B-Block Cubes

	S_1	S_0	O/P
r_1	*x*	1	0
r_2	1	0	1
r_3	0	0	M

r_{11}: 0 1 / 0

r_{12}: 1 1 / 0

Assumes Tri-State Capability

	S_1	S_0	O/P	
$r_{12} \cap r_2$	1	D	$\overline{D}$	
$r_{12} \cap r_2$	1	$\overline{D}$	D	
$r_{11} \cap r_2$	$\overline{D}$	D	$\overline{D}$	
$r_{11} \cap r_2$	D	$\overline{D}$	D	
$r_{11} \cap r_3$	0	D	$\overline{D}$	(1)
$r_{11} \cap r_3$	0	$\overline{D}$	D	(1)
$r_{12} \cap r_3$	D	D	$\overline{D}$	(1)
$r_{12} \cap r_3$	$\overline{D}$	$\overline{D}$	D	(1)
$r_2 \cap r_3$	D	0	D	(0)
$r_2 \cap r_3$	$\overline{D}$	0	$\overline{D}$	(0)

- For S_OP Faults, the CUT must be first ***Initialized***
- **Initialization** is achieved by *Line Justification* (D-Alg) or by proper *Initial Objective* in PODEM
- The fault Should not Cause False Initialization → If a Line is Stuck at 0 (**1**), a 1 (**0**) Should not be Applied to it During Initialization

Initial State To Show Fault

Fault Activation: PDFF Line 1 = 0

Implication: Line 3 = DB, Line 5 = 0 = Line 8,

D-Frontier: {G1} → t2= t1 ∩ PDC(G1)

t2=(0, *x*, DB, *x*, 0, *x*, *x*, 0, *x*) ∩ (x2=1)

=(0, 1, DB, 1, 0, 1, D, 0, (*x*) → = **D Provided F** Initialized at **0**

Initialization: Faulty & Fault-Free CUT Must Give the Same Result, i.e. Fault *Must not MASK the Initialization Value*

Initialization: Since t2= (0, 1, DB, 1, 0, 1, D, 0, ***D***)

Line Justify: (*x*, *x*, 1 , *x*, *x*, *x*, *x*, *x*, **0**)

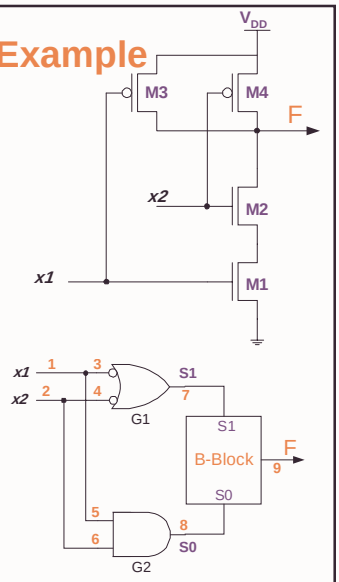
Prevents Fault Masking

Line Justify: (*x*, *x*, 1 , *x*, *x*, *x*, **0**, **1**, **0**)

This Yields : (1, 1, 1 , 1, 1, 1, **0**, **1**, **0**)

The Test Sequence becomes : (11/0) (01/1)

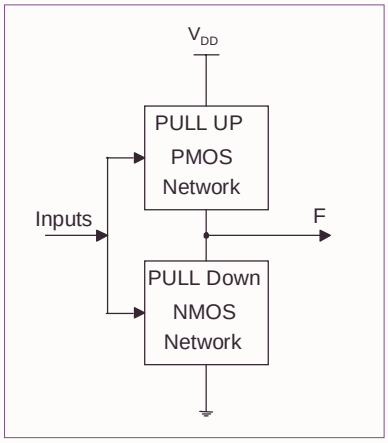
Example



Fault M3 S_OP ≡ 3/1

Test Generation for Transistor-Level Faults of Static CMOS Logic Connection Graph Method (Chiang-Vransic)

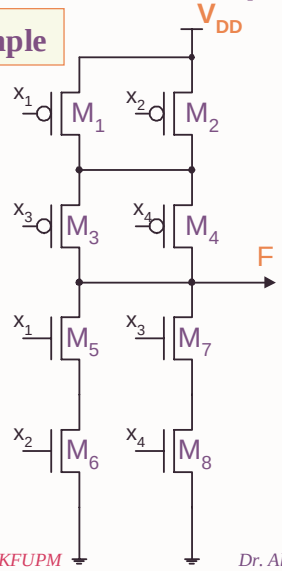
- **A Connection Graph is Drawn for both the Pullup and Pull Down Networks**
- **The Connection Graph Models the Transmission Function between the Source and the Sink Node**
- **The Transmission Function Expresses the Conditions under which Paths are Enabled between the Source and Sink Nodes**



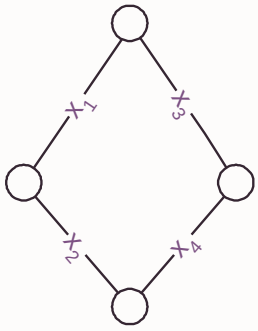
COE – KFUPM *Dr. Alaaeldin Amin (COE 545)* *Slide Number 26*

Test Generation for Static CMOS Logic Connection Graph Method (Chiang-Vransic)

Example



Source F

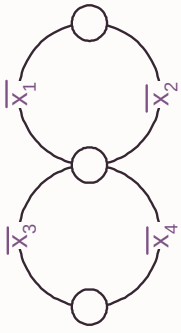


Sink V_{SS}

Connection Graph for NMOS NW

$F_n = x_1x_2 + x_3x_4 = \overline{F_P}$

Source V_{DD}



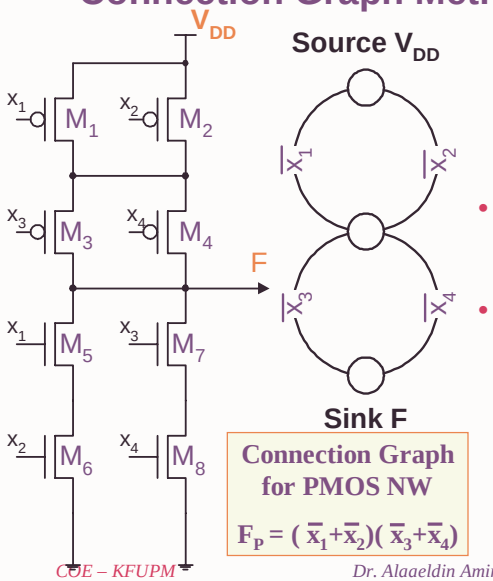
Sink F

Connection Graph for PMOS NW

$F_P = (\overline{x_1}+\overline{x_2})(\overline{x_3}+\overline{x_4})$

COE – KFUPM *Dr. Alaaeldin Amin (COE 545)* *Slide Number 27*

Test Generation for Static CMOS Logic Connection Graph Method (Chiang-Vransic)

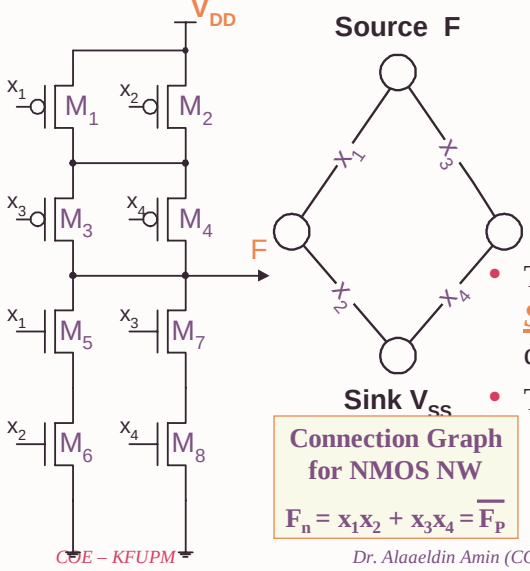


Connection Graph for PMOS NW

$F_P = (\overline{x}_1+\overline{x}_2)(\overline{x}_3+\overline{x}_4)$

- The Set of TVs that ***Enable*** a ***SINGLE*** Path in the Pull up PMOS Network is :
- TV(1) ={0101 , 0110 , 1001 , 1010}

Test Generation for Static CMOS Logic Connection Graph Method (Chiang-Vransic)

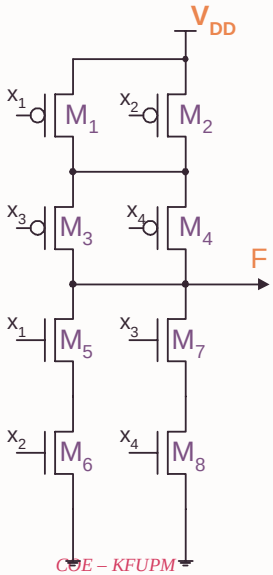


Connection Graph for NMOS NW

$F_n = x_1x_2 + x_3x_4 = \overline{F_P}$

- The Set of TVs that ***Enable*** a ***SINGLE*** Path in the Pull down NMOS Network is :
- TV(0) ={1100 , 0011}

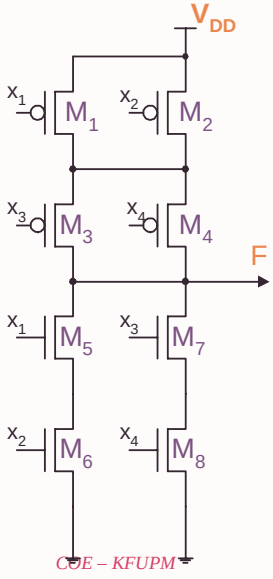
Test Generation for Static CMOS Logic Connection Graph Method (Chiang-Vransic)



- TV(1) ={0101 , 0110 , 1001 , 1010}
- TV(0) ={1100 , 0011}
- Choose the smallest Subset of TVs which Covers Paths between Source & Sink that Go Through All Transistors
  - T (1) ={0101 , 1010}
  - T (0) ={1100 , 0011}
- Choose a test Sequence that Alternates TVs from T(1) and T(0) → Enables all Conduction Paths between F and V_{DD} or V_{SS} alternately.
- Test Sequence = T = { **0101** , 1100 , 1010 , 0011 , **0101** }

COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 30

Test Generation for Static CMOS Logic Connection Graph Method (Chiang-Vransic)



- **T** = { **0101** , 1100 , 1010 , 0011 , **0101** }
- **F** = { **1 , 0 , 1 , 0 , 1** }
- The First TV Serves as Initialization, thus it is Repeated again at the end of the sequence.
- **T** Detects ALL Single S_OP Faults in either the Pull up or the Pull Down Networks.
- *A TV Detecting Given S_OP Faults, ALSO Detects S_ON Fault of the Corresponding Transistors in the Dual Network*
- **Thus**: 1100 Detects **S_OP** Faults of **M5** & **M6**
- With a **BICS** in place, the same TV (1100) Also **Detects** S_ON Faults of M1 & M2

COE – KFUPM

CMOS Test Invalidation (Test Robustness)

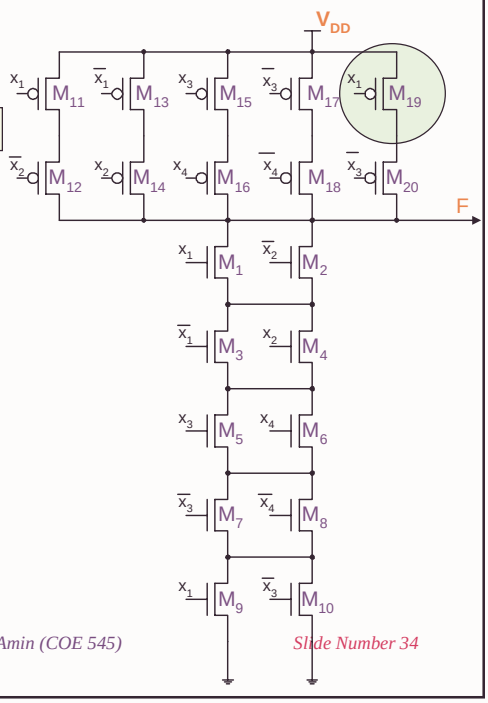
- 2 Mechanisms May Prevent a TV Sequence from Detecting the S_OP Faults it is Designed to Detect:
  1. Circuit Delays, and
  2. Charge Sharing (Only Dynamic Logic)
- Tests that **Can be** Invalidated are called non-Robust Tests
- Tests that Cannot be Invalidated are called Robust Tests

CMOS Test Invalidation (Test Robustness)

- 2 Mechanisms May Prevent a TV Sequence from Detecting the S_OP Faults it is Designed to Detect:
  1. Circuit Delays, and
  2. Charge Sharing (Only Dynamic Logic)
- Tests that **Can be** Invalidated are called non-Robust Tests
- Tests that Cannot be Invalidated are called Robust Tests

Test Robustness
- Example

$F = \overline{x}_1 x_2 + x_1 \overline{x}_2 + \overline{x}_3 \overline{x}_4 + x_3 x_4 + \overline{x}_1 x_3$



Fault *f = M19 S_OP*

T(0) = { 0001 , 1101 , 1110 }

T(1) = { 0010}

- **Only 3 Test Sequences are Possible to Detect *f*.**

Test Robustness
- Example

$F = \overline{x}_1 x_2 + x_1 \overline{x}_2 + \overline{x}_3 \overline{x}_4 + x_3 x_4 + \overline{x}_1 x_3$

Fault *f = M19 S_OP*

T(0) = { 0001 , 1101 , 1110 }

T(1) = { 0010}

- Only 3 Test Sequences are Possible to Detect *f*.
- Transient TVs always Pass by either 0000 or 0011
- *For this Fault*, There is No Robust 2-Pattern Test Sequence → This CUT is *not Robustly Testable*
- For a Robust test Sequence, Alternating TVs in the Sequence Must have a Hamming Distance of 1