Test without Fault Model

- Introduction
- Boolean Tests without Fault Model
  - Toggle Test
  - Design Verification
  - Exhaustive Test
  - Pseudo Exhaustive Test (PET)
    - Individual Output Verification
      - Dependence Matrix
      - Minimum IOV test
    - Segment Verification
- Conclusions

Examples in this PPT are courtesy of McCluskey's lecture note at Stanford U.

© National Taiwan University

Exhaustive Test

- Apply all possible input combination
  - Combinational circuits with *n* inputs
    - Test length = 2^n

- Advantages of Exhaustive Test
  - + No ATPG required
  - + No fault simulation
  - + Suitable for BIST (no storage needed)
  - + Very high fault coverage
- Disadvantages
  - – Long test length

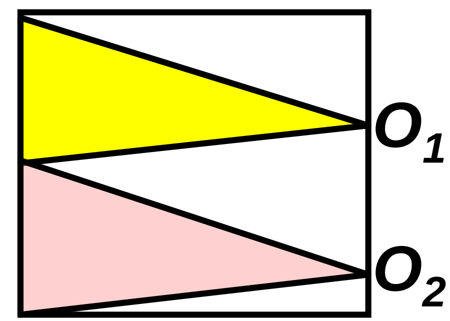
But… Really Need 2^n Patterns?

© National Taiwan University

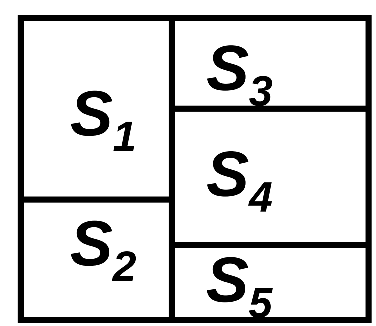
Pseudo Exhaustive Test (PET) [McCluskey 84]

- Idea
  - Do not need exhaustive test for whole circuit
  - Test each circuit partition exhaustively

- Two categories:
  1. *Individual Output Verification (IOV)*
     - Exhaustive test of each output
     - This video



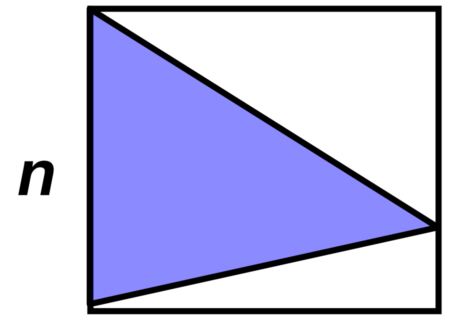
  2. *Segment Verification*
     - Exhaustive test of each segment
     - Next video



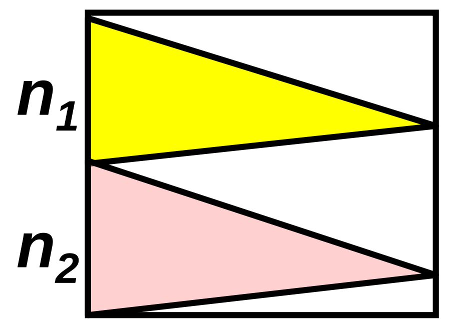
© National Taiwan University

IOV Test, Circuit Classification

- ***Full Dependence Circuit*** **- FDC**
  - **Some Outputs Depend on All Inputs**
  - **IOV not effective**
    - **Exhaustive test length: 2^n**
    - **IOV test length: 2^n**



- ***Partial Dependence Circuit*** **- PDC**
  - **No output depends on all inputs**
  - **IOV effectively reduce test length**
    - $2^{n1+n2} \gg (2^{n1} + 2^{n2})$

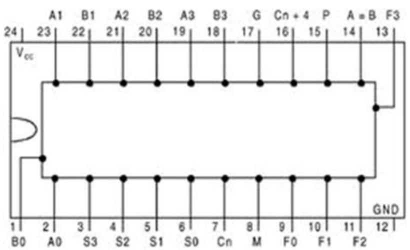


IOV Test is Useful to PDC, not FDC

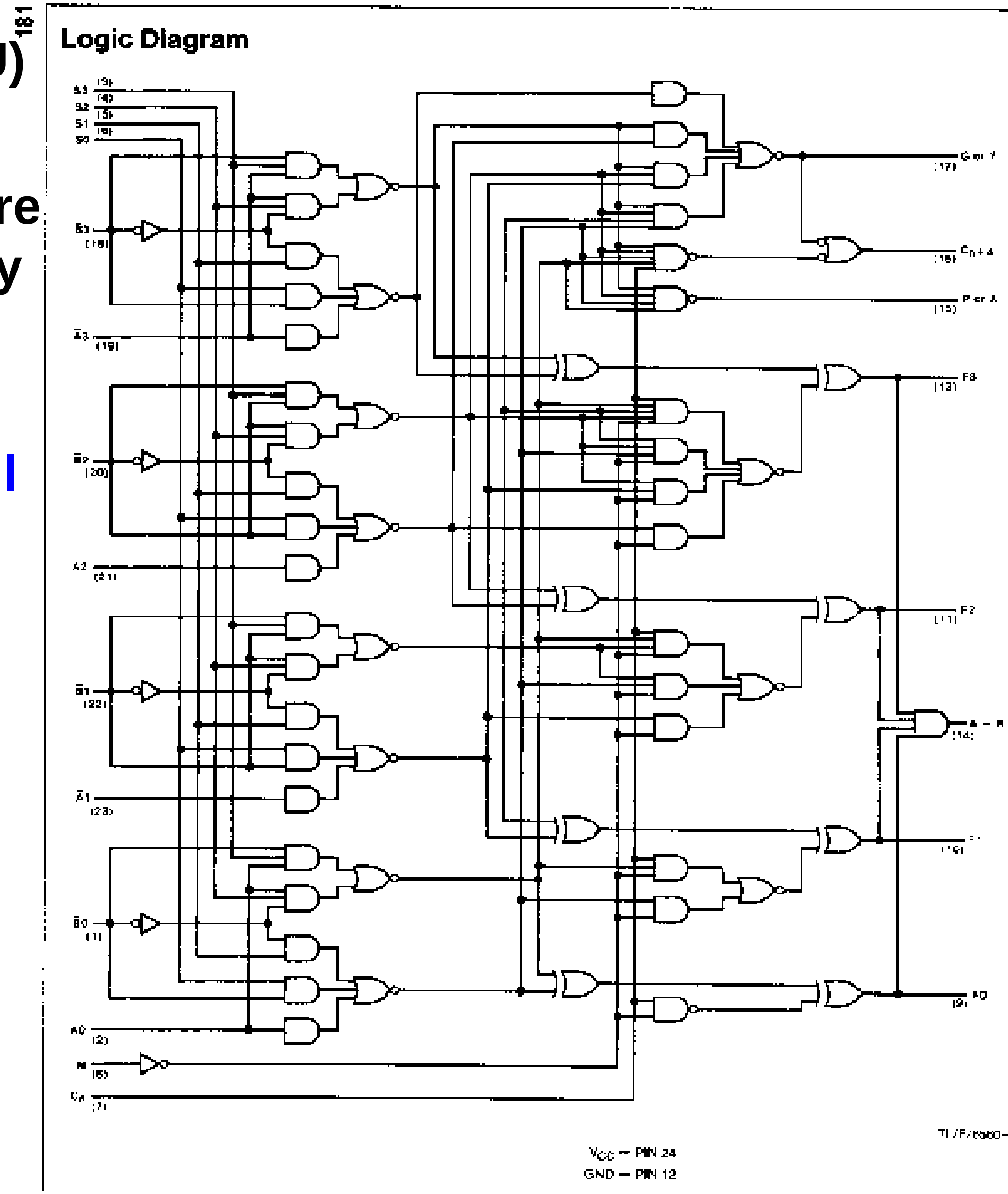
© National Taiwan University

FDC Example - 74181 ALU

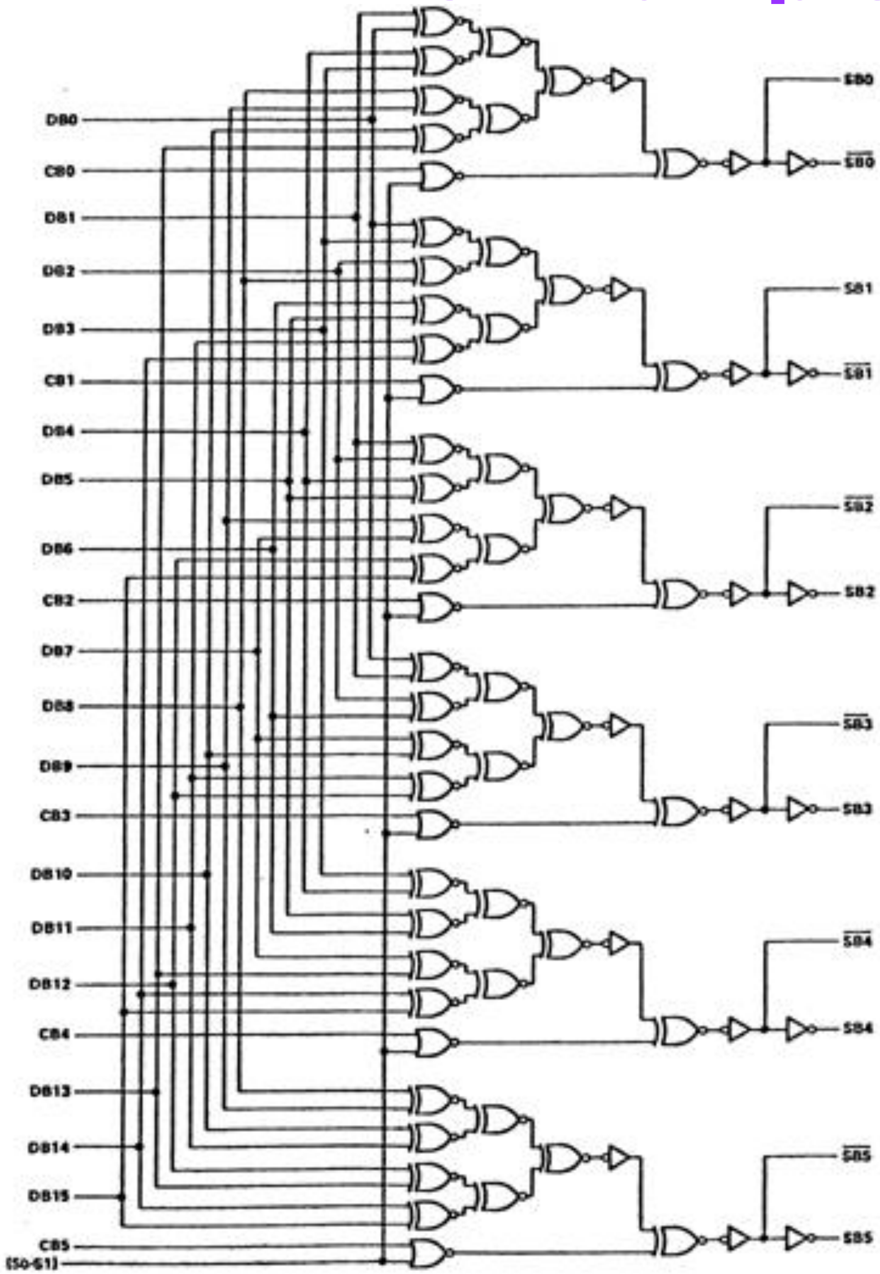
- **4-bit arithmetic logic unit (ALU)**
  - **First ALU on single chip**
  - **Used as arithmetic/logic core in CPUs of many historically significant mini-computers**
- **14 inputs, 8 outputs**
  - **Some outputs depend on all inputs**
  - **IOV =2^{14}=exhaustive**



IOV not Useful to FDC



PDC Example – 74LS630



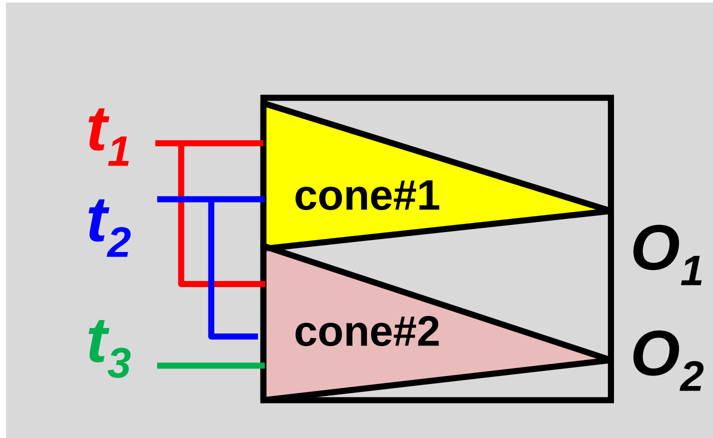
- 16-BIT parallel error detection and correction IC
- 24 inputs, 6 outputs
  - Each output depends on 10 inputs
  - Exhaustive test = 2^{24}
  - IOV test = 6×2^{10}

Can Do Better than 6×2^{10} ?

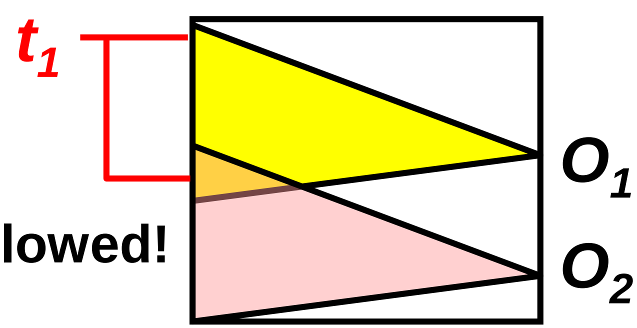
© National Taiwan University

We Can Share Test Signals !

- Share *test signals* to test each output simultaneously can further reduce TL



- If cones do NOT overlap
  - Can arbitrarily share test signals with inputs of different output
  - TL = $2^{n1}+2^{n2} \rightarrow 2^{max(n1,n2)}$
    - e.g. n_1=2, n_2=3; TL=2^3
- If cones overlap
  - CANNOT share test signals with inputs of same output



not allowed!

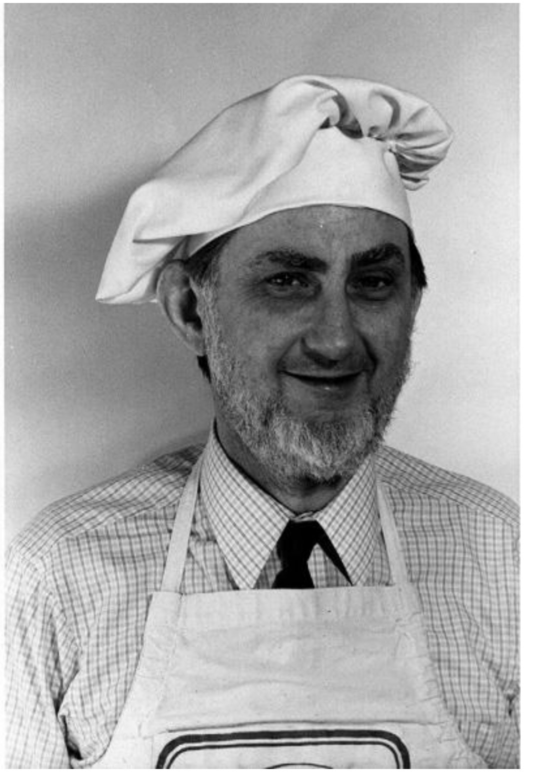
Cannot Share Inputs of Same Cone

© National Taiwan University

Test without Fault Model

- Introduction
- Boolean Tests without Fault Model
  - Toggle Test
  - Design Verification
  - Exhaustive Test
  - Pseudo Exhaustive Test (PET)
    - Individual Output Verification
      - Dependence Matrix
      - Minimum IOV test
    - Segment Verification
- Conclusions

How Many Test Signals We Can Share?

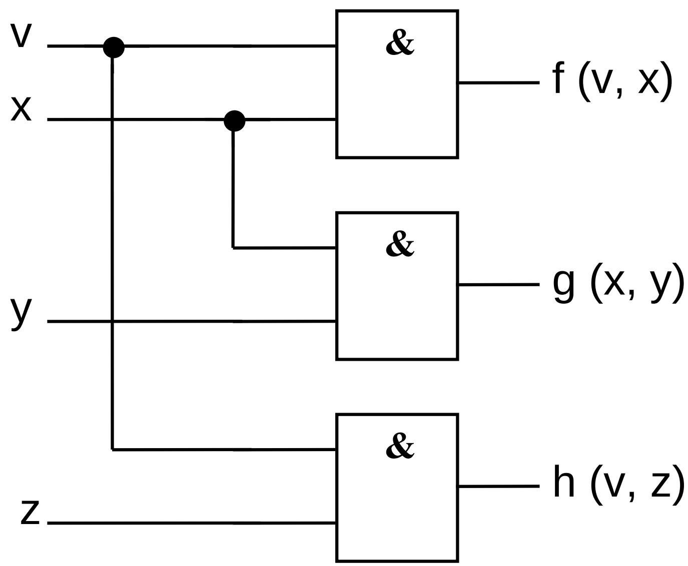


McCluskey and his collection of hats

© National Taiwan University

Dependence Matrix

- *Dependence matrix*
  - Each row is an output; each column is an input
  - D_{ij} = 1 iff output i depends on input j
  - *Row weight* = sum of each row
  - *Max Row Weight* = w = Max number of inputs to an output
- Example
  - w=2



Outputs	Inputs v	x	y	z	Row weight
f (v, x)	1	1	0	0	2
g (x, y)	0	1	1	0	2
h (v, z)	1	0	0	1	2

Max row weight =2=w

© National Taiwan University

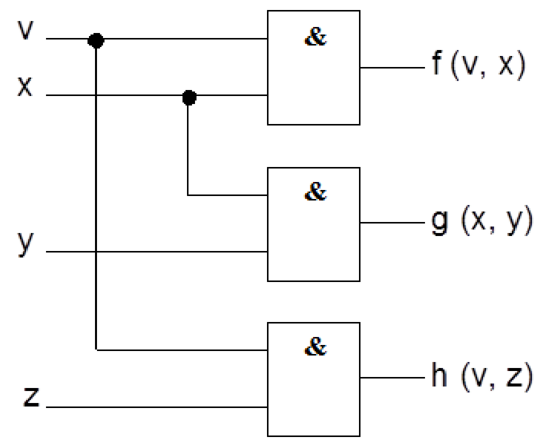
Partitioned Dependence Matrix

- ***Partitioned dependence matrix***
  - Each row of a partition has **at most one 1 entry**
    - Do not share test signals with inputs of same output
  - **Number of partitions = *p*** = Number of test signals needed
- **Example**
  - ***p*=3**

Outputs	v	x	y	z
f (v, x)	1	1	0	0
g (x, y)	0	1	1	0
h (v, z)	1	0	0	1

  - ***p*=2**

Outputs	v	y	x	z
f (v, x)	1	0	1	0
g (x, y)	0	1	1	0
h (v, z)	1	0	0	1



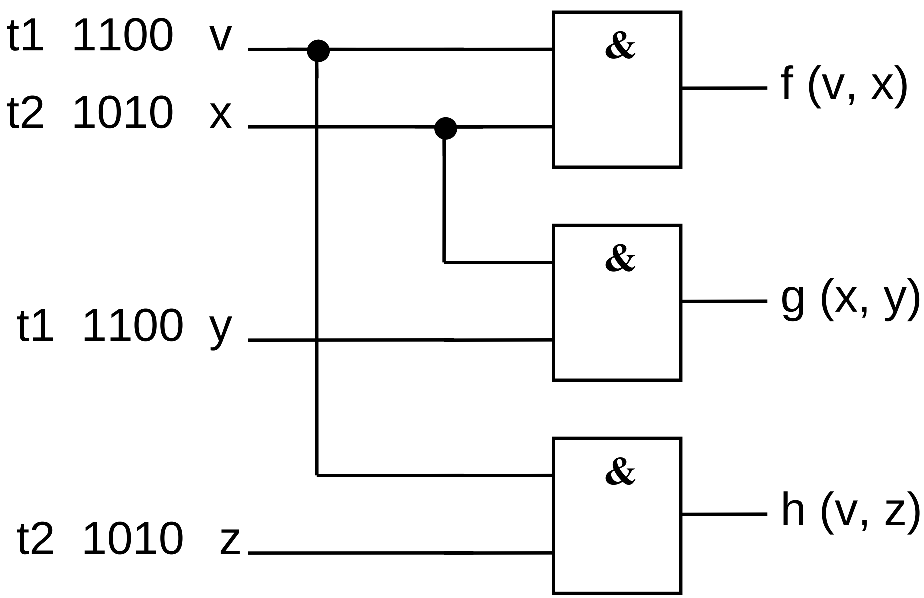
***p* ≥ *w*, Why?**

- **PDM is not unique. Smaller *p* is preferred**

© National Taiwan University

Minimum Test Length =?

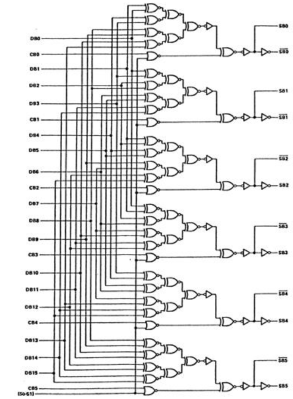
- Example: p=2; w=2
  - Number of test signals required = p = 2
  - Shortest test length = 2^w =4
  - Each output is exhaustively tested



Outputs	Inputs			
	v	y	x	z
f (v, x)	1	0	1	0
g (x, y)	0	1	1	0
h (v, z)	1	0	0	1

© National Taiwan University

DM of 74LS630

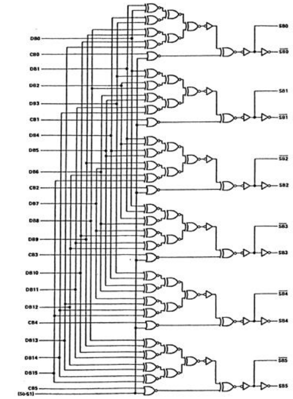


- **74LS630**
  - **24 inputs, 6 outputs, *w*=10**

	DATA BITS																CHECK BITS							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	1	2	3	4	5	S0.S1	rw
SB0	1	1	0	1	1	0	0	0	1	1	1	0	0	1	0	0	1	0	0	0	0	0	1	10
SB1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	10
SB2	0	1	1	0	1	1	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	0	1	10
SB3	1	1	1	0	0	0	1	1	0	0	1	1	1	0	0	0	0	0	0	1	0	0	1	10
SB4	0	0	0	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	1	0	1	10
SB5	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	10

© National Taiwan University

PDM of 74LS630



- p=w=10
- Minimum 10 test signals required
- Minimum test length = $6 \times 2^{10} \rightarrow 2^{10}$

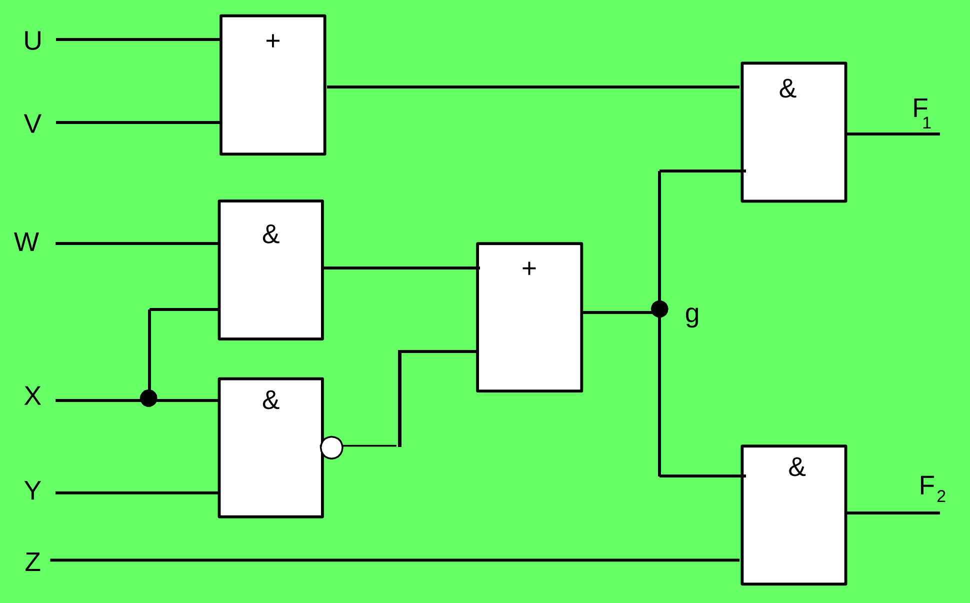
	DATA BITS																CHECK BITS						S0.S1
	0	15	1	14	3	12	4	11	8	7	9	6	10	5	13	2	0	1	2	3	4	5	
SB0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	1
SB1	1	0	0	1	1	0	0	1	1	0	0	1	0	1	0	1	0	1	0	0	0	0	1
SB2	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	1	0	0	1	0	0	0	1
SB3	1	0	1	0	0	1	0	1	0	1	0	1	1	0	0	1	0	0	0	1	0	0	1
SB4	0	1	0	1	1	0	1	0	0	1	0	1	0	1	1	0	0	0	0	0	1	0	1
SB5	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	0	0	0	0	0	1	1

p = 10, w = 10

© National Taiwan University

Quiz

Q: Find the partitioned dependence matrix for this circuit. p=? w=?



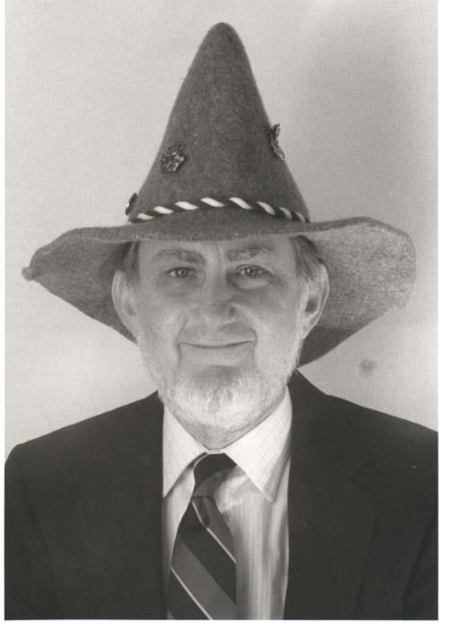
	W	X	Y	Z	U	V	Row Weight
F1							
F2							

© National Taiwan University

Test without Fault Model

- Introduction
- Boolean Tests without Fault Model
  - Toggle Test
  - Design Verification
  - Exhaustive Test
  - Pseudo Exhaustive Test (PET)
    - Individual Output Verification
      - Dependence Matrix
      - Minimum IOV Test
    - Segment Verification
- Conclusions

**Now We Have *p & w*,
What is Mini IOV Test?**

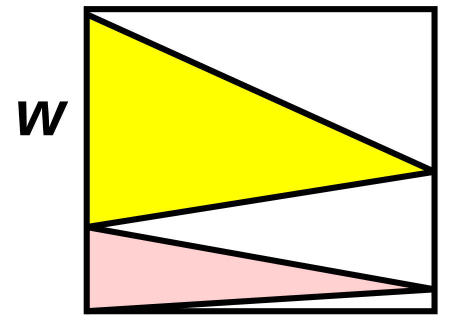


McCluskey and his collection of hats

© National Taiwan University

Minimum IOV Test Length

- Minimum IOV test length must $\geq 2^w$
  - why?



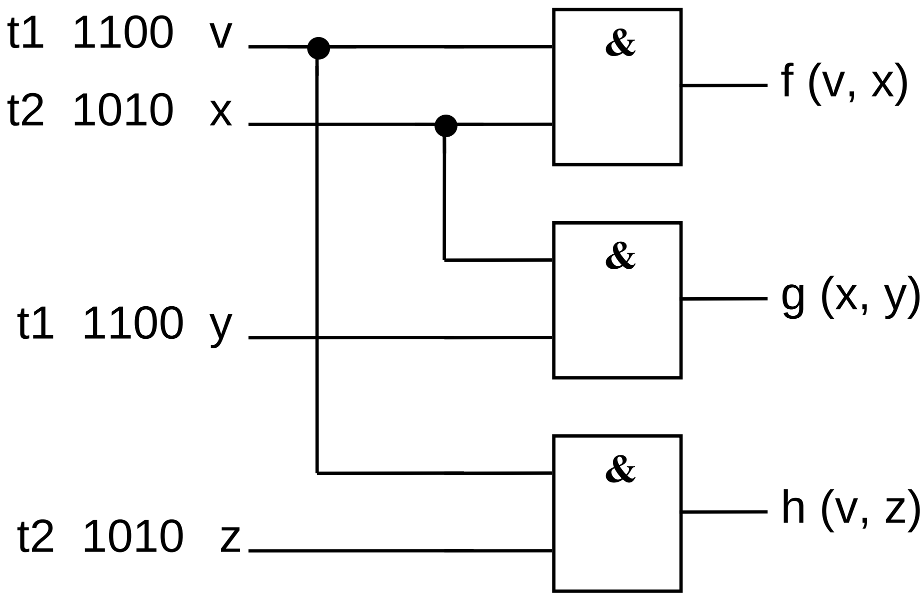
- Three cases:
  - ① $p = w$
    - Minimum IOV test length $= 2^w$ always possible
  - ② $p = w + 1$
    - Minimum IOV test length $= 2^w$ always possible
  - ③ $p > w + 1$
    - Minimum IOV test length $\geq 2^w$
    - 2^w may or may not achievable

Mini IOV TL Has 3 Cases

© National Taiwan University

① Example: *p=w*

- **Example: *p*=2, *w*=2**
  - **Minimum IOV test length = $2^w = 2^2 = 4$**



DP(N)

Outputs	Inputs v	y	x	z
f (v, x)	1	0	1	0
g (x, y)	0	1	1	0
h (v, z)	1	0	0	1

① Minimum IOV TL = 2^w

© National Taiwan University

② Example: $p=w+1$

- Given w test signals $t_1, t_2, \ldots t_w$
  - XOR them to generate $t_{w+1} = t_1 \oplus t_2 \oplus \ldots t_w$
- Example: $p = 3, w = 2$
  - $z = x \oplus y$
  - Any pair is exhaustive: (x,y) (y,z) (x,z). This is *orthogonal array*

dependency matrix

	x	y	z
f1	1	1	0
f2	0	1	1
f3	1	0	1

IOV Test Patterns

x	y	z=x⊕y
0	0	0
1	1	0
0	1	1
1	0	1

② Minimum IOV TL = 2^w

© National Taiwan University

Orthogonal Array [Rao 1946]

- Entries are a finite *symbol set*, like {0,1}
- For every selection of *p* columns
  - All ordered *w-tuples* of symbols appear same number of times
  - *w* is called *strength* of orthogonal array
- Example: OA with symbol set {1,0}, strength *w*=2
  - (0,0)(1,0)(0,1)(1,1) appear exactly once for every 2 columns
  - OA is NOT unique

x	y	z
0	0	0
1	1	0
0	1	1
1	0	1

x	y	z
1	1	1
1	0	0
0	1	0
0	0	1

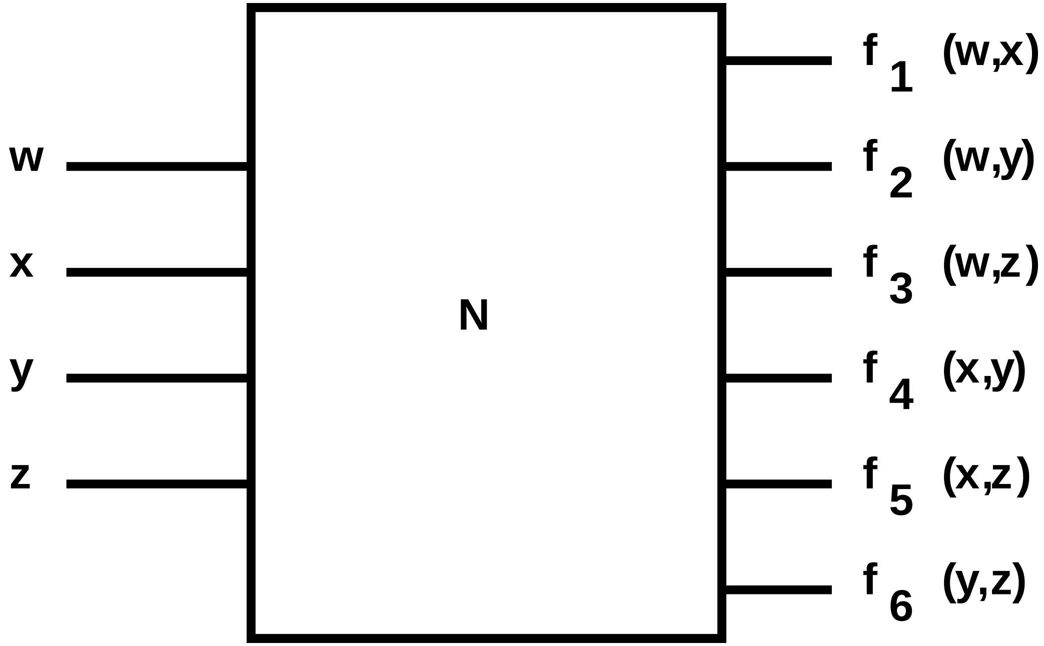
OA Provides IOV Test for *p=w+1*

© National Taiwan University

③ Example: $p>w+1$

- Example: $p=4$, $w=2$
  - Minimum IOV test length = row = $5 > 2^w$

w	x	y	z
1	1	1	0
1	1	0	1
1	0	1	1
0	1	1	1
0	0	0	0



How to Find IOV Test for Given $p>w+1$?

© National Taiwan University

Universal IOV Test Table [McCluskey 84]

- **This table provides general solution for *p > w*+1**
- **U[*i,j*] represents p-column matrix with**
  - **all possible weight *i* and all possible weight *j* rows**

w	U[i, j]		Range	Number of Tests
2	U[0, p-1]	U[1, p]	p > 3	p+1
3	U[1, p-1]		p > 4	2p
4	U[1, p-2]	U[2, p-1]	p > 5	p(p+1)/2
5	U[2, p-2]		p > 6	p(p-1)
6	U[2, p-3]	U[3, p-2]	p > 8	B(p+1, 3)
6	U[1, 4, 7]		p = 8	p(p+1)/2
7	U[3, p-3]		p > 9	2 B(p, 3)
7	U[0, 3, 6, 9]		p = 9	170
8	U[3, p-4]	U[4, p-3]	p > 11	B(p+1, 4)
8	U[0, 3, 6, 9]	U[1, 4, 7, 10]	p = 10	341
8	U[0, 4, 8]	U[3, 7, 9]	p = 11	496
9	U[4, p-4]		p > 12	2 B(p, 4)
9	U[1, 4, 7, 10]		p = 11	682
9	U[0, 4, 8, 12]		p = 12	992
10	U[4, p-5]	U[5, p-4]	p > 14	B(p+1, 5)
10	U[1, 4, 7, 10]	U[2, 5, 8, 11]	p = 12	1365
10	U[0, 4, 8, 12]	U[1, 5, 9, 13]	p = 13	2016
10	U[0, 5, 10]	U[4, 9, 14]	p = 14	3004

B(i,j)* is *binomial coefficient, i* things taken *j

© National Taiwan University

Example: p=4, w=2

w	U[i, j]		Range	Number of Tests
2	U[0, p-1]	U[1, p]	p > 3	p+1
3	U[1, p-1]		p > 4	2p
4	U[1, p-2]	U[2, p-1]	p > 5	p(p+1)/2
5	U[2, p-2]		p > 6	p(p-1)
6	U[2, p-3]	U[3, p-2]	p > 8	B(p+1, 3)
6	U[1, 4, 7]		p = 8	p(p+1)/2
7	U[3, p-3]		p > 9	2 B(p, 3)
7	U[0, 3, 6, 9]		p = 9	170
8	U[3, p-4]	U[4, p-3]	p > 11	B(p+1, 4)
8	U[0, 3, 6, 9]	U[1, 4, 7, 10]	p = 10	341
8	U[0, 4, 8]	U[3, 7, 9]	p = 11	496
9	U[4, p-4]		p > 12	2 B(p, 4)
9	U[1, 4, 7, 10]		p = 11	682
9	U[0, 4, 8, 12]		p = 12	992
10	U[4, p-5]	U[5, p-4]	p > 14	B(p+1, 5)
10	U[1, 4, 7, 10]	U[2, 5, 8, 11]	p = 12	1365
10	U[0, 4, 8, 12]	U[1, 5, 9, 13]	p = 13	2016
10	U[0, 5, 10]	U[4, 9, 14]	p = 14	3004

- **Choice 1**
- **U[0, p-1]=U[0,3]**

t_1	t_2	t_3	t_4
0	0	0	0
1	1	1	0
0	1	1	1
1	1	0	1
1	0	1	1

RW=0

RW=3

- **Choice 2**
- **U[1, p]=U[1,4]**

t_1	t_2	t_3	t_4
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	1	1	1

RW=1

RW=4

© National Taiwan University

③ Example: $p > w + 1$

- Universal IOV table may **NOT be optimal** solution
  - Sometimes, we can do better than universal IOV
- Example: $p=5$, $w=3$
  - Minimum IOV TL = 8 = $2^w < 2p$

w	U[i, j]		Range	Number of Tests
2	U[0, p-1]	U[1, p]	p > 3	p+1
3	U[1, p-1]		p > 4	2p
4	U[1, p-2]	U[2, p-1]	p > 5	p(p+1)/2
5	U[2, p-2]		p > 6	p(p-1)
6	U[2, p-3]	U[3, p-2]	p > 8	B(p+1, 3)
6	U[1, 4, 7]		p = 8	p(p+1)/2

Dependence Matrix

	u	v	X	Y	z
f_1 (uvx)	1	1	1	0	0
f_2 (uvy)	1	1	0	1	0
f_3 (uxz)	1	0	1	0	1
f_4 (uyz)	1	0	0	1	1
f_5 (vxy)	0	1	1	1	0
f_6 (vxz)	0	1	1	0	1
f_7 (wyz)	0	1	0	1	1
f_8 (xyz)	0	0	1	1	1

Mini IOV Test

Test Patterns

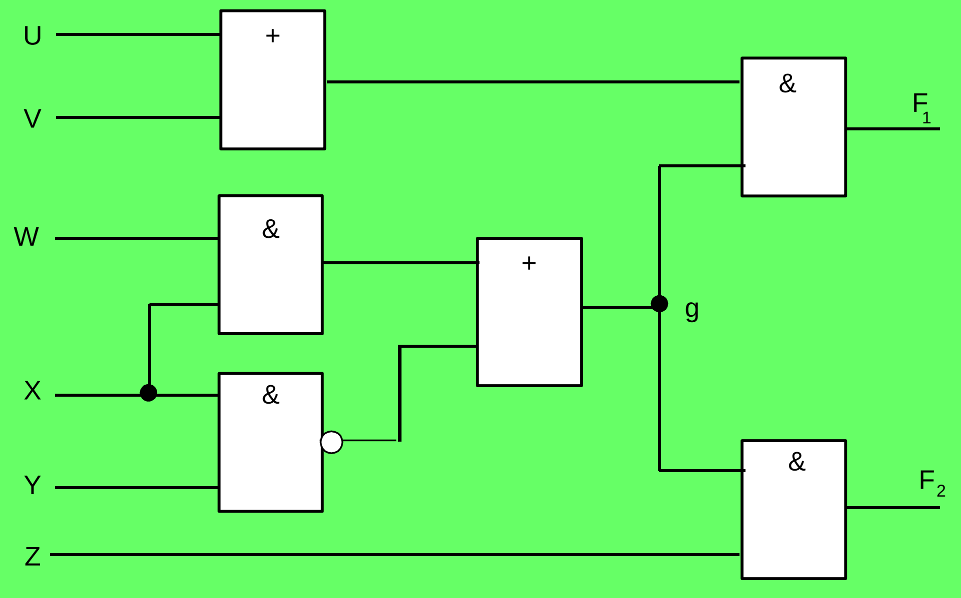
u	v	X	Y	z
1	0	0	0	0
0	0	0	1	1
0	0	1	0	1
1	0	1	1	0
1	1	0	0	1
0	1	0	1	0
0	1	1	0	0
1	1	1	1	1

© National Taiwan University

Quiz

Q1: (Cont'd) *p*=5 *w*=5. Which case is this?

Q2: What is minimum IOV test length?

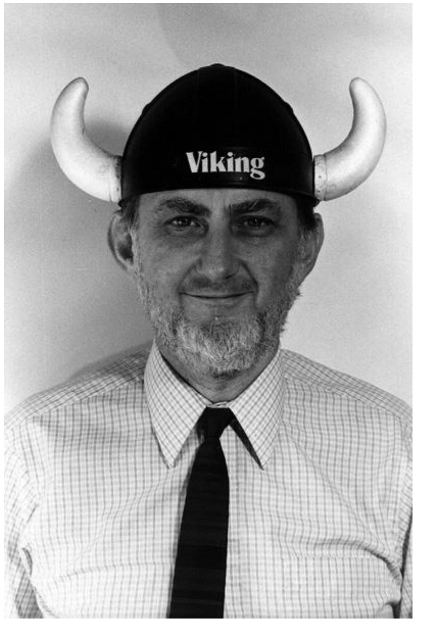


Can We Do Better ? See Next Video

© National Taiwan University

Summary

- Individual Output Verification (IOV)
  - Test every output exhaustively
- Q1: How many test signals?
  - Partitioned dependence matrix
  - w =max row weight
  - p = partition
- Q2: What is minimum IOV TL?
  - $p = w$: TL $= 2^w$
  - $p = w + 1$: TL $= 2^w$
    - Orthogonal Array
  - $p > w + 1$: TL $\geq 2^w$
    - Universal IOV test table



McCluskey and his collection of hats

© National Taiwan University

FFT

- Q: $p \geq w$, why?

- Three cases:
  - $p = w$
    - Minimum IOV test length $= 2^w$
  - $p = w + 1$
    - Minimum IOV test length $= 2^w$
  - $p > w + 1$
    - Minimum IOV test length $\geq 2^w$

© National Taiwan University

Orthogonal Array [Rao 1946]

- ***N x k*** **array**
- **Entries are a finite *symbol set*, {1,2,...,*s*}**
- **For every selection of *t* columns**
  - **All *t-tuples* of symbols (s^t) appear same number of times**
  - **t is called *strength* of orthogonal array**
- **Example: OA with symbol set {1,0}, strength =2**
  - **N=4, k=3, s=2, t =2**
  - **(0,0)(1,0)(0,1)(1,1) appear exactly once for every 2 columns**

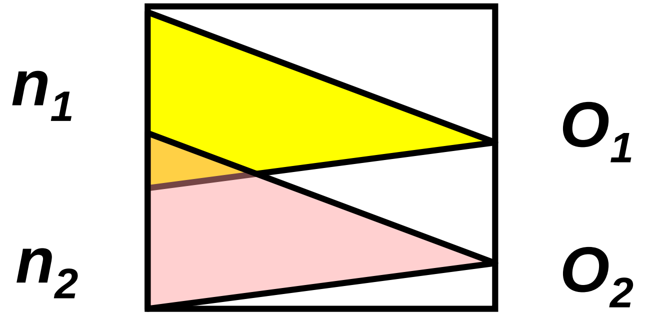
1	1	1
0	0	1
1	0	0
0	1	0

© National Taiwan University

Individual Output Verification

- Apply all possible input combination to each *individual output*

- Reduce test length
  - Exhaustive test length: 2^{n1+n2}
  - IOV test length: $2^{n1} + 2^{n2}$



IOV Reduce TL

© National Taiwan University