International
Journal
Of Advanced Research in Engineering & Management (IJAREM)
ISSN: 2456-2033 || PP. 05-09

IMPLEMENTATIONOF TEST ARCHITECTURE FOR 64-BIT WALSH SEQUENCE GENERATOR

Ashwini G V,
PG Student, UTL Technologies Ltd.,

Nayana M,
Lecturer, UTL Technologies Ltd.,

Dr. Siva Yellampalli,
Principal, UTL Technologies Ltd.,

Abstract: Walsh codes are error correcting orthogonal codes with good cross correlation property. Because of this property they are used for error free communication. The Channel Identification in CDMA is one of the applications of Walsh sequence. This paper presents ATPG Test Architecture for 64-bit Walsh Sequence Generator. 10 scan chains are created using existing flops. For the experiment 100% fault coverage is achieved for static faults.
Keywords:CDMA, ATPG, Walsh sequences, SDR

I. Introduction

The Walsh code is a linear code which maps binary strings of length *n* to binary code words of length *2n*. Walsh codes are mutually orthogonal error correcting codes. Walsh code have many interesting mathematical properties and vital applications in communication systems. WALSH functions are a complete set of periodic two valued {+1,-1} orthogonal functions, which is closed in a standard interval (0, 1) and every function takes the values {+1,-1} except the final number of discrete points, which is zero. Walsh code have vast applications in the field of communications, fast Fourier transforms, audio and video signal processing, filtering and multiplexing have been widely reported specially for multi-mode radio SDR (Software Defined Radio) and Reconfigurable Radio as in CDMA standard [1]. However, they have aroused great interest in recent years in wireless communication as they are used as channelization code in many standards such as CDMA2000 [1].

Walsh code stands for the elimination or the reduction of interference within the users and within the channels and furthermore for their identification in literature [2] and [3] where the individual channels are distinguished from each other by mutual orthogonality of signals. It is therefore, necessary to create mutually orthogonal sequences, through which information is transmitted by various channels spread [4].

II. DESIGN OF WALSH GENERATOR

There are two different kinds of Walsh function generators are in use [4]. The first one generates only one Walsh function at a time out of a large possible number. And the second method generates a complete set of Walsh functions simultaneously. The Walsh generator used in this paper generates only one Walsh function at a time. The Walsh Generator provides total of 64 Walsh codes, all of which are orthogonal. It can be obtained by the equation shown 2.1, here; "i" represents the index which ranges from 0 to 63. Fig 1 shows the flowchart for Walsh generation.

$Walsh\,[i]$=[(α0XORα1)XOR (α2XORα3)XOR (α4 XOR α5)]……2.1

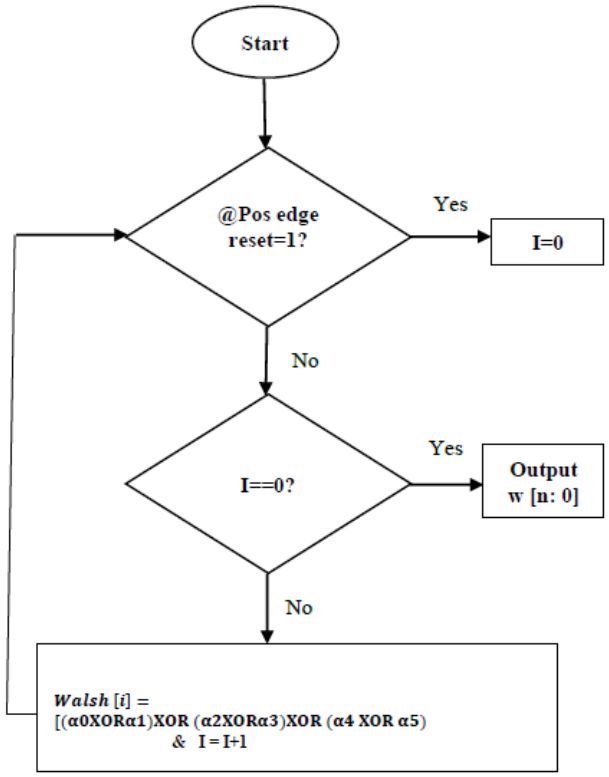

Fig 1: Flow Chart for Walsh Generation

Steps:

1. Initialize the Walsh index i=0;
2. In each cycle index will be incremented by 1.
3. Walsh output bit will be calculated and that will be stored for associated index.
4. Once index reaches the value n, Walsh generator produce the n bit Walsh Sequence.
5. Once again I value reaches zero and all steps will be repeated.

III. TEST ARCHITECTURE

There are different Design for Test (DFT) methods such as JTAG, ATPG and BIST [8].

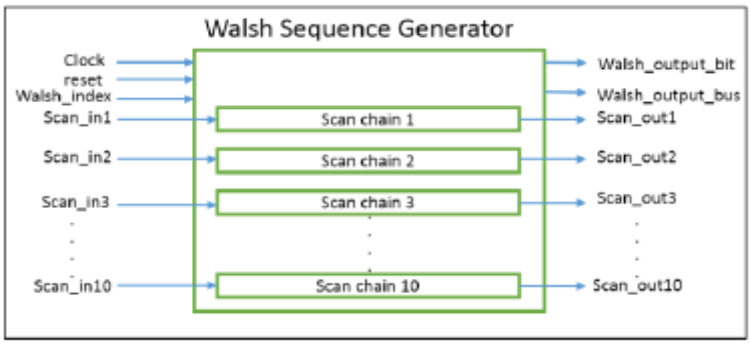

Fig 2: Test architecture for Walsh sequence generator

Fig 2 shows the Test architecture for Walsh sequence generator. The ATPG technique is used for the test architecture of Walsh sequence generator.

A. ATPG flow

The below are the steps followed in ATPG.

1. Build Model and Test Mode

The library used for synthesis is TSMC 180 nm and there are no black box created and got the full coverage. This build model is used to identify the static faults in test mode. The full scan test mode has been generated and this is industry compatible. The elapsed time is 3.46 ns.

2. Build Fault Model

The tested mode is full scan mode and the elapsed time is 0.08 ns. Build fault model takes input from previously created model and applies a fault model to the appropriate nodes by placing faults in the circuit. It allows in applying static fault model to the circuit.

3. Verify Test Structure

Verify test structure method traces scan chain from the scan_out pin by applying inputs to scan_in to find-out
Observable and controllable scan chain in the design. The test mode has been selected as full scan mode for scan chain identifications which gives information about broken chain for debugging and to find possible test coverage.

B. ATPG tests

The ATPG engine has been used for generating test patterns to test the structural integrity of the design. There are two methods of test coverage have been carried out:

1. Scan Test:

In order to identify the manufacturing defects, scan test generated patterns are used to verify the design by shifting the data from scan_in (SI) to scan_out (SO) pins. The scan chain patterns are useful to find out fault coverage and global coverage of test sequences which was created in full scan mode. The fault coverage for one sequence is 36%.

2. Logic Tests:

The logic test model is used to create pattern to cover struck-at model. The ATPG engine is used for generating test patterns that will test the structural integrity of the design in terms of resultant fault coverage in full scan test mode, global coverage and generated test sequences. The fault coverage is 100%.

IV. EXPERIMENTAL SETUP AND ANALYSIS

The 64 bit Walsh generator is implemented and verified its simulation results using cadence NC Sim. The simulation result is shown in the figure 3. To build the test architecture 10 scan chains are created with each of length 14. Cadence RC is used for synthesis. The TSMC 180 nm technological

Library is used for synthesis. Target faults are 3300 and the test sequence is 143. The time taken for verification is 4.5 ns. Cadence ET tool is used for ATPG analysis. The power and area reports are shown in figure 4 and 5 respectively. The figure 6 shows fault coverage obtained for the static fault.

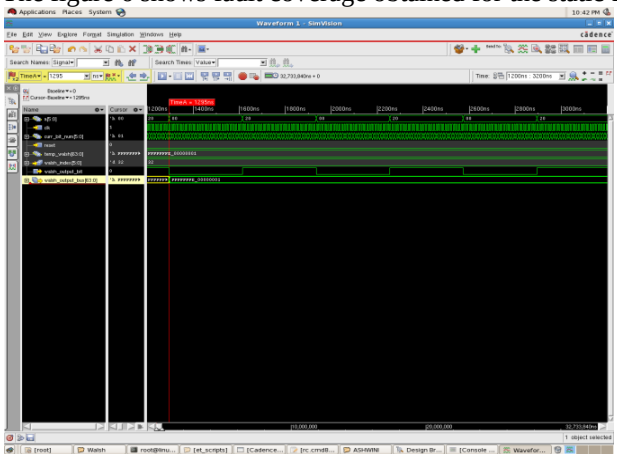

Fig 3: Simulation results of Walsh generator

International
Journal
Of Advanced Research in Engineering & Management (IJAREM)
ISSN: 2456-2033 || PP. 05-09

```
============================================================  Generated by:           Encounter(R) RTL Compiler v14.20-s038_1
Generated on:           May 31 2017  05:45:19 pm  Module:                demod  Technology library:     tsmc18 1.0  Operating
conditions:   fast (balanced_tree)  Wireload mode:          enclosed  Area mode:              timing library
============================================================                                          Leakage     Dynamic       Total
          Instance           Cells Power(nW)   Power(nW)     Power(nW)
-------------------------------------------------------------------------------demod                          59299 52198.013 399476817.436
399529015.449   corr_a                  27640 23611.472 199734890.167 199758501.640     csa_tree_a..._1_I64_groupi  1560  4167.872
 40319098.965  40323266.837     csa_tree_s..._1_I64_groupi  3044  4038.969  39931487.670  39935526.639     csa_tree_a..._1_I64_groupi
2183  3530.058  47733462.500  47736992.558     csa_tree_s..._1_I64_groupi  2163  3190.472  47837110.213  47840300.685   corr_b
           27640 23611.472 182686960.077 182710571.549     csa_tree_a..._1_I64_groupi  1560  4167.872  36579129.186  36583297.058
 csa_tree_s..._1_I64_groupi  3044  4038.969  36064745.516  36068784.484     csa_tree_a..._1_I64_groupi  2183  3530.058  45222011.579
45225541.637     csa_tree_s..._1_I64_groupi  2163  3190.472  44353912.734  44357103.206   pseudonoisegen               898   521.481
 1554834.030   1555355.511   walsh32                          307   245.215   1084750.396   1084995.611   csa_tree_add_90_33_groupi
312   192.746     38124.047     38316.793   csa_tree_add_91_33_groupi    312   192.746     38123.270     38316.017
```

Fig 4: Static power of test architecture

From the above Fig 4, the total power obtained is 1084750.396 nw

```
============================================================  Generated by:           Encounter(R) RTL Compiler v14.20-s038_1
Generated on:           May 31 2017  05:46:32 pm  Module:                demod  Technology library:     tsmc18 1.0  Operating
conditions:   fast (balanced_tree)  Wireload mode:          enclosed  Area mode:              timing library
============================================================            Instance             Cells  Cell Area  Net Area  Total Area
Wireload     ------------------------------------------------------------------------------------------demod
59299    1653191         0     1653191    <none> (D)   corr_b                          27640     731575         0      731575
<none> (D)     csa_tree_sub_45_1_I64_groupi   3044     110892         0      110892    <none> (D)     csa_tree_add_43_1_I64_groupi
1560     101046         0      101046    <none> (D)     csa_tree_add_49_1_I64_groupi   2183      84171         0       84171
<none> (D)     csa_tree_sub_51_1_I64_groupi   2163      83612         0       83612    <none> (D)   corr_a
27640     731575         0      731575    <none> (D)     csa_tree_sub_45_1_I64_groupi   3044     110892         0      110892
<none> (D)     csa_tree_add_43_1_I64_groupi   1560     101046         0      101046    <none> (D)     csa_tree_add_49_1_I64_groupi
2183      84171         0       84171    <none> (D)     csa_tree_sub_51_1_I64_groupi   2163      83612         0       83612
<none> (D)   pseudonoisegen                    898     23032         0       23032    <none> (D)   walsh32
307     10322         0       10322    <none> (D)   csa_tree_add_91_33_groupi         312      7840         0        7840    <none>
(D)   csa_tree_add_90_33_groupi        312      7840         0        7840    <none> (D)  (D) = wireload is default in technology
library
```

Fig 5: Area of test architecture

From the above Fig 5, the area obtained is 10322

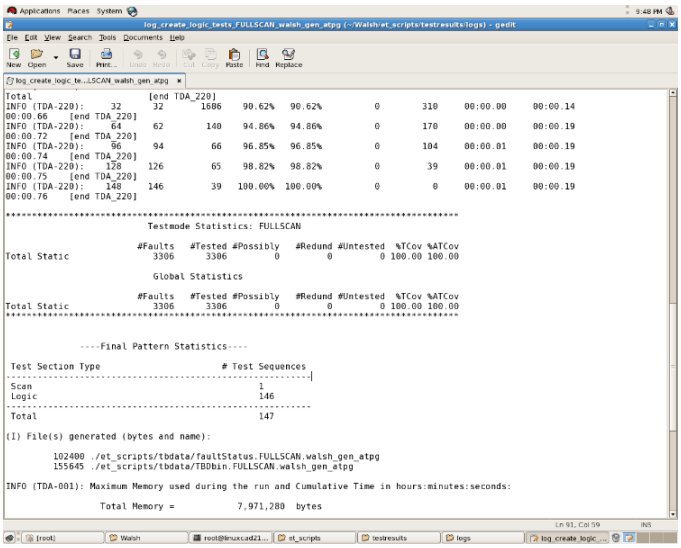

Fig 6: Fault coverage of test architecture

From the above Fig 6, the static fault coverage is 100%

V. Conclusion

The 64-bit Walsh sequence generator is implemented and is verified. Scan chains are created to increase the controllability and observability of the test architecture. ATPG technique is adopted to implement architecture. For the experiment 100% fault coverage is achieved for static fault.

References

[1]. Lee, J. S., Miller L. E.,CDMA System Engineering Handbook, London: Artech House, 1998, pp.68–73..
[2]. Jos, S., Nair, J.P. , Sen, D. , Naniyat, A. "Method of Generating Multiple Sets of Orthogonal Codes with Wide Choice of Spreading Factors" Wireless Communications Letters, IEEE, Jul. 9 2012, pp. 492- 495.
[3]. Physical Layer Standard for cdma2000 Spread Spectrum Systems. Available: [online] http://www.3gpp2.org/public_html/specs/c.s0002-d_v1.0_021704.pdf, Mar.-2013..
[4]. Gaurav Purohit., V.K Chaubey., Kota Solomon Divya Vyas, V.K Chaubey "FPGA Implementation & Power Analysis of Parameterized Walsh sequences" Proceeding of the 2014 IEEE Students' Technology Symposium.
[5]. Adam Powell , Christos Savvas-Bouganis, Peter,Y.K. Cheung "High-level power and performanceestimation of FPGA-based soft processors and itsapplication to design space exploration" ,Journalof Systems Architecture ,Vol. 59 , Issue 10 , pp. 1144–1156,2013.
[6]. H. Harmuth, Transmission of Information by Orthogonal Functions.2nd New York: Springer, 1972.
[7]. Michael John Sebastian Smith, "Application Specific Integrated Circutis", Pearsion Education Inc, 12th impression, 2013
[8]. Nigel Horspool, Peter Gorman, ASIC Handbook, PrinticeHall , 2011
[9]. Cadence NCLaunch User Guide, Product Version 14.1, June 2014