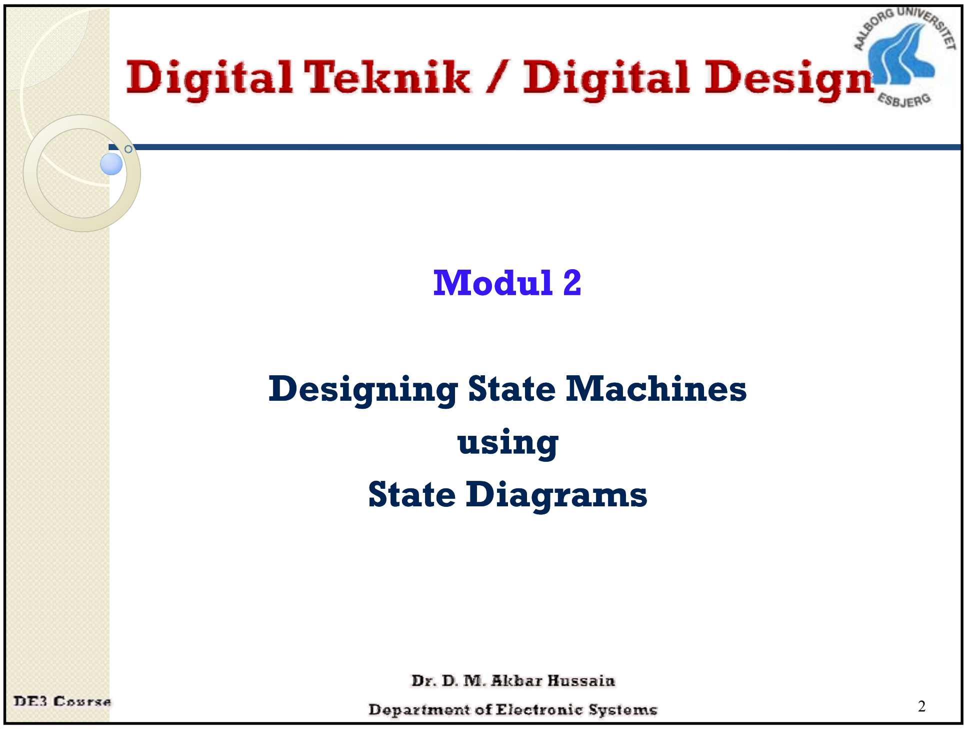
Digital Teknik II / Digital Design II

Dr. D. M. Akbar Hussain
Department of Electronic Systems

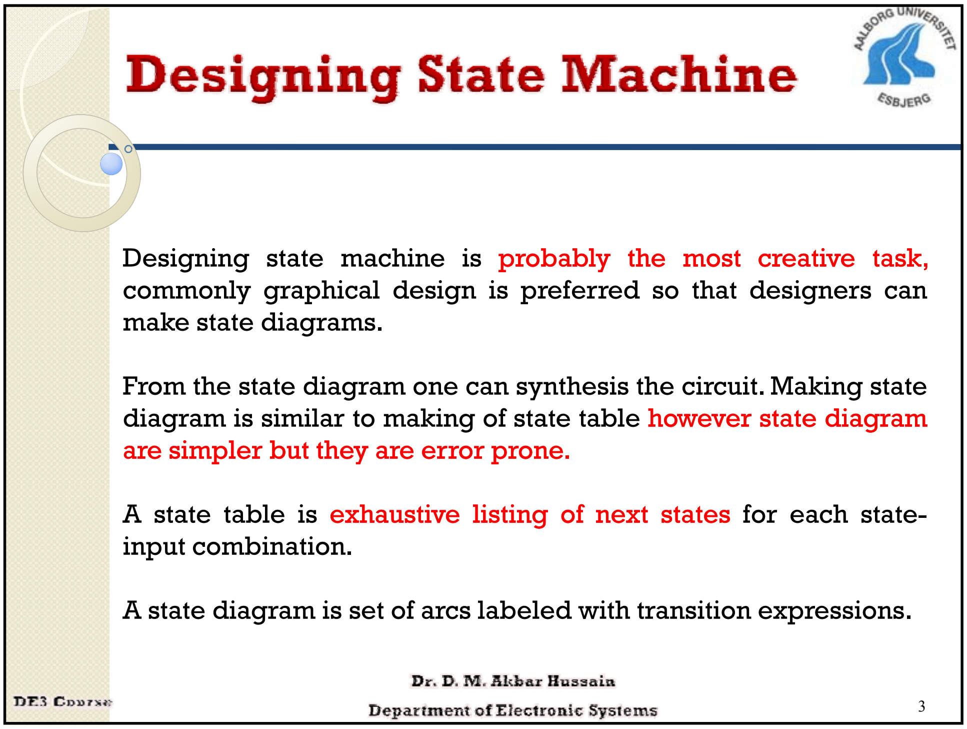


Digital Teknik / Digital Design

Modul 2

**Designing State Machines
using
State Diagrams**

Dr. D. M. Akbar Hussain
Department of Electronic Systems



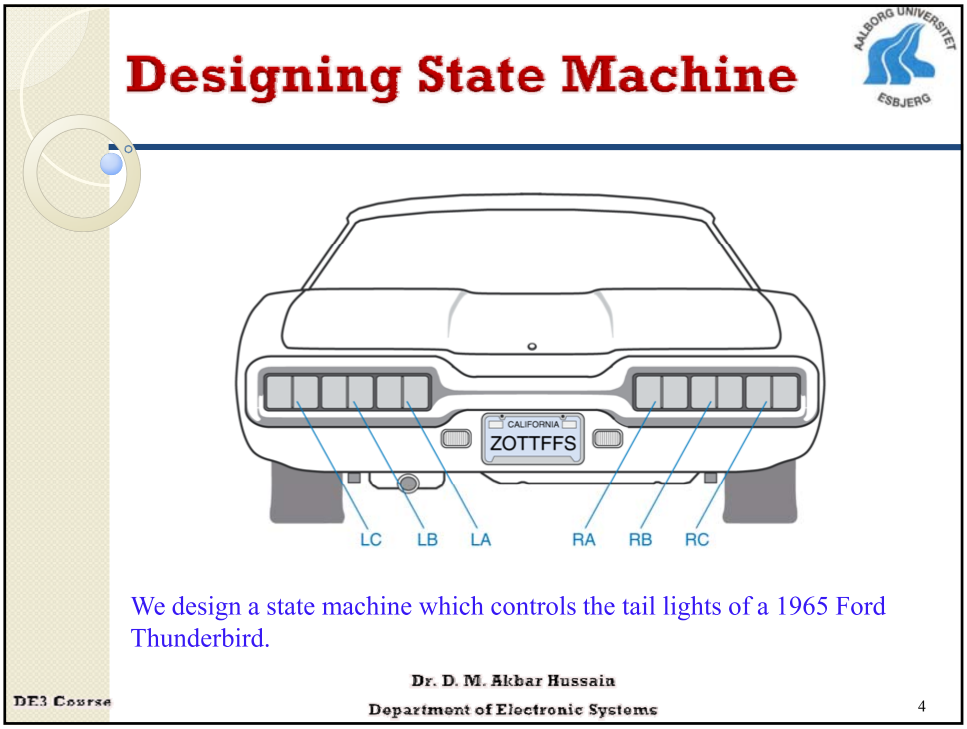
Designing State Machine

Designing state machine is probably the most creative task, commonly graphical design is preferred so that designers can make state diagrams.

From the state diagram one can synthesis the circuit. Making state diagram is similar to making of state table however state diagram are simpler but they are error prone.

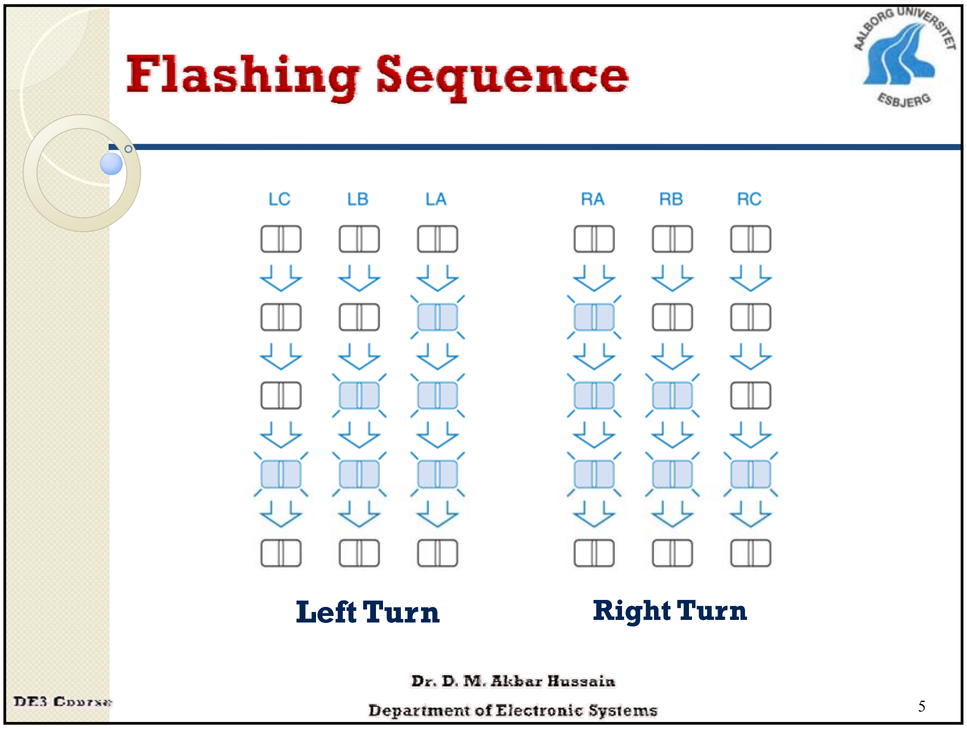
A state table is exhaustive listing of next states for each state-input combination.

A state diagram is set of arcs labeled with transition expressions.



Designing State Machine

We design a state machine which controls the tail lights of a 1965 Ford Thunderbird.



Flashing Sequence

Left Turn

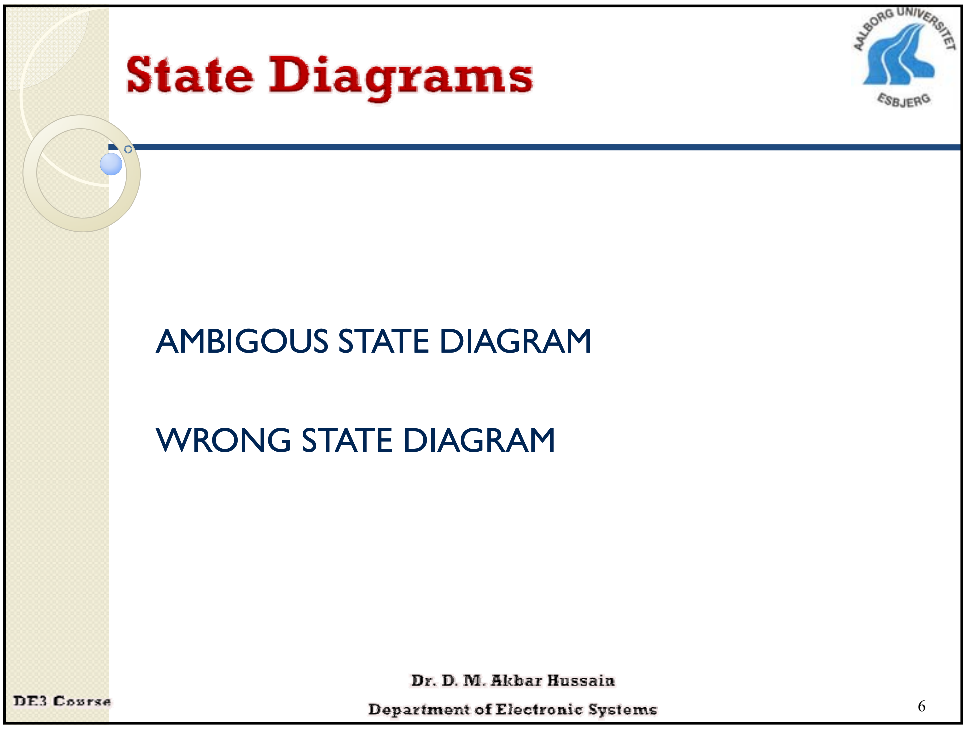
Right Turn

Dr. D. M. Akbar Hussain

Department of Electronic Systems

DE3 Course

5



State Diagrams

AMBIGOUS STATE DIAGRAM

WRONG STATE DIAGRAM

Dr. D. M. Akbar Hussain

Department of Electronic Systems

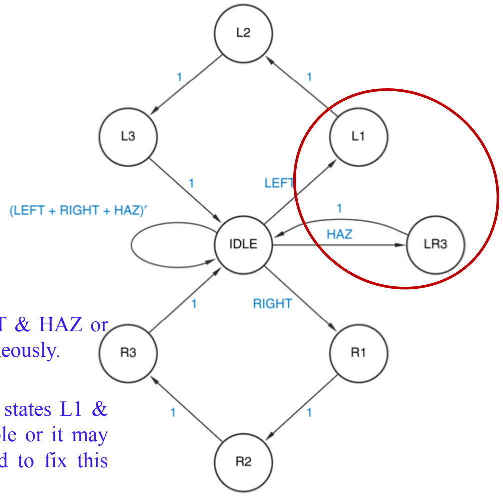
DE3 Course

6

Initial State Diagram & Truth Table

Output Table

State	LC	LB	LA	RA	RB	RC
IDLE	0	0	0	0	0	0
L1	0	0	1	0	0	0
L2	0	1	1	0	0	0
L3	1	1	1	0	0	0
R1	0	0	0	1	0	0
R2	0	0	0	1	1	0
R3	0	0	0	1	1	1
LR3	1	1	1	1	1	1



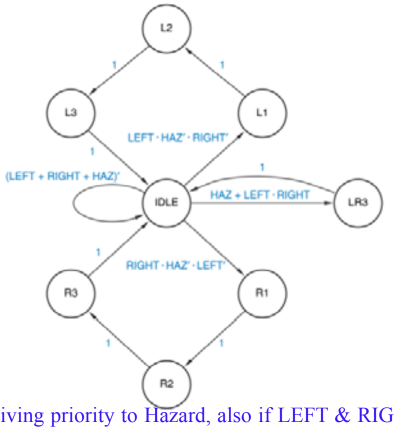
Problem if in the IDLE state LEFT & HAZ or RIGHT & HAZ is asserted simultaneously.

So for the LEFT case it goes to 2 states L1 & LR3 which is wrong and impossible or it may go to an known state. So we need to fix this problem.

Dr. D. M. Akbar Hussain

Department of Electronic Systems

Corrected State Diagram

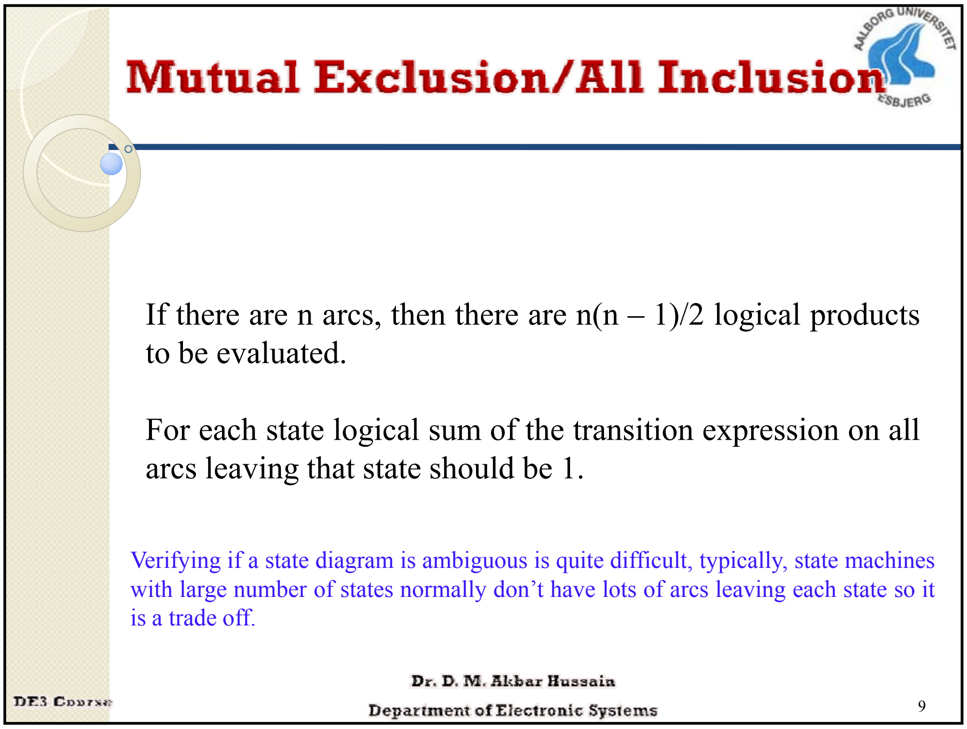


Here we fix it by giving priority to Hazard, also if LEFT & RIGHT both are asserted it will also be considered as hazard.

One problem is that if LEFT or RIGHT is asserted and than hazard is asserted the system will circle through left or right sequence and finally go to hazard.

Dr. D. M. Akbar Hussain

Department of Electronic Systems



Mutual Exclusion/All Inclusion

If there are n arcs, then there are n(n – 1)/2 logical products to be evaluated.

For each state logical sum of the transition expression on all arcs leaving that state should be 1.

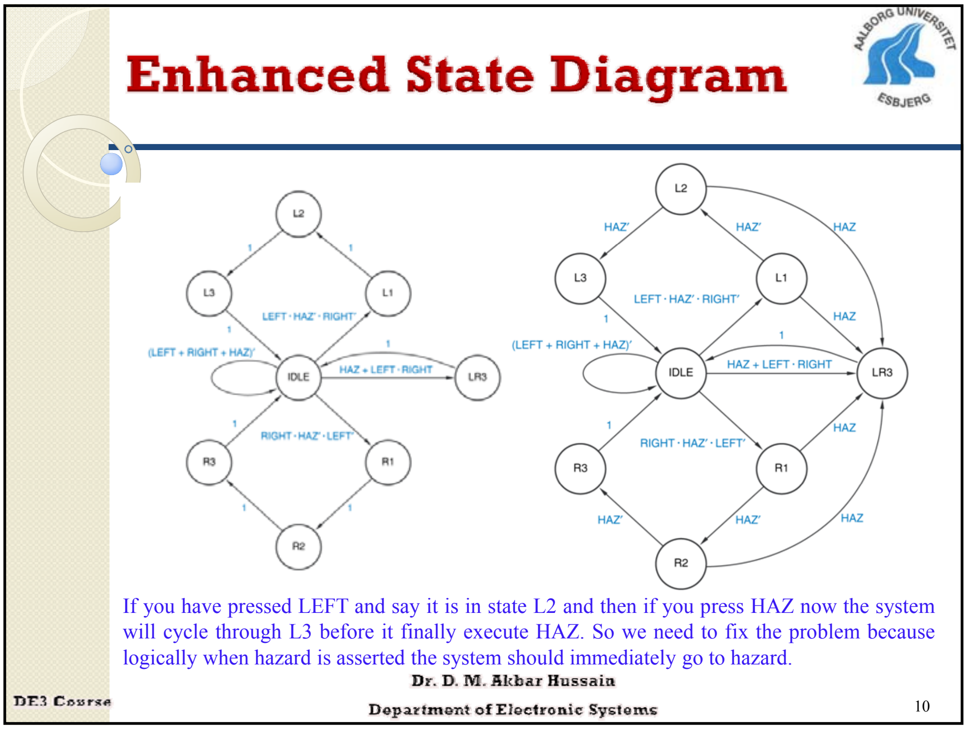
Verifying if a state diagram is ambiguous is quite difficult, typically, state machines with large number of states normally don't have lots of arcs leaving each state so it is a trade off.

DE3 Course

Dr. D. M. Akbar Hussain

Department of Electronic Systems

9



Enhanced State Diagram

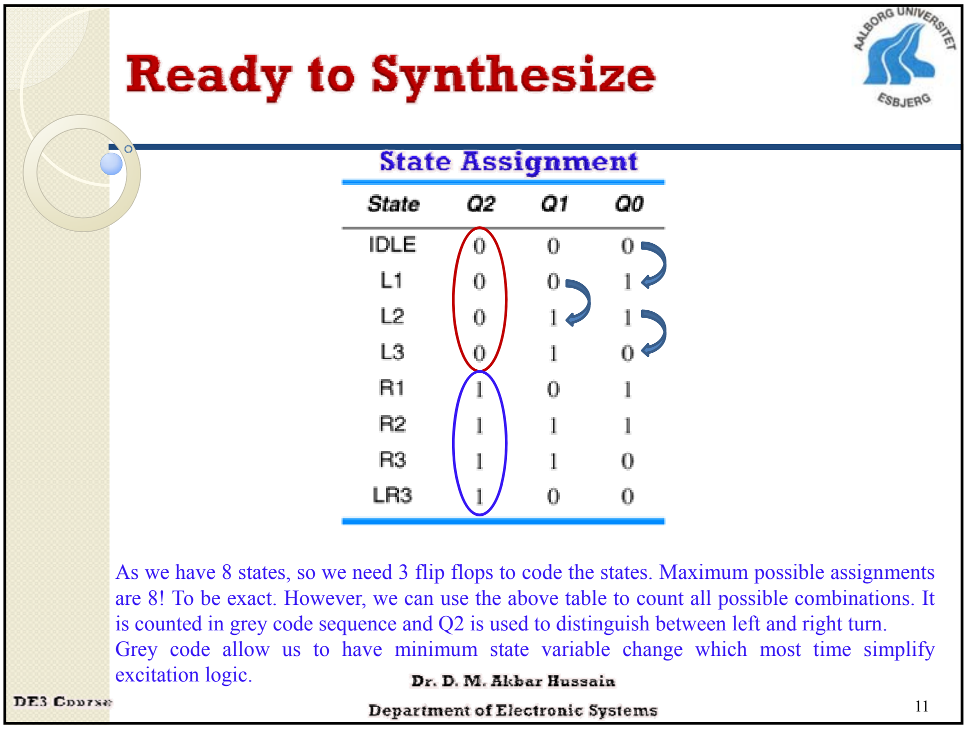
If you have pressed LEFT and say it is in state L2 and then if you press HAZ now the system will cycle through L3 before it finally execute HAZ. So we need to fix the problem because logically when hazard is asserted the system should immediately go to hazard.

Dr. D. M. Akbar Hussain

DE3 Course

Department of Electronic Systems

10



Ready to Synthesize

State Assignment

State	Q2	Q1	Q0
IDLE	0	0	0
L1	0	0	1
L2	0	1	1
L3	0	1	0
R1	1	0	1
R2	1	1	1
R3	1	1	0
LR3	1	0	0

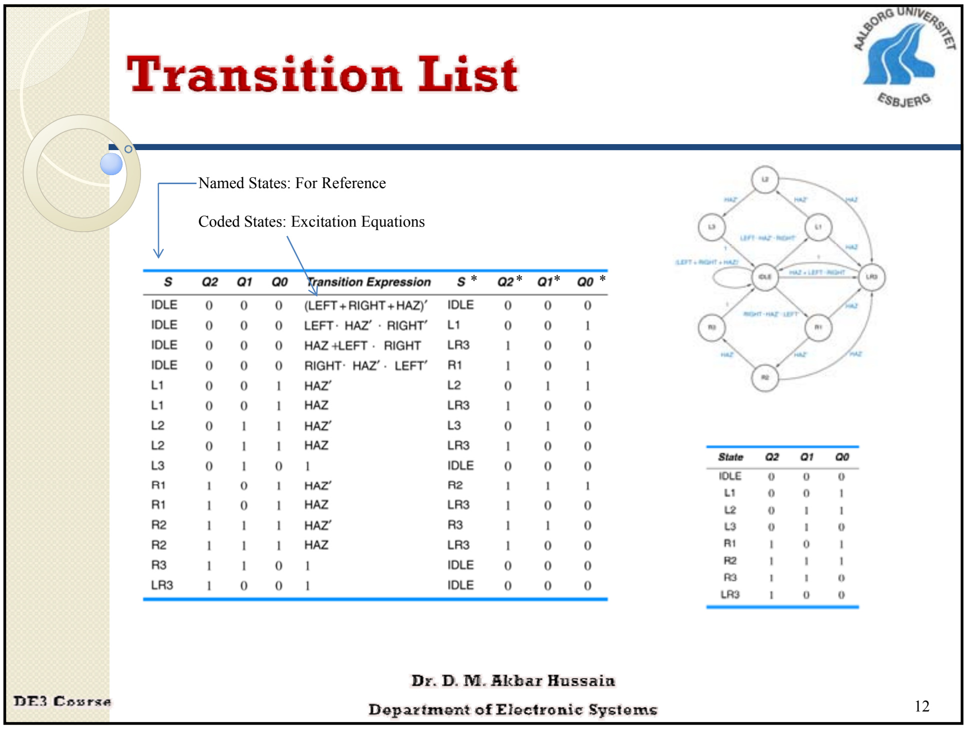
As we have 8 states, so we need 3 flip flops to code the states. Maximum possible assignments are 8! To be exact. However, we can use the above table to count all possible combinations. It is counted in grey code sequence and Q2 is used to distinguish between left and right turn. Grey code allow us to have minimum state variable change which most time simplify excitation logic.

Dr. D. M. Akbar Hussain

Department of Electronic Systems

DE3 Course

11



Transition List

Named States: For Reference

Coded States: Excitation Equations

S	Q2	Q1	Q0	Transition Expression	S *	Q2 *	Q1*	Q0 *
IDLE	0	0	0	(LEFT + RIGHT + HAZ)′	IDLE	0	0	0
IDLE	0	0	0	LEFT · HAZ′ · RIGHT′	L1	0	0	1
IDLE	0	0	0	HAZ +LEFT · RIGHT	LR3	1	0	0
IDLE	0	0	0	RIGHT · HAZ′ · LEFT′	R1	1	0	1
L1	0	0	1	HAZ′	L2	0	1	1
L1	0	0	1	HAZ	LR3	1	0	0
L2	0	1	1	HAZ′	L3	0	1	0
L2	0	1	1	HAZ	LR3	1	0	0
L3	0	1	0	1	IDLE	0	0	0
R1	1	0	1	HAZ′	R2	1	1	1
R1	1	0	1	HAZ	LR3	1	0	0
R2	1	1	1	HAZ′	R3	1	1	0
R2	1	1	1	HAZ	LR3	1	0	0
R3	1	1	0	1	IDLE	0	0	0
LR3	1	0	0	1	IDLE	0	0	0

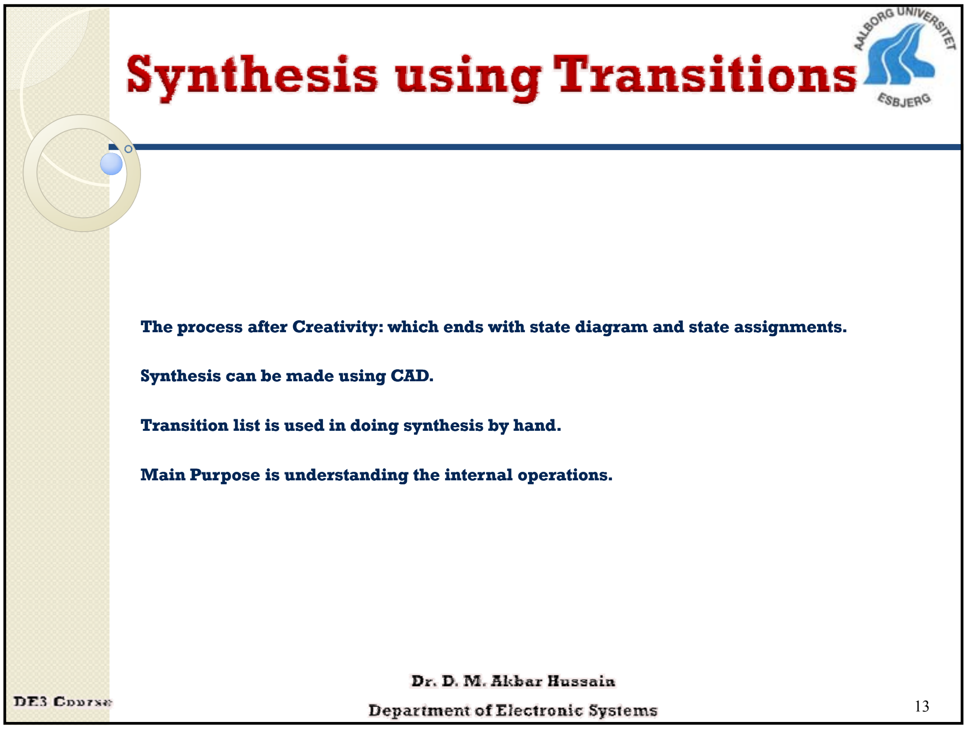
State	Q2	Q1	Q0
IDLE	0	0	0
L1	0	0	1
L2	0	1	1
L3	0	1	0
R1	1	0	1
R2	1	1	1
R3	1	1	0
LR3	1	0	0

Dr. D. M. Akbar Hussain

Department of Electronic Systems

DE3 Course

12



Synthesis using Transitions

The process after Creativity: which ends with state diagram and state assignments.

Synthesis can be made using CAD.

Transition list is used in doing synthesis by hand.

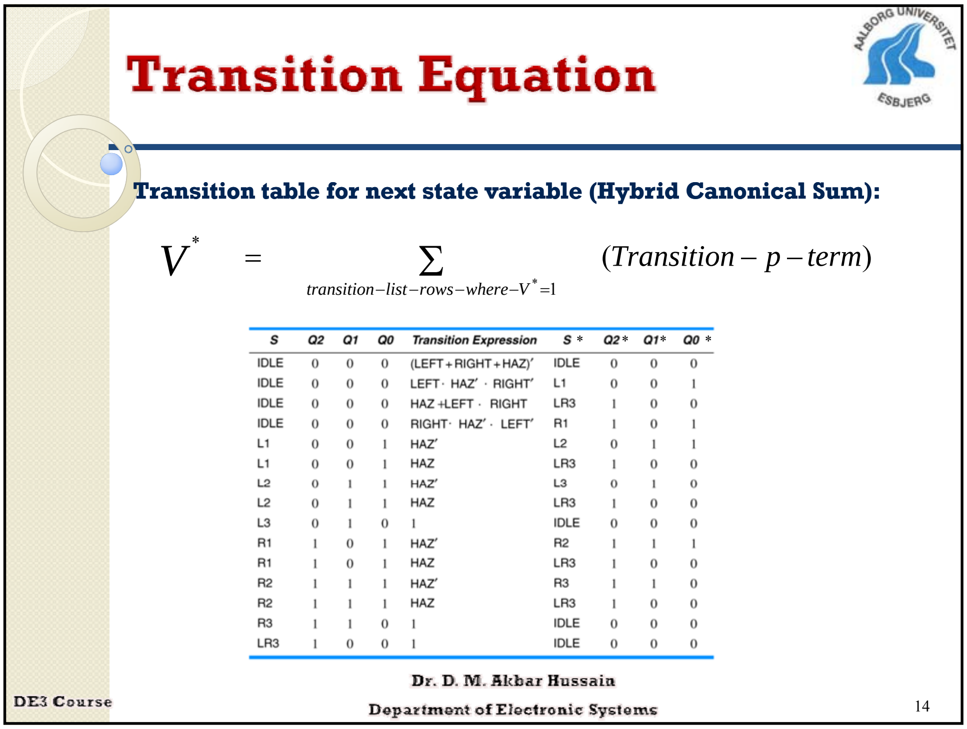
Main Purpose is understanding the internal operations.

DE3 Course

Dr. D. M. Akbar Hussain

Department of Electronic Systems

13



Transition Equation

Transition table for next state variable (Hybrid Canonical Sum):

$$V^{*} = \sum_{transition-list-rows-where-V^{*}=1} (Transition - p - term)$$

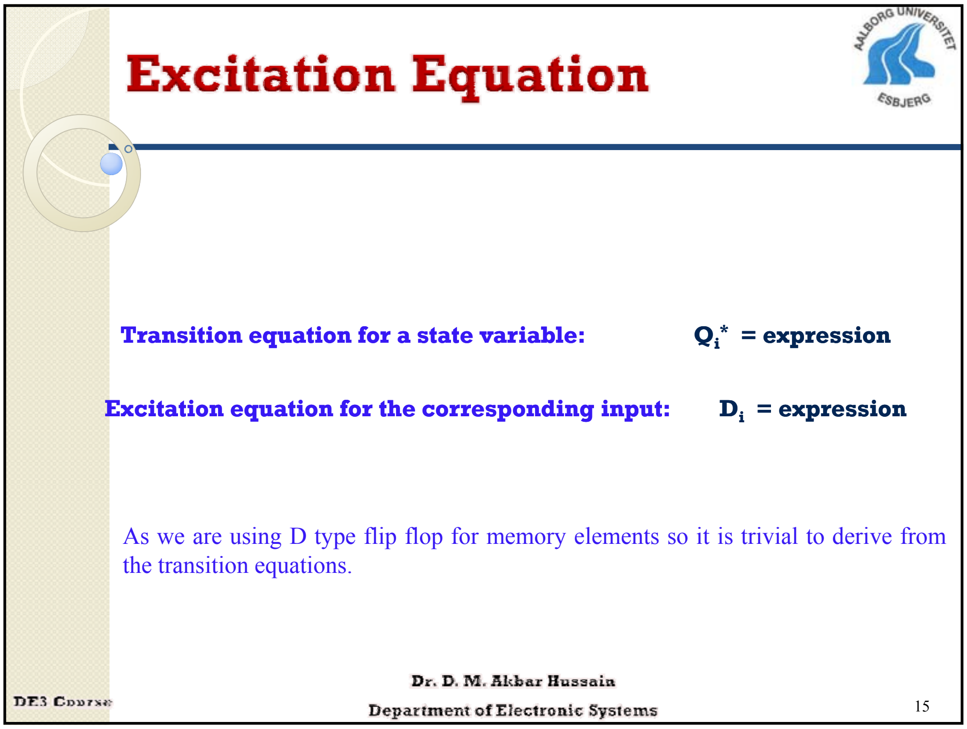
S	Q2	Q1	Q0	Transition Expression	S *	Q2 *	Q1*	Q0 *
IDLE	0	0	0	(LEFT + RIGHT + HAZ)′	IDLE	0	0	0
IDLE	0	0	0	LEFT · HAZ′ · RIGHT′	L1	0	0	1
IDLE	0	0	0	HAZ +LEFT · RIGHT	LR3	1	0	0
IDLE	0	0	0	RIGHT · HAZ′ · LEFT′	R1	1	0	1
L1	0	0	1	HAZ′	L2	0	1	1
L1	0	0	1	HAZ	LR3	1	0	0
L2	0	1	1	HAZ′	L3	0	1	0
L2	0	1	1	HAZ	LR3	1	0	0
L3	0	1	0	1	IDLE	0	0	0
R1	1	0	1	HAZ′	R2	1	1	1
R1	1	0	1	HAZ	LR3	1	0	0
R2	1	1	1	HAZ′	R3	1	1	0
R2	1	1	1	HAZ	LR3	1	0	0
R3	1	1	0	1	IDLE	0	0	0
LR3	1	0	0	1	IDLE	0	0	0

DE3 Course

Dr. D. M. Akbar Hussain

Department of Electronic Systems

14

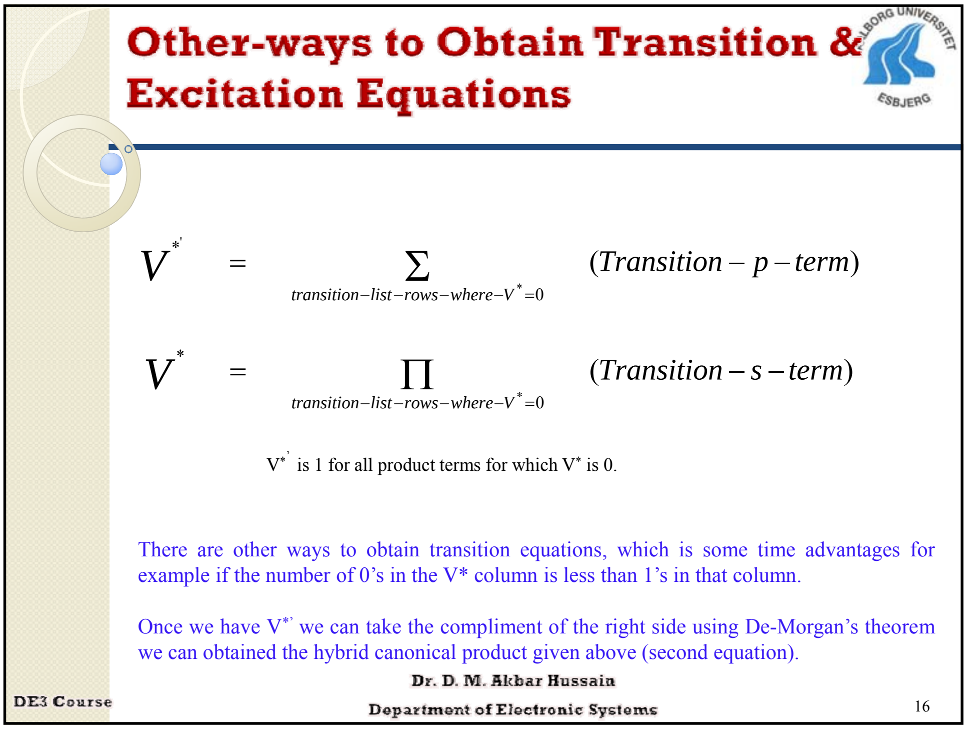


Excitation Equation

Transition equation for a state variable: $Q_i^* = \text{expression}$

Excitation equation for the corresponding input: $D_i = \text{expression}$

As we are using D type flip flop for memory elements so it is trivial to derive from the transition equations.



Other-ways to Obtain Transition & Excitation Equations

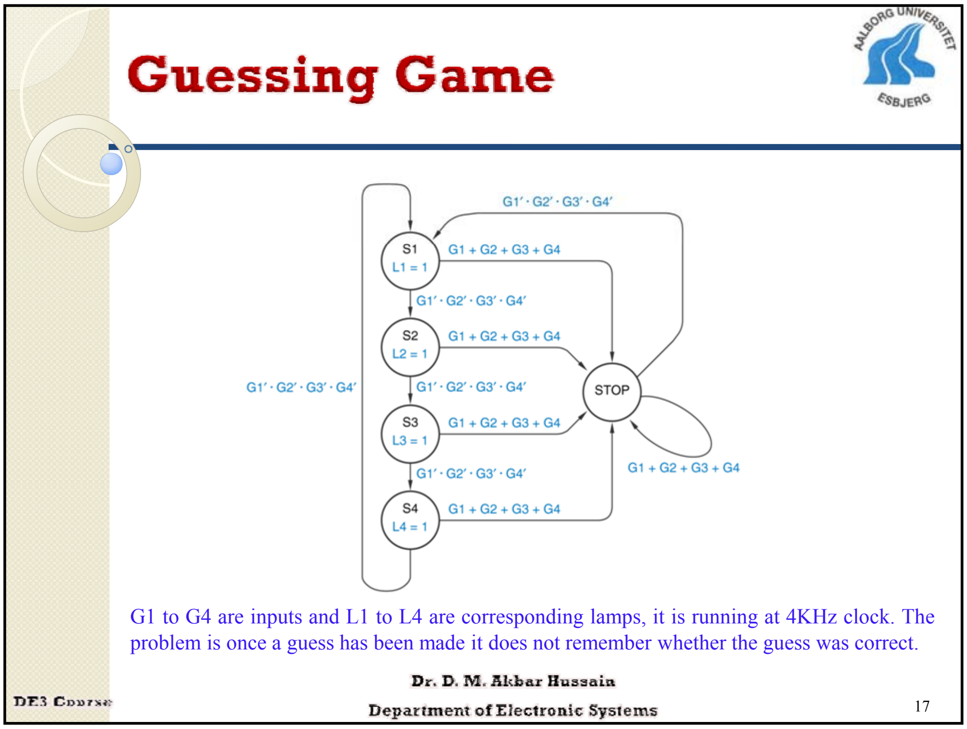
$$V^{*'} = \sum_{transition-list-rows-where-V^*=0} (Transition - p - term)$$

$$V^{*} = \prod_{transition-list-rows-where-V^*=0} (Transition - s - term)$$

$V^{*'}$ is 1 for all product terms for which V^* is 0.

There are other ways to obtain transition equations, which is some time advantages for example if the number of 0's in the V* column is less than 1's in that column.

Once we have $V^{*'}$ we can take the compliment of the right side using De-Morgan's theorem we can obtained the hybrid canonical product given above (second equation).

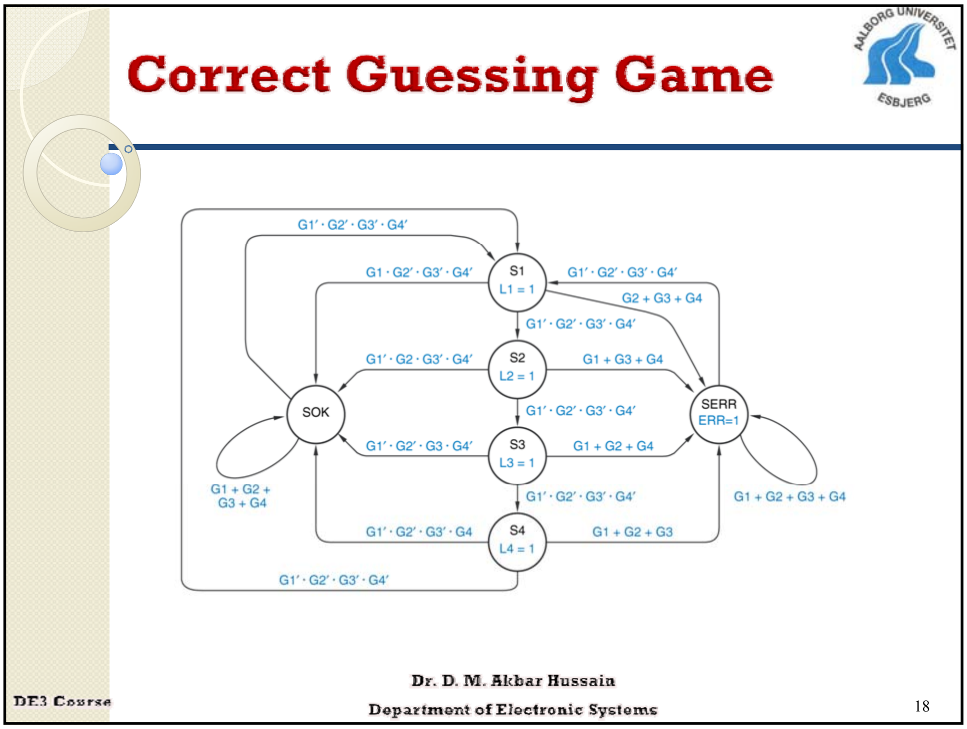


Guessing Game

G1 to G4 are inputs and L1 to L4 are corresponding lamps, it is running at 4KHz clock. The problem is once a guess has been made it does not remember whether the guess was correct.

Dr. D. M. Akbar Hussain

Department of Electronic Systems



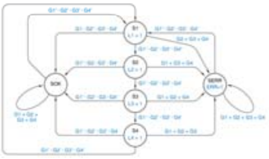
Correct Guessing Game

Dr. D. M. Akbar Hussain

Department of Electronic Systems

Transition List

Current State					Next State				Output				
S	Q2	Q1	Q0	Transition Expression	S	Q2	Q1	Q0	L1	L2	L3	L4	ERR
S1	0	0	0	G1' · G2' · G3' · G4'	S2	0	0	1	1	0	0	0	0
S1	0	0	0	G1 · G2' · G3' · G4'	SOK	1	0	0	1	0	0	0	0
S1	0	0	0	G2 + G3 + G4	SERR	1	0	1	1	0	0	0	0
S2	0	0	1	G1' · G2' · G3' · G4'	S3*	0 *	1 *	1 *	0	1	0	0	0
S2	0	0	1	G1' · G2 · G3' · G4'	SOK	1	0	0	0	1	0	0	0
S2	0	0	1	G1 + G3 + G4	SERR	1	0	1	0	1	0	0	0
S3	0	1	1	G1' · G2' · G3' · G4'	S4	0	1	0	0	0	1	0	0
S3	0	1	1	G1' · G2' · G3 · G4'	SOK	1	0	0	0	0	1	0	0
S3	0	1	1	G1 + G2 + G4	SERR	1	0	1	0	0	1	0	0
S4	0	1	0	G1' · G2' · G3' · G4'	S1	0	0	0	0	0	0	1	0
S4	0	1	0	G1' · G2' · G3' · G4	SOK	1	0	0	0	0	0	1	0
S4	0	1	0	G1 + G2 + G3	SERR	1	0	1	0	0	0	1	0
SOK	1	0	0	G1 + G2 + G3 + G4	SOK	1	0	0	0	0	0	0	0
SOK	1	0	0	G1' · G2' · G3' · G4'	S1	0	0	0	0	0	0	0	0
SERR	1	0	1	G1 + G2 + G3 + G4	SERR	1	0	1	0	0	0	0	1
SERR	1	0	1	G1' · G2' · G3' · G4'	S1	0	0	0	0	0	0	0	1



As there are 6 states so we need 3 flip flop to design such a machine, some (2) states will be unused.

Write down the Transition equations for Q0*, Q1* & Q2*.

For example: Q0* = Q2' . Q1' . Q0 . (G1' . G2' . G3' . G4') + Q2' . Q1 . Q0 . (G1' . G2' . G3' . G4') [corresponding to the line 4 & 7 where Q0* is 1]

Write down also the output equations as it is Moore machine so output equations are independent of transition expressions:

For example L1 = Q2' . Q1' . Q0'

Guessing Game

Unused States: We assume that they have 0 in the corresponding columns when we write equations for Q* as a sum of p-terms.

As a consequence all such states are coded as 1 0 0 corresponding to SOK, which is a safe option.

For small problems when we develop excitation equations using karnaugh map for minimization and putting a d in all don't care cells is ok, however, larger problems cannot be taken cared by karnaugh map, commercially available logic minimization tools are available to deal with larger problems.

Output Coded State Assignments

Current State							Next State					
S	*L1*	*L2*	*L3*	*L4*	*ERR*	*Transition Expression*	*S*	*L1**	*L2**	*L3**	*L4**	*ERR**
S1	1	0	0	0	0	G1′ · G2′ · G3′ · G4′	S2	0	1	0	0	0
S1	1	0	0	0	0	G1· G2′ · G3′ · G4′	SOK	0	0	0	0	0
S1	1	0	0	0	0	G2 + G3 + G4	SERR	0	0	0	0	1
S2	0	1	0	0	0	G1′ · G2′ · G3′ · G4′	S3	0	0	1	0	0
S2	0	1	0	0	0	G1′ · G2· G3′ · G4′	SOK	0	0	0	0	0
S2	0	1	0	0	0	G1 + G3 + G4	SERR	0	0	0	0	1
S3	0	0	1	0	0	G1′ · G2′ · G3′ · G4′	S4	0	0	0	1	0
S3	0	0	1	0	0	G1′ · G2′ · G3· G4′	SOK	0	0	0	0	0
S3	0	0	1	0	0	G1 + G2 + G4	SERR	0	0	0	0	1
S4	0	0	0	1	0	G1′ · G2′ · G3′ · G4′	S1	1	0	0	0	0
S4	0	0	0	1	0	G1′ · G2′ · G3′ · G4	SOK	0	0	0	0	0
S4	0	0	0	1	0	G1 + G2 + G3	SERR	0	0	0	0	1
SOK	0	0	0	0	0	G1 + G2 + G3 + G4	SOK	0	0	0	0	0
SOK	0	0	0	0	0	G1′ · G2′ · G3′ · G4′	S1	1	0	0	0	0
SERR	0	0	0	0	1	G1 + G2 + G3 + G4	SERR	0	0	0	0	1
SERR	0	0	0	0	1	G1′ · G2′ · G3′ · G4′	S1	1	0	0	0	0

As the machine is a Moore machine meaning output is dependent on state only, we can use the output as state variables this type of output coded state assignment could result in simpler excitation equations. For example for L1*

L1* = L1’ . L2’ . L3’. L4 . ERR’ . (G1’ . G2’ . G3’ . G4’) + L1’ . L2’ . L3’. L4’ . ERR’ . (G1’ . G2’ . G3’ . G4’) + L1’ . L2’ . L3’. L4’ . ERR . (G1’ . G2’ . G3’ . G4’)

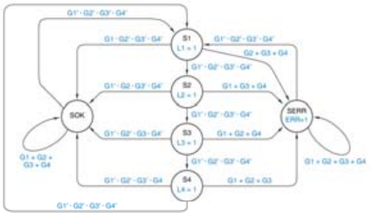
Compare with earlier, ERR is the worst having 16 terms.

Dr. D. M. Akbar Hussain

Department of Electronic Systems

Current Assignments using Don’t care

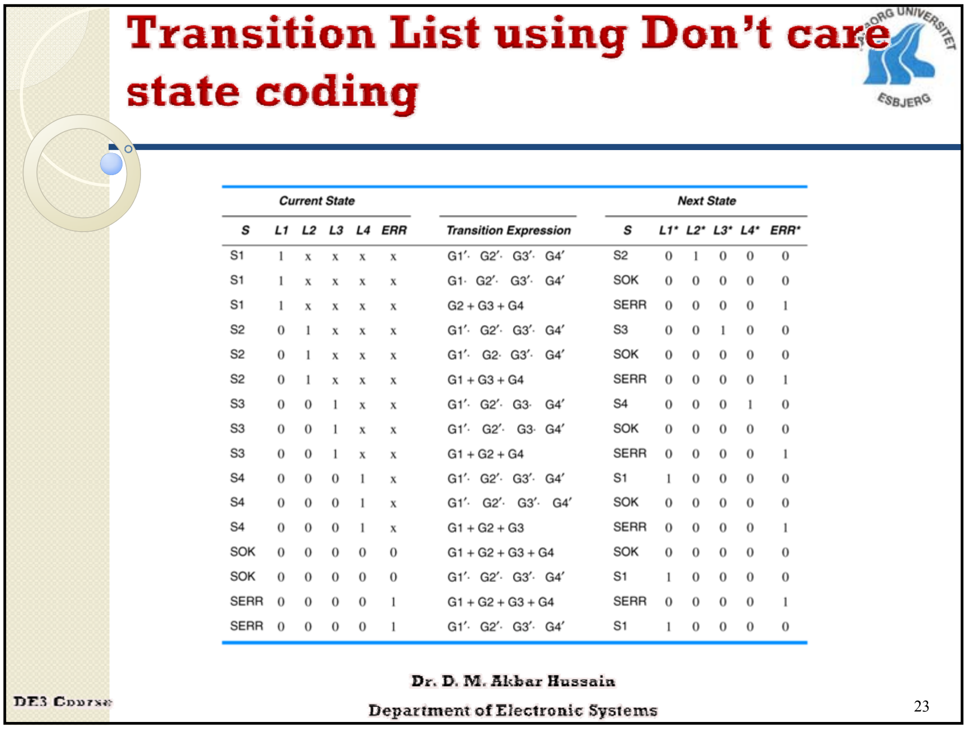
State	*L1*	*L2*	*L3*	*L4*	*ERR*
S1	1	x	x	x	x
S2	0	1	x	x	x
S3	0	0	1	x	x
S4	0	0	0	1	x
SOK	0	0	0	0	0
SERR	0	0	0	0	1



As there are 5 state variables so possible states are 32 but we are only using 6, so rest of the states are unused and have a next state 00000 (SOK).

Dr. D. M. Akbar Hussain

Department of Electronic Systems



Transition List using Don't care state coding

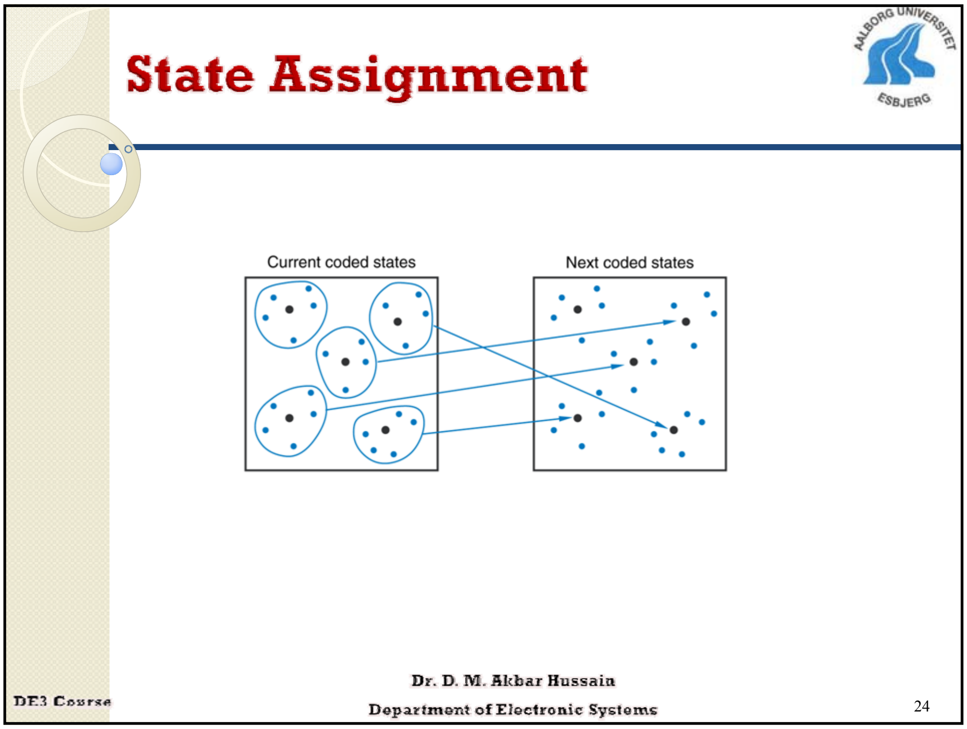
Current State							Next State					
S	*L1*	*L2*	*L3*	*L4*	*ERR*	*Transition Expression*	*S*	*L1**	*L2**	*L3**	*L4**	*ERR**
S1	1	x	x	x	x	$G1' \cdot G2' \cdot G3' \cdot G4'$	S2	0	1	0	0	0
S1	1	x	x	x	x	$G1 \cdot G2' \cdot G3' \cdot G4'$	SOK	0	0	0	0	0
S1	1	x	x	x	x	$G2 + G3 + G4$	SERR	0	0	0	0	1
S2	0	1	x	x	x	$G1' \cdot G2' \cdot G3' \cdot G4'$	S3	0	0	1	0	0
S2	0	1	x	x	x	$G1' \cdot G2 \cdot G3' \cdot G4'$	SOK	0	0	0	0	0
S2	0	1	x	x	x	$G1 + G3 + G4$	SERR	0	0	0	0	1
S3	0	0	1	x	x	$G1' \cdot G2' \cdot G3 \cdot G4'$	S4	0	0	0	1	0
S3	0	0	1	x	x	$G1' \cdot G2' \cdot G3 \cdot G4'$	SOK	0	0	0	0	0
S3	0	0	1	x	x	$G1 + G2 + G4$	SERR	0	0	0	0	1
S4	0	0	0	1	x	$G1' \cdot G2' \cdot G3' \cdot G4'$	S1	1	0	0	0	0
S4	0	0	0	1	x	$G1' \cdot G2' \cdot G3' \cdot G4'$	SOK	0	0	0	0	0
S4	0	0	0	1	x	$G1 + G2 + G3$	SERR	0	0	0	0	1
SOK	0	0	0	0	0	$G1 + G2 + G3 + G4$	SOK	0	0	0	0	0
SOK	0	0	0	0	0	$G1' \cdot G2' \cdot G3' \cdot G4'$	S1	1	0	0	0	0
SERR	0	0	0	0	1	$G1 + G2 + G3 + G4$	SERR	0	0	0	0	1
SERR	0	0	0	0	1	$G1' \cdot G2' \cdot G3' \cdot G4'$	S1	1	0	0	0	0

DE3 Course

Dr. D. M. Akbar Hussain

Department of Electronic Systems

23



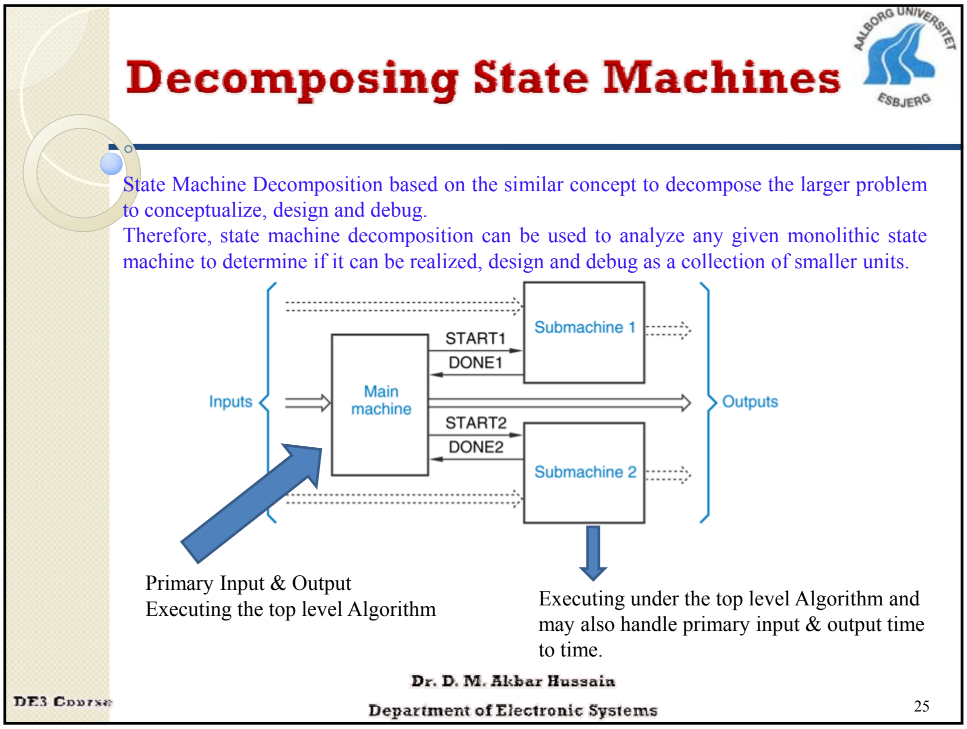
State Assignment

DE3 Course

Dr. D. M. Akbar Hussain

Department of Electronic Systems

24



Decomposing State Machines

State Machine Decomposition based on the similar concept to decompose the larger problem to conceptualize, design and debug.

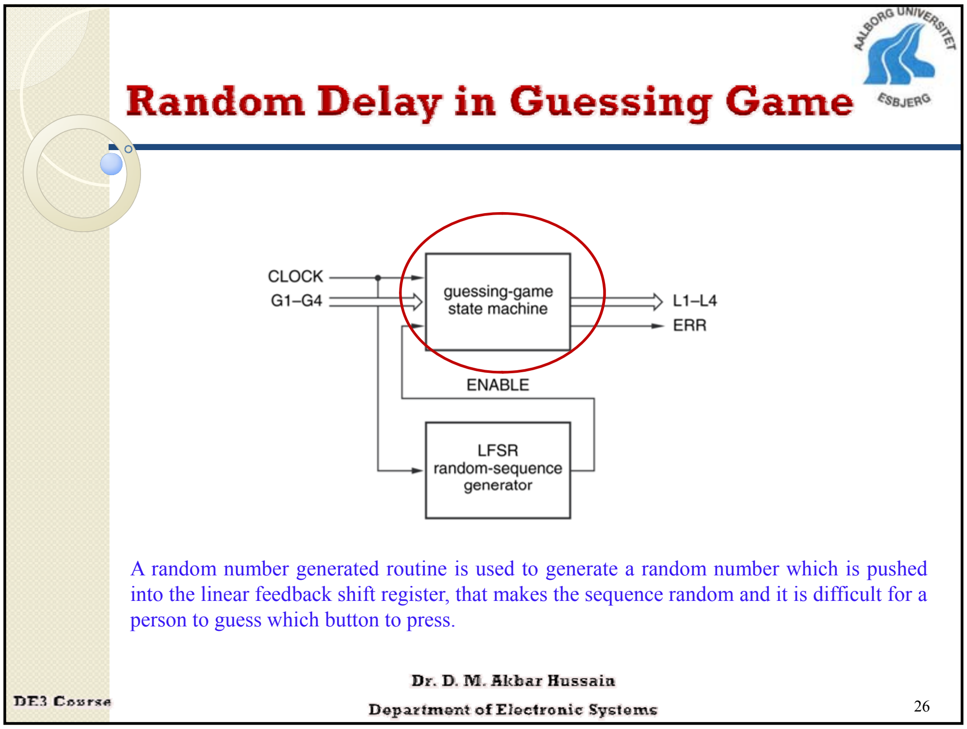
Therefore, state machine decomposition can be used to analyze any given monolithic state machine to determine if it can be realized, design and debug as a collection of smaller units.

Primary Input & Output
Executing the top level Algorithm

Executing under the top level Algorithm and may also handle primary input & output time to time.

Dr. D. M. Akbar Hussain

Department of Electronic Systems

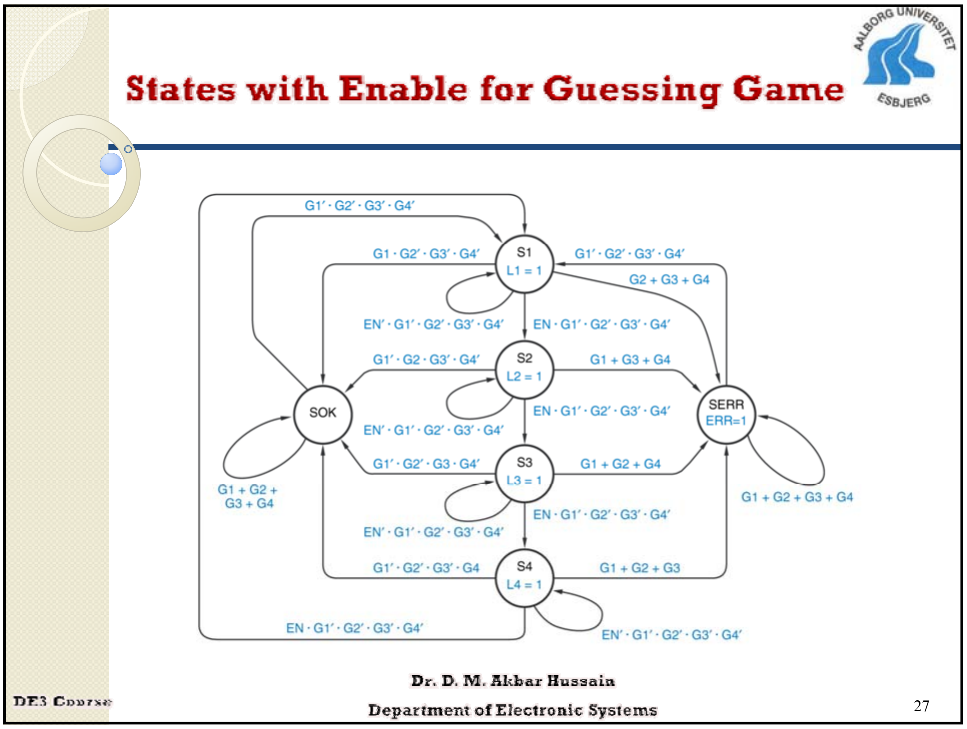


Random Delay in Guessing Game

A random number generated routine is used to generate a random number which is pushed into the linear feedback shift register, that makes the sequence random and it is difficult for a person to guess which button to press.

Dr. D. M. Akbar Hussain

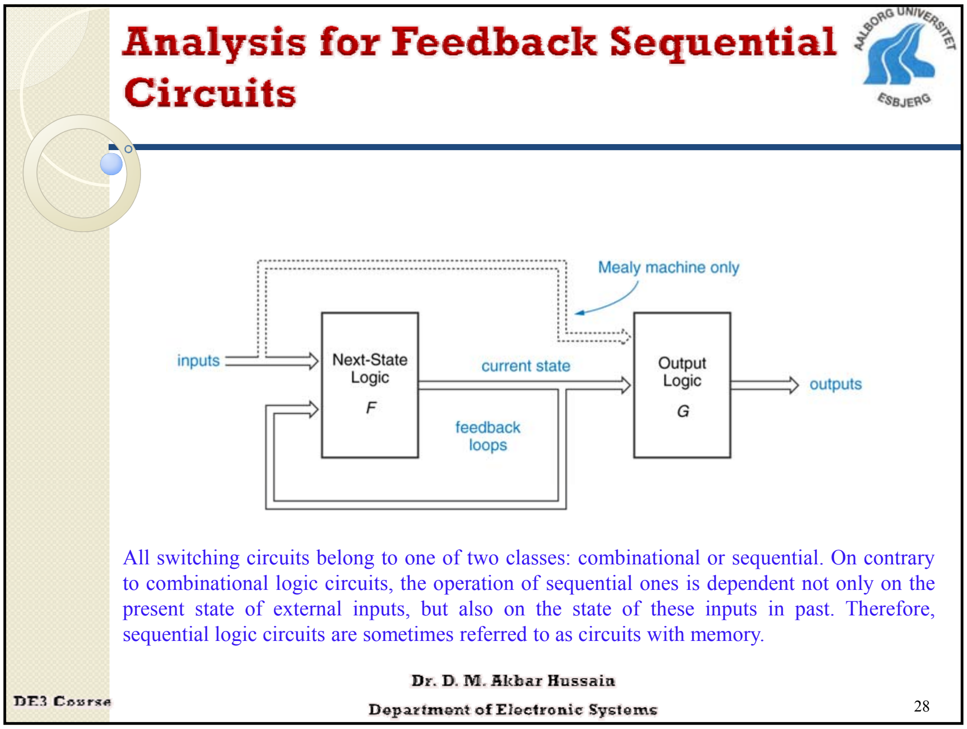
Department of Electronic Systems



States with Enable for Guessing Game

Dr. D. M. Akbar Hussain

Department of Electronic Systems

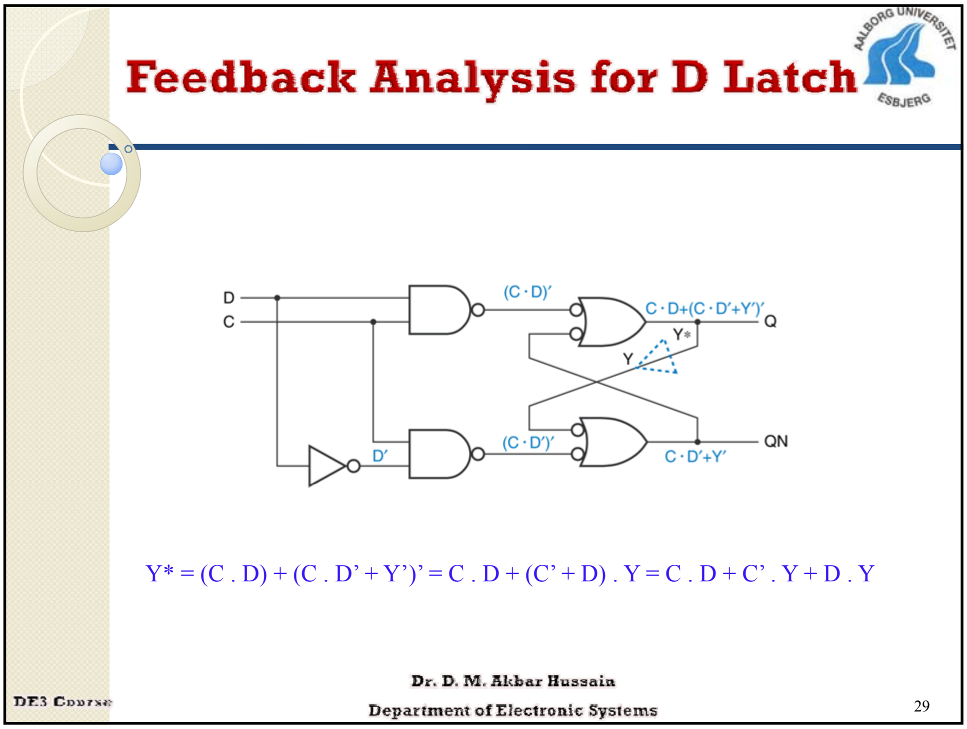


Analysis for Feedback Sequential Circuits

All switching circuits belong to one of two classes: combinational or sequential. On contrary to combinational logic circuits, the operation of sequential ones is dependent not only on the present state of external inputs, but also on the state of these inputs in past. Therefore, sequential logic circuits are sometimes referred to as circuits with memory.

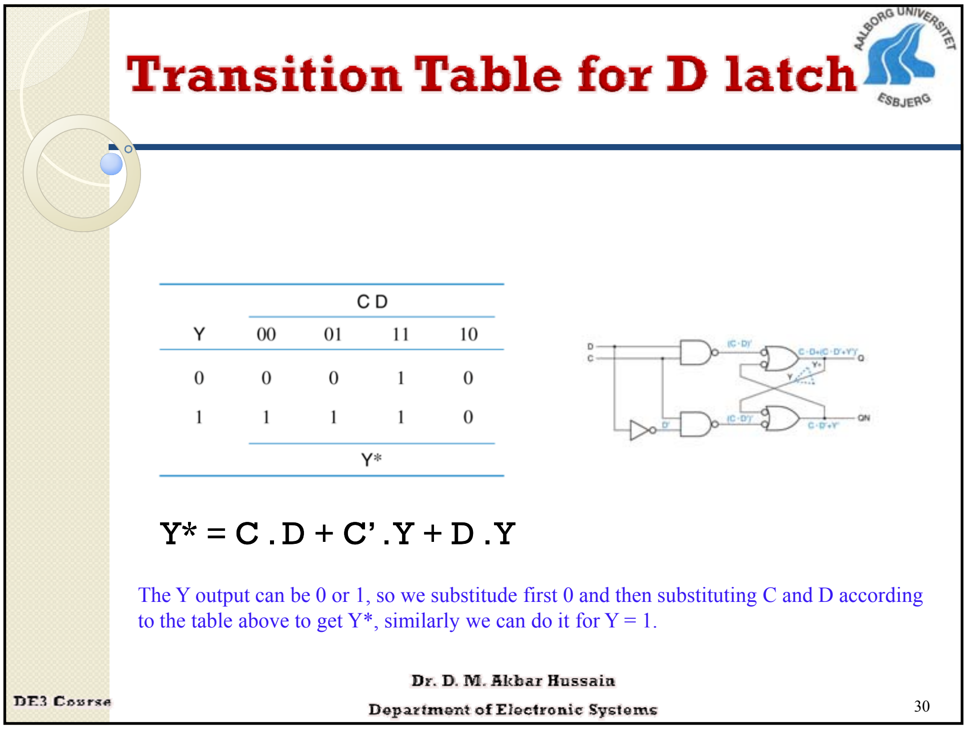
Dr. D. M. Akbar Hussain

Department of Electronic Systems



Feedback Analysis for D Latch

$$Y^* = (C \,.\, D) + (C \,.\, D' + Y')' = C \,.\, D + (C' + D) \,.\, Y = C \,.\, D + C' \,.\, Y + D \,.\, Y$$

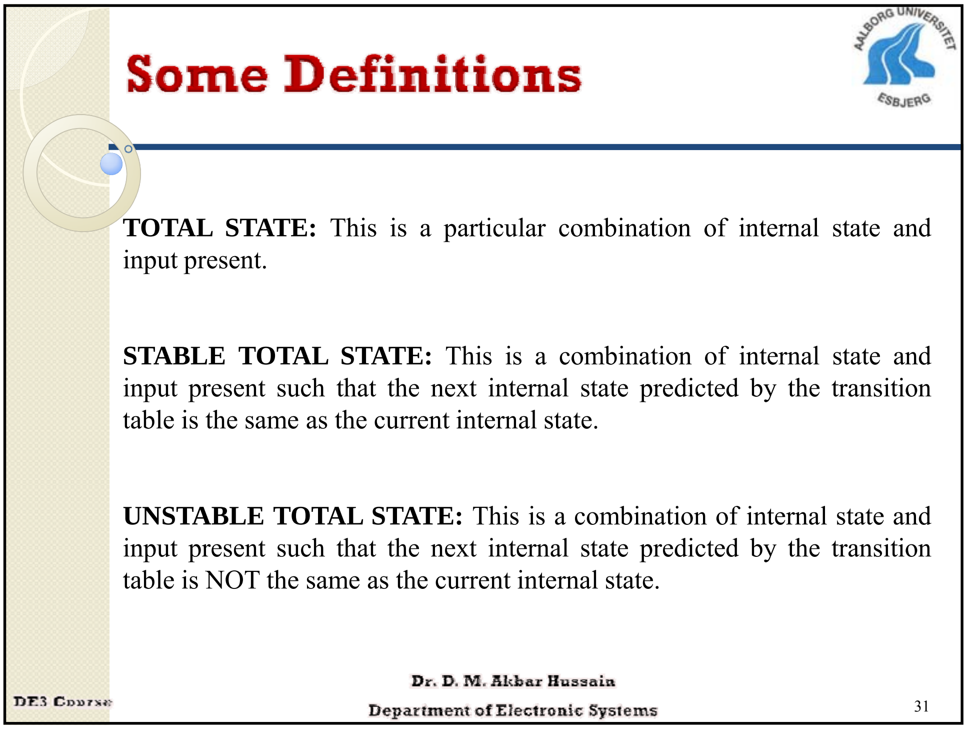


Transition Table for D latch

	CD			
Y	00	01	11	10
0	0	0	1	0
1	1	1	1	0
	Y*			

$$Y^* = C \,.\, D + C' \,.\, Y + D \,.\, Y$$

The Y output can be 0 or 1, so we substitude first 0 and then substituting C and D according to the table above to get Y*, similarly we can do it for Y = 1.

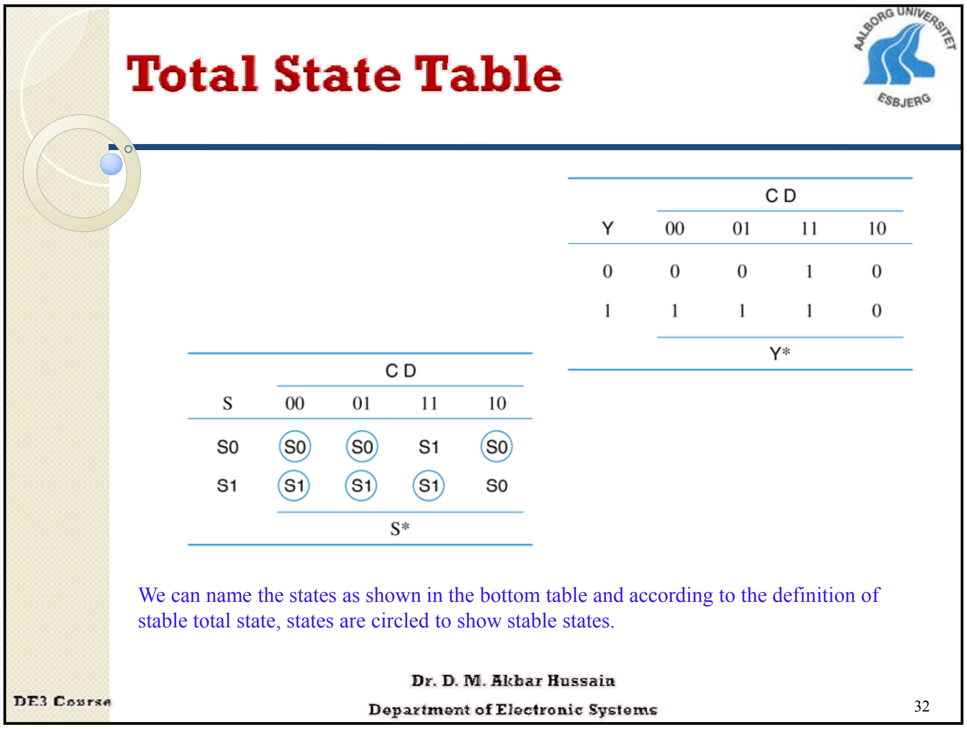


Some Definitions

TOTAL STATE: This is a particular combination of internal state and input present.

STABLE TOTAL STATE: This is a combination of internal state and input present such that the next internal state predicted by the transition table is the same as the current internal state.

UNSTABLE TOTAL STATE: This is a combination of internal state and input present such that the next internal state predicted by the transition table is NOT the same as the current internal state.

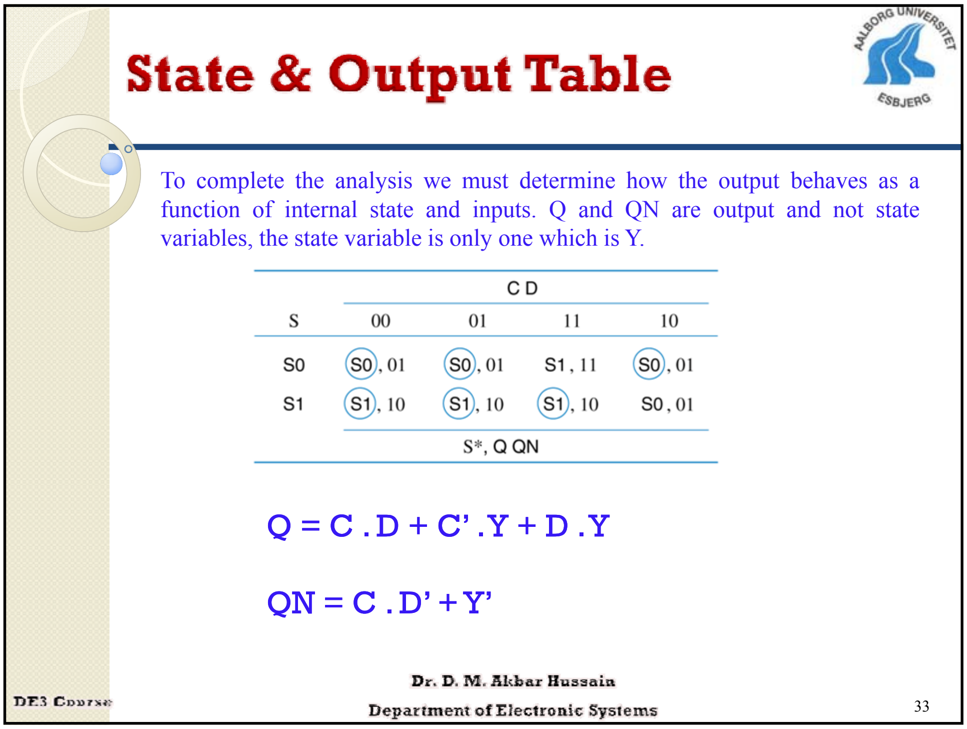


Total State Table

	C D			
Y	00	01	11	10
0	0	0	1	0
1	1	1	1	0
	Y*			

	C D			
S	00	01	11	10
S0	S0	S0	S1	S0
S1	S1	S1	S1	S0
	S*			

We can name the states as shown in the bottom table and according to the definition of stable total state, states are circled to show stable states.



State & Output Table

To complete the analysis we must determine how the output behaves as a function of internal state and inputs. Q and QN are output and not state variables, the state variable is only one which is Y.

	C D			
S	00	01	11	10
S0	S0, 01	S0, 01	S1, 11	S0, 01
S1	S1, 10	S1, 10	S1, 10	S0, 01
	S*, Q QN			

$$Q = C \,.\, D + C' \,.\, Y + D \,.\, Y$$

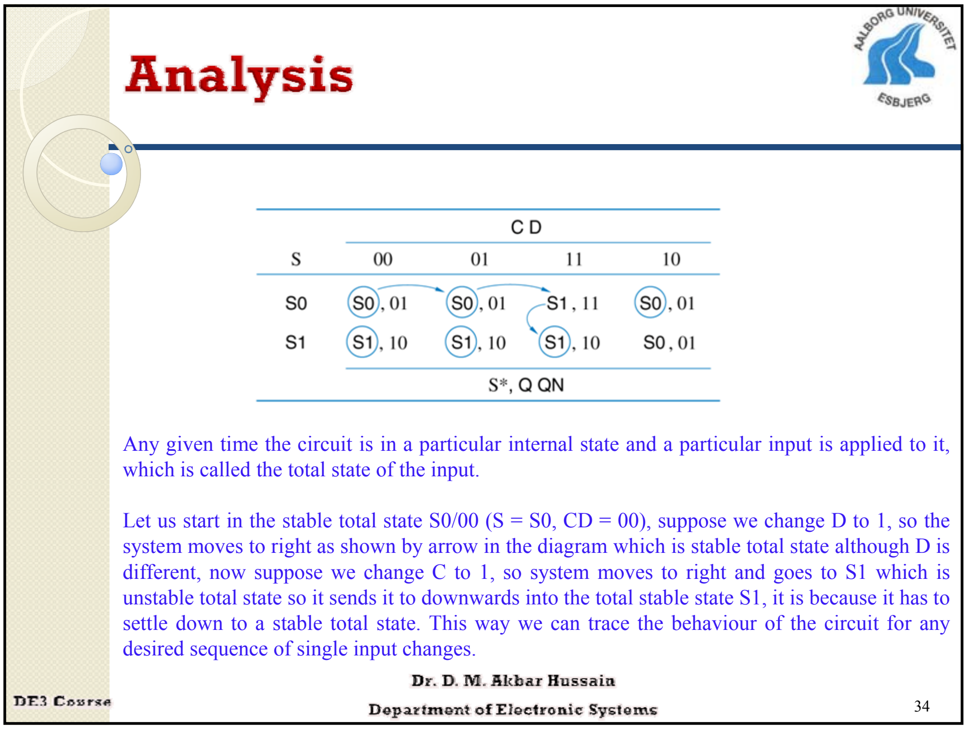
$$QN = C \,.\, D' + Y'$$

DE3 Course

Dr. D. M. Akbar Hussain

Department of Electronic Systems

33



Analysis

	C D			
S	00	01	11	10
S0	S0, 01	S0, 01	S1, 11	S0, 01
S1	S1, 10	S1, 10	S1, 10	S0, 01
	S*, Q QN			

Any given time the circuit is in a particular internal state and a particular input is applied to it, which is called the total state of the input.

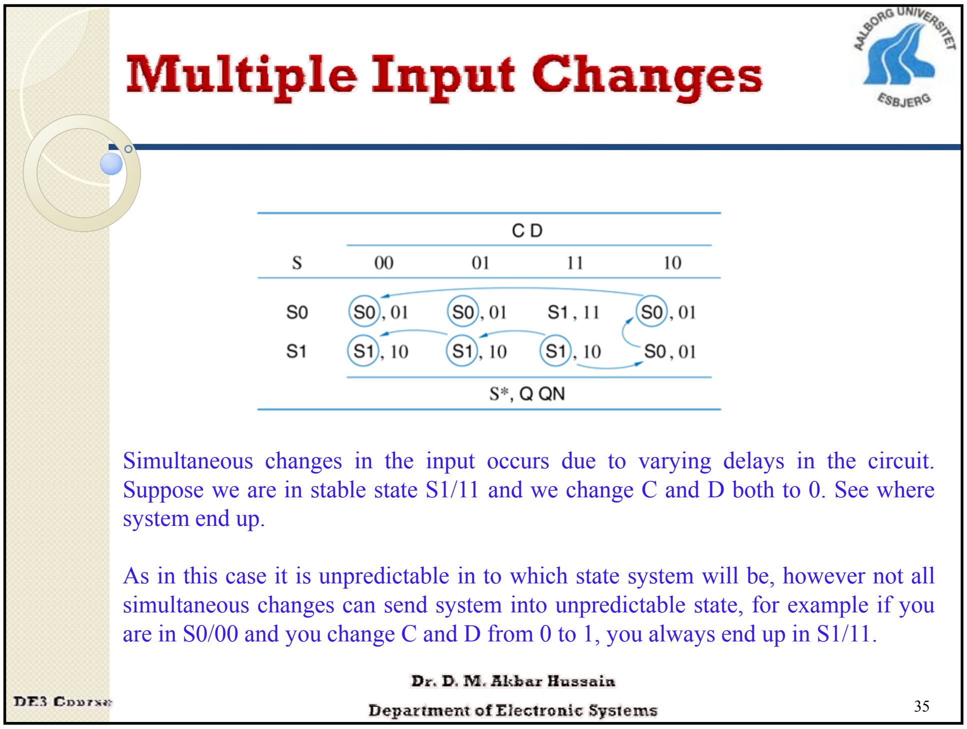
Let us start in the stable total state S0/00 (S = S0, CD = 00), suppose we change D to 1, so the system moves to right as shown by arrow in the diagram which is stable total state although D is different, now suppose we change C to 1, so system moves to right and goes to S1 which is unstable total state so it sends it to downwards into the total stable state S1, it is because it has to settle down to a stable total state. This way we can trace the behaviour of the circuit for any desired sequence of single input changes.

Dr. D. M. Akbar Hussain

DE3 Course

Department of Electronic Systems

34

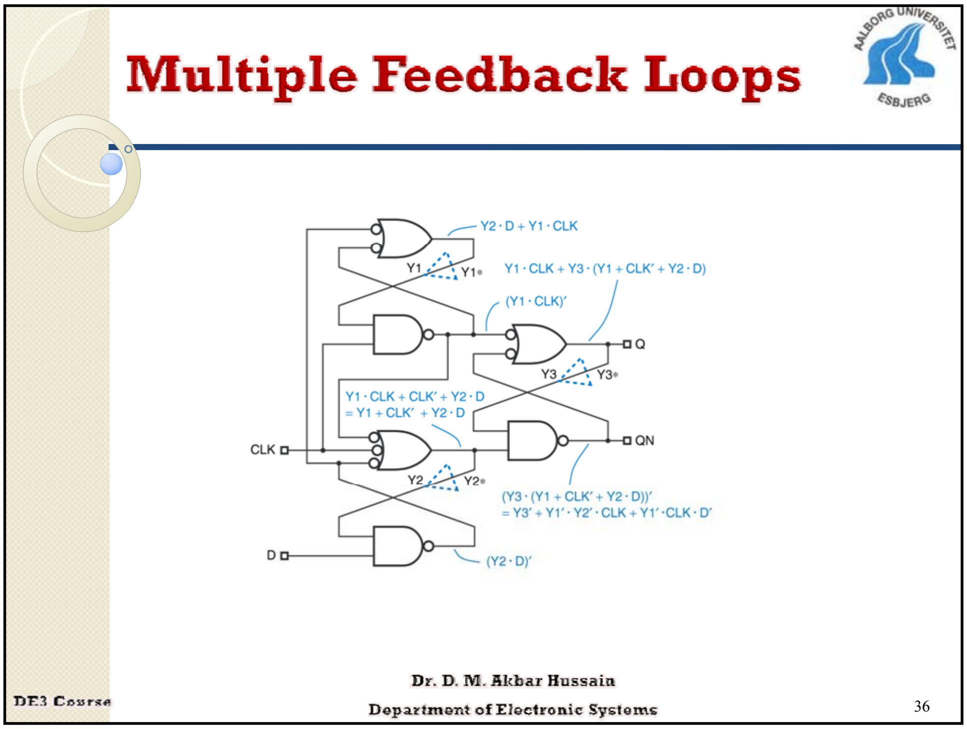


Multiple Input Changes

	C D			
S	00	01	11	10
S0	S0 , 01	S0 , 01	S1 , 11	S0 , 01
S1	S1 , 10	S1 , 10	S1 , 10	S0 , 01
	S*, Q QN			

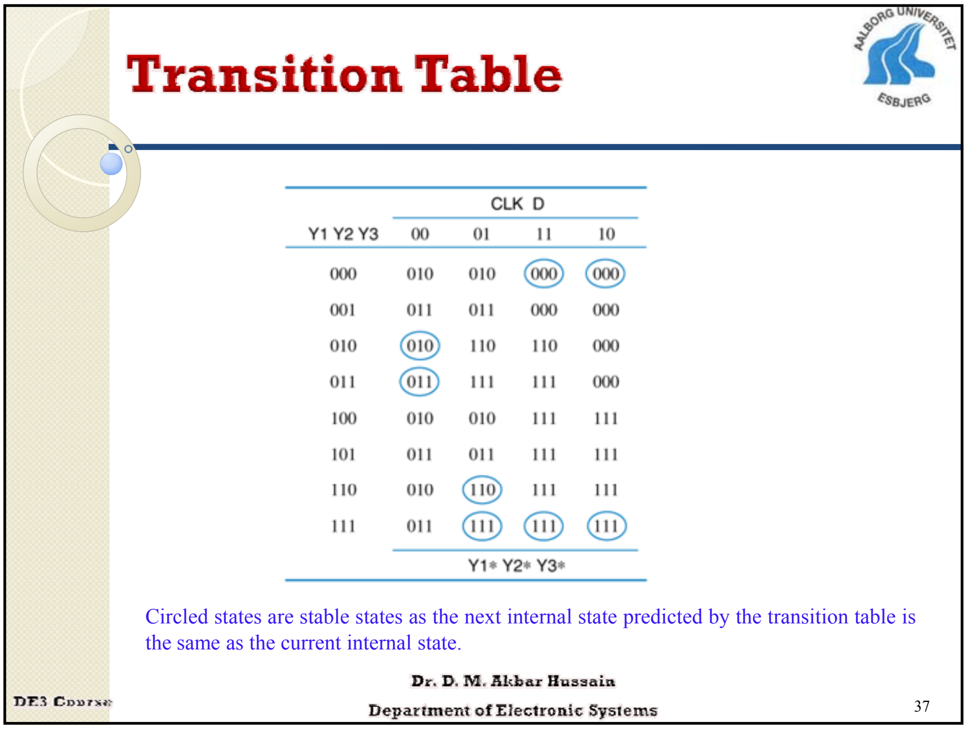
Simultaneous changes in the input occurs due to varying delays in the circuit. Suppose we are in stable state S1/11 and we change C and D both to 0. See where system end up.

As in this case it is unpredictable in to which state system will be, however not all simultaneous changes can send system into unpredictable state, for example if you are in S0/00 and you change C and D from 0 to 1, you always end up in S1/11.



Multiple Feedback Loops

Y2 · D + Y1 · CLK
Y1
Y1*
Y1 · CLK + Y3 · (Y1 + CLK′ + Y2 · D)
(Y1 · CLK)′
Q
Y3
Y3*
Y1 · CLK + CLK′ + Y2 · D
= Y1 + CLK′ + Y2 · D
CLK
QN
Y2
Y2*
(Y3 · (Y1 + CLK′ + Y2 · D))′
= Y3′ + Y1′ · Y2′ · CLK + Y1′ ·CLK · D′
D
(Y2 · D)′



Transition Table

	CLK D			
Y1 Y2 Y3	00	01	11	10
000	010	010	000	000
001	011	011	000	000
010	010	110	110	000
011	011	111	111	000
100	010	010	111	111
101	011	011	111	111
110	010	110	111	111
111	011	111	111	111
	Y1* Y2* Y3*			

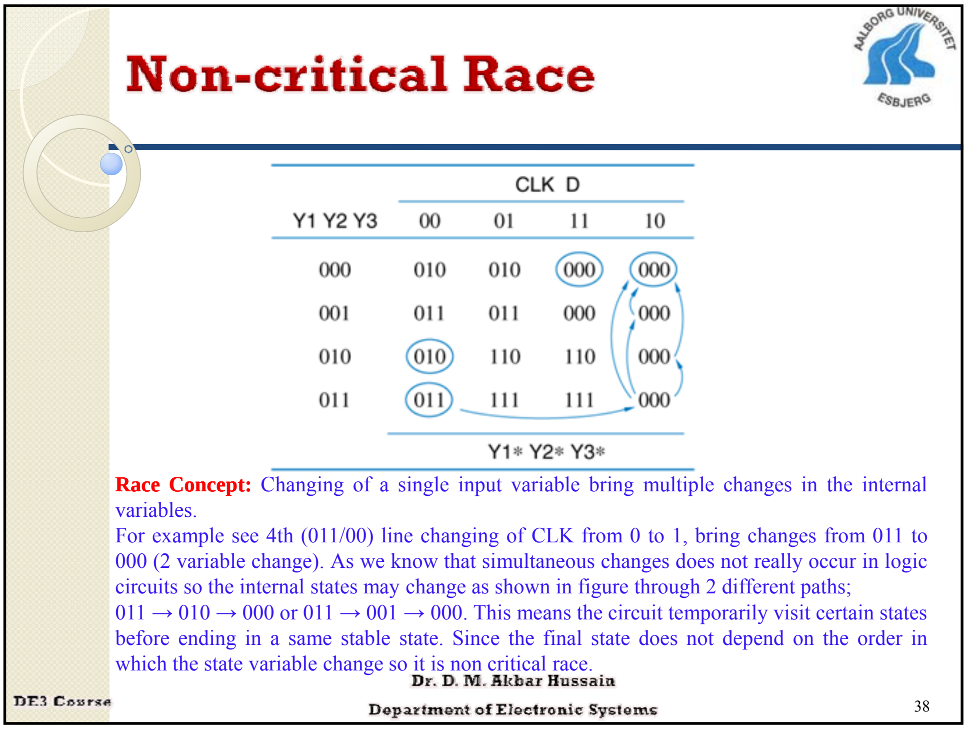
Circled states are stable states as the next internal state predicted by the transition table is the same as the current internal state.

Dr. D. M. Akbar Hussain

DE3 Course

Department of Electronic Systems

37



Non-critical Race

	CLK D			
Y1 Y2 Y3	00	01	11	10
000	010	010	000	000
001	011	011	000	000
010	010	110	110	000
011	011	111	111	000
	Y1* Y2* Y3*			

Race Concept: Changing of a single input variable bring multiple changes in the internal variables.

For example see 4th (011/00) line changing of CLK from 0 to 1, bring changes from 011 to 000 (2 variable change). As we know that simultaneous changes does not really occur in logic circuits so the internal states may change as shown in figure through 2 different paths;

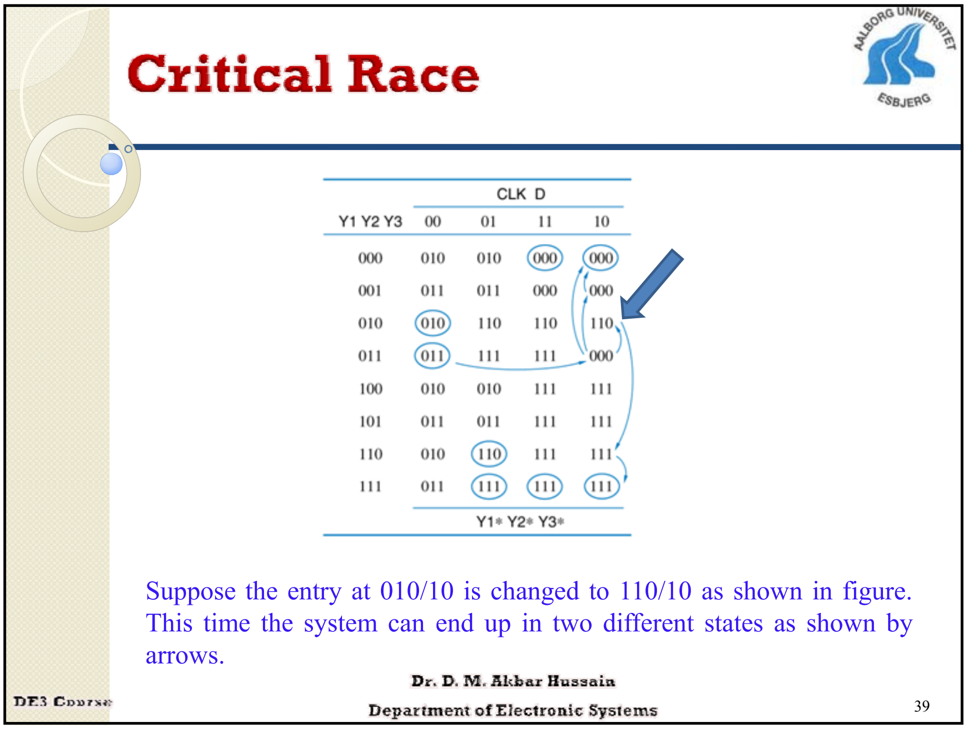
011 → 010 → 000 or 011 → 001 → 000. This means the circuit temporarily visit certain states before ending in a same stable state. Since the final state does not depend on the order in which the state variable change so it is non critical race.

Dr. D. M. Akbar Hussain

DE3 Course

Department of Electronic Systems

38



Critical Race

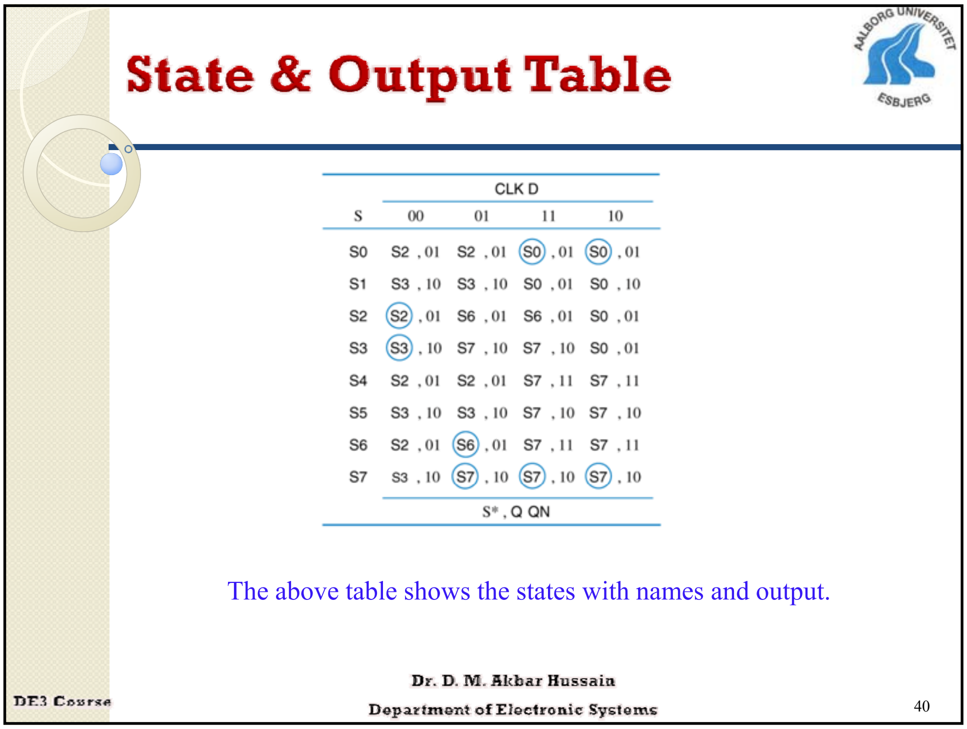
	CLK D			
Y1 Y2 Y3	00	01	11	10
000	010	010	000	000
001	011	011	000	000
010	010	110	110	110
011	011	111	111	000
100	010	010	111	111
101	011	011	111	111
110	010	110	111	111
111	011	111	111	111
	Y1* Y2* Y3*			

Suppose the entry at 010/10 is changed to 110/10 as shown in figure. This time the system can end up in two different states as shown by arrows.

Dr. D. M. Akbar Hussain

Department of Electronic Systems

DE3 Course



State & Output Table

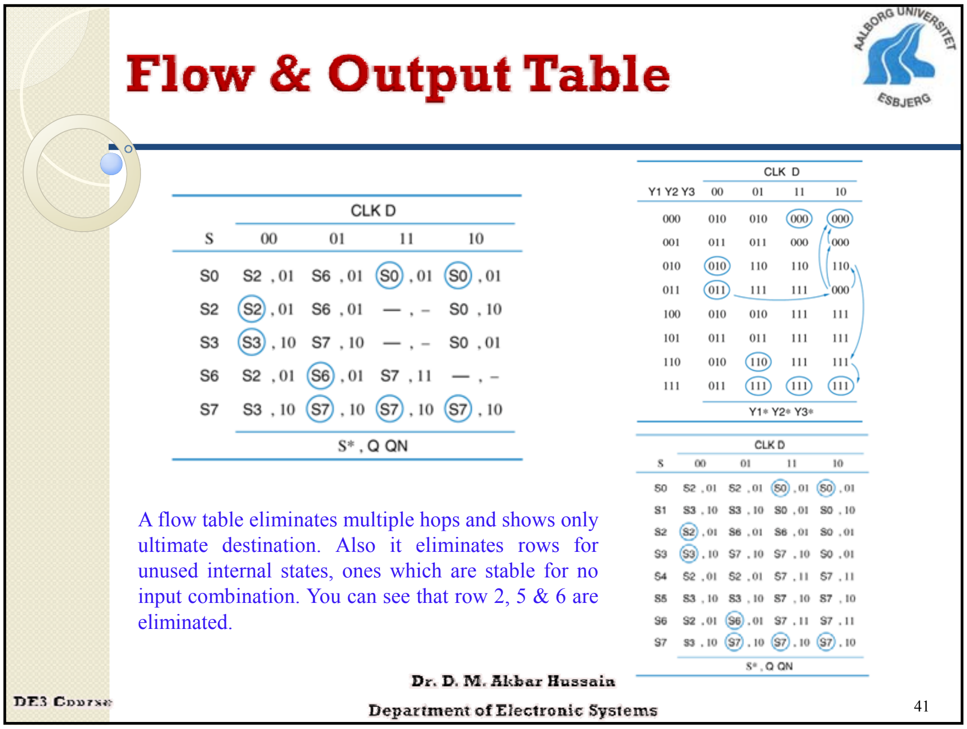
	CLK D			
S	00	01	11	10
S0	S2 , 01	S2 , 01	S0 , 01	S0 , 01
S1	S3 , 10	S3 , 10	S0 , 01	S0 , 10
S2	S2 , 01	S6 , 01	S6 , 01	S0 , 01
S3	S3 , 10	S7 , 10	S7 , 10	S0 , 01
S4	S2 , 01	S2 , 01	S7 , 11	S7 , 11
S5	S3 , 10	S3 , 10	S7 , 10	S7 , 10
S6	S2 , 01	S6 , 01	S7 , 11	S7 , 11
S7	S3 , 10	S7 , 10	S7 , 10	S7 , 10
	S* , Q QN			

The above table shows the states with names and output.

Dr. D. M. Akbar Hussain

Department of Electronic Systems

DE3 Course



Flow & Output Table

	CLK D			
S	00	01	11	10
S0	S2 , 01	S6 , 01	S0 , 01	S0 , 01
S2	S2 , 01	S6 , 01	— , –	S0 , 10
S3	S3 , 10	S7 , 10	— , –	S0 , 01
S6	S2 , 01	S6 , 01	S7 , 11	— , –
S7	S3 , 10	S7 , 10	S7 , 10	S7 , 10
	S* , Q QN			

	CLK D			
Y1 Y2 Y3	00	01	11	10
000	010	010	000	000
001	011	011	000	000
010	010	110	110	110
011	011	111	111	000
100	010	010	111	111
101	011	011	111	111
110	010	110	111	111
111	011	111	111	111
	Y1* Y2* Y3*			

	CLK D			
S	00	01	11	10
S0	S2 , 01	S2 , 01	S0 , 01	S0 , 01
S1	S3 , 10	S3 , 10	S0 , 01	S0 , 10
S2	S2 , 01	S6 , 01	S6 , 01	S0 , 01
S3	S3 , 10	S7 , 10	S7 , 10	S0 , 01
S4	S2 , 01	S2 , 01	S7 , 11	S7 , 11
S5	S3 , 10	S3 , 10	S7 , 10	S7 , 10
S6	S2 , 01	S6 , 01	S7 , 11	S7 , 11
S7	S3 , 10	S7 , 10	S7 , 10	S7 , 10
	S* , Q QN			

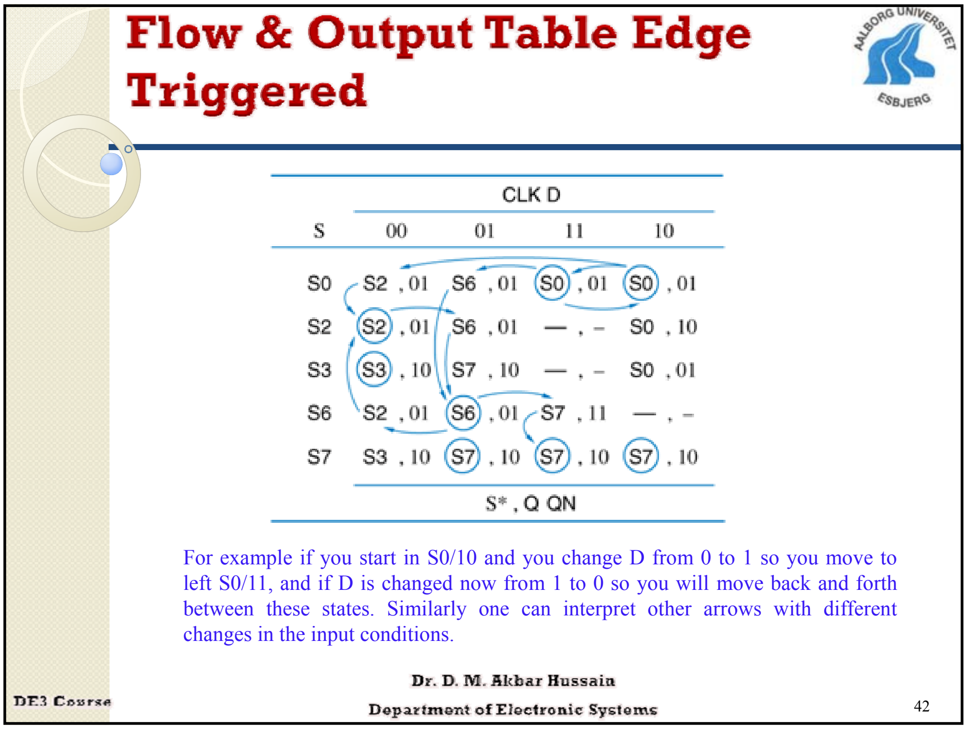
A flow table eliminates multiple hops and shows only ultimate destination. Also it eliminates rows for unused internal states, ones which are stable for no input combination. You can see that row 2, 5 & 6 are eliminated.

Dr. D. M. Akbar Hussain

DE3 Course

Department of Electronic Systems

41



Flow & Output Table Edge Triggered

	CLK D			
S	00	01	11	10
S0	S2 , 01	S6 , 01	S0 , 01	S0 , 01
S2	S2 , 01	S6 , 01	— , –	S0 , 10
S3	S3 , 10	S7 , 10	— , –	S0 , 01
S6	S2 , 01	S6 , 01	S7 , 11	— , –
S7	S3 , 10	S7 , 10	S7 , 10	S7 , 10
	S* , Q QN			

For example if you start in S0/10 and you change D from 0 to 1 so you move to left S0/11, and if D is changed now from 1 to 0 so you will move back and forth between these states. Similarly one can interpret other arrows with different changes in the input conditions.

Dr. D. M. Akbar Hussain

DE3 Course

Department of Electronic Systems

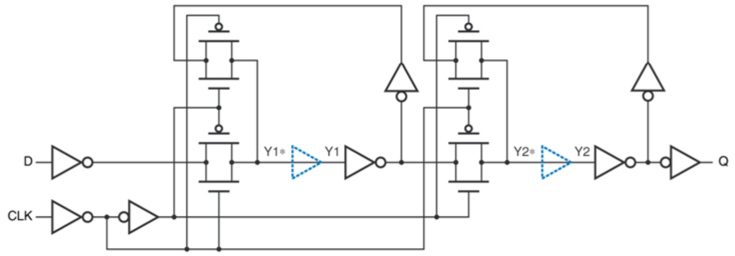
42

Reduced Flow & Output Table

	CLK D			
S	00	01	11	10
SB	SB , 01	S6 , 01	SB , 01	SB , 01
S3	S3 , 10	S7 , 10	— , –	SB , 01
S6	SB , 01	S6 , 01	S7 , 11	— , –
S7	S3 , 10	S7 , 10	S7 , 10	S7 , 10
	S* , Q QN			

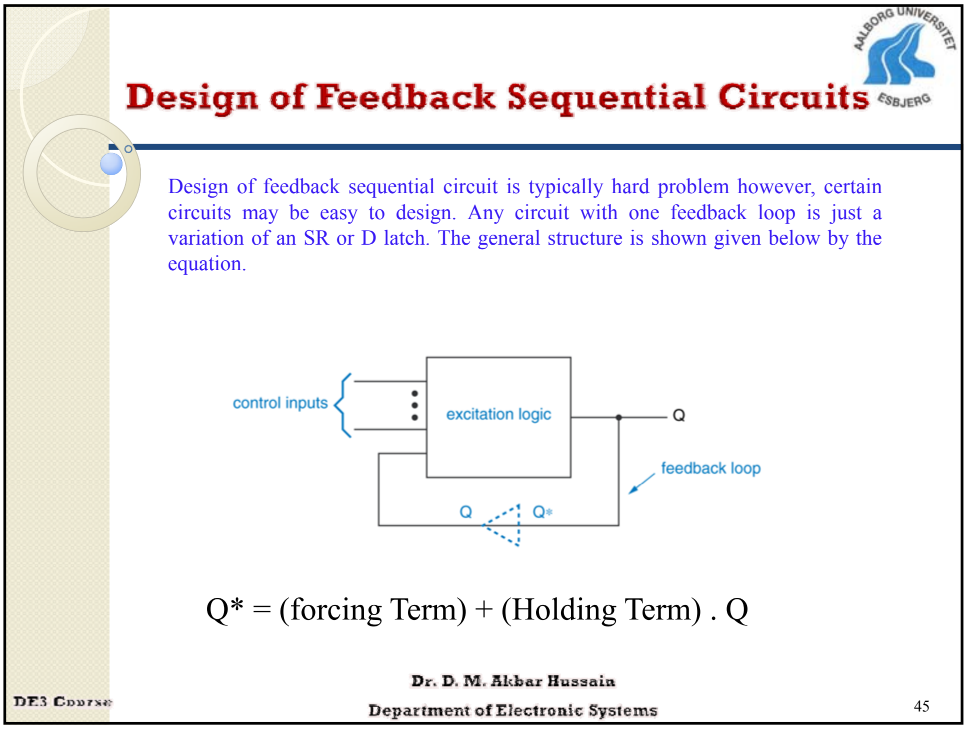
Although flow and output tables is simple, it can further improve as reduced flow table by minimization through combining compatible states, so in this case row 1 and row 2 of the flow table is combined as above.

Analysis



Y1* = CLK’ . D’ + CLK . Y1
Y2* = CLK . Y1’ + CLK’ . Y2

Similarly circuit like above can also be analysed in similar fashion described earlier.

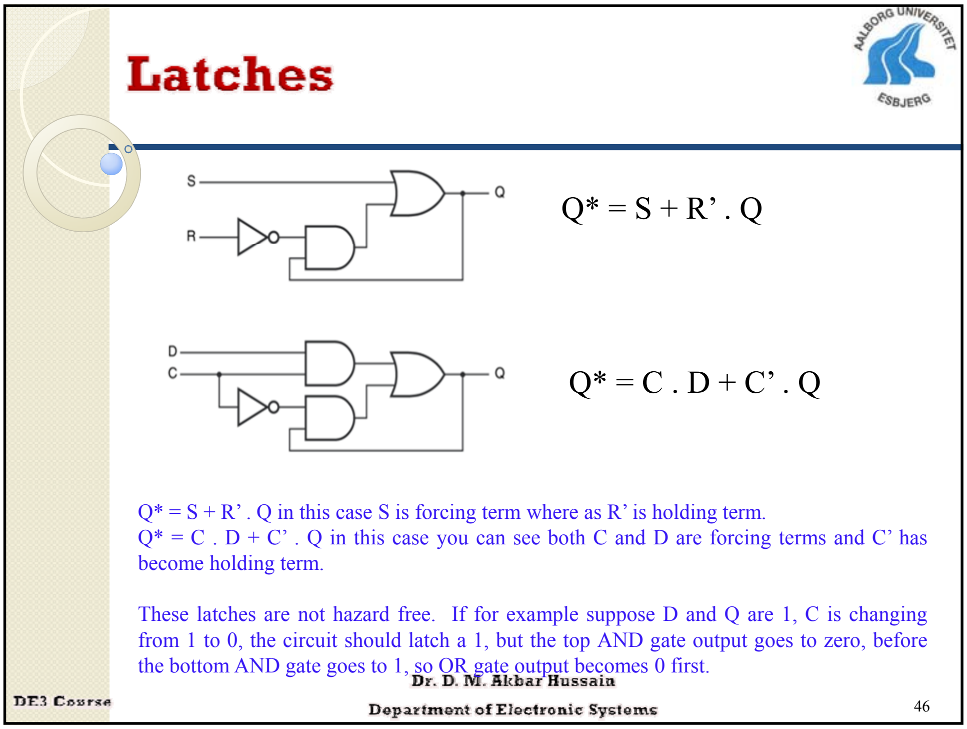


Design of Feedback Sequential Circuits

Design of feedback sequential circuit is typically hard problem however, certain circuits may be easy to design. Any circuit with one feedback loop is just a variation of an SR or D latch. The general structure is shown given below by the equation.

control inputs
excitation logic
Q
feedback loop
Q
Q*

Q* = (forcing Term) + (Holding Term) . Q



Latches

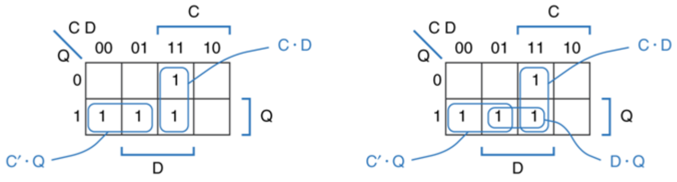
Q* = S + R’ . Q

Q* = C . D + C’ . Q

Q* = S + R’ . Q in this case S is forcing term where as R’ is holding term.
Q* = C . D + C’ . Q in this case you can see both C and D are forcing terms and C’ has become holding term.

These latches are not hazard free. If for example suppose D and Q are 1, C is changing from 1 to 0, the circuit should latch a 1, but the top AND gate output goes to zero, before the bottom AND gate goes to 1, so OR gate output becomes 0 first.

Karnaugh Maps

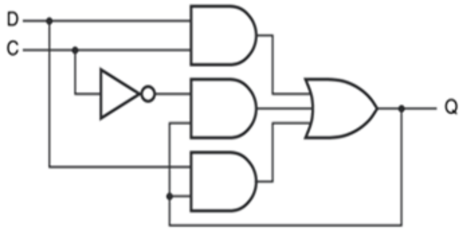


$Q* = C \cdot D + C' \cdot Q$

$Q* = C \cdot D + C' \cdot Q + D \cdot Q$

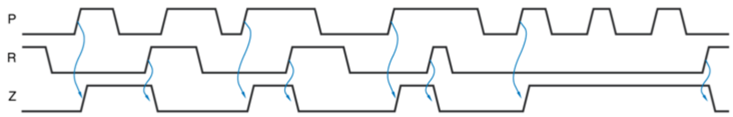
This latch can be made Hazard free by using the Karnaugh map as above on right, with the introduction of the extra term in the equation guarantees proper operation.

Hazard Free Latch Diagram



$$Q* = C \,.\, D + C' \,.\, Q + D \,.\, Q$$

Designing Fundamental Mode Flow Table



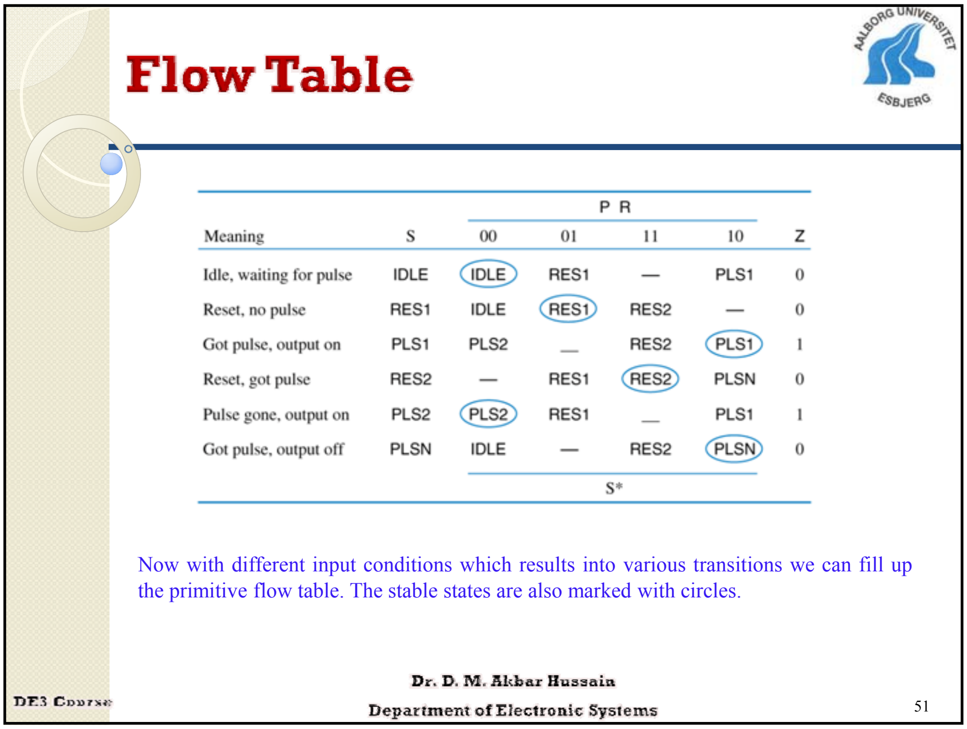
Pulse Catching Circuit Timing Diagram

Feedback sequential circuits are obviously complex and difficult to design compared with latches, typically the word description is converted into flow table. Let us design the circuit if we are given the above timing diagram. Intially it is assumed that both P and R are 0, so it is an IDLE state.

Flow Table

		P R				
Meaning	S	00	01	11	10	Z
Idle, waiting for pulse	IDLE					0
Reset, no pulse	RES1					0
Got pulse, output on	PLS1					1
Reset, got pulse	RES2					0
Pulse gone, output on	PLS2					1
Got pulse, output off	PLSN					0

RES1: When R goes from 0 to 1, the system may still be in the IDLE state, which means that there would be 2 stable states in the same row, typically one avoids that in the primitive flow table, so a new state RES1.
PLS1: In the IDLE state when P goes from 0 to 1, system goes to PLS1 state and output is 1.
RES2: When R is 1 and P goes from 0 to 1.
PLS2: This state occurs when P goes away (1 to 0).
PLSN: This state occurs when RESET goes away



Flow Table

		P R				
Meaning	S	00	01	11	10	Z
Idle, waiting for pulse	IDLE	IDLE	RES1	—	PLS1	0
Reset, no pulse	RES1	IDLE	RES1	RES2	—	0
Got pulse, output on	PLS1	PLS2	—	RES2	PLS1	1
Reset, got pulse	RES2	—	RES1	RES2	PLSN	0
Pulse gone, output on	PLS2	PLS2	RES1	—	PLS1	1
Got pulse, output off	PLSN	IDLE	—	RES2	PLSN	0

S*

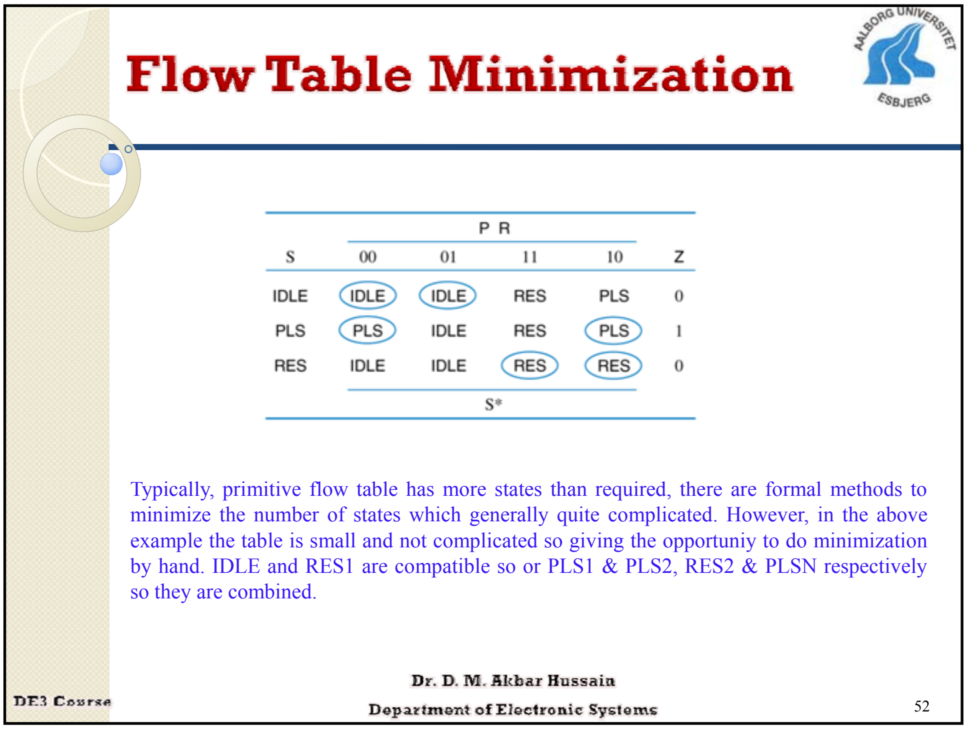
Now with different input conditions which results into various transitions we can fill up the primitive flow table. The stable states are also marked with circles.

Dr. D. M. Akbar Hussain

Department of Electronic Systems

DE3 Course

51



Flow Table Minimization

	P R				
S	00	01	11	10	Z
IDLE	IDLE	IDLE	RES	PLS	0
PLS	PLS	IDLE	RES	PLS	1
RES	IDLE	IDLE	RES	RES	0

S*

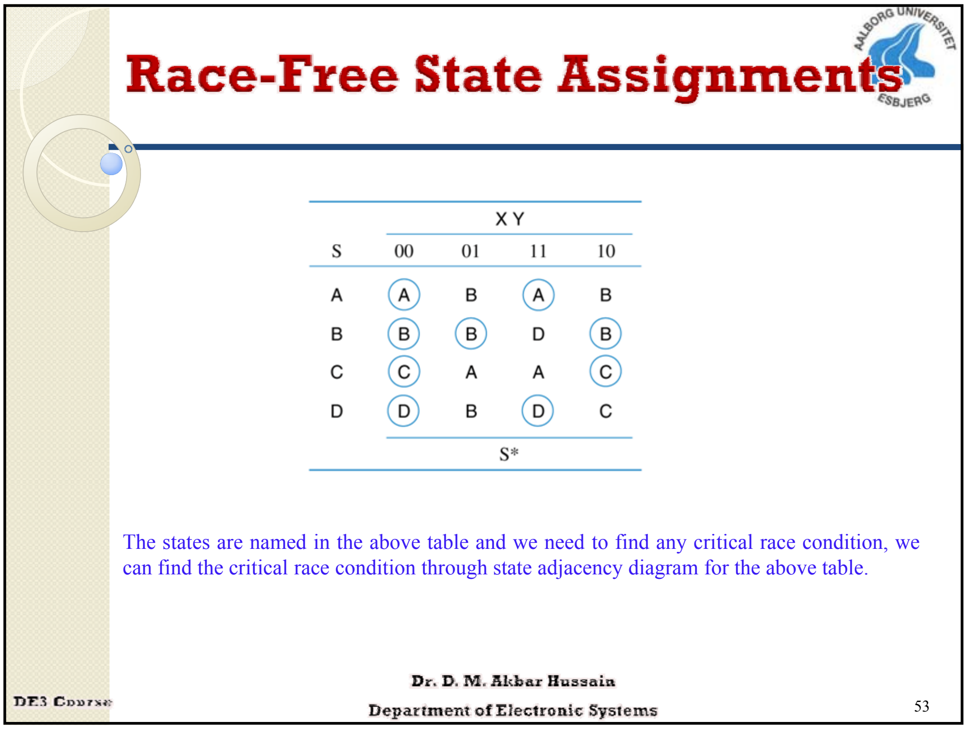
Typically, primitive flow table has more states than required, there are formal methods to minimize the number of states which generally quite complicated. However, in the above example the table is small and not complicated so giving the opportuniy to do minimization by hand. IDLE and RES1 are compatible so or PLS1 & PLS2, RES2 & PLSN respectively so they are combined.

Dr. D. M. Akbar Hussain

Department of Electronic Systems

DE3 Course

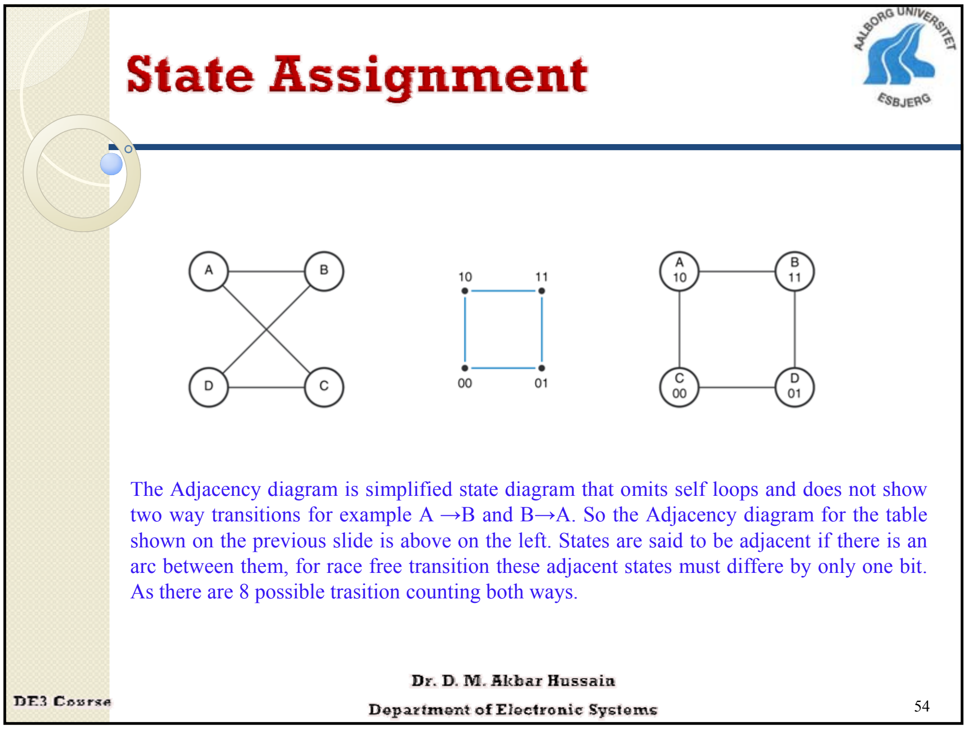
52



Race-Free State Assignments

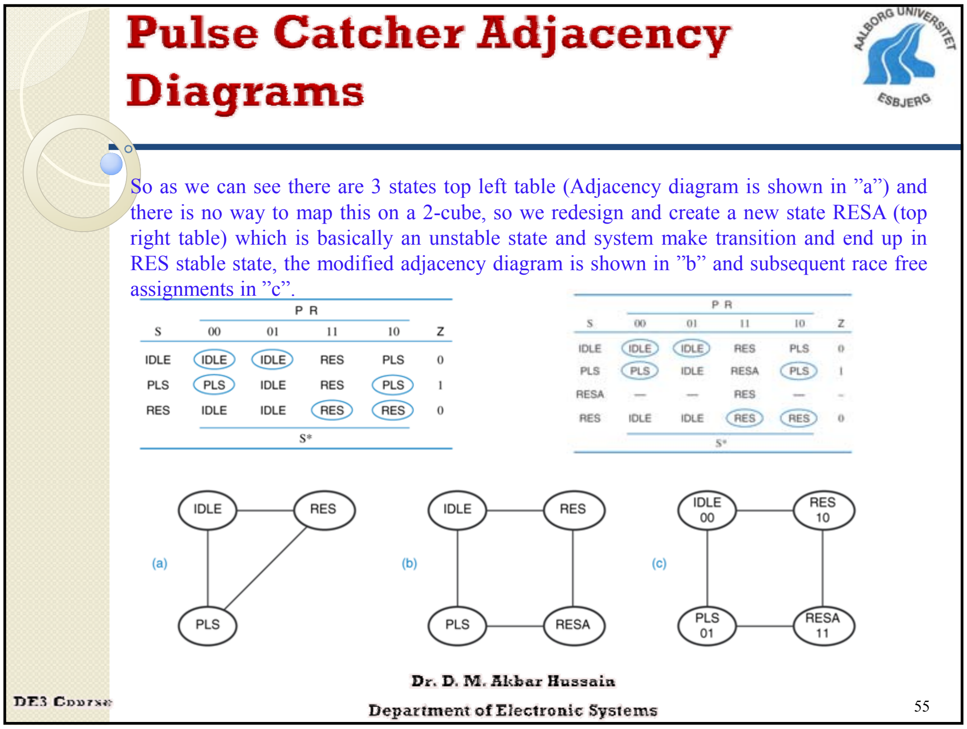
	XY			
S	00	01	11	10
A	Ⓐ	B	Ⓐ	B
B	Ⓑ	Ⓑ	D	Ⓑ
C	Ⓒ	A	A	Ⓒ
D	Ⓓ	B	Ⓓ	C
	S*			

The states are named in the above table and we need to find any critical race condition, we can find the critical race condition through state adjacency diagram for the above table.



State Assignment

The Adjacency diagram is simplified state diagram that omits self loops and does not show two way transitions for example A →B and B→A. So the Adjacency diagram for the table shown on the previous slide is above on the left. States are said to be adjacent if there is an arc between them, for race free transition these adjacent states must differe by only one bit. As there are 8 possible trasition counting both ways.

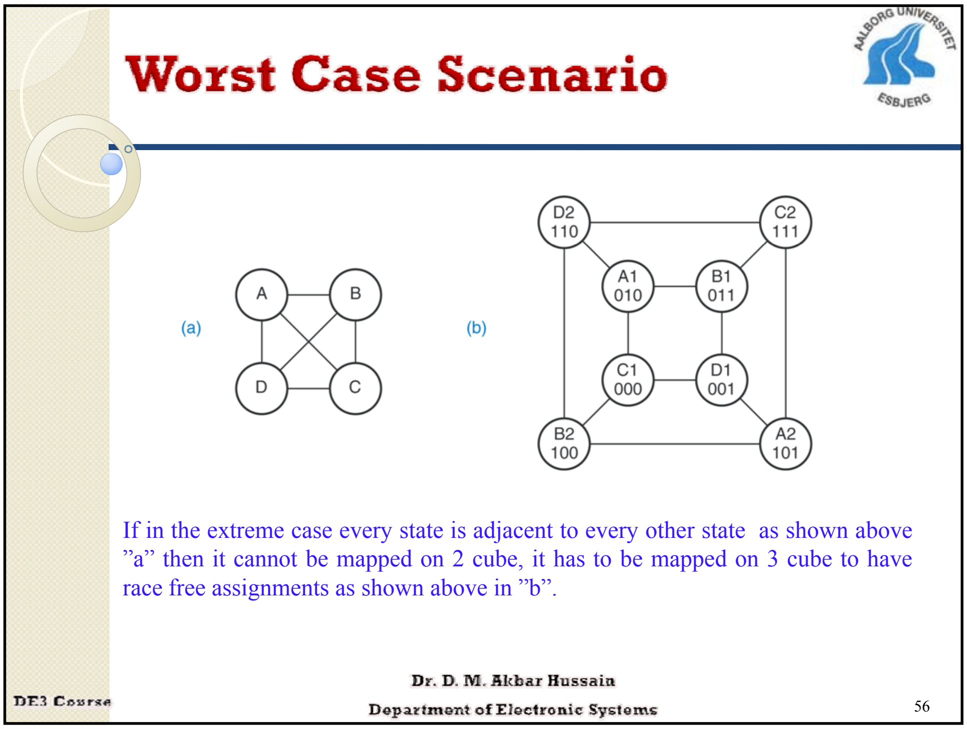


Pulse Catcher Adjacency Diagrams

So as we can see there are 3 states top left table (Adjacency diagram is shown in ”a”) and there is no way to map this on a 2-cube, so we redesign and create a new state RESA (top right table) which is basically an unstable state and system make transition and end up in RES stable state, the modified adjacency diagram is shown in ”b” and subsequent race free assignments in ”c”.

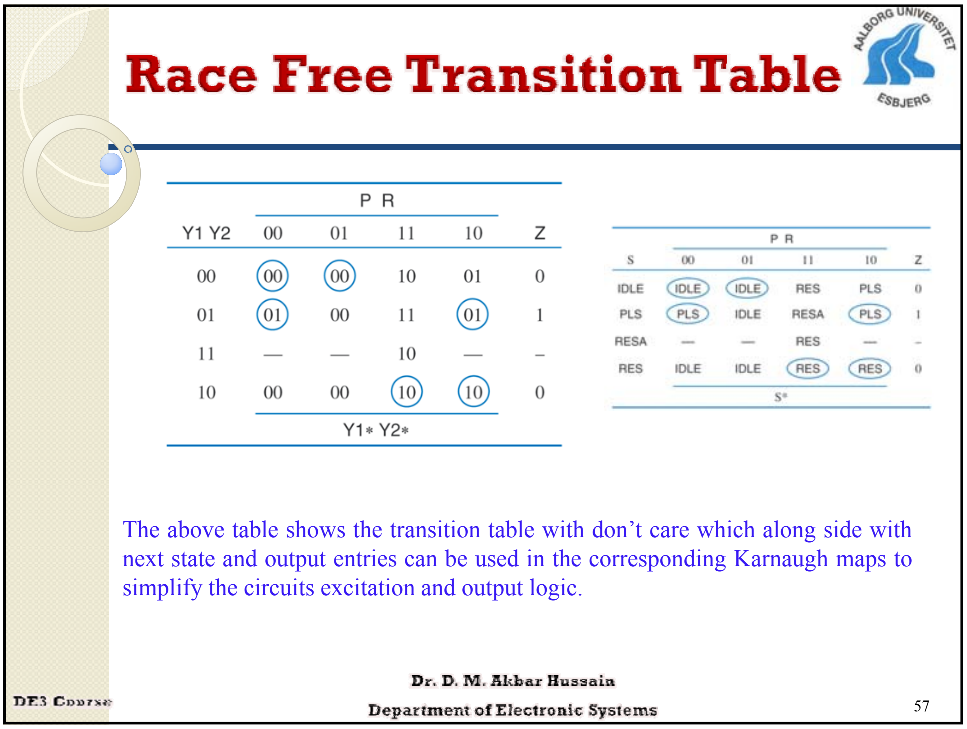
	P R				
S	00	01	11	10	Z
IDLE	IDLE	IDLE	RES	PLS	0
PLS	PLS	IDLE	RES	PLS	1
RES	IDLE	IDLE	RES	RES	0
	S*				

	P R				
S	00	01	11	10	Z
IDLE	IDLE	IDLE	RES	PLS	0
PLS	PLS	IDLE	RESA	PLS	1
RESA	—	—	RES	—	-
RES	IDLE	IDLE	RES	RES	0
	S*				



Worst Case Scenario

If in the extreme case every state is adjacent to every other state as shown above ”a” then it cannot be mapped on 2 cube, it has to be mapped on 3 cube to have race free assignments as shown above in ”b”.



Race Free Transition Table

	P R				
Y1 Y2	00	01	11	10	Z
00	00	00	10	01	0
01	01	00	11	01	1
11	—	—	10	—	–
10	00	00	10	10	0
	Y1* Y2*				

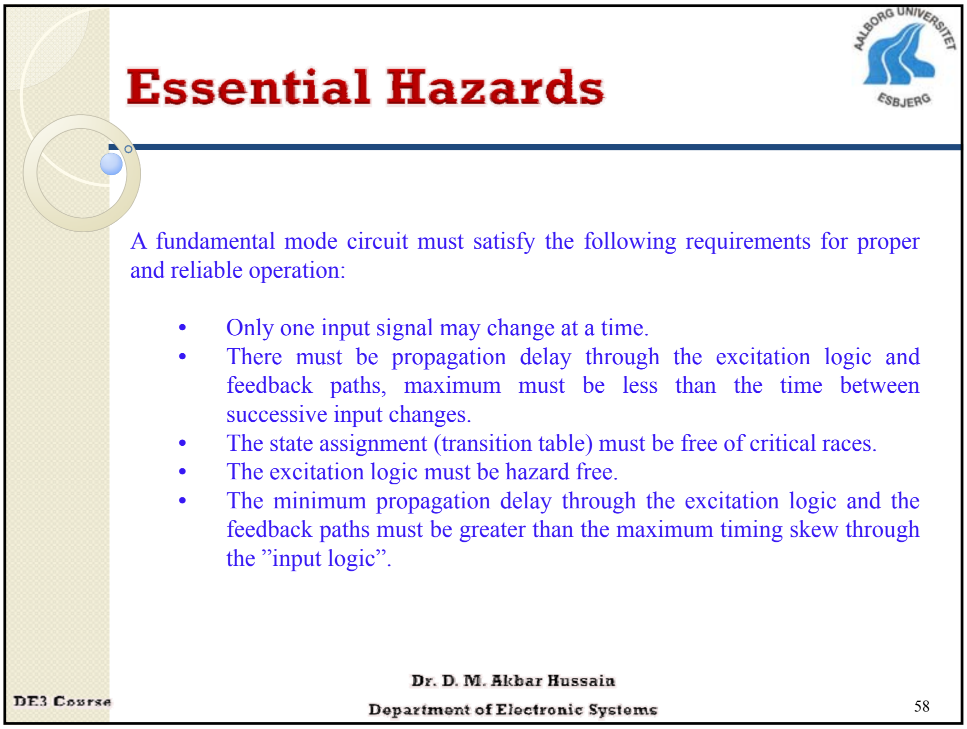
	P R				
S	00	01	11	10	Z
IDLE	IDLE	IDLE	RES	PLS	0
PLS	PLS	IDLE	RESA	PLS	1
RESA	—	—	RES	—	–
RES	IDLE	IDLE	RES	RES	0
	S*				

The above table shows the transition table with don't care which along side with next state and output entries can be used in the corresponding Karnaugh maps to simplify the circuits excitation and output logic.

DE3 Course

Dr. D. M. Akbar Hussain
Department of Electronic Systems

57



Essential Hazards

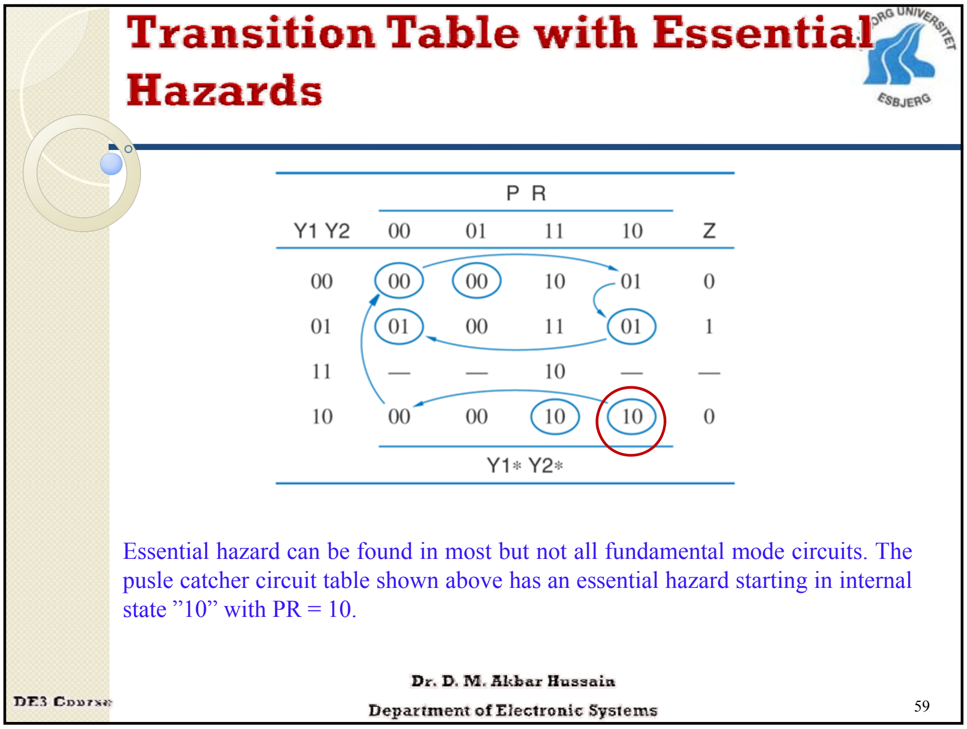
A fundamental mode circuit must satisfy the following requirements for proper and reliable operation:

- Only one input signal may change at a time.
- There must be propagation delay through the excitation logic and feedback paths, maximum must be less than the time between successive input changes.
- The state assignment (transition table) must be free of critical races.
- The excitation logic must be hazard free.
- The minimum propagation delay through the excitation logic and the feedback paths must be greater than the maximum timing skew through the ”input logic”.

DE3 Course

Dr. D. M. Akbar Hussain
Department of Electronic Systems

58



Transition Table with Essential Hazards

	P R				
Y1 Y2	00	01	11	10	Z
00	00	00	10	01	0
01	01	00	11	01	1
11	—	—	10	—	—
10	00	00	10	10	0
	Y1* Y2*				

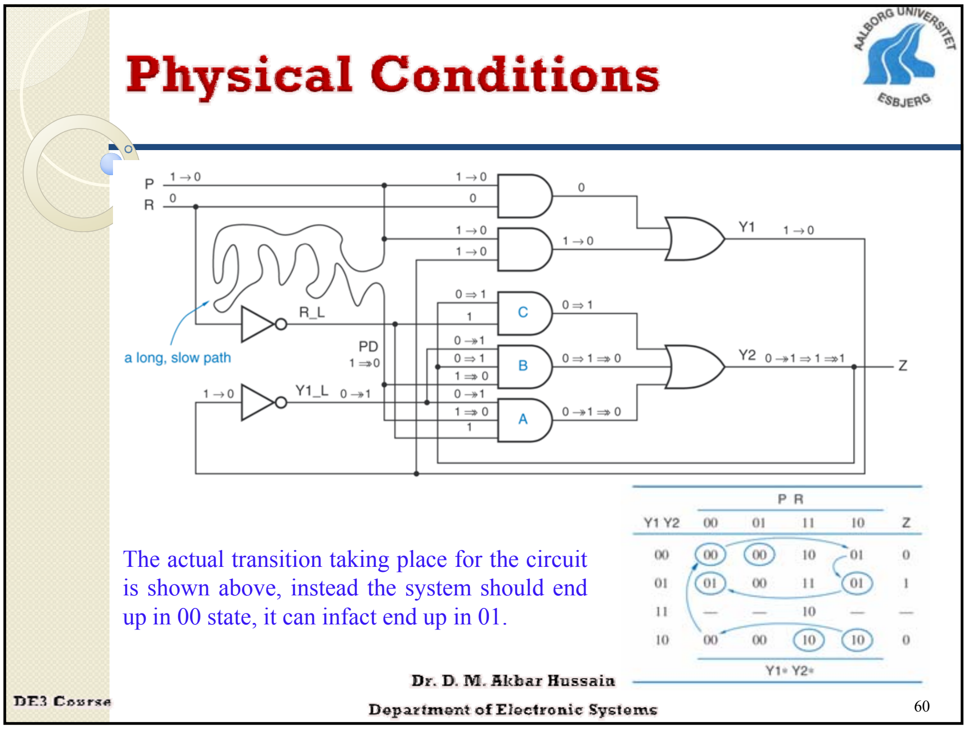
Essential hazard can be found in most but not all fundamental mode circuits. The pusle catcher circuit table shown above has an essential hazard starting in internal state ”10” with PR = 10.

DE3 Course

Dr. D. M. Akbar Hussain

Department of Electronic Systems

59



Physical Conditions

	P R				
Y1 Y2	00	01	11	10	Z
00	00	00	10	01	0
01	01	00	11	01	1
11	—	—	10	—	—
10	00	00	10	10	0
	Y1* Y2*				

The actual transition taking place for the circuit is shown above, instead the system should end up in 00 state, it can infact end up in 01.

DE3 Course

Dr. D. M. Akbar Hussain

Department of Electronic Systems

60

Spørgsmål
Opgaver

DE3 Course

Dr. D. M. Akbar Hussain

Department of Electronic Systems

61