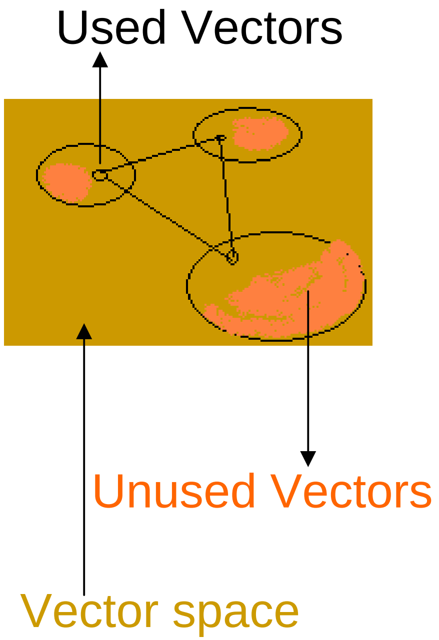
Vector Generation using Spectral Methods for better fault coverage

Ayoush M Dixit
dixitam@auburn.edu
November 16, 2004

Overview

- Inputs have to be meaningful for better coverage.
- While using random vectors it is impossible to visit the whole vector space.
- Spectral coefficients give us links from one vector space to another space which could have vectors within close range that will be able to give better fault coverage.
- Spectral analysis helps concentrate only on the vectors that are relevant thus reduces the *white noise*.
- Most importantly sequential circuit tests are not random so this method helps as this co-relates the vectors to one another.



Papers reviewed

- Ashish Giani, Shuo Sheng and Michael S. Hsiao, and Vishwani D. Agrawal, "Novel Spectral Methods for Built-In Self-Test in a System-on-a-Chip Environment," *Proc. IEEE VLSI Test Symposium*, April 2001, pp. 163-167.
- Ashish Giani, Shuo Sheng, Michael S. Hsiao, and Vishwani D. Agrawal, "Efficient Spectral Techniques for Sequential ATPG," *Design, Automation and Test in Europe, 2001. Conference and Exhibition 2001. Proceedings*, 13-16 March 2001, pp. 204 – 208.
- Xiaodeng Chen and Michael S. Hsiao, "Characteristic Faults and Spectral Information for Logic BIST," *Proc. IEEE/ACM ICCAD*, November 2002, pp. 294-298.
- Ganapathy Kasturirangan, and Michael S. Hsiao, "Spectrum-Based BIST in Complex SOCs," Proceedings of the 20th IEEE VLSI Test Symposium (VTS'02).
- A. K. Susskind, "Testing by Verifying Walsh Coefficients," *IEEE Trans. Computers*, vol. C-32, no. 2, Feb., 1983, pp. 198-201.
- Ten-Chuan Hsiao, and Sharad C. Seth, "The use of Rademacher-Walsh Spectrum in Testing and Design of Digital Circuits,"
- Bogdan J. Falkowski, "Parallelization of Methods to Calculate Walsh Spectra for Logic Functions," *J. of Multi-Valued Logic & Soft Computing.*, Vol. 10, pp. 91-127.
- Bogdan J. Falkowski, Ingo Schafer, and Marek A. Perkowski, "Effective Computer Methods for the Calculation of Rademacher-Walsh Spectrum for Completely and Incompletely Specified Boolean Functions," *IEEE Transactions on Computer-Aided Design*, vol. 11, issue no. 10, Oct. 1992, pp. 1207-1226.
- Prakash Ranganathan and Rajesh G. Kavasseri, "A Method for Flicker Severity Evaluation using the Hadamard Transform," *36th Annual Frontier International Power Conference, Oklahoma State University (OSU)*, Stillwater, 28th Oct '03.
- Benjamin Jacoby PhD, "Walsh Functions: A Digital Fourier Series."

Contd.. Hadamard - Walsh Transform for *N=4*

Explicitly written as,

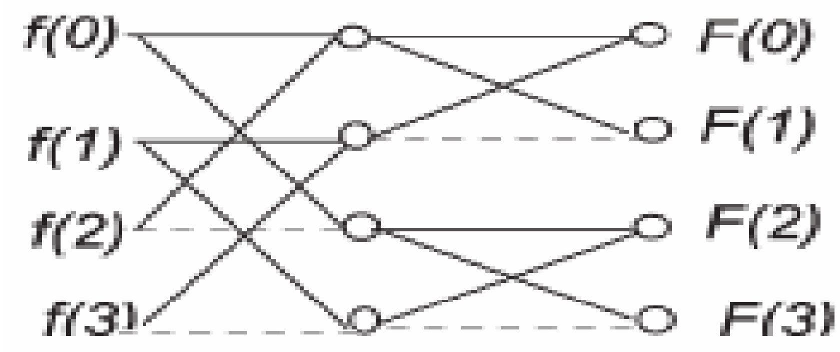
$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \end{bmatrix} = \begin{bmatrix} a_{H2} & a_{H2} \\ a_{H2} & -a_{H2} \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{bmatrix} \qquad a_{H2} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Matrix equality of previous matrix

$$\begin{bmatrix} F(0) \\ F(1) \end{bmatrix} = a_{1/2} \begin{bmatrix} f(0) + f(2) \\ f(1) + f(3) \end{bmatrix} = a_{1/2} \begin{bmatrix} f_1(0) \\ f_1(1) \end{bmatrix} = \begin{bmatrix} f_1(0) + f_1(1) \\ f_1(1) - f_1(1) \end{bmatrix}$$

$$\begin{bmatrix} F(2) \\ F(3) \end{bmatrix} = a_{1/2} \begin{bmatrix} f(0) - f(2) \\ f(1) - f(3) \end{bmatrix} = a_{1/2} \begin{bmatrix} f_1(2) \\ f_1(3) \end{bmatrix} = \begin{bmatrix} f_1(2) + f_1(3) \\ f_1(2) - f_1(3) \end{bmatrix}$$

Data flow graph of previous matrix



Continuous line represents +1, dotted line represents -1

Therefore,

F(0) = f(0) + f(1) + f(2) + f(3)
F(1) = f(0) - f(1) + f(2) - f(3)
F(2) = f(0) + f(1) - f(2) - f(3)
F(3) = f(0) - f(1) - f(2) + f(3)

Similar to,

$$H(1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Results in H(2) = [H(1)   H(1); H(1)   -H(1)]

“Testing by Verifying Walsh Coefficients” Susskind ’83

F(x1, x2, x3) = x1*x2 + x1*x3+x2*x3

x1	x2	x3	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Fw	w0	w1	w2	w3	w12	w13	w23	w123
-1	+1	-1	-1	-1	+1	+1	+1	-1
-1	+1	-1	-1	+1	+1	-1	-1	+1
-1	+1	-1	+1	-1	-1	+1	-1	+1
+1	+1	-1	+1	+1	-1	-1	+1	-1
-1	+1	+1	-1	-1	-1	-1	+1	+1
+1	+1	+1	-1	+1	-1	+1	-1	-1
+1	+1	+1	+1	-1	+1	-1	-1	-1
+1	+1	+1	+1	+1	+1	+1	+1	+1

	w0 Fw	w1 Fw	w2 Fw	w3 Fw	w12 Fw	w13 Fw	w23 Fw	w123Fw
	-1	+1	+1	+1	-1	-1	-1	+1
	-1	+1	+1	-1	-1	+1	+1	-1
	-1	+1	-1	+1	+1	-1	+1	-1
	+1	-1	+1	+1	-1	-1	+1	-1
	-1	-1	+1	+1	+1	+1	-1	-1
	+1	+1	-1	+1	-1	+1	-1	-1
	+1	+1	+1	-1	+1	-1	-1	-1
	+1	+1	+1	+1	+1	+1	+1	+1
C	0	4	4	4	0	0	0	-4

C_{all} in the paper is sum of all the C. C_{all} for F(x1, x2, x3) = 8

Hadamard Transform (output spectral coefficient of F)

1	1	1	1	1	1	1	1	-1		0
1	-1	1	-1	1	-1	1	-1	-1		-4
1	1	-1	-1	1	1	-1	-1	-1		-4
1	-1	-1	1	1	-1	-1	1	1	=	0
1	1	1	1	-1	-1	-1	-1	-1		-4
1	-1	1	-1	-1	1	-1	1	1		0
1	1	-1	-1	-1	-1	1	1	1		0
1	-1	-1	1	-1	1	1	-1	1		4

"Effective Computer Methods for the Calculation of Rademacher-Walsh Spectrum for Completely and Incompletely Specified Boolean Functions" Falkowski '92

U=1 in this region for x1

x3x4 / x1x2	00	01	11	10
00	-	0	1	0
01	1	1	1	1
11	1	1	1	1
10	1	0	1	1

a = # of true min-terms of F in the area for which both the function and the standard trivial function u have logical value 1

b = # of false min-terms of F in the area for which function has a logical value 0 and trivial function u has 1

c = # of true min-terms of F in the area for which F=1 and U=0

d = # of false min-terms of F in the area for which F=U=0

e = # of don't care min-terms of F in the area for which U=1

f = # of don't care min-terms of F in the area for which U=0

Some definitions,

$S0 = 2^n - 2a0 - e0$

$S1 = 2(a^1+d^1)+e^1+f^1-2^n$

.

.

$Sn = 2(a_n+d_n)+e_n+f_n-2^n$

For U0:		For x1:		For $x3 \oplus x4$:	
a0 =12	d0=0	**a1 =7**	**d1=2**	**a23 =5**	**d23=0**
e0 =1	f0=0	**e1 =0**	**f1=1**	**e23 =0**	**f23=1**
S0=16-24-1=-9		**S1=18+1-16=3**		**S34 = 10+1-16=-5**	

Spectral coefficients for

*F(x1, x2, x3) = x1*x2 + x1*x3+x2*x3 using this algorithm*

x2x3 / x1	00	01	11	10
0	**0**	**0**	**1**	**0**
1	**0**	**1**	**1**	**1**

u0:
a=4 d=0
e=0 f=0
s0=8-8-0=**0**

x2:
a=3 d=3
e=0 f=0
s0=12+0-8=4

x1⊕ x2:
a=2 d=2
e=0 f=0
s0=8+0-8=**0**

x2⊕ x3:
a=2 d=2
e=0 f=0
s0=8+0-8=**0**

x1:
a=3 d=3
e=0 f=0
s1=12+0-8=**4**

x3:
a=3 d=3
e=0 f=0
s0=12+0-8=4

x1⊕ x3:
a=2 d=2
e=0 f=0
s0=8+0-8=**0**

x1⊕ x2⊕ x3:
a=1 d=1
e=0 f=0
s0=4+0-8=**-4**

C	**0**	**4**	**4**	**4**	**0**	**0**	**0**	**-4**

"Novel Spectral Methods for Built-In Self-Test in a System-on-a-Chip Environment" Gianni et al… '01

Circuit	Total Faults	Faults detected (highest coverage shown in boldface)			
		Weighted-random patterns		Spectral patterns	
		Ideal weights	*Rounded-off weights*	$H(4)$	$H(5)$
s382	399	329	116	**364**	**364**
s400	428	306	106	**384**	**384**
s526	555	95	94	451	**454**
s713	581	**476**	**476**	**476**	**476**
s1196	1242	1233	1228	**1237**	1236
s1238	1355	1276	1270	**1282**	1279
s1423	1515	1319	1167	**1414**	1413
s1488	1486	1442	1410	**1444**	**1444**
s1494	1506	1451	1418	**1453**	**1453**
s5378	4603	3127	3083	**3596**	3521
b01	135	**133**	**133**	**133**	**133**
b04	1346	**1168**	**1168**	**1168**	**1168**
b08	489	461	438	**463**	**463**
b11	1089	937	898	**1004**	**1004**
b12	3102	663	636	**1621**	1615
Total	19831	14416	13641	**16490**	16407

“Efficient Spectral Techniques for Sequential ATPG” Gianni et al…’01

Circuit	HITEC [14]			STRATEGATE [15]			PROPTEST [9]			Spectral-Based		
	Det	Vec	Time	Det	Vec	Time	Det	Vec	Time	Det	Vec	Time
s382	301	1463	90.0	364	1486	8.1	364	572	0.5	364	567	1.0
s400	341	1845	72.0	384	2424	15.5	382	677	0.7	382	588	1.5
s713	476	173	0.1	476	176	1.3	476	104	0.1	476	89	0.4
s1196	1239	435	0.1	1239	574	1.5	1239	224	0.4	1239	244	1.2
s1238	1283	475	0.1	1282	624	1.5	1283	235	0.4	1283	255	1.1
s1423	723	150	834.0	1414	3943	76.2	1416	1049	8.8	1416	927	16.3
s1488	1444	1170	16.5	1444	593	7.5	1444	426	1.9	1444	384	3.2
s1494	1453	1245	9.6	1453	540	7.6	1453	454	2.2	1453	388	2.9
s5378	3231	912	1104.0	3639	11571	2268.0	3643	672	35.6	3643	734	43.5
b01	-	-	-	133	86	0.1	-	-	-	133	50	0.1
b04	-	-	-	1168	2680	10.3	-	-	-	1168	158	0.8
b08	-	-	-	463	1567	0.7	-	-	-	463	387	1.5
b11	-	-	-	1003	4883	50.0	1004	419	1.6	1004	962	2.5
b12	-	-	-	1488	33113	9659.0	1470	3697	27.5	1645	4464	24.2

Time reported in minutes Different platforms were used for different test generators

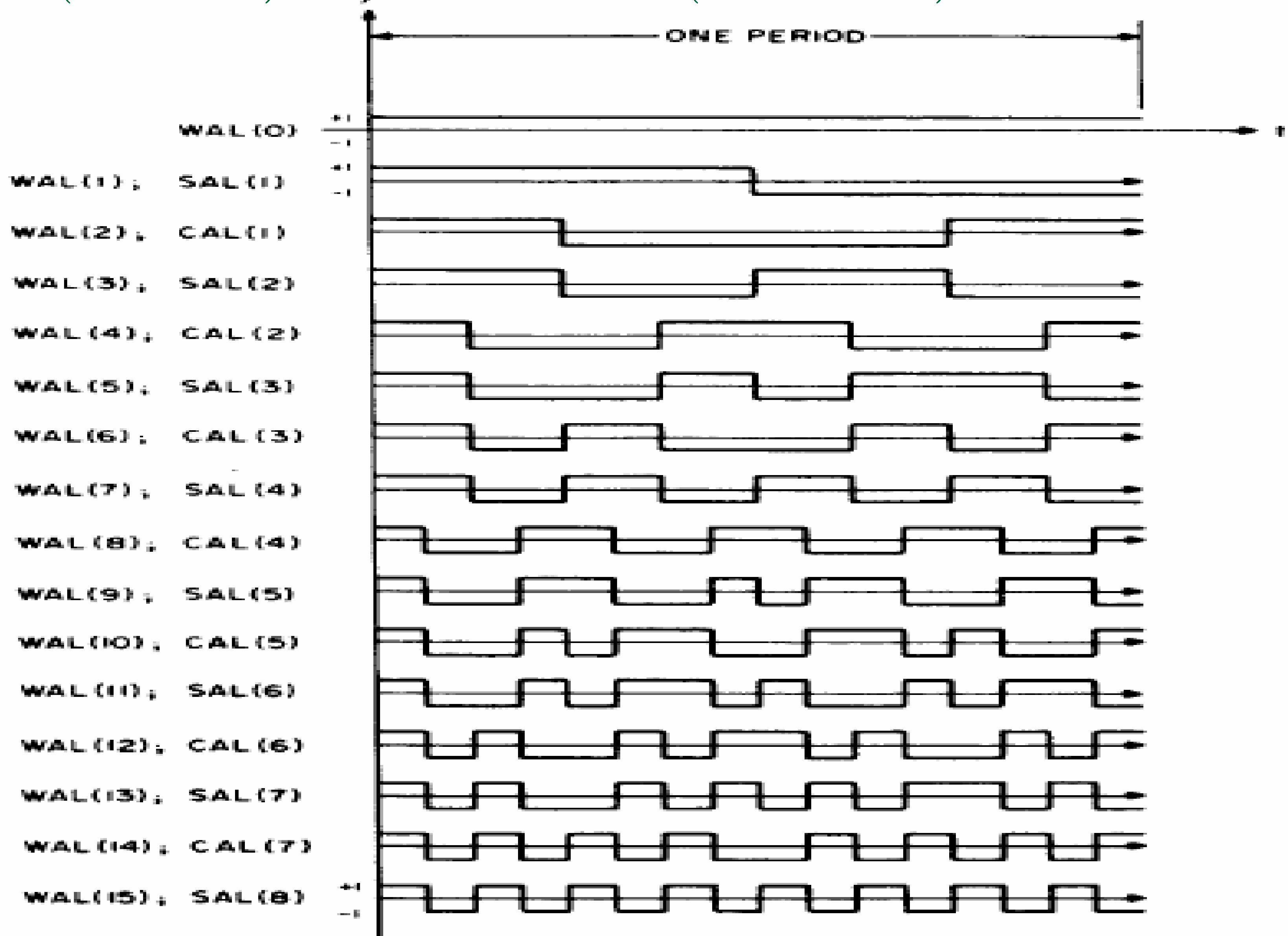
"Characteristic Faults and Spectral Information for Logic BIST" Chen and Hsiao, 02

Ckt	Total Faults	# Faults Detected by STRATEGATE [21]	# Faults Detected by Non-Scan BIST			
			Weighted-Random Patterns		Spectral Patterns	
			Ideal Weights	Rounded-off Weights	[16]	Ours
s382	399	**364**	329	116	**364**	357
s400	428	**384**	306	106	**384**	376
s526	555	**454**	95	94	**454**	442
s713	581	**476**	476	476	**476**	**476**
s1196	1242	**1239**	1233	1228	*1237*	1227
s1238	1355	**1282**	1276	1270	**1282**	1266
s1423	1515	**1414**	1319	1167	**1414**	**1414**
s1488	1486	**1444**	1442	1410	**1444**	**1444**
s1494	1506	**1453**	1451	1418	**1453**	**1453**
s5378	4603	**3639**	3127	3083	3596	*3611*
b01	135	**133**	**133**	**133**	**133**	**133**
b04	1346	**1168**	**1168**	**1168**	**1168**	**1168**
b08	489	**463**	461	438	**463**	**463**
b11	1089	1003	937	898	**1004**	**1004**
b12	3102	1488	663	636	1621	**1648**

Only **one** characteristic fault is used under "Ours" column

Walsh Functions

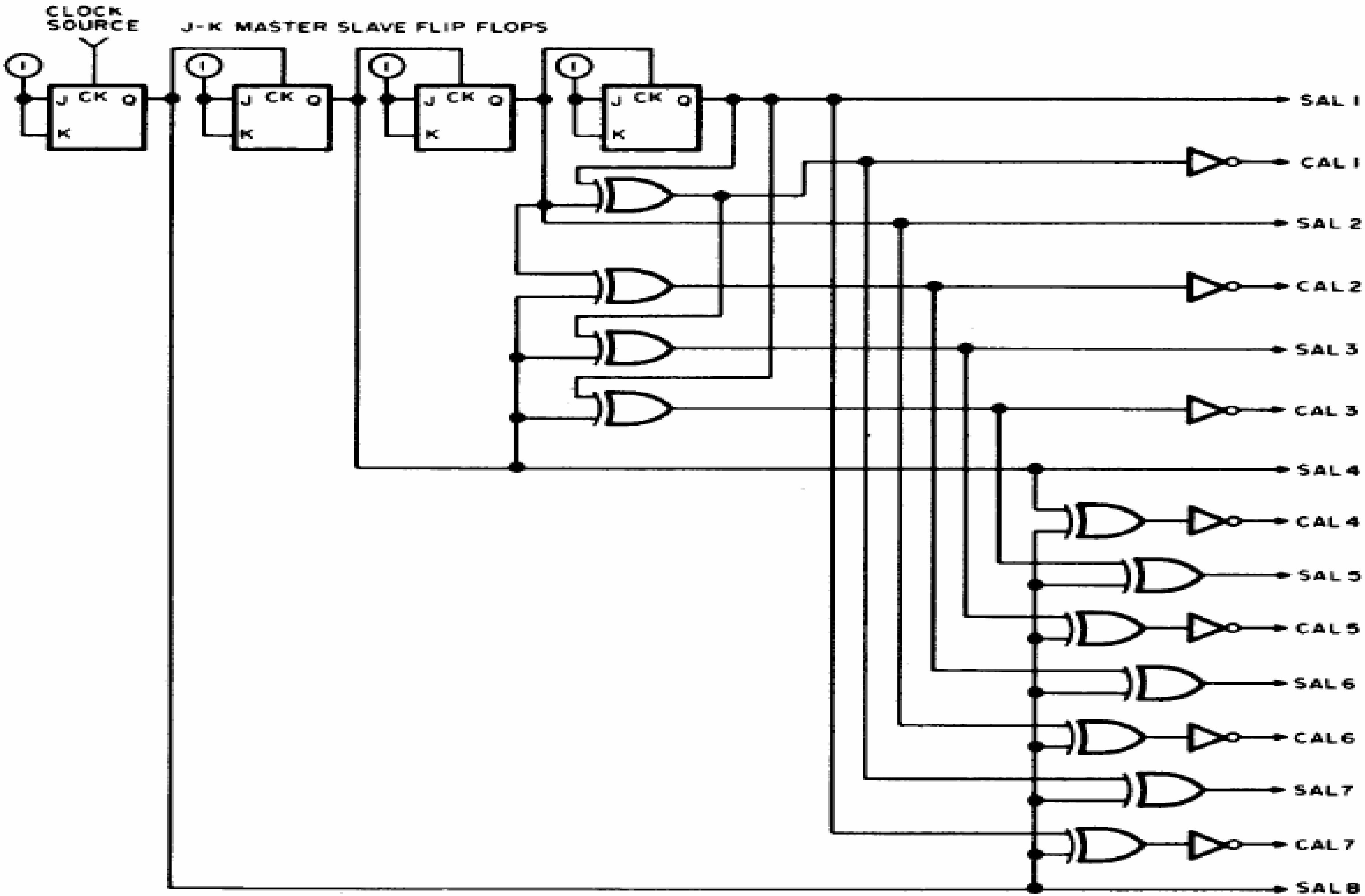
SAL (Sine wALsh) ⇔Fourier Sine & CAL (Cosine wALsh) ⇔Fourier Cosine



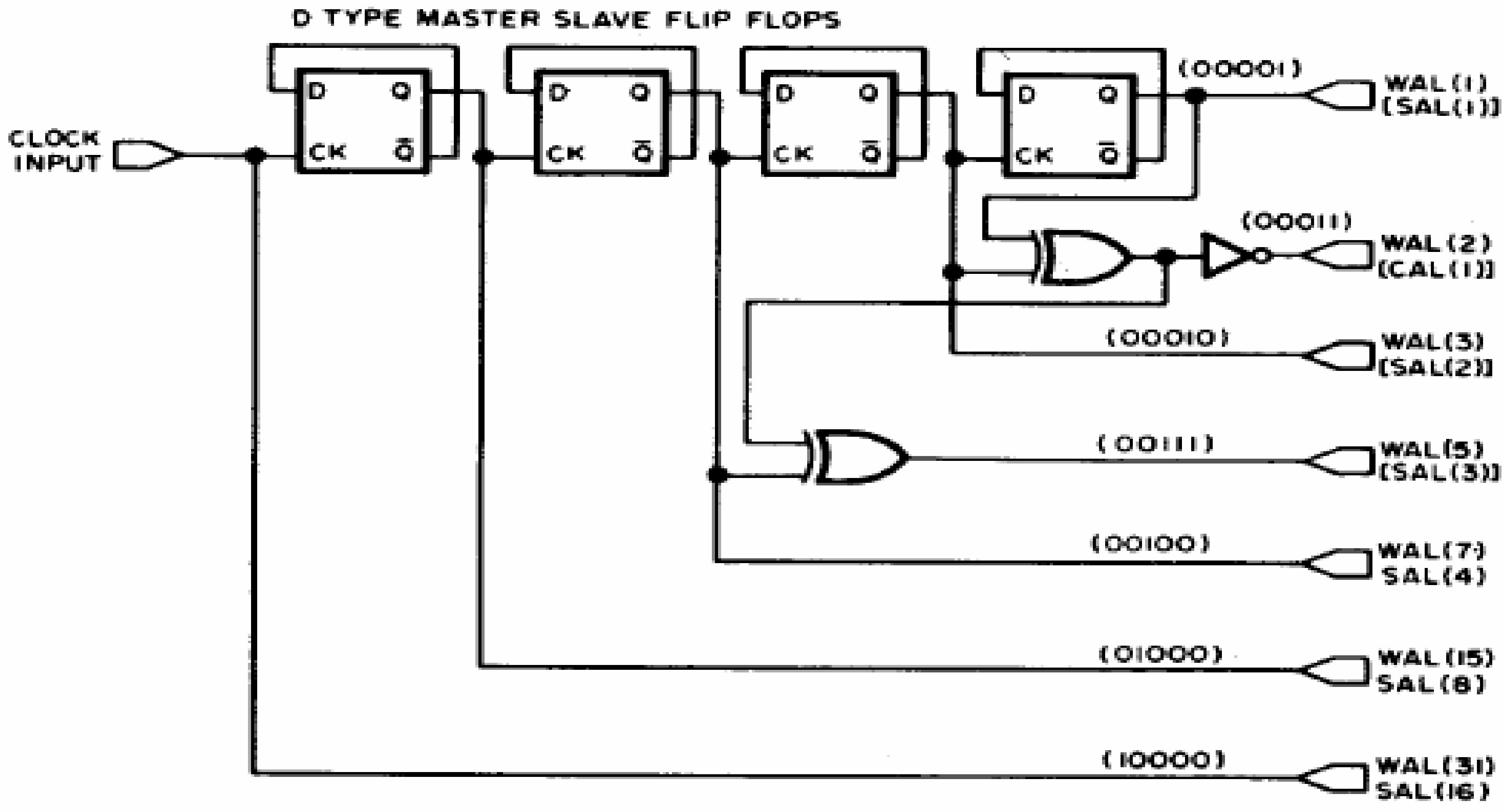
Why use Walsh Functions?

- Spectral methods based on Walsh transform provide very efficient representation of logic functions that uncovers the properties unseen in their original domain.
- The exploitation of the spectral data provides a sound mathematical basis for logic function synthesis.
- Drawback
  - The large complexity associated with the calculation of the spectrum of a logic function.
- Different sequential algorithms have been developed for calculation of the Walsh spectral coefficients.
- The binary nature of the Walsh transform makes if advantageous over Fourier Transform because of the computational simplicity.
- The transform matrices are made up of only ±1’s and the required computations in any digital circuit are only addition and subtraction.

Walsh function generating circuit



Selected Walsh function generating circuit



Algorithm that will be used for analysis (for a vector sequence)

```
T = generate random vectors
i = 0; done = 0; Smax=0;
while (not done) and (i<max_iteration)
  fault simulate Ti;
  Ki = k last-detected faults;
  Ci = Ti filtered for Ki;
  perform Hadamard Transform on Ci to obtain spectral coefficients Hi;
  Ei = new vectors generated using coeff. Hi;
  Si = faults detected by Ei;
  if Si > Smax
      characteristic fault set = Ki;
      Smax = Si;
  i++;
  if no improvements for n iterations
      done = 1;
```

S_i set of faults that have similar spectral characteristics as K

This algorithm is a modification of Chen and Hsiao's algorithm presented in "Characteristic Faults and Spectral Information for Logic BIST"

Algorithm that will be used for analysis (output spectral characteristics w.r.t inputs)

- Perform logic simulation and get the outputs
  Generate random vectors
  Perform fault simulation
  Perform vector compaction
  Calculate spectral characteristics (Susskind's method)
  i = 0; *done* = 0; max_iteration = n;
  while (not *done*) and (*i* < max_iteration)
  for required coverage not reached or maximum iterations not reached
  generate random vectors
  use the coefficients to get one vector
  fault simulation
  if new faults detected
  { collect the vector;
  generate new fault list;
  }
  i++;
  if no improvements for n iterations
  done = 1;

New algorithm proposed using ideas from the papers reviewed.

max_iteration can be any number depending on the complexity and resources available for simulation purposes.

Conclusion

- A newer algorithm is proposed to generate vectors that will most probably give better fault coverage.
- Knowing the spectral coefficients of the output of a system with respect to the inputs helps preserve the characteristics of the system. It is definitely something that should be looked into.
- After seeing some of the results presented by Gianni and colleagues prospect for spectral BIST looks good for sequential circuits.
- New vector generation time is greatly reduced as number of calculation reduces from $n \times n$ to $n \times Log_2 n$
- Different other orthonormal functions can also be looked in depth to get better spectral coefficients which will give better fault coverage.
- Characteristic faults can be targeted and looked into more closely for better fault coverage using spectral analysis.

Questions and Comments

Happy BISTING!!