on A	OKA0	OK00	OK00	OKA1	A1	1
OK, got a 1 on A	OKA1	A0	OKA0	OK11	OK11	1
						S*

ใช้แทนได้

Same → cancelได้

ใช้แทนได้เปลี่ยนได้ bec. เป็นแค่ชื่อ (can change)

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

∴ เป็น equivalent กัน
bec. ● equivalent กัน

Meaning	S	A B 00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK00	OK00	A1	A1	0
Got a 1 on A	A1	A0	A0	OK11	OK11	0
Got 00 on A	OK00	OK00	OK00	A001	A1	1
Got 11 on A	OK11	A0	A110	OK11	OK11	1
Got 001 on A, B=1	A001	A0	AE10	OK11	OK11	1
Got 110 on A, B=1	A110	OK00	OK00	AE01	A1	1
Got bb...10 on A, B=1	AE10	OK00	OK00	AE01	A1	1
Got bb...01 on A, B=1	AE01	A0	AE10	OK11	OK11	1
		S*				

State Assignment

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

- A coded state is the binary combination assigned to a particular state.
- The total number of state in a machine with n flip-flops is 2^n, so the number of flip-flops needed to code s states is $\lceil \log_2 s \rceil$, the smallest integer greater than or equal to $log_2 s$.

⌈ ⌉ = "ceiling" f" = ปัดขึ้น
⌊ ⌋ = "flooring" f". ปัดลง

S	A B: 00	01	11	10	Z
INIT	A0	A0	A1	A1	0
A0	OK0	OK0	A1	A1	0
A1	A0	A0	OK1	OK1	0
OK0	OK0	OK0	OK1	A1	1
OK1	A0	OK0	OK1	OK1	1
	S*				

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

ไม่จำเป็นต้องเป็นวิธีที่ดีที่สุด

State name	Assignment: Simplest Q1 - Q3	Decomposed Q1 - Q3	One-hot Q1 - Q5	Almost one-hot Q1 - Q4
INIT	000	000	00001	0000
A0	001	100	00010	0001
A1	010	101	00100	0010
OK0	011	110	01000	0100
OK1	100	111	10000	1000

อย่างน้อยมี state ที่เป็น 1

เกิดขึ้นครั้งเดียว

เหลือ

Binary counting order

Q1 is for stating the initial state (occur only once)

use one bit per state

"no-hot" for initial state

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

- The simplest state assignment does not always lead to the simplest excitation equations, output equations, and resulting logic circuit.

- Most digital designers rely on experience and several practical guidelines for making reasonable state assignments.

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

- Choose an initial coded state into which the machine can easily be forced at reset (00 … 00 or 11 … 11 in typical circuits).
- Minimize the number of state variable that change on each transition.
- If there are unused states, then choose the "best" of the available state-variable combinations to achieve the foregoing goals; *i.e.*, do not limit the choice of coded states to the first *s* *n*-bit integers.

counting ขากลุด "อย่าทำ"

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

- Decompose ออกมาเป็นกลุ่ม the set of state variables into individual bits or fields where each bit or field has a well-defined meaning with respect to the input effects or output behavior of the machine.
- Consider using more than minimum number of state variables to make a decomposed assignment possible.
- Etc.

pakorn.wa@kmitl.ac.th

When the number of states available with *n* flip-flops, 2^n, is greater than the number of states required, *s*.

Two possible approaches:

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

- Minimal risk.
  - Assume that it is possible for the state machine somehow to get into one of the unused (or "illegal") states : hardware failure, an unexpected input, or a design error.
  - Therefore, all of the unused state-variable combination are identified, and explicit next state entries are made so that, for any input combination, the unused states go to the "initial" state, the "idle" state, or some other "safe" state.

ไม่รู้ไปไหน → อยู่เฉยๆ

pakorn.wa@kmitl.ac.th

- Minimal cost.
  - Assume that the machine will never enter an unused state.
  - Therefore, in the transition and excitation tables, the next-state entries of the unused states can be marked as "don't-cares."
    In most case, this simplifies the excitation logic.
  - However, the machine's behavior if it ever does enter an unused state may be pretty weird.

เท่าอะไรออกอย่างนั้น ไม่ต้องทำอะไรเพิ่ม

Synthesis Using D Flip-Flops

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

เลือก " Decompose"

	A B				
S	00	01	11	10	Z
INIT	A0	A0	A1	A1	0
A0	OK0	OK0	A1	A1	0
A1	A0	A0	OK1	OK1	0
OK0	OK0	OK0	OK1	A1	1
OK1	A0	OK0	OK1	OK1	1
	S*				

State / output table

	A B				
Q1 Q2 Q3	00	01	11	10	Z
000	100	100	101	101	0
100	110	110	101	101	0
101	100	100	111	111	0
110	110	110	111	101	1
111	100	110	111	111	1
	Q1* Q2* Q3*				

Transition / output table
(possibly minimized)

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

	A B				
Q1 Q2 Q3	00	01	11	10	Z
000	100	100	101	101	0
100	110	110	101	101	0
101	100	100	111	111	0
110	110	110	111	101	1
111	100	110	111	111	1
	Q1*	Q2*	Q3*		

Transition / output table
(possibly minimized)

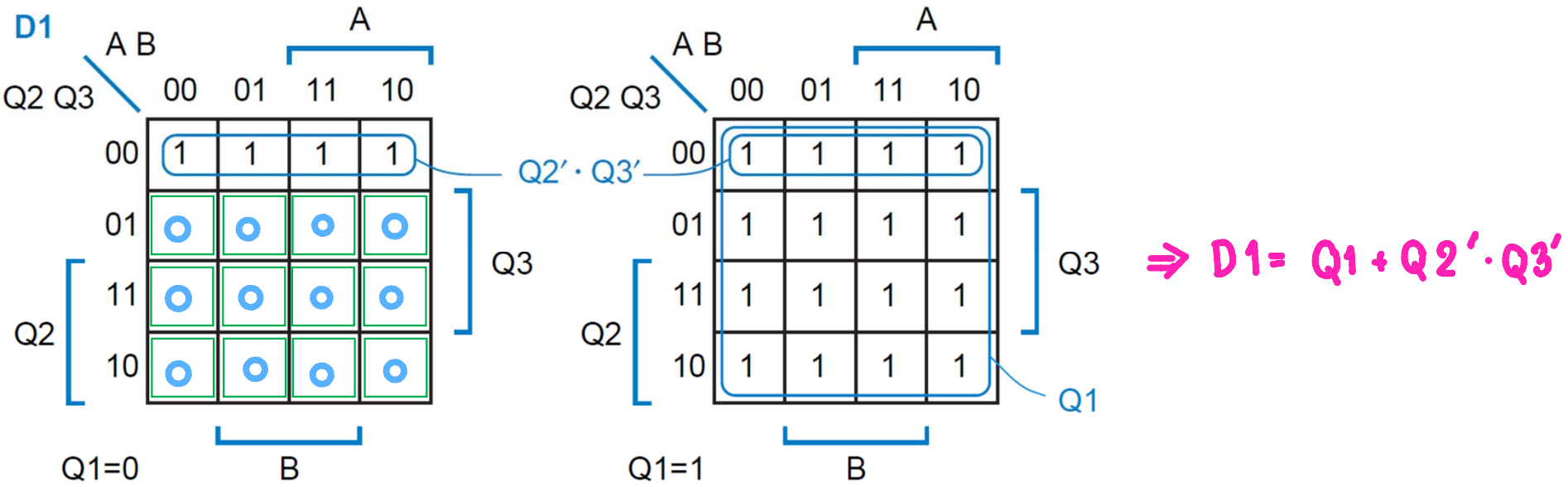
	A B				
Q1 Q2 Q3	00	01	11	10	Z
000	100	100	101	101	0
100	110	110	101	101	0
101	100	100	111	111	0
110	110	110	111	101	1
111	100	110	111	111	1
	D1	D2	D3		

Excitation / output table

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

	A B				
Q1 Q2 Q3	00	01	11	10	Z
000	100	100	101	101	0
100	110	110	101	101	0
101	100	100	111	111	0
110	110	110	111	101	1
111	100	110	111	111	1
	D1	D2	D3		

Excitation / output table

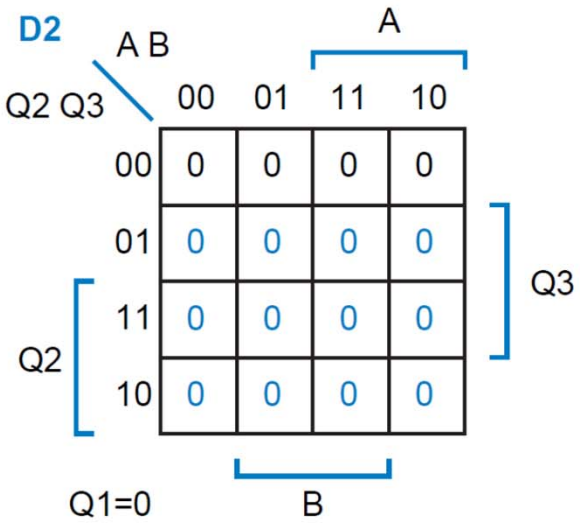


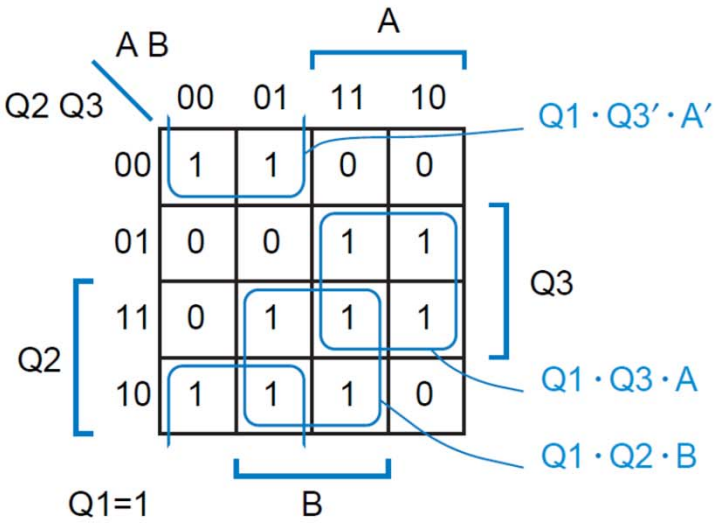
$\Rightarrow D1 = Q1 + Q2' \cdot Q3'$

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

Q1 Q2 Q3	A B 00	01	11	10	Z
000	100	100	101	101	0
100	110	110	101	101	0
101	100	100	111	111	0
110	110	110	111	101	1
111	100	110	111	111	1
	D1	D2	D3		

Excitation / output table



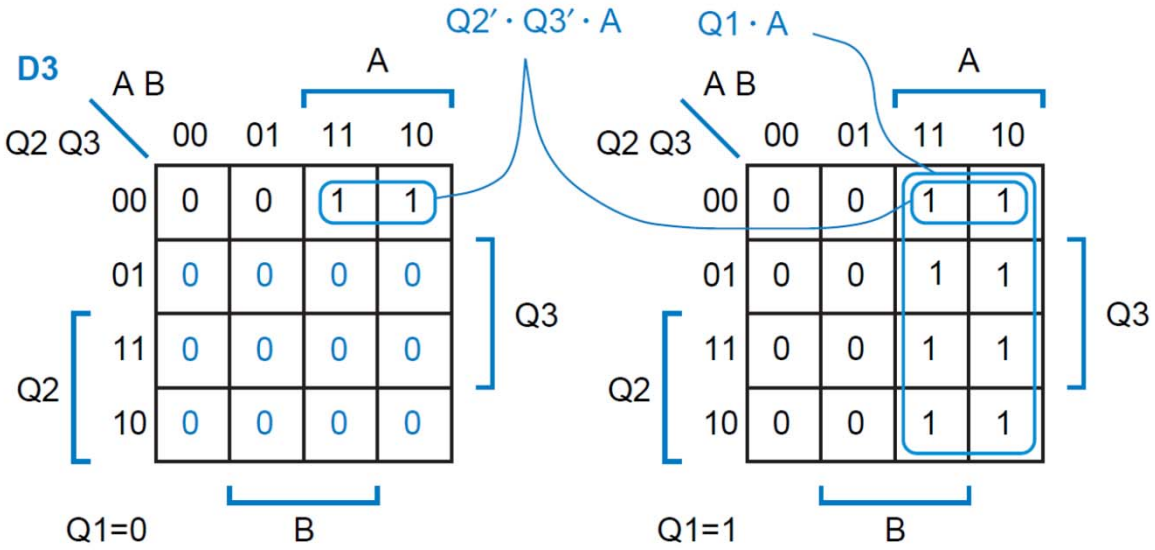


$$D_2 = Q1 \cdot Q3' \cdot A + Q1 \cdot Q3 \cdot A + Q1 \cdot Q2 \cdot B$$

	A B				
Q1 Q2 Q3	00	01	11	10	Z
000	100	100	101	101	0
100	110	110	101	101	0
101	100	100	111	111	0
110	110	110	111	101	1
111	100	110	111	111	1
	D1 D2 D3				

Excitation / output table

$Z = Q1 \cdot Q2 \cdot Q3' + Q1 \cdot Q2 \cdot Q3 + Q1 \cdot Q2$

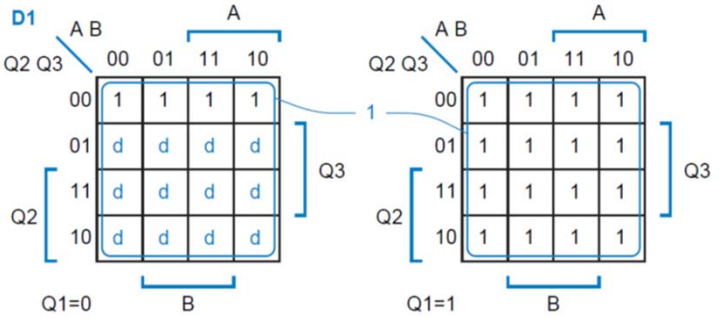


$D3 = Q1 \cdot A + Q2' \cdot Q3' \cdot A$

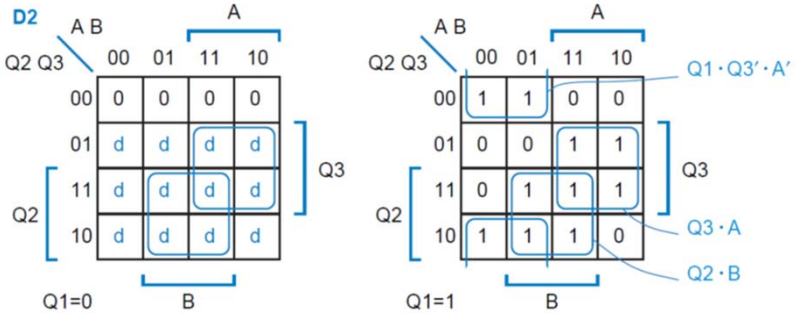
KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

In case "don't-cares" are used in the unused states.

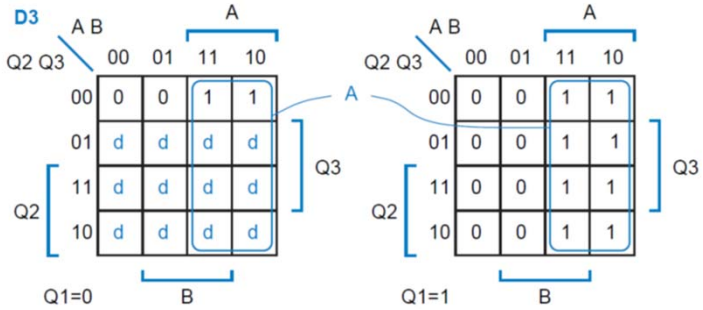
KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th



$D1 = 1$



$D2 = Q1 \cdot Q3' \cdot A + Q3 \cdot A + Q2 \cdot B$



$D3 = A$

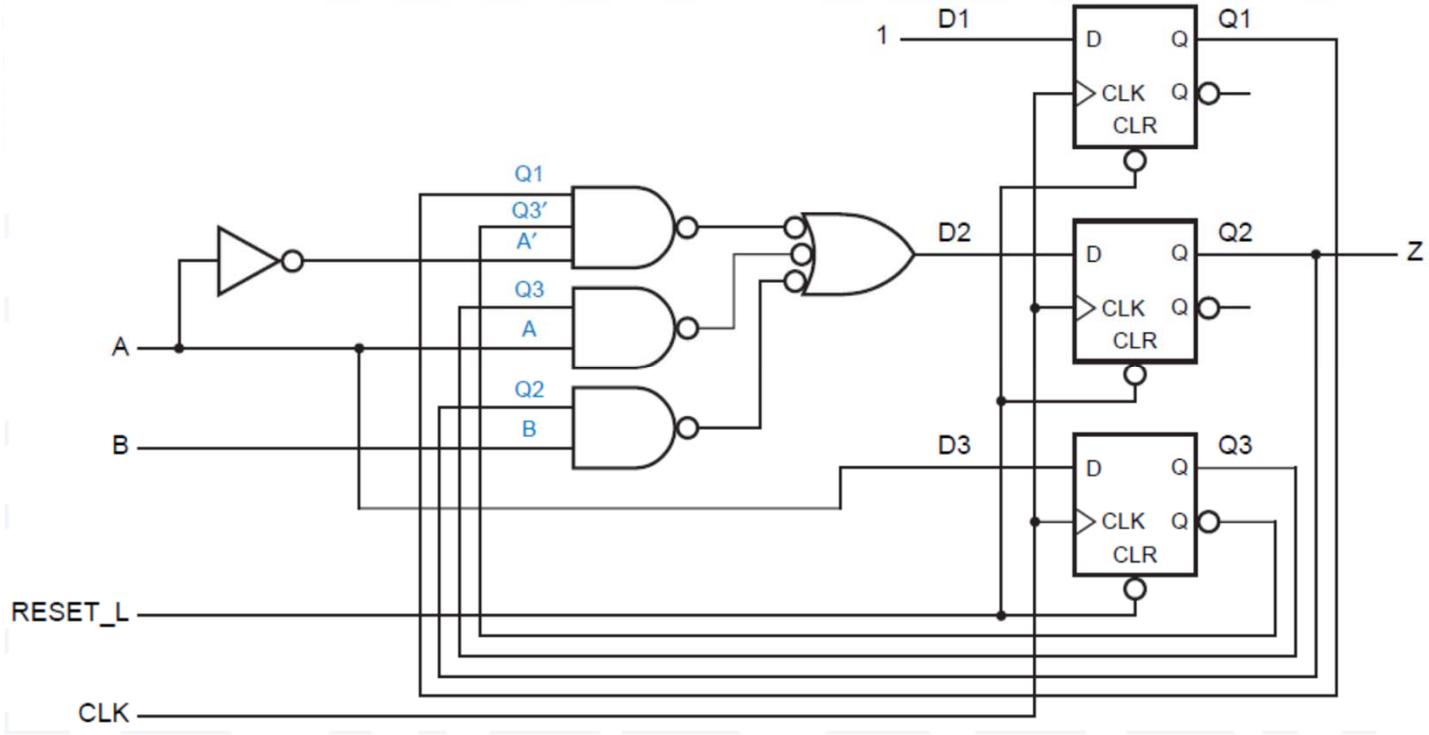
D1 = 1

D2 = Q1·Q3'·A + Q3·A + Q2·B

D3 = A

Z = Q1·Q2

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th



Synthesis Using J-K Flip-Flops

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

Q	Q*	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

ขึ้นอยู่กับ FF ที่เราเลือก

	A B				
Q1 Q2 Q3	00	01	11	10	Z
000	100	100	101	101	0
100	110	110	101	101	0
101	100	100	111	111	0
110	110	110	111	101	1
111	100	110	111	111	1
		Q1* Q2* Q3*			

Transition / output table
(possibly minimized)

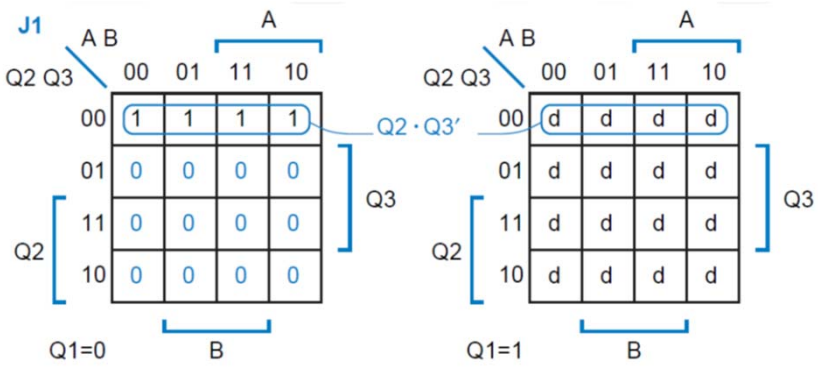
	A B				
Q1 Q2 Q3	00	01	11	10	Z
000	1d , 0d , 0d	1d , 0d , 0d	1d , 0d , 1d	1d , 0d , 1d	0
100	d0 , 1d , 0d	d0 , 1d , 0d	d0 , 0d , 1d	d0 , 0d , 1d	0
101	d0 , 0d , d1	d0 , 0d , d1	d0 , 1d , d0	d0 , 1d , d0	0
110	d0 , d0 , 0d	d0 , d0 , 0d	d0 , d0 , 1d	d0 , d1 , 1d	1
111	d0 , d1 , d1	d0 , d0 , d1	d0 , d0 , d0	d0 , d0 , d0	1
		J1 K1 , J2 K2 , J3 K3			

Excitation / output table

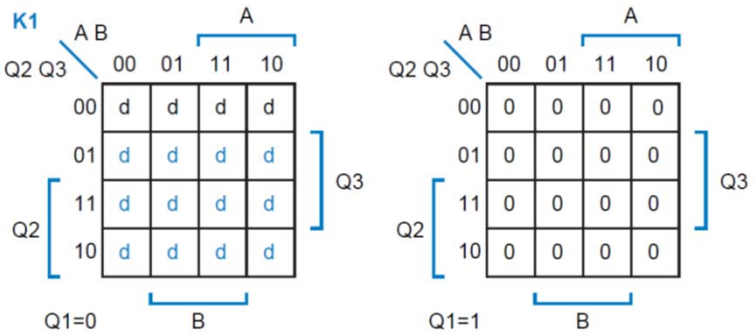
Excitation maps for J1, K1

Assuming that unused states go to state 000

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th



$$J1 = Q2' \cdot Q3'$$

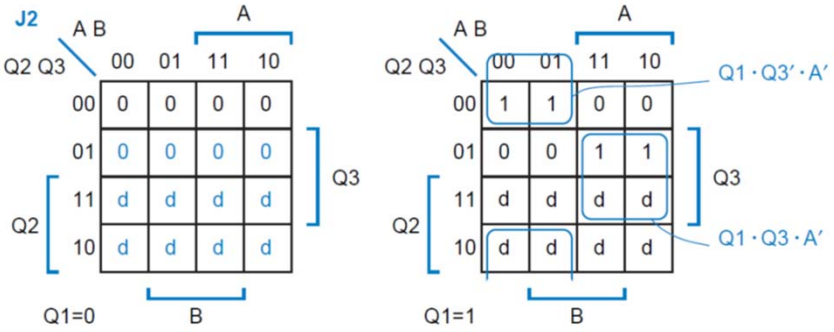


$$K1 = 0$$

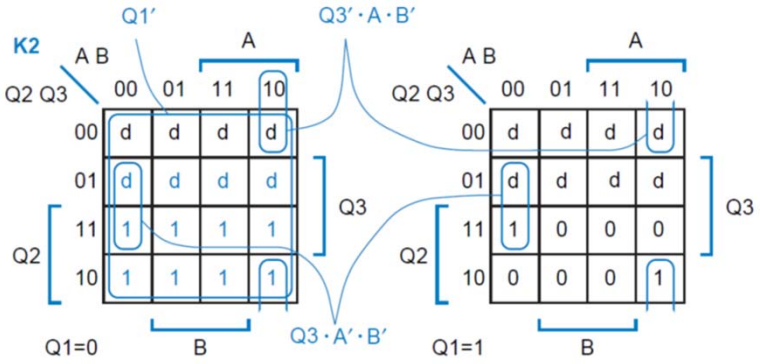
Excitation maps for J2, K2

Assuming that unused states go to state 000

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th



$$J2 = Q1Q3' \cdot A' + Q1 \cdot Q3 \cdot A$$

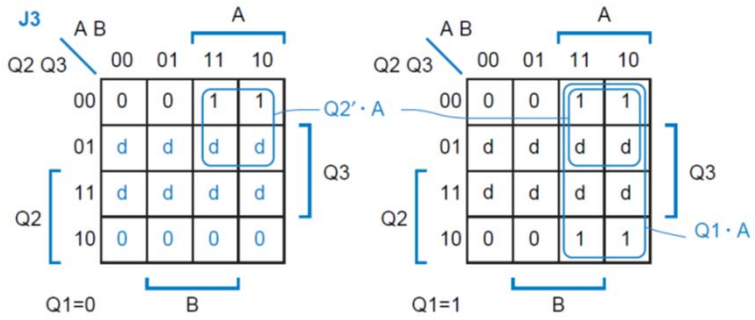


$$K2 = Q1' + Q3' \cdot A \cdot B' + Q3 \cdot A' \cdot B'$$

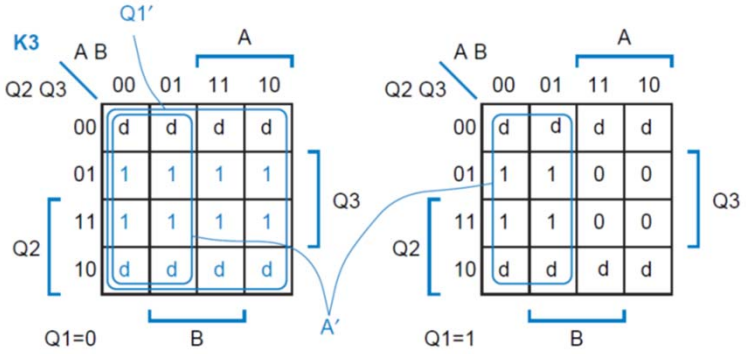
Excitation maps for J3, K3

Assuming that unused states go to state 000

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th



$$J3 = Q2' \cdot A + Q1 \cdot A$$



$$K3 = Q1' + A'$$

ขั้นตอน FF ที่เราใช้

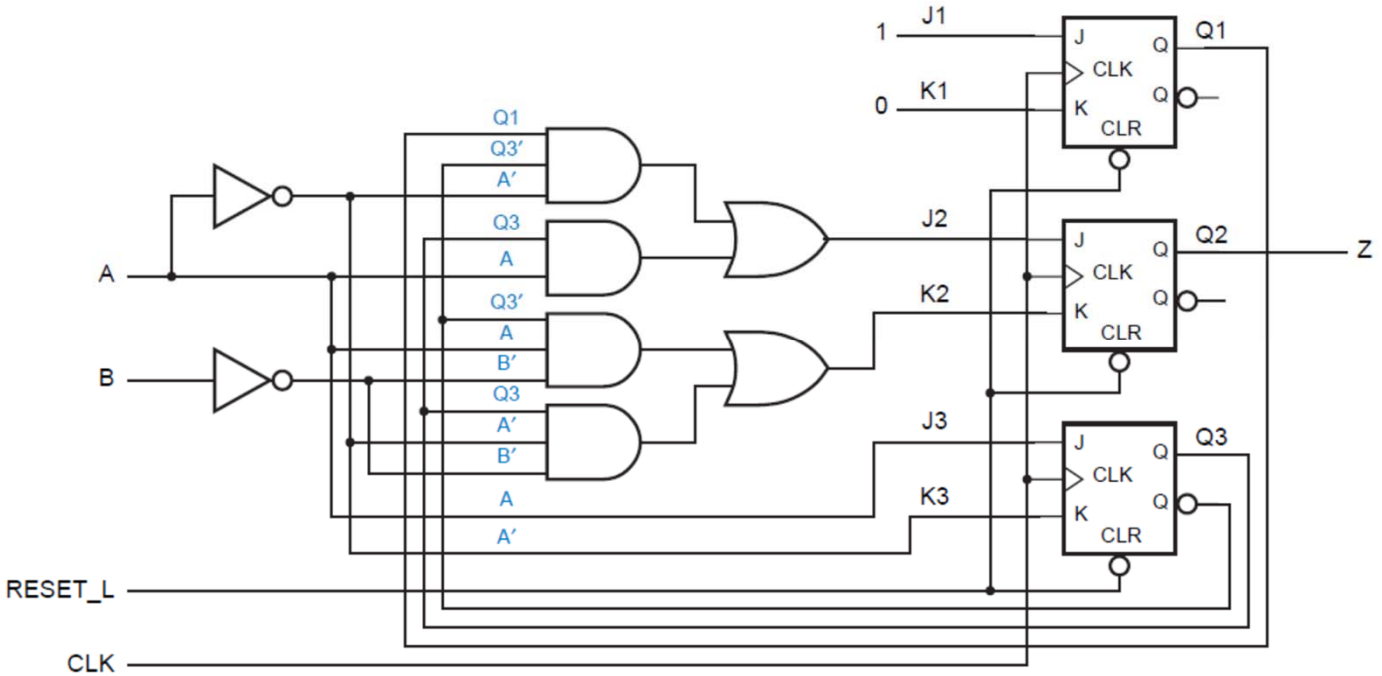
Excitation equations using J-K flip-flops

by putting **d's** in all of the unused state entries

KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

J1 = 1
J2 = Q1·Q3·A' + Q3·A
J3 = A

K1 = 0
K2 = Q3'·A·B' + Q3·A'·B'
K3 = A'



KMITL
พระจอมเกล้าลาดกระบัง
pakorn.wa@kmitl.ac.th

Q & A