DIGITAL SYSTEM TESTING
COE -545

Lecture - 09

Path Oriented DEcision Making
PODEM (Goel – 1981)

- **Goel Noticed that**:
  - Search in D-Alg. Involves Every Node in the Circuit.
  - Search Can be Limited only to PIs Since All Internal Nodes are Function of the PIs.
  - IF only PIs are Considered No Hidden Conflict will Arise
  - A simple "backtrace" heuristic is used to select PI assignments

Path Oriented DEcision Making PODEM (Goel – 1981)

D-Algorithm	PODEM
• Assignment of Values Allowed To *Any Internal* Line	• Assignment of Values Allowed *Only To* **PIs** →Values are then Propagated to Internal Lines by Forward Implication (Simulation).
• **Backtracking** May Occur at Any Internal Line	• **Backtracking** May Only Occur at **PIs** (**PI Remake**).
• **Conflict** May Arise -- An implied value different from assigned value	• **No Conflict** is Possible, Only Objectives May not be Met.
• Requires Line Justification	• Line Justification is Not Needed

Improvement

- Improvements in the Original Paper Were:
  - **X-path Check**
    - Checks That at Least One Gate in D-frontier Has a Path to a PO Such That the Node Values on That Path Are All X's → *Otherwise No Solution is Possible*.
  - **Heuristics for Selecting an Objective in Backtrace (***Better Directed Search***)**
    - Among the *Required* Objectives, Choose the *Hardest* Objective.
    - Among *Alternative* Objectives, Choose the *Easiest* Objective.

Algorithm Outline

- To Generate a TV for a *Target* Fault
  1. **PODEM** tries to Solve a Set of *Objectives* *One at a Time*. Each Objective Consists of Justifying a Logic Value $v \in \{0,1, D, DB\}$ to some Line **L** → The Objective is **Coded as (L, *v*)**
  2. An Objective **(L, *v*)** is *Mapped* into Objective(s) at the Input(s) of the Gate Driving **L** by a Procedure called **Backtrace**
  3. If the Gate Input is a **PI**, The Objective is Satisfied by Assigning the Required Value to this **PI**
  4. Each **PI**-Assignment is Followed by a Forward Implication (*Simulation*) Step. If Simulation Indicates that an Objective cannot be met, The *Last* Assigned **PI** value is *Complemented* and Simulated (PI-Remake)
  5. The Process Continues Till Objectives Are Met or All PI Combinations are Exhausted.

Algorithm Outline

- Internal Line Values are Computed Only Through Simulation (Forward Implication) of PI assignments → Values are *Always Self-Consistent* and All Values are Justified. Thus:
  - *Consistency Checks are not Required* (Only Objective Values are Checked)
  - *Backward Implication are not Needed*
  - *No Explicit Backtracking* is Required (Only **PI-Remake**) → Backtracking is Implicitly Done By simulation
- Since the Circuit State Depends Only on PI Values (Changes in State Computed By Simulation), and Since No Explicit Backtracking is Required → *No Circuit State Saving/Restoring is Required*

Choice of Gate Inputs

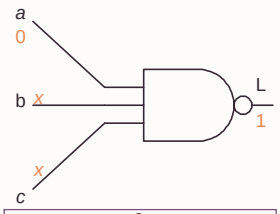
Assumptions:

a is Easiest to Control

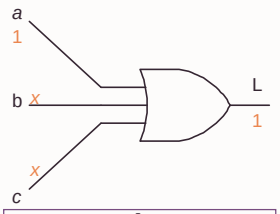
c is Hardest to Control

•To Meet (L, *v*) → Only One I/P Must Be Set to the Controlling Value

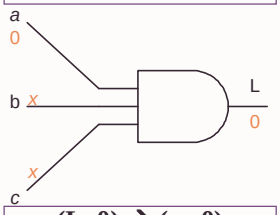
•Choose the I/p Easiest to Control (a)



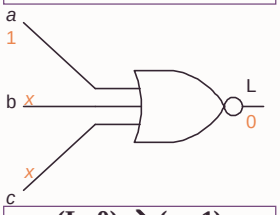
(L, 1) → (a , 0)
a is Easiest to Control
(*Has Min CC0 Value*)



(L, 1) → (a , 1)
a is Easiest to Control
(*Has Min CC1 Value*)



(L, 0) → (a , 0)
a is Easiest to Control
(*Has Min CC0 Value*)



(L, 0) → (a , 1)
a is Easiest to Control
(*Has Min CC1 Value*)

COE – KFUPM

Choice of Gate Inputs

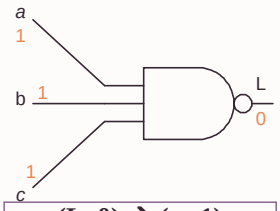
Assumptions:

a is Easiest to Control

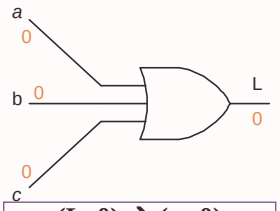
c is Hardest to Control

•To Meet (L, *v*) → ALL I/Ps Must Be Set to the NonControlling Value

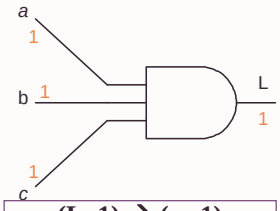
•Start with Objective on I/p HARDEST to Control (c)



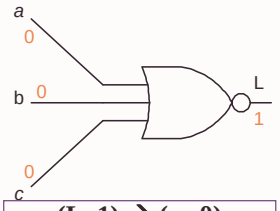
(L, 0) → (c , 1)
c is Hardest to Control
(*Has Max CC1 Value*)



(L, 0) → (c , 0)
c is Hardest to Control
(*Has Max CC0 Value*)



(L, 1) → (c , 1)
c is Hardest to Control
(*Has Max CC1 Value*)



(L, 1) → (c , 0)
c is Hardest to Control
(*Has Max CC0 Value*)

COE – KFUPM

Backtrace (s, v_s) Pseudo-Code

```
--Maps Objective into PI  Assignment
v = v_s;
while (S is a gate output)
    if (S is NAND or INVERTER or NOR) v = v̄;
    if (objective requires setting all inputs)
       Select Hardest to Control (CC_v) Unassigned Input of S → (a)
    else
       Select Easiest to Control (CC_v) Unassigned Input of S → (a)
    S = a;
-- S is Now a PI
return (S, v) /* PI and value to be assigned */;
```


Objective Selection Code

```
Objective(g, v )
{
/* Target fault is Gate g stuck-at- v */
if (g = x) return (g , v̄ ) ;
Select gate (P) from D-frontier;
Select  Input  ( L ) of  P  with  value  x;
c = 0;
if (gate P  has controlling value)  then
        c = controlling value of P;
if (gate g is XOR / XNOR)  then
        if (0 value easier to get at input)  then  c = 1;
    return ( L , c̄) ;-- for Fault Propagation
}
```

Pseudo-Code *for PODEM* (*fault*, v_{fault})

IF (PO = D/DB) **return** (SUCCESS);

IF (***xpathcheck*** (D-frontier)– *Returns True if an x-Path to PO exists*

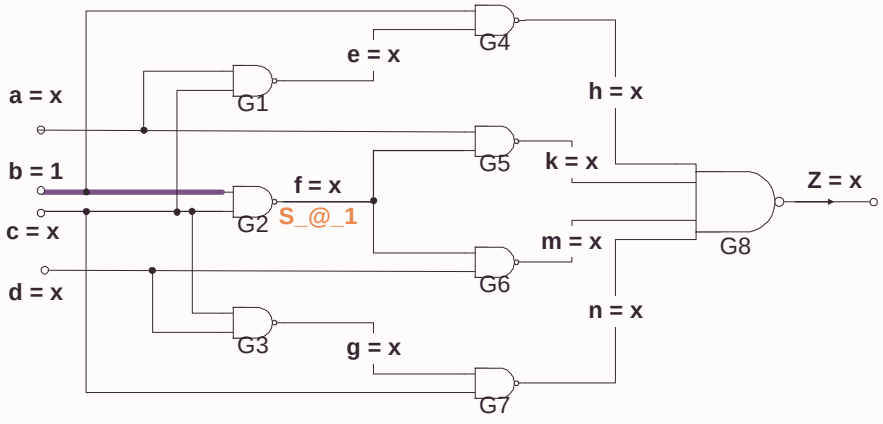
(l, v_l) = ***Objective*** (*fault*, v_{fault});
(PI, v_{pi}) = ***Backtrace*** (l, v_l); -- *Returns a PI and Assignment* v_{pi}
Imply (PI, v_{pi}); -- *Simulation Step & D-Frontier Update*
if (***PODEM*** (*fault*, v_{fault}) == SUCCESS) **return** (SUCCESS);
Imply (PI, $\overline{v_{pi}}$); -- *Reverse PI Assignment (PI-Remake)*
if (***PODEM*** (*fault*, v_{fault}) == SUCCESS) **return** (SUCCESS);
Imply (PI, "X"); -- *Earlier Assigned PI Blocks Fault Propagat.*

return (FAILURE)

Example

Excite (Provoke) the Fault (INITIAL OBJECTIVE)

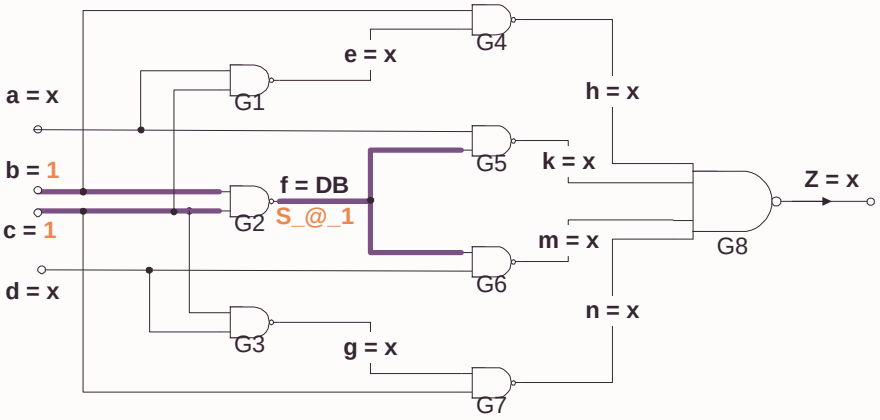
	a	b	c	d	e	f	g	h	k	m	n	z
	x	x	x	x	x	x	x	x	x	x	x	x
Initial Objective (I.O)	f=DB --> (f , 0)											
BackTrace- Objectives	(b , 1) , (C , 1) - B & C are PI's (equally hard)											
Current Objective	(b , 1)											
Simulate	x	1	x	x	x	x	x	x	x	x	x	x
	Initial Objective NOT MET ------> BackTrace											



Example

Initial Objective Not Met
New Current Objective

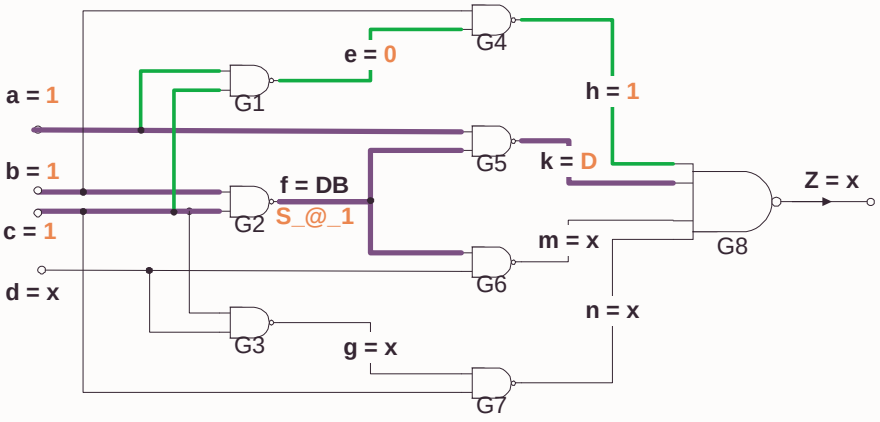
	a	b	c	d	e	f	g	h	k	m	n	z
tc(1)	x	1	x	x	x	x	x	x	x	x	x	x
Current Objective	(C , 1) - C is PI											
SIMULATE	x	1	1	x	x	DB	x	x	x	x	x	x
	Initial Objective Met (f, DB)											



Example

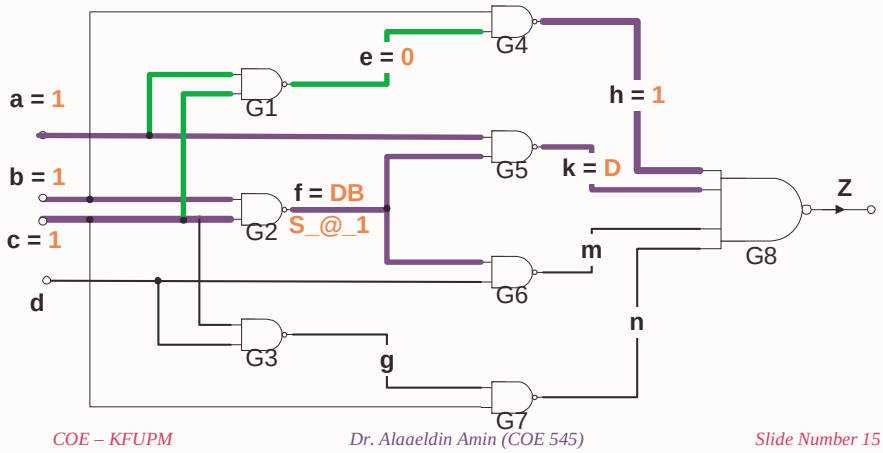
Initial Objective Not Met
New Current Objective

	a	b	c	d	e	f	g	h	k	m	n	z
tc(2)	x	1	1	x	x	DB	x	x	x	x	x	x
D-Frontier	{G5 , G6 }											
Initial Objective	(K, D) { K and M are Equally Observable }											
Backtrace - **CO**	(a, 1) - a is PI											
SIMULATE tc(3)	1	1	1	x	0	DB	x	1	D	x	x	x
	Initial Objective Met (K, D)											



Example

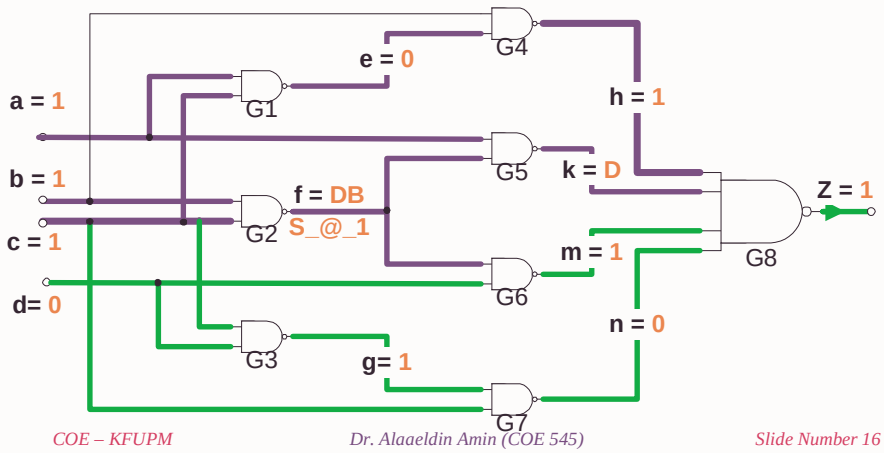
	a	b	c	d	e	f	g	h	k	m	n	z
tc(3)	1	1	1	x	0	DB	x	1	D	x	x	x
D-Frontier	{G8 , G6 } - *G8 Chosen - Higher*											
Initial Objective	*Observability*											
Backtrace - CO	(Z, DB) --> (m, 1), (n, 1)											
CO	Choose (m, 1) {n& m equally hard to Control}											
	(d , 0) - d is a PI											



Example

PO=1 → No Possible Test

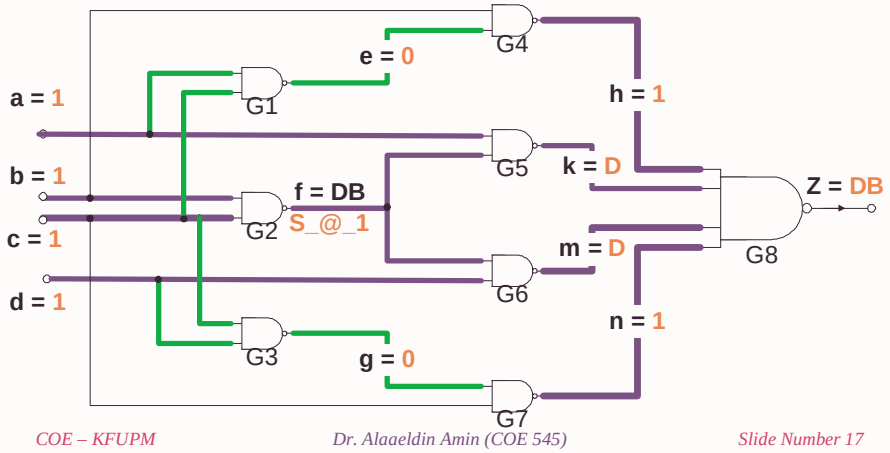
	a	b	c	d	e	f	g	h	k	m	n	z
tc(3)	1	1	1	0	0	DB	1	1	D	1	0	1
	Z = 1 --> No Test Possible											
	PI-Remake -- (d,1)											



Example

	a	b	c	d	e	f	g	h	k	m	n	z
tc(3)	1	1	1	1	0	DB	0	1	D	D	1	DB
D-Frontier												
Initial Objective	Met (Z, DB)											

PO=DB →Test Generated

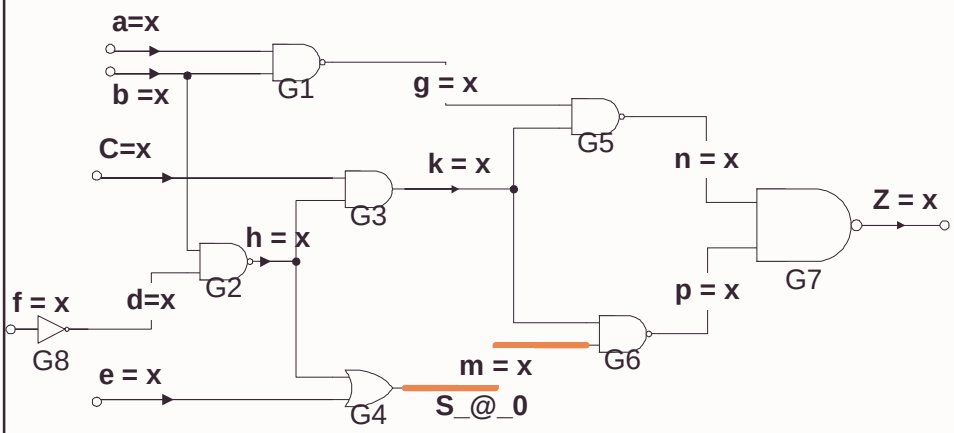


COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 17

Example 2

Excite (Provoke) the Fault (INITIAL OBJECTIVE)

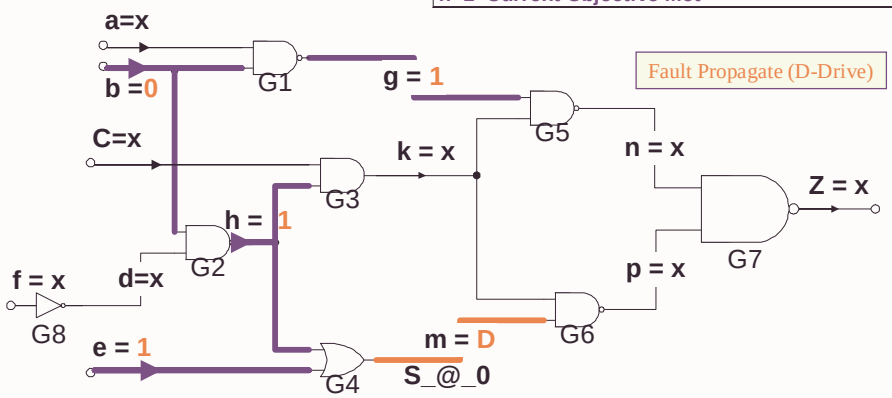
	a	b	c	d	e	f	g	h	k	m	n	p	z
	x	x	x	x	x	x	x	x	x	x	x	x	x
Initial Objective (I.O)	(m , D)												
BackTrace- Objective	(h , 1) OR (e , 1) - *e Easier to Control*												
Current Objective	(e , 1) -- PI Assignment												
Simulate	x	x	x	x	1	x	x	x	x	D	x	x	x
	INITIAL Objective Met												



COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 18

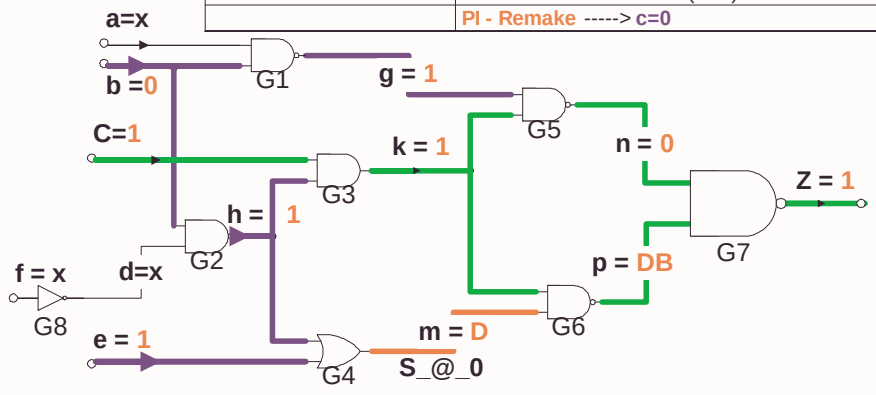
Example 2

	a	b	c	d	e	f	g	h	k	m	n	p	z
	x	x	x	x	1	x	x	x	x	D	x	x	x
D-Frontier	{G6}												
Initial Objective (I.O)	(p , DB)												
BackTrace- C. Objective	(k , 1) ---> **BOTH** (C , 1) AND (h , 1)												
Current Objective	(h , 1) ---- **Harder** To Control												
BackTrace	(b , 0) OR (d , 0) ----> (b , 0) **Easier** to Control												
Current Objective	(b , 0) -- *PI Assignment*												
SIMULATE	x	0	x	x	1	x	1	1	x	D	x	x	x
	h=1 Current Objective Met												



Example 2

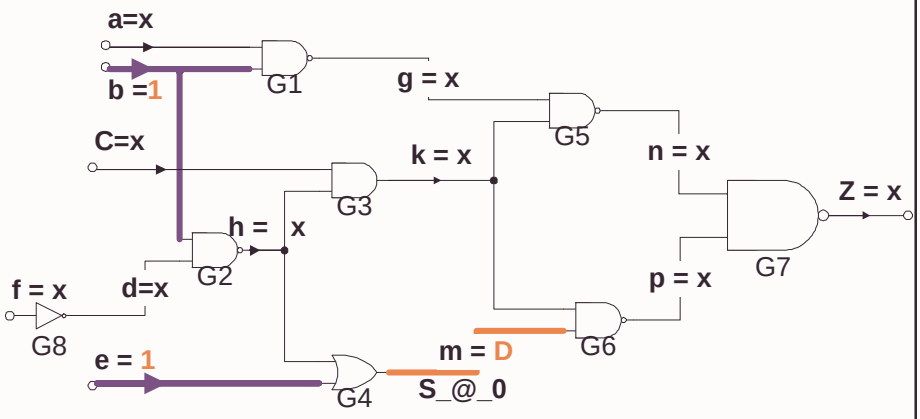
	a	b	c	d	e	f	g	h	k	m	n	p	z
	x	0	x	x	1	x	1	1	x	D	x	x	x
D-Frontier	{G6}												
Initial Objective (I.O)	(p , DB)												
BackTrace- C. Objective	(k , 1) ---> BOTH (C , 1) AND (h , 1)												
	(h , 1) ---- OBJECTIVE MET												
Current Objective	(C , 1) -- PI Assignment												
Simulate	x	0	1	x	1	x	1	1	1	D	0	DB	1
Fault Propagate (D-Drive	Initial Objective Met (p , DB)												
	Test is NOT POSSIBLE (Z=1)												
	PI - Remake -----> c=0												



Example 2

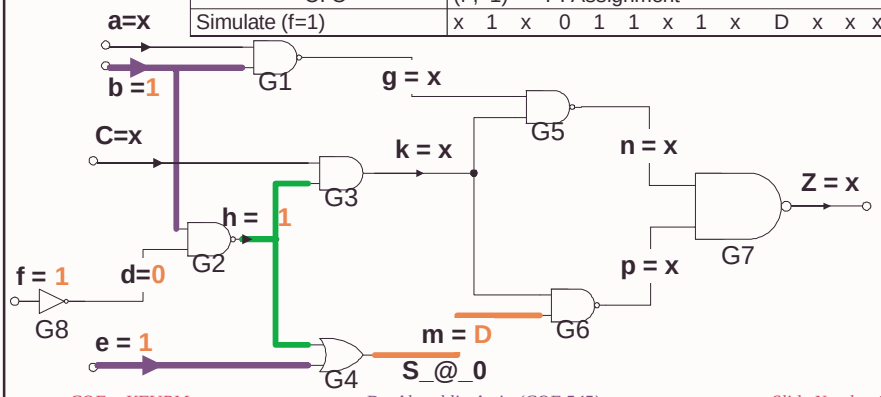
	a	b	c	d	e	f	g	h	k	m	n	p	z
PI-Remake (C = 0)	x	0	0	x	1	x	1	1	0	D	1	1	0
	TEST NOT POSSIBLE -- > PI Remake (b = 1)												
Simulate (C=x, b = 1)	x	1	x	x	1	x	x	x	x	D	x	x	x

PI-Remake & Simulate



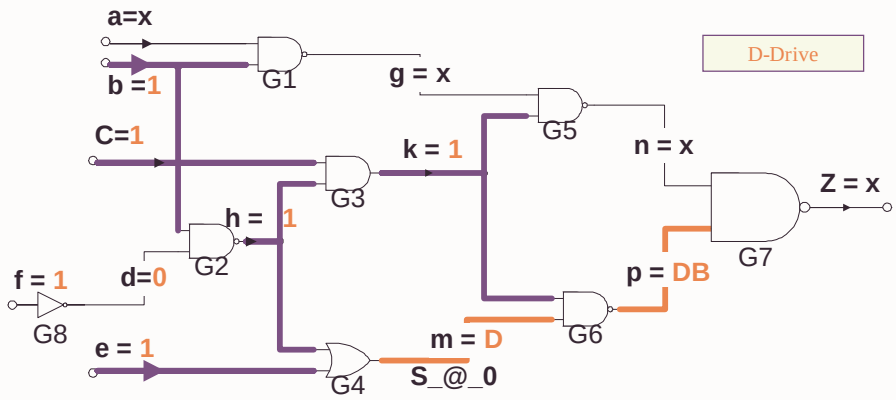
Example 2

	a	b	c	d	e	f	g	h	k	m	n	p	z
PI-Remake (b = 1)	x	1	x	x	1	x	x	x	x	D	x	x	x
D- Frontier	{ G6 }												
Initial Objective	(p , DB)												
D-Drive BackTrace - C. O.	(k , 1)												
C. O	(h , 1) AND (c , 1) --- (h , 1) Harder To Control												
C. O	(h , 1)												
C. O	(d , 0)												
C. O	(f , 1) -- PI Assignment												
Simulate (f=1)	x	1	x	0	1	1	x	1	x	D	x	x	x



Example 2

	a b c d e f g h k m n p z
D- Frontier	(p , DB)
Initial Objective	(k , 1)
BackTrace - C. O.	(h , 1) **AND** (c , 1) --- *(h , 1) Harder To Control*
C. O	(h , 1) -- Objective Met
C. O	(c , 1) -- *PI Assignment*
Simulate (C, 1)	x 1 1 0 1 1 x 1 1 D x DB x
	Initial Objective Met (p , DB)



Example 2

	a b c d e f g h k m n p z
	x 1 1 0 1 1 x 1 1 D x DB x
D- Frontier	{ G7 }
Initial Objective	(Z , D)
BackTrace - C. O.	(n , 1)
C. O	(g , 0)
C. O.	(a , 1) -- PI Assignment
Simulate -- (a=1)	1 1 1 0 1 1 0 1 1 D 1 DB D
	Test Vector Generated (PO = D)

D-Drive