Test without Fault Model

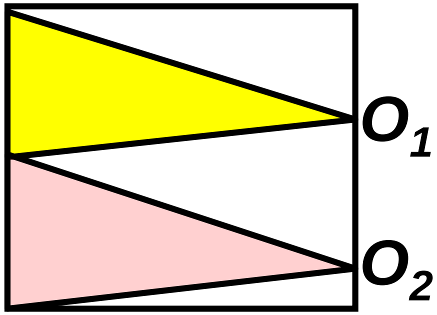
- Introduction
- Boolean Tests without Fault Model
  - Toggle Test
  - Design Verification
  - Exhaustive Test
  - Pseudo Exhaustive Test (PET)
    - Individual Output Verification
    - Segment Verification
      - Path sensitization
      - MUX Insertion
- Conclusion

Some examples in this PPT are from McCluskey's original lecture notes @Stanford University

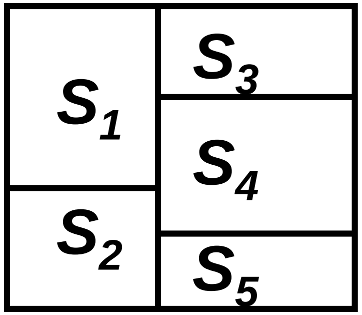
Review: Pseudo Exhaustive Test

- 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
     - Last video

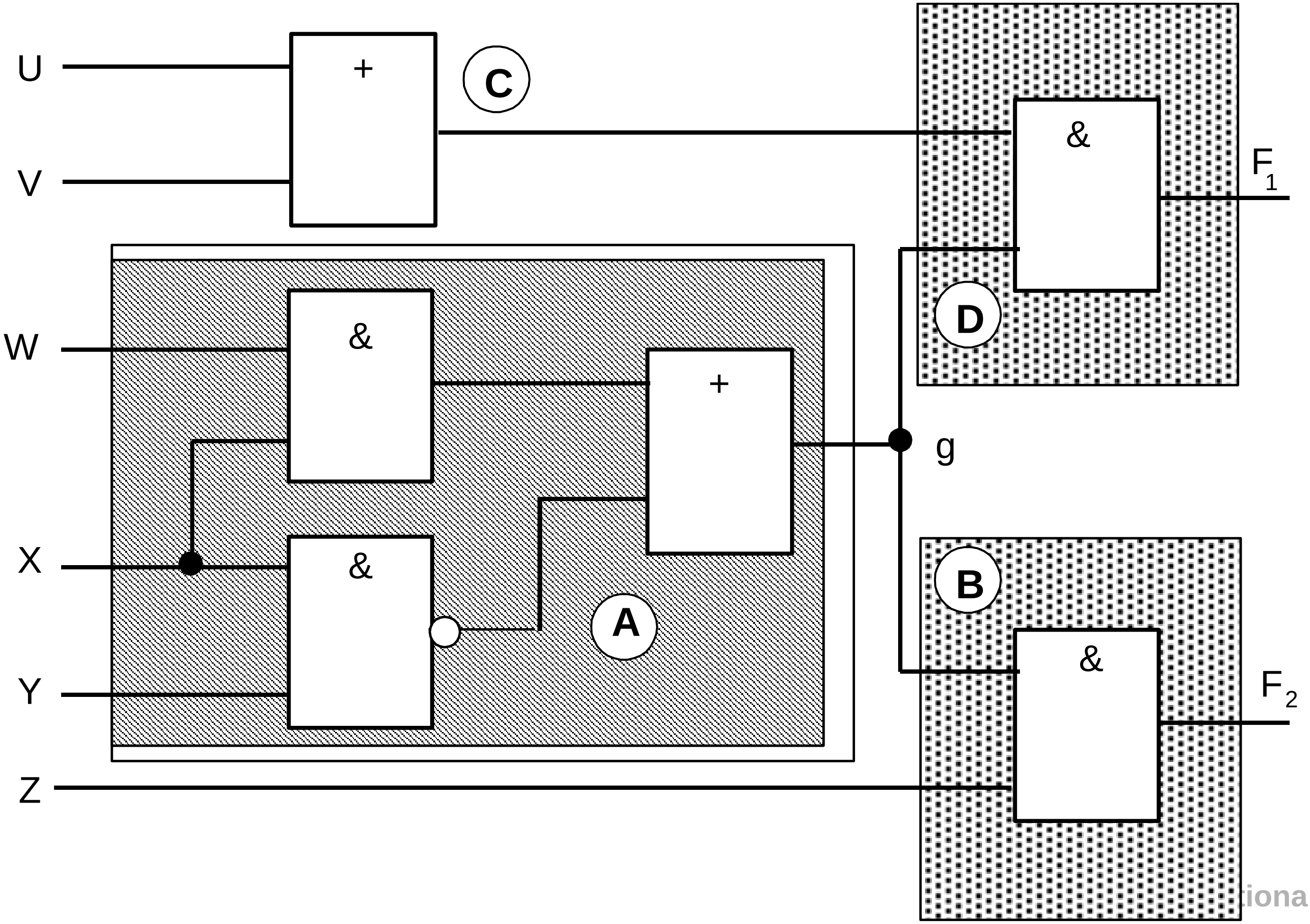


  2. *Segment Verification*
     - Exhaustive test of each segment
     - This video



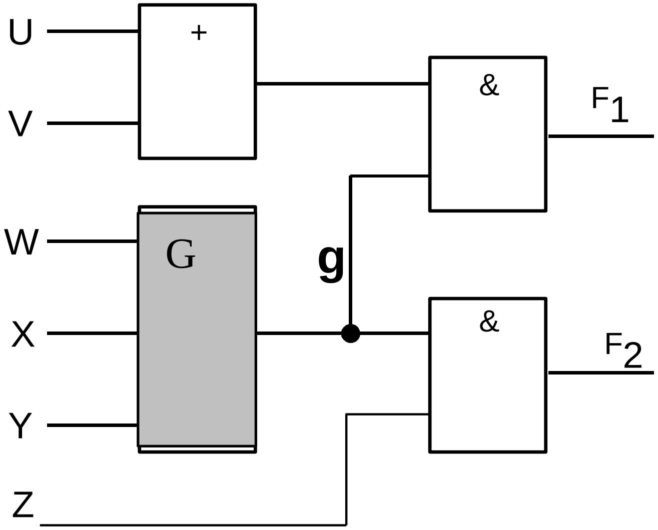
Segment Verification [McCluskey 81]

- Idea: Partition circuit, test each segment exhaustively
- Example: Partitioned into 4 segments: A, B, C, D



Path Sensitization (1) - Test A

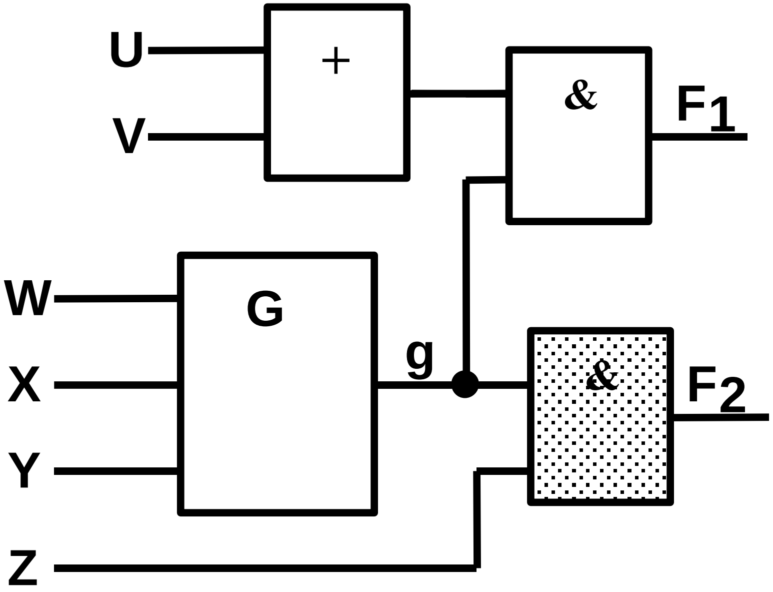
- **Sensitize to F2: Set Z to 1**
- **8 test patterns**



	U	V	W	X	Y	Z	g	F_1	F_2
1			0	0	0	1	1		1
2			0	0	1	1	1		1
3			0	1	0	1	1		1
4			0	1	1	1	0		0
5			1	0	0	1	1		1
6			1	0	1	1	1		1
7			1	1	0	1	1		1
8			1	1	1	1	1		1

Path Sensitization (2) - Test B

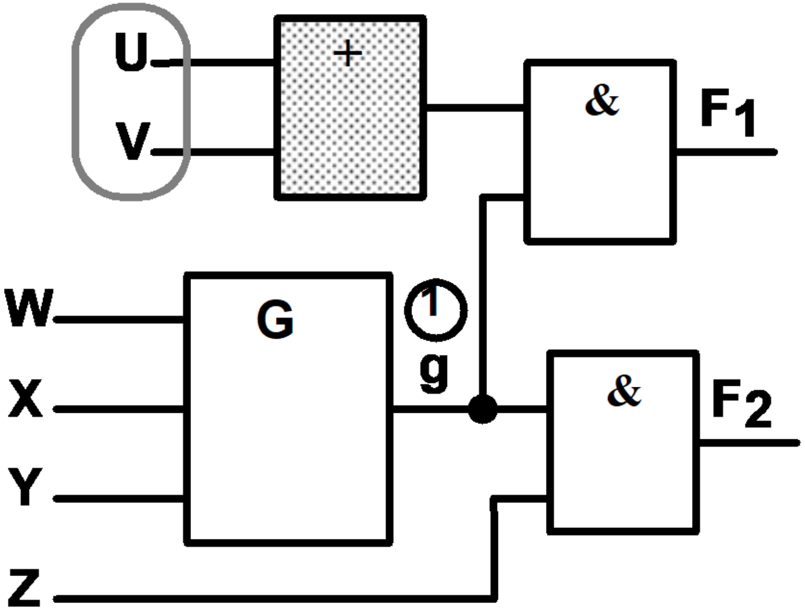
- Add two extra patterns



UV	WXY	Z	g	F_1	F_2
	000	1	1		1
	001	1	1		1
	010	1	1		1
	011	1	0		0
	100	1	1		1
	101	1	1		1
	110	1	1		1
	111	1	1		1
	111	0	1		0
	011	0	0		0

Path Sensitization (3) - Test C

- Sensitized to F1
- No extra pattern needed

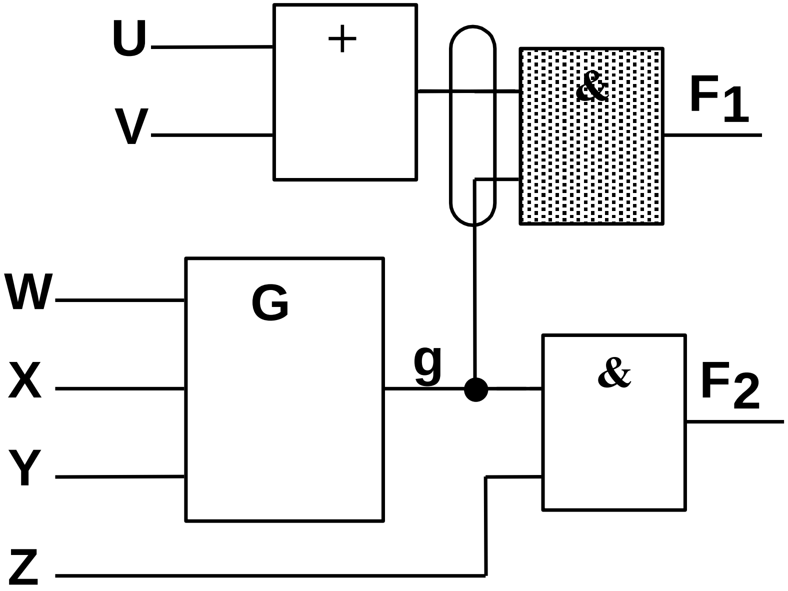


UV OR gate
sensitized
to F1   by g = 1.

UV	WXY	Z	g	F_1	F_2
	000	1	1		1
	001	1	1		1
	010	1	1		1
	011	1	0		0
00	100	1	1	0	1
01	101	1	1	1	1
10	110	1	1	1	1
11	111	1	1	1	1
	111	0	1		0
	011	0	0		0

Path Sensitization (4) - Test D

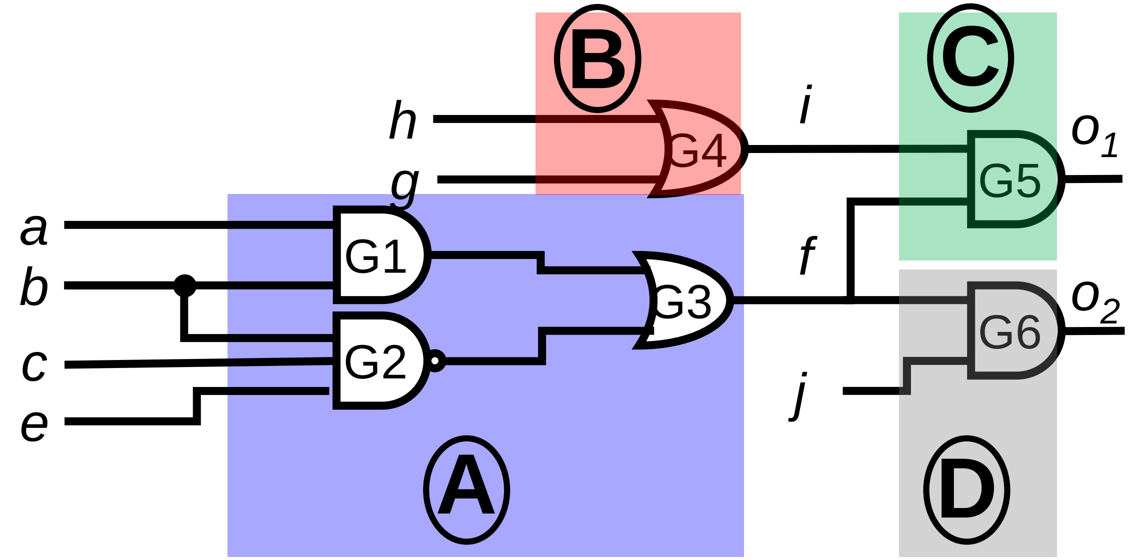
- No extra pattern needed
- Total 10 patterns



UV	WXY	Z	g	F_1	F_2
	000	1	1		1
	001	1	1		1
	010	1	1		1
00	011	1	0	0	0
00	100	1	1	0	1
01	101	1	1	1	1
10	110	1	1	1	1
11	111	1	1	1	1
	111	0	1		0
11	011	0	0	0	0

Quiz

Q: Exhaustive test length is 2^7=128.
Find minimum SV test for this circuit.
Test length =?
(Hint: ABCD 4 partitions.)



	a	b	c	e	f	g	h	i	j
1	0	0	0	0					
2	0	0	0	1					
3	0	0	1	0					
4	0	0	1	1					
5	0	1	0	0					
6	0	1	0	1					
7	0	1	1	0					
8	0	1	1	1					
9	1	0	0	0					
10	1	0	0	1					
11	1	0	1	0					
12	1	0	1	1					
13	1	1	0	0					
14	1	1	0	1					
15	1	1	1	0					
16	1	1	1	1					
17									
18									
19									
20									

Test without Fault Model

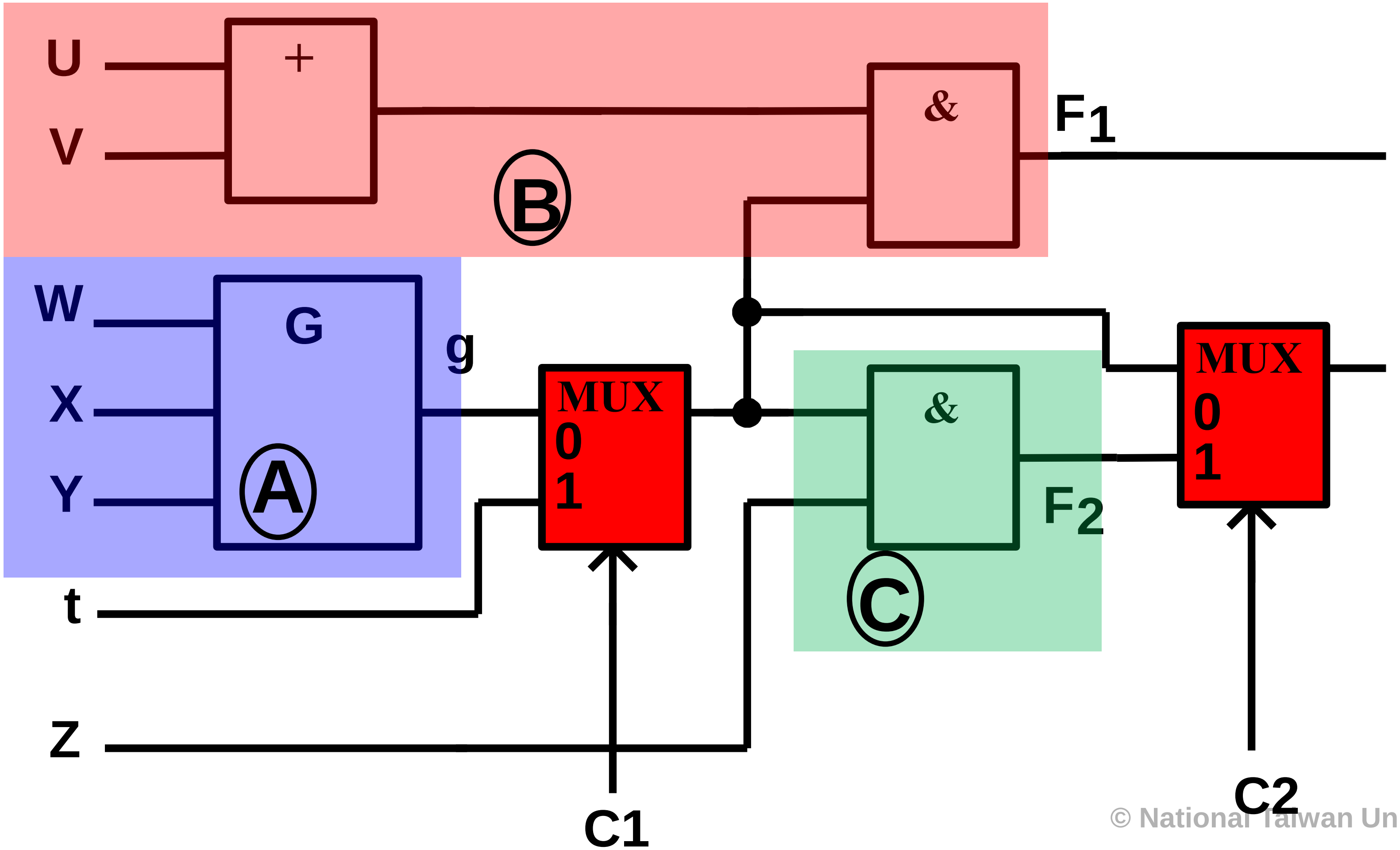
- Introduction
- Boolean Tests without Fault Model
  - Toggle Test
  - Design Verification
  - Exhaustive Test
  - Pseudo Exhaustive Test (PET)
    - Individual Output Verification
    - Segment Verification
      - Path sensitization
      - MUX Insertion
- Conclusions

McCluskey and his collection of hats

Any Simpler Method?

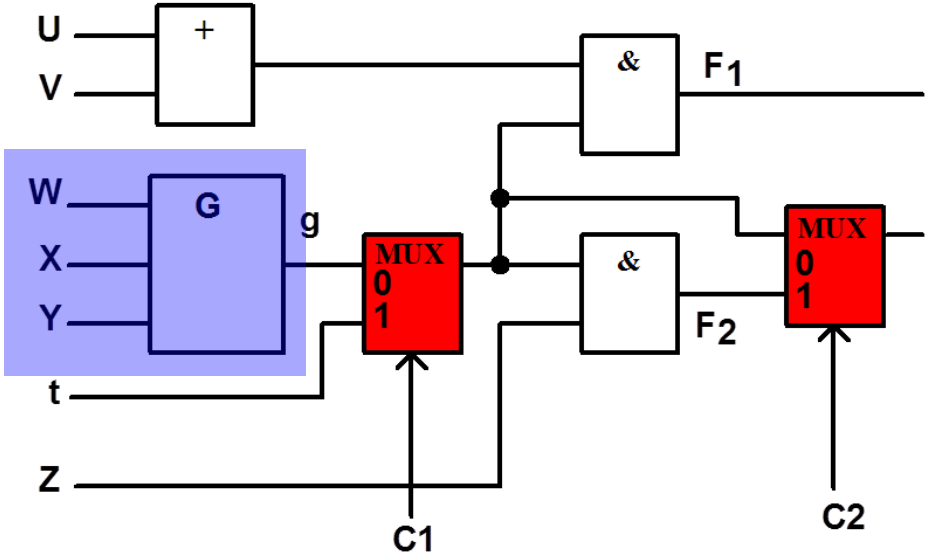
MUX Insertion

- Partition into 3 segments: A, B, C
- Area overhead: 3 test signals, 2 MUX
- Fully control and observe each segment



Test A

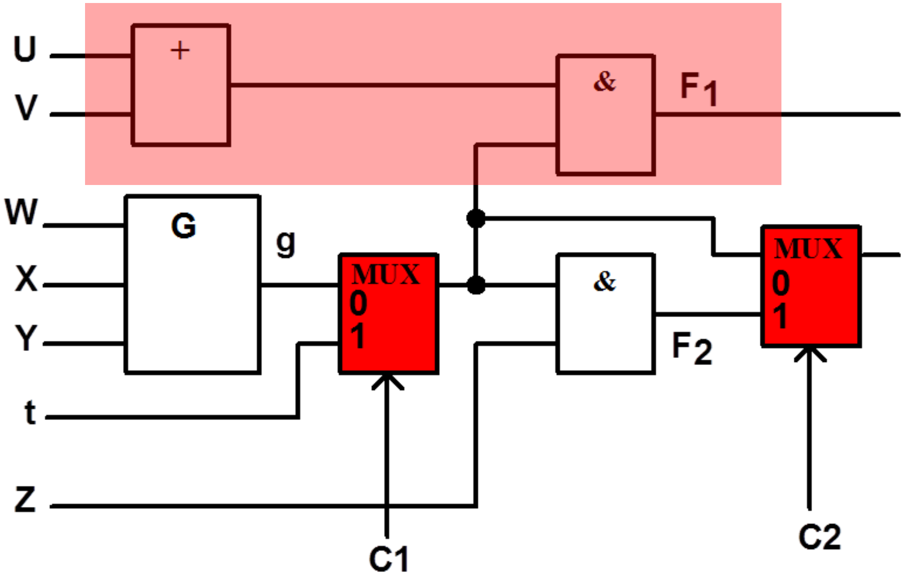
- 8 test patterns



	U	V	W	X	Y	Z	g	t	C_1	C_2
1			0	0	0		1		0	0
2			0	0	1		1		0	0
3			0	1	0		1		0	0
4			0	1	1		0		0	0
5			1	0	0		1		0	0
6			1	0	1		1		0	0
7			1	1	0		1		0	0
8			1	1	1		1		0	0
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										

Test B

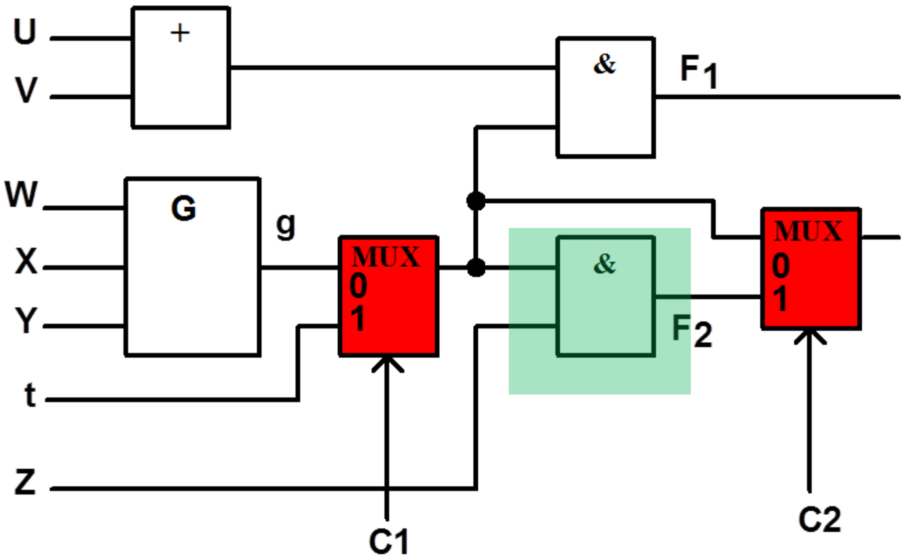
- add 8 test patterns



	U	V	W	X	Y	Z	g	t	C_1	C_2
1			0	0	0		1		0	0
2			0	0	1		1		0	0
3			0	1	0		1		0	0
4			0	1	1		0		0	0
5			1	0	0		1		0	0
6			1	0	1		1		0	0
7			1	1	0		1		0	0
8			1	1	1		1		0	0
9	0	0						0	1	
10	0	0						1	1	
11	0	1						0	1	
12	0	1						1	1	
13	1	0						0	1	
14	1	0						1	1	
15	1	1						0	1	
16	1	1						1	1	
17										
18										
19										
20										

Test C

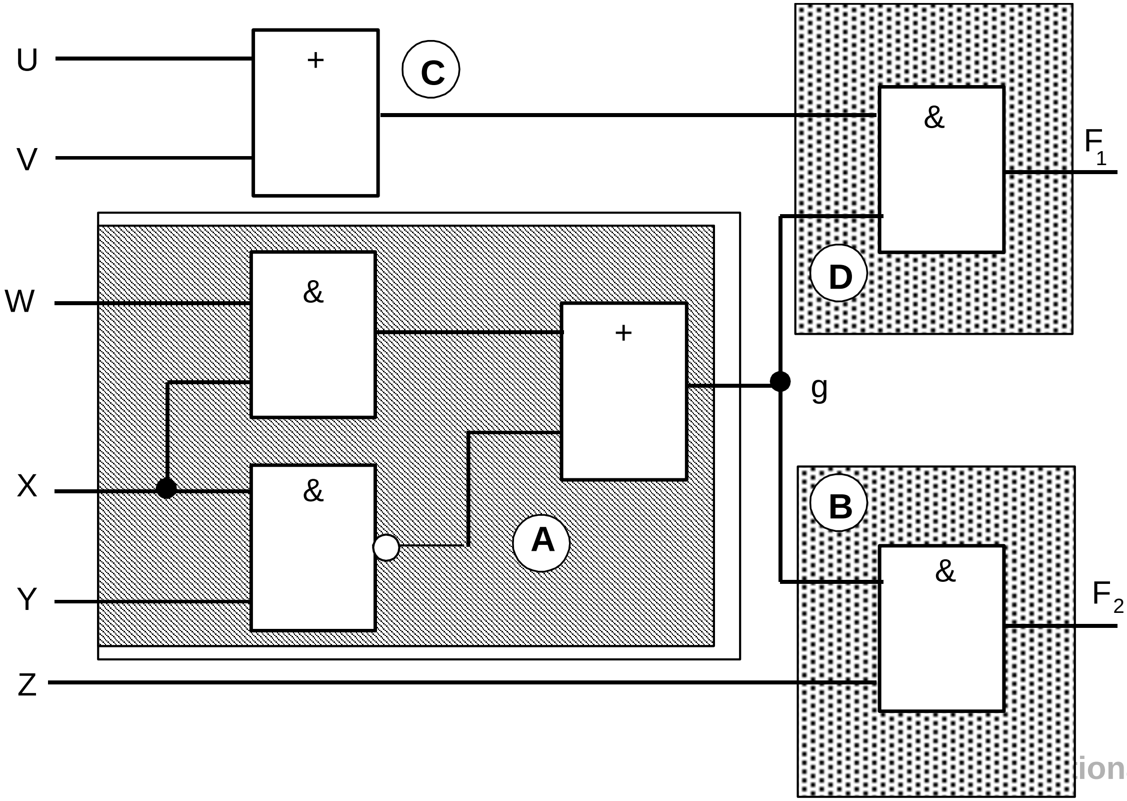
- add 4 test patterns
- totally test length = 20



	U	V	W	X	Y	Z	g	t	C_1	C_2
1			0	0	0		1		0	0
2			0	0	1		1		0	0
3			0	1	0		1		0	0
4			0	1	1		0		0	0
5			1	0	0		1		0	0
6			1	0	1		1		0	0
7			1	1	0		1		0	0
8			1	1	1		1		0	0
9	0	0						0	1	
10	0	0						1	1	
11	0	1						0	1	
12	0	1						1	1	
13	1	0						0	1	
14	1	0						1	1	
15	1	1						0	1	
16	1	1						1	1	
17						0		0	1	1
18						0		1	1	1
19						1		0	1	1
20						1		1	1	1

Conclusion

- Exhaustive test = 64
- PET Individual Output Verification = 32
- PET Segment Verification
  - Path sensitization = 10
  - MUX = 20



Summary

- Pseudo Exhaustive Test (PET)
  - Individual Output Verification
    - Test each output exhaustively
  - Segment Verification
    - Test each segment exhaustively
- Two SV techniques

① Path sensitization
  - Sensitize by test pattern
  - No hardware but difficult to find test

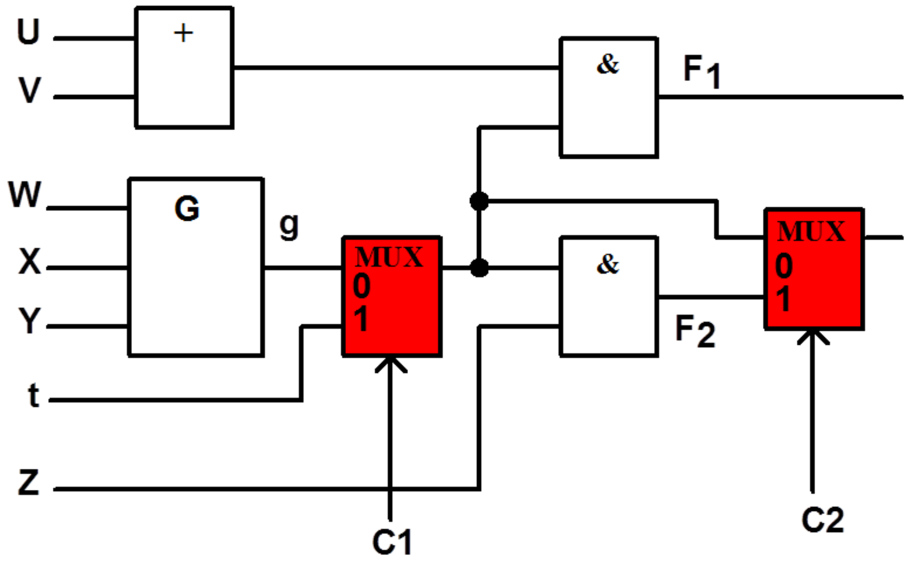
② MUX Insertion
  - Add MUX to control/observe
  - Need hardware but easier to find test

McCluskey and his collection of hats

- PET effectively reduce test length. Good for BIST

FFT

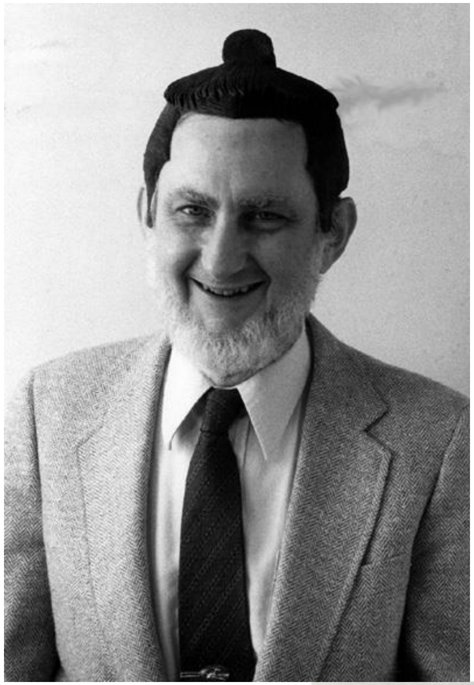
- **Q: Can we do better than 20 ?**

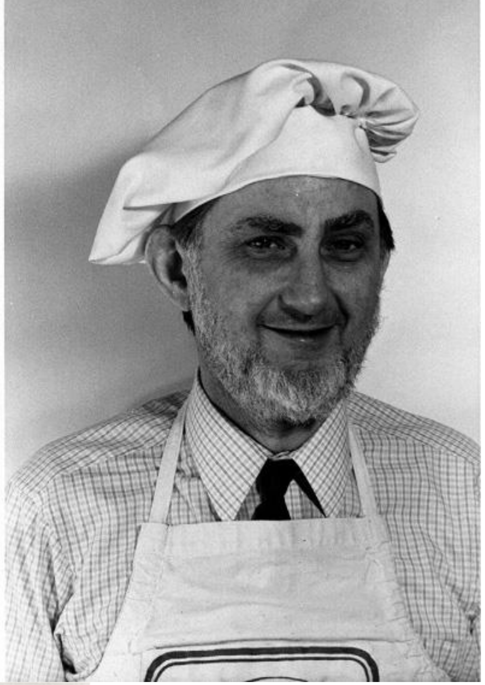


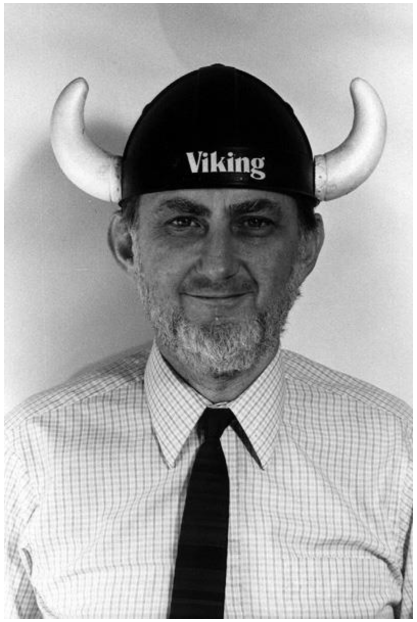
	U	V	W	X	Y	Z	g	t	C_1	C_2
1			0	0	0		1		0	0
2			0	0	1		1		0	0
3			0	1	0		1		0	0
4			0	1	1		0		0	0
5			1	0	0		1		0	0
6			1	0	1		1		0	0
7			1	1	0		1		0	0
8			1	1	1		1		0	0
9	0	0						0	1	
10	0	0						1	1	
11	0	1						0	1	
12	0	1						1	1	
13	1	0						0	1	
14	1	0						1	1	
15	1	1						0	1	
16	1	1						1	1	
17						0		0	1	1
18						0		1	1	1
19						1		0	1	1
20						1		1	1	1

References

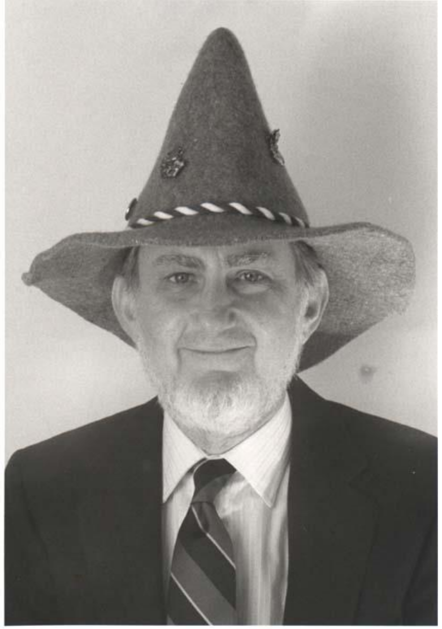
- [Hennie 64] F.C. Hennie “Fault detection experiments for sequential circuits”, Symposium on switching and automata theory, 1964.
- [Kime 66] C. R. Kime, “An organization for checking experiments on sequential circuits, IEEE Trans. Electron,” Comput. EC-U, 113-115, 1966.
- [McCluskey 81] E.J. McCluskey, S. Bozorgui-Nesbat, “Design for autonomous test”, IEEE Trans. on Ckt. and System, Volume: 28, Issue: 11, Nov 1981.
- [McCluskey 84] E.J. McCluskey, “Verification Testing – A Pseudo exhaustive Test Technique,” IEEE Trans. On Computers, C-33(6), pp541-546, 1984.
- [Moore 56] E. F. Moore, "Gedanken experiments on sequential machines," in Automata Studies. Princeton, 1956.
- [Friedman 71] A. D. Friedman, P. R. Menon, *Fault detection in digital circuits*, Prentice-Hall, 1971.





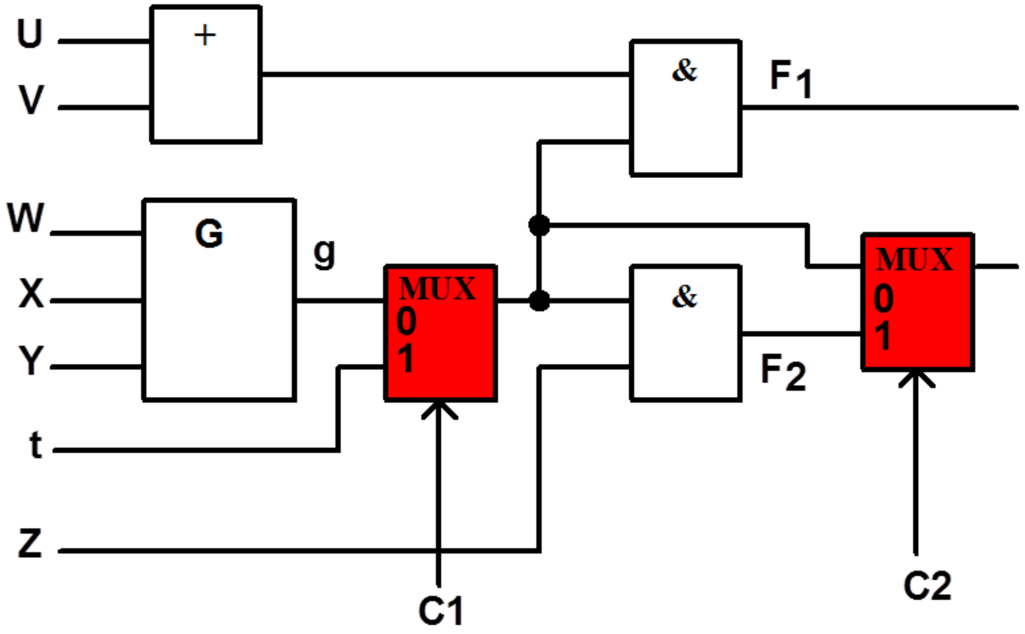


Viking



Exercise

- Show PET test for this circuit


U
V
+
&
F1
W
G
X
Y
g
MUX
0
1
&
MUX
0
1
F2
t
Z
C1
C2