FPGA Implementation of Low Power Testing Using Razor Based Processor

R. Karthick[1] and M. Sundararajan[2]

[1] *Research Scholar, Bharath University, E-mail: karthickkiwi@gmail.com*
[2] *Dean R&D Department, Bharath University, E-mail: msrajan69@gmail.com*

Abstract: A low power broad test set is developed from a functional broadside sets, the derivation of skewed test load cubes in Built in Self Test circuits. The indefinite value performed in tests are considered as to as sustained with the functional criteria. The twin effects of programmable truncated multiplications are implemented in a Digital Signal Processor having a fault tolerant capability. Thus the supply voltage is decreased in beyond the critical timing level. Timing modulation characteristics of truncated multiplications are studied. The performance of the designs is improved to attain the effective fault tolerant designs in order to reduce the error correction problems and to increase the scale operating voltage. The low power test schemes using Razor techniques are implemented in FPGA. The only drawbacks are the degradation of the output signal-to-noise ratio.

Index: Digital Signal processing, BIST, Razor technique.

I. INTRODUCTION

In VLSI circuits, the voltage scaling is implemented to decrease the dynamic power consumption and to achieve the static power management. The progress in their CMOS technology exploits in scaling to avoid issues from Process Voltage Temperature (PVT) deviations. DSP functions are preserved in to employing the Voltage over Scaling (VOS) levels in digital signal processing. The main features are offered in contrasts to timing constraints to presentation of appraisal of subsystem that delivers estimation, when fault is identified in skills that vary the data captures by supplementing the latches or flip flops. PVT variations, circuit design and data input regulate the power conservation obtained through fault tolerant techniques. Numerous timing distributions can be obtained by truncated multipliers which can offer power and area consumption. Functional broadside tests can be used to overcome the power consumption issues in the scan-based test. Hence the switching activity is minimized [1].

The power dissipation of a system in test mode is more than in normal mode. Low correlation between consecutive tests happens when applying low correlated patterns to scan chains. Increasing a switching activity in scan chain results its increased power consumption in scan chain and combinational block. This extra power consumption (average or peak) can create problems such as instantaneous power surge that cause circuit damage, formation of hot spots, difficulty in performance verification, and reduction of the product yield and lifetime. Different types of techniques are presented in the literature to control the power consumption. These mainly

includes algorithms for test scheduling with minimum power, techniques to reduce average and peak power, techniques for reducing power during scan testing and BIST(built-in-self- test)technique [2]. The skewed-load test is maintained so that the switching activity during their fast functional clock cycles is confined by the maximum switching activity of a functional broadside test. Initially, the transformation of skewed-load test is achieved by complementing some input values. The frequency of fault detection is increased without associating the switching activity with that of the functional broadside test.

The resultant target faults are grouped as CBRD. The CBRD forms a low-power broadside test set. The skewed-load test cubes derived from functional broadside tests are brought under skewed-load test. A mixed low-power test is the combination of broadside and skewed-load tests. The major task is to extract the skewed-load test cubes. Leakage power is decreased by 18.7 % in low-power vs. high-speed mode (on average); dynamic power is reduced by 99.4 %. Leakage power in sleep mode is 18.1% lower than that in high-speed mode [3].

With the purpose of attaining low power consumption, a new configuration known as I-PTMAC is proposed by combining this low power barrel shifter with RPTMAC. In this configuration, the Leakage power is reduced by 18.7 % in low-power vs. high-speed mode (on average); dynamic power is reduced by 99.5 %. Leakage power in sleep mode is 18.2 % lower than that in high-speed mode. Then I-RPTMAC with reconfigurable signed pipelined array multiplier is configured. The signed pipelined array multiplier is employed in the place of conventional data memory in PTMAC. Finally, Sleep Mode Approach is used with PTMAC architectures. For LTPTMAC, Leakage power is reduced by 23.2 %in low-power Vs. high-speed mode (on average); dynamic power is reduced by 99.6 % and Leakage power in sleep mode is 22.8% lower than that in high-speed mode [4].

II. RELATED WORKS

In this section, overview of related works, it focuses on the literature of profile based personalization and privacy protection in this PWS system.

(a) Built in self-test (BIST) circuit

A built-in self-test (BIST) or built-in test (BIT) is an automated self-test on a machine. The advantage is independence (pattern-programmed) and cost reduction. Affordability is increased by decreasing test-cycle duration, simple test/ probe setup, and by minimizing I/O signals that are essentially being investigated under tester control [5].

Design for Test ("Design for Testability" or "DFT") offers some testability features to a hardware product design. The manufacturing tests for the hardware design ensure that the product hardware does not contain manufacturing defects. Tests can be performed at any stage of the hardware manufacturing flow and for the maintenance. The test is either conducted by Automatic Test Equipment (ATE) or within the assembled system. The analytical information is transferred to detect the reason for a failure.

DUT (device under test) is associated with the standard circuit and the response of vectors (patterns) is observed. The equality denotes the perfection of the device. Fig. 1 shows General BIST circuit [6].
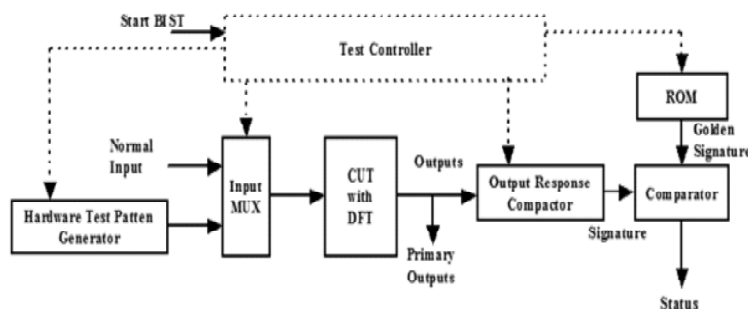

Figure 1: General BIST circuit

DUT is an effective interface for test application and diagnostics. The effective DUT rules make Automatic Test Pattern Generation (ATPG) simple. Several advanced BIST techniques have been studied and applied. The first class is the LFSR tuning. Girard *et al.* analyzed the Impact of an LFSR's polynomial and seed selection on the CUT's switching activity, and proposed a method to select the LFSR seed for energy reduction.

The second class is low-power TPGs. One approach is to design low-transition TPGs. Wang and Gupta used two LFSRs of different speeds to control those inputs that have elevated transition densities [7].

(b) Structural testing, Scan Test

The structural test is an inbuilt design in a VLSI circuit. The logic primitives (gates, flip-flops) are accessed within the circuit. Another common method is scan test. The whole registers of the circuit are combined with a shift register chain (scan-chain) in a special test mode. Fig. 2 shows that the separated combinational sub circuit contains I/O stage connected with the scan chain. In the scan chain, the scan controllable register and the scan observable register form the pseudo primary inputs and forms the pseudo primary outputs.

The testing is affected by the changes in connections of register cells. If various short register chains are formed, the test duration can be saved. However, test pattern generation and response study is complex. If a partial scan method is chosen rather than full scan method, its effect on the design, cost and the critical path is avoided. The shift register chain does not include all register cells. Also the remaining logic consists sequential blocks. If the number of vectors of the partial scan is high, more clock cycles of the target fault. Since non-scan latches consumes less space than scan latches, partial scan consumes less silicon area. In addition, the less routing resources required by the partial scan provide shorter scan path. The non-scan elements should be carried forward in sequential operation to keep the circuit under control [8]. Current CAD-tools performs the default transfer of an existing design into a scan design fit for test.
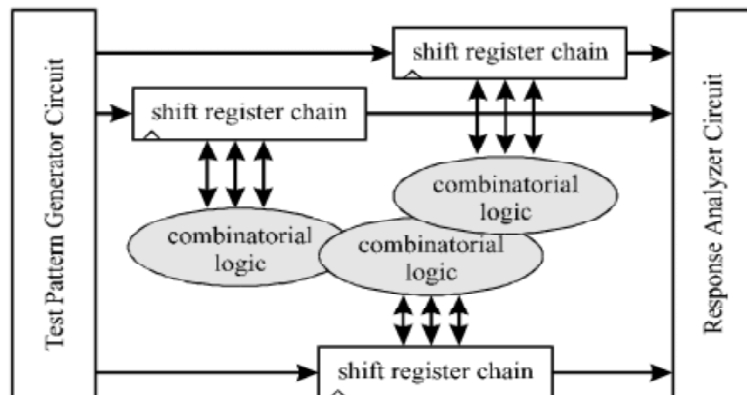

Figure 2: Principle of the scan test

(c) Basic test procedure

The architecture of the test setup is depicted in fig. 3. The initial step is the adaptation of the circuit into a test pattern to perform the required operation. Secondly, the accessing of the test pattern is done and finally the circuit's response is examined. The test procedure is repeated with different test patterns.

(d) Test pattern generation

A new weighted random pattern design for testability is described where the shift register latches distributed throughout the chip are modified so that they can generate biased pseudo-random patterns upon demand. A two-
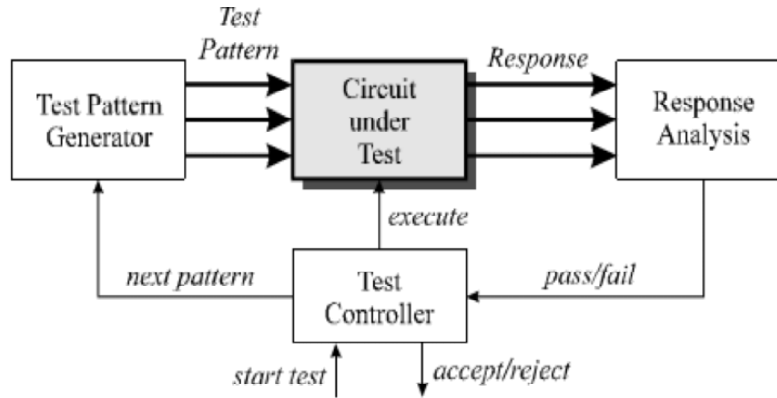
Figure 3: Architecture of a typical test setup

bit code is transmitted to each weighted random pattern shift register latches to determine its specific weight. The weighted random pattern test is then divided into groups, where each group is activated with a different set of weights. The weights are dynamically adjusted during the course of the test to "go after" the remaining untested faults [9].

The main challenge is to make the optimal sequence of test patterns to meet the several criteria such as, detection of every defect assumed in the fault model, comfort of generation/storage (low overhead), compactness (short test duration).

(d) Combinational vs. sequential logic

The task of a combinational circuit, a circuit without storage features, is completely described by its truth table. A combinational fault is a fault in a time-invariant logic truth table. It can be sensed with a single test stimulus. An improved technique reduces the number of patterns further more. An algorithm called as D-algorithm is imported to track all detectable faults. The runtime of the in-depth algorithm may be large in complex applications. Updated techniques are there to reduce time required for generating test and size of the test set. In case of sequential logic circuits, the test becomes hectic and requires more test patterns in different states [10].
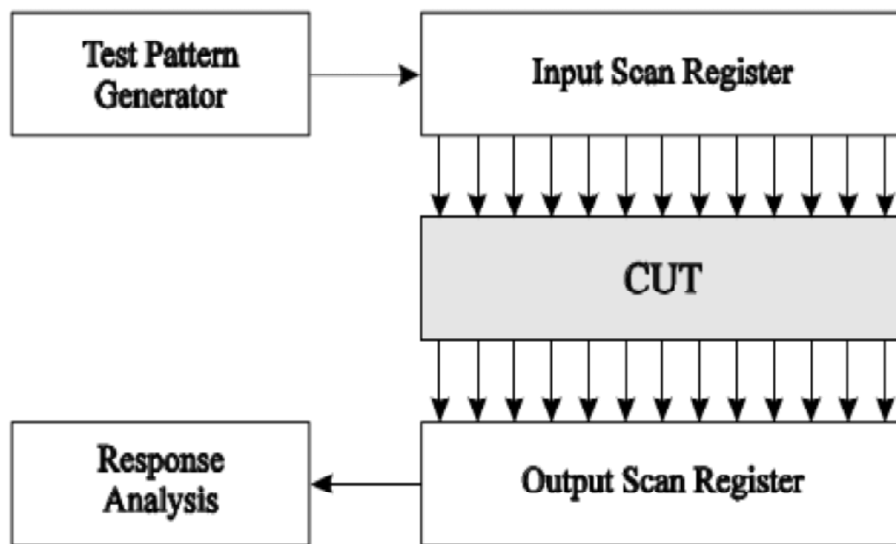

Figure 4: Basic test-per-scan architecture

III. TEST PATTERN GENERATOR

In ROM, the sequence of the pattern is released here to produce outstanding fault coverage along with the conventional deterministic algorithm. The sequential length increases with ROM size, but leads to restricted area overhead for built-in test. In the processors where test pattern calculation is performed, a test called program is activated to measure a suitable sequence of test patterns. Counter (exhaustive test) is the modest kind of test pattern generation that is employed in exhaustive and pseudo-exhaustive testing. The stored pattern test method comes under offline test pattern generation, whereas processor, counter and pseudo-random generator are classified as concurrent test pattern generation methods.

(i) Test pattern application

The function of clock in a scan structure is scanning in and out of the test pattern and its application. One clock cycle is enough for a full scan of the test pattern application, whereas in partial scan, the sequential logic left in the CUT is clocked frequently.

(ii) Test-per-scan

The generated test pattern is fed to the CUT input in a synchronous circuit. The test pattern is shifted to the scan chain and clocking is triggered. Then the CUT output pattern is latched and terminated to the scan chain for further processing. Parallel scanning of previous and next test pattern demanded clock cycles for an n-stage scan path, which extends the duration of the test pattern processing by the CUT.

(iii) Razor and Fault Tolerance

The Razor technique is an approach of dynamic voltage scaling with dynamic detection and rectification of circuit timing errors. With the support of error rate detection, the supply voltage is regulated to achieve the need of conservative timing analysis. Razor is practiced in a high-speed real-time finite-impulse response (FIR) filter in a system with reduced area and timing overheads. The efficiency of Razor and the V_{dd} scaling limits are governed by the circuit timing distribution. The reduced timing strategy will improve the performance of the Razor as well as the truncated multiplier [12].

(a) *Truncated Multiplication*

If a system does not require the implementation of sections of the least significant part of the partial product matrix, then the truncated multiplier promises to accomplish the necessary outputs. Generally product values generated by fixed width N × N bit multipliers are truncated or rounded back to the original bit width in the latter stages of the algorithm flow. Truncation uses by a smaller compensation circuit to replace the lower parts of the partial product matrix. Programmable and configurable methods for truncated multiplication utilize fixed-width structures that can function with reduced resolutions by deactivating parts of the partial product generation.

IV. RAZOR IMPLEMENTATION

The programmable truncated multiply and accumulate (PTMAC) architecture acts as a mean to employ PTM in biomedical applications which requires simple DSP, such as ECG filtering or fall detection [13]. The Testing Based Low Power with PTMAC is introduced as an outspread of BIST to support general DSP architecture. LT-PTMAC is inexpensive towards implementing PTM with low power applications. The total control unit functions under five stage program, memory blocks and pipeline in a multi-bus Harvard configuration. The proposed CUT for Testing based Razor based PTMAC architecture (LT-PTMAC) is structured as shown in Fig. 7. On performing timing analysis, the critical path is experienced with the MAC design belonging to arithmetic unit. The arithmetic unit reduces power consumption obtained through the voltage scaling method using the arithmetic unit. An

experiment is done to link the delay-modulation abilities of programmable truncation and to aid the fault tolerance. The proposed amplified cells are structured and accumulated as library cells for post synthesis insertion. These cells respond to the unique implementation that is Razor implementation, where the Razor registers containing the shadow latch is replaced with shadow-flip-flop. Hence it avoids combination issues. The meta-stability sensor required in Razor implementations is programmed as the delay of an inverter, which acts as a limitation to the hold time of the Razor accumulator.
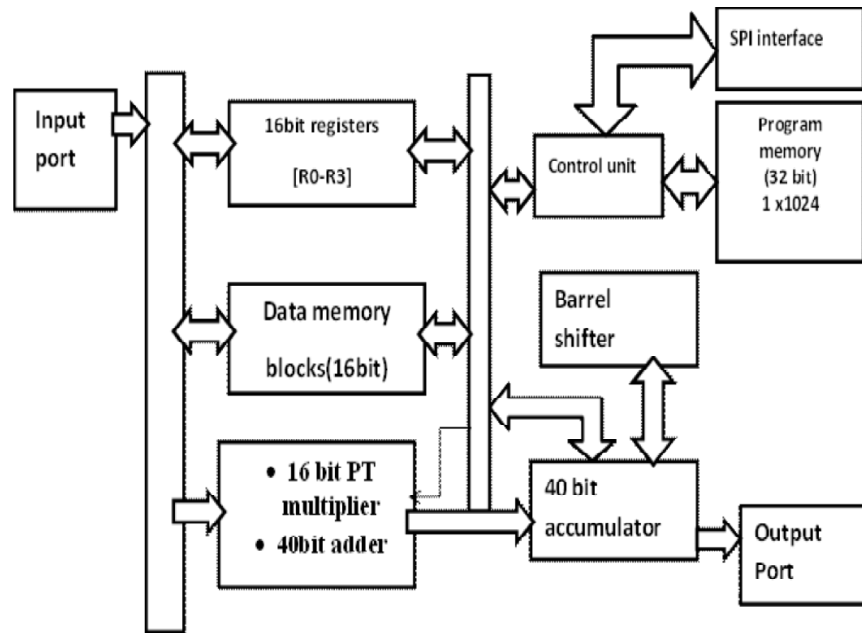

Figure 5: Low power top level Testing based PTMAC
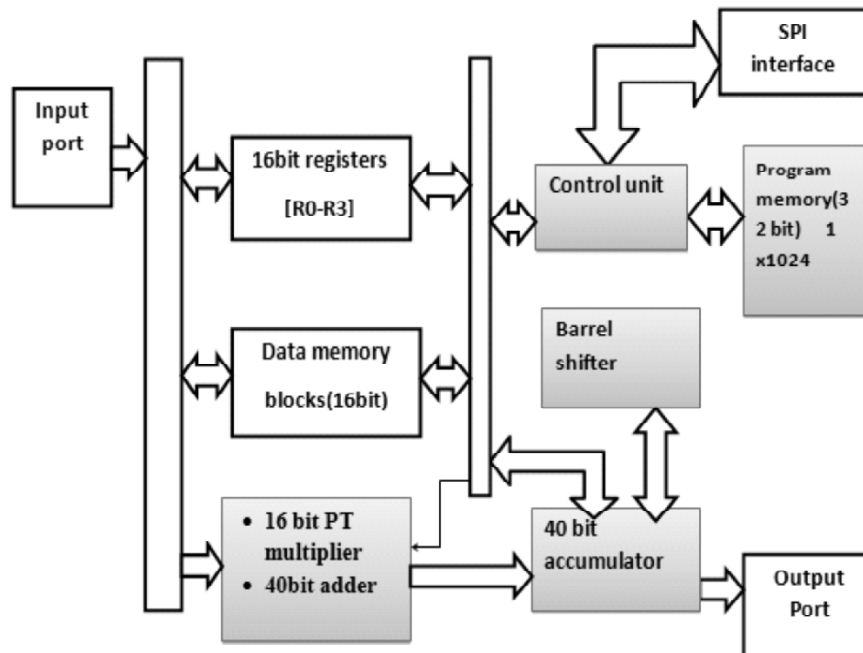

Figure 6: Top level CUT for LT-PTMAC

The architecture has to be modified in a Razor implementation for error rectification by allocating a special clock cycle for the data error. The instruction of an execution cycle in the Sleep Mode Low Power Razor-augmented PTMAC (LTPTMAC) can thus be divided into four possible stages. The fault tolerance is accomplished by some alteration of the PTMAC unit. The accumulator unit of the PTMAC was replaced by a fault tolerant version named Razor Accumulator in which the original flip-flops were substituted by a version of the Razor registers [14]. The Meta stability detector in Razor implementations was designed as the delay of an inverter added as a constraint to the hold time of the Razor accumulator. Static timing analysis of PTMAC proved that the only registers situated at potentially critical paths within PTMAC were located in the accumulator. The multiplication and accumulation of the input data are restricted within a clock cycle. When the error detection phase is shortened, it detects that an obstacle formed by transparent latches is located between the compression tree of the multiplier and adder blocks.

A functional broadside test cube that detects a fault creates signal transitions occurring during functional operation in a sub-circuit around the site off. Thus, fault is detected with functional operation conditions in a sub circuit close to it when the test cube is combined with other test cubes to form a test. For the discussion in this brief, the primary input sequences are considered as unconstrained. A low complexity sequential test is performed to generate functional test sequences.

The test cubes, thus create functional operation conditions throughout the circuit. The execution cycle of an instruction on the Razor-augmented PTMAC can be thus divided into four possible stages as follows,

a) **EP:** The initial half clock cycle is referred as the execution phase in which the instruction initiates its execution. However, it fails to attain the augmented registers due to hold time needs.

b) **AP:** The second half clock cycle is referred as the arrival phase (AP) in which the instruction completes its execution and data reach the destination registers. Instructions that fail to do will generate either an Error or a System Failure.

c) **EDP:** The third half clock cycle is referred as the error detection phase, in which the failed signals can complete their execution by creating a Razor error. Instructions failing to finish during the third stage will cause a System Failure, limiting the minimum supply voltage applicable to the system.

d) **ECP:** The fourth stage is referred as the error correction phase (ECP). When an Error is flagged in EDP, a multiplexer will supply the output of the value which formerly taken by the Shadow Latch into the input of the main flip-flop, and the error signal will be cleared.
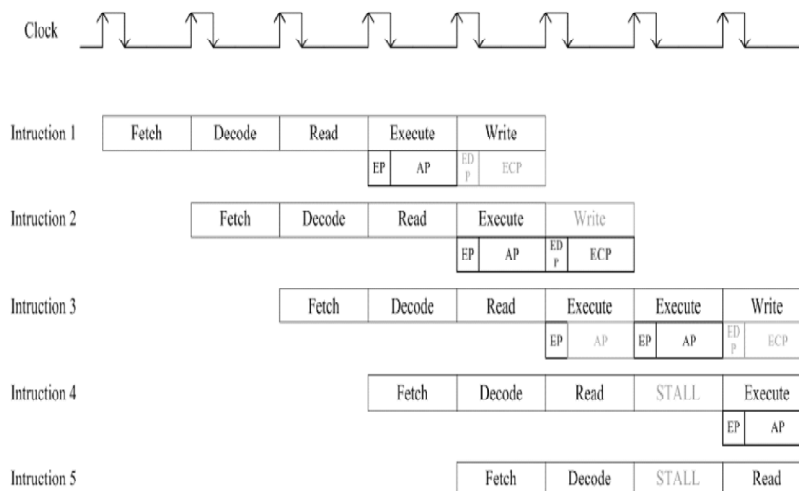

Figure 7: Execution of five instructions in the Razor PTMAC pipeline

EP and AP stages represent a regular execution stage of a pipeline, while the last two stages (EDP and ECP) overlap with the execution stage of the next instruction in Fig. 7. Detecting an error in the EDP phase causes either the ECP of the faulty instruction or the AP phase of the following one to update the output of the Razor registers.

Table 1
Comparison Results of Various PTMAC Architectures

PTMAC	I-PTMAC	LT-PTMAC
AREA		
Amount of slice registers =228	Amount of slice registers= 119	Amount of slice registers= 96
Amount of Slice LUTs= 534	Amount of Slice LUTs = 421	Amount of slice LUTs = 86
HDL Synthesis Report		
Adder/subtractors = 24	Adder/subtractors = 22	Adder/ subtractors = 15
Registers= 36	Registers = 18	Registers = 11
Latches =29	Latches = 22	Latches = 11
Comparators=8	Comparators = 6	Comparators=3
Multiplexers = 120	Multiplexers = 110	Multiplexers = 87
Tristates = 1	Tristates = 0	Tristates = 0
Xors = 145	Xors =122	Xors =89

From the experimental analysis, we got results on the area, no of circuits, power, energy, time and speed for the three architectures. The results are compared and depicted as a bar chart as follows. PTMAC requires large area for slice registers and slice LUTs. On the other hand, LR-PTMAC requires small area. Also the area for slice registers is more than that for slice LUTs in PTMAC and I-PTMAC. But in LT-PTMAC number of slice LUTs is more than the number of slice registers. No. of adders, registers required for PTMAC is more than that required for others. I-PTMAC don't have latches while LT-PTMAC has more latches than PTMAC. Unlike PTMAC and I- PTMAC, LT-PTMAC has more comparators.

By comparing the required no of multiplexers and registers, it is clear that in all the three architectures the no. of registers is more than the no. of multiplexers used.

From the experimental analysis, we got results on the area, no of circuits, power, energy, time and speed for the three architectures. The results are compared and depicted as a bar chart as follows. PTMAC requires large area for slice registers and slice LUTs. On the other hand, LR-PTMAC requires small area. Also the area
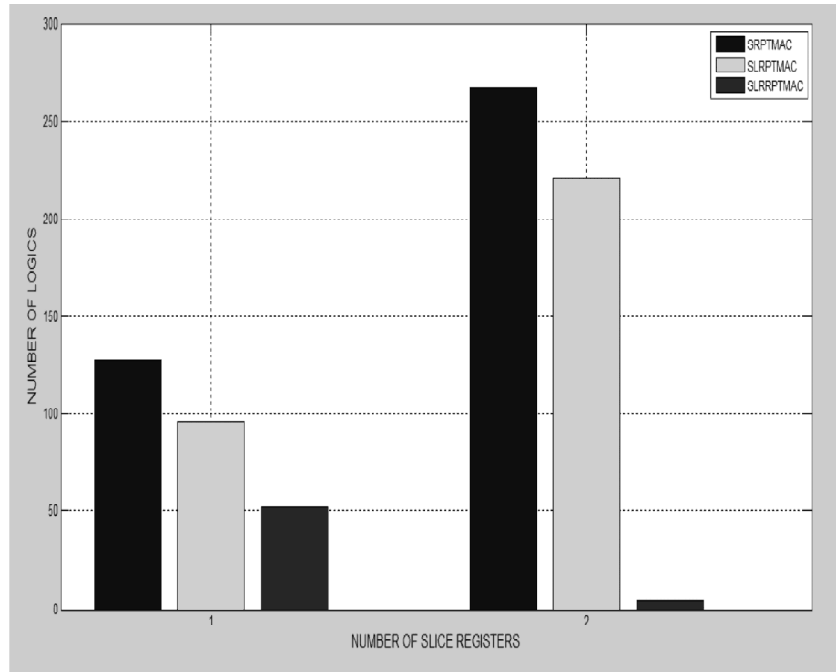
Figure 8: Area of slice registers for PTMAC

Table 2
Comparison of PTMAC Power

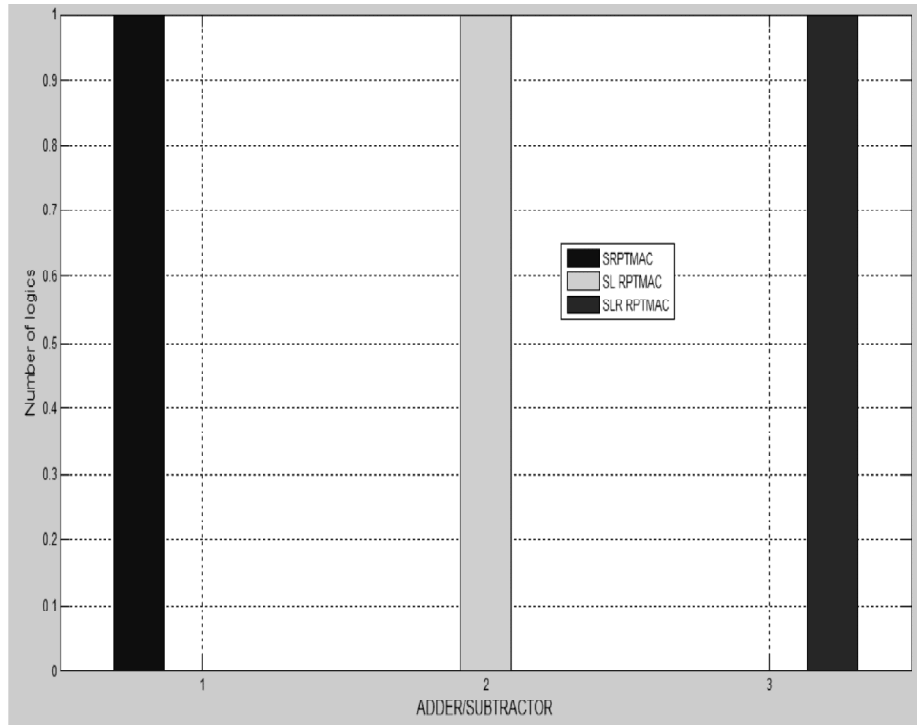| Power		
Logic = 0.926	Logic = 0.848	Logic = 0.712
IOs = 84.6816	IOs = 84.6816	IOs = 65.4896
Speed		
Minimum period = 2.432 ns	Minimum period = 2.101ns	Minimum period = 1.140 ns
Minimum input arrival time before clock = 1.912 ns	Minimum input arrival time before clock = 1.499 ns	Minimum input arrival time before clock = 1.302 ns
Maximum output required time after clock = 3.408 ns Maximum combinational path delay = 3.003 ns	Maximum output required time after clock = 3.408 ns Maximum combinational path delay = 3.003 ns	Maximum output required time after clock = 0.822ns Maximum combinational path delay = 0.145ns
Total Real time to Xst completion = 18.38 secs	Total Real time to Xst completion = 17.42 secs	Total Real time to Xst completion = 8.02 secs
Total CPU time to XST completion = 19.06 secs	Total CPU time to XST completion = 13.24secs	Total CPU time to XST completion = 8.13 secs

Figure 9: Adder/Subtractor analysis

Table.3
Comparison of PTMAC Speed

Speed		
Minimum period = 2.432 ns	Minimum period = 2.101ns	Minimum period = 1.140 ns
Minimum input arrival time before clock = 1.912 ns	Minimum input arrival time before clock = 1.499 ns	Minimum input arrival time before clock = 1.302 ns
Maximum output required time after clock = 3.408 ns	Maximum output required time after clock = 3.408 ns	Maximum output required time after clock = 0.822ns
Maximum combinational path delay = 3.003 ns	Maximum combinational path delay = 3.003 ns	Maximum combinational path delay = 0.145ns
Total Real time to Xst completion = 18.38 secs	Total Real time to Xst completion = 17.42 secs	Total Real time to Xst completion = 8.02 secs
Total CPU time to XST completion = 19.06 secs	Total CPU time to XST completion = 13.24secs	Total CPU time to XST completion = 8.13 secs

for slice registers is more than that for slice LUTs in PTMAC and I-PTMAC. But in LT-PTMAC number of slice LUTs is more than the number of slice registers. No. of adders, registers required for PTMAC is more than that required for others. I-PTMAC don't have latches while LT-PTMAC has more latches than PTMAC. Unlike PTMAC and I-PTMAC, LT-PTMAC has more comparators. It is also seen that in all the three architectures, number of registers used is greater than number of adders, latches, comparators used.
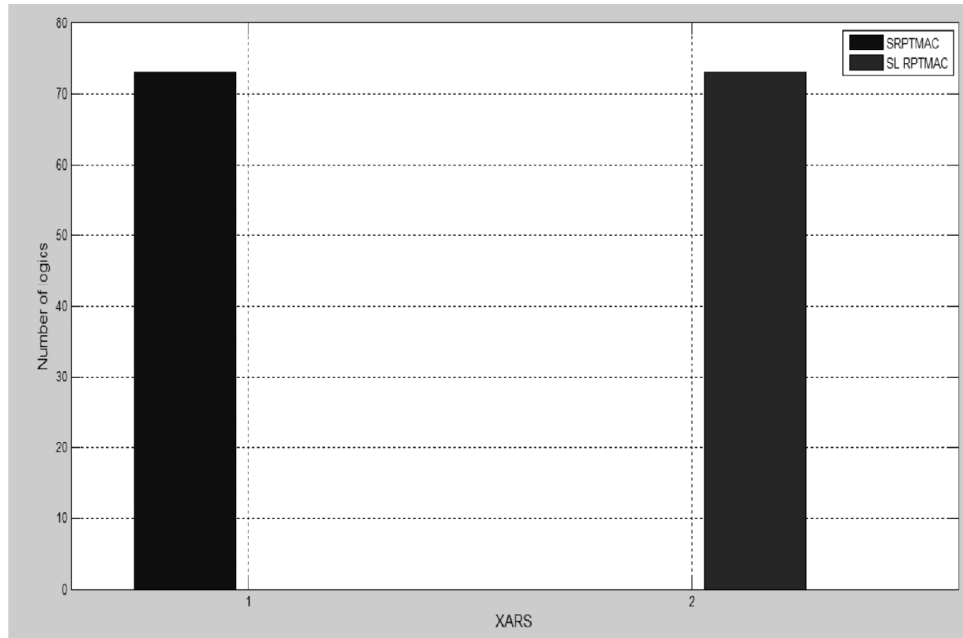

Figure 10: Number of XORs for PTMAC and LT-PTMAC

From counter and comparator analysis, it is shown that in PTMAC and I-PTMAC the no. of comparators are less than counters. But in I-PTMAC more comparators are used than counters. Thus it is clear that the no of circuits required is less in LT-PTMAC (except counters). Considering the logic power, the high value corresponds to PTMAC and low value corresponds to LT-RPTMAC. Equal IOs power is used by PTMAC and I-PTMAC. For LT-PTMAC IOs power is low. Among the three architectures LT-PTMAC has low logic power and low IOs power.
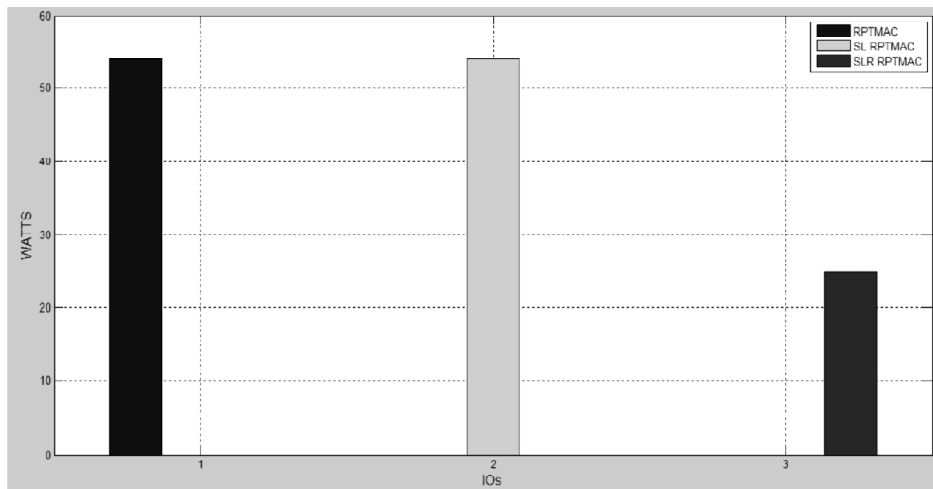

Figure 11: Total Real Time for Xst Completion

From the graph it can be obviously said that the minimum period, minimum input arrival time, maximum output required time maximum combinational path delay are low for I-PTMAC. So it is proved that the delay is very low in LT-PTMAC. Total real time and CPU time needed for Xst completion is high for I-PTMAC and low for LT-PTMAC.

V.   CONCLUSION

The use of Razor on a PTMAC structure has been tested at a post synthesis simulation level to study the effect and interactions of both energy reducing techniques on a previously tested DSP design. The timing and power effects of VOS with error correction and the application of programmable truncated multiplication resulted in significant power reductions. It describes a test generation procedure briefly that produces a compact low-power skewed-load test set by merging of skewed-load test cubes that are derived from functional broadside tests. Test cube merging was implemented in a way that would ensure that the fault coverage of the final test set will not be limited by the fault coverage of functional broadside tests.

Thus, the proposed method indicates that the delay-modulation properties of truncated multiplication and BIST using testable circuits can be exploited to improve the energy consumption of fault tolerant DSP architectures where multipliers are involved in the critical path of the circuit. Various power reduction values of novel design are obtained experimentally and given. Leakage power is decreased by 18.7% in low-power vs. high-speed mode (on average); dynamic power is reduced by 99.4% up to. Leakage power in sleep mode is 18.1% lower than that in high-speed mode.

REFERENCES

[1]   Whitemouth, Paul N., Siddhartha Das, and David M. Bull. "A low-power 1-ghz razor fir accelerator with time-borrow tracking pipeline and approximate error correction in 65-nm CMOS." IEEE Journal of Solid-State Circuits 49, no. 1. 2014.

[2]   Kuang, Shiann-Rong, and Jiun-Ping Wang. "Design of power-efficient configurable booth multiplier." IEEE Transactions on Circuits and Systems I: Regular Papers 57, no. 3. 2010.

[3]   De la Guia Solaz, Manuel, Wei Han, and Richard Conway. "A flexible low power DSP with a programmable truncated multiplier." IEEE Transactions on Circuits and Systems I: no. 11 2012.

[4]   Petra, Nicola, Davide De Caro, Valeria Garofalo, Ettore Napoli, and Antonio GM Strollo. "Truncated binary multipliers with variable correction and minimum mean square error." IEEE Transactions on Circuits and Systems I: Regular Papers 57, no. 6, 2010.

[5]   Fojtik, Matthew, David Fick, Yejoong Kim, Nathaniel Pinckney, David Harris, David Blaauw, and Dennis Sylvester. "Bubble Razor: An architecture-independent approach to timing-error detection and correction." In 2012 IEEE International Solid-State Circuits Conference, 2012.

[6]   Pomeranz, Irith. "Static test compaction for delay fault test sets consisting of broadside and skewed-load tests." 29th VLSI Test Symposium. IEEE, 2011.

[7]   Shidhartha Das, David Roberts, Seokwoo Lee, Sanjay Pant, David Blaau, Todd Austin, Krisztián Flautner and Trevor Mudge , "Self-Tuning DVS Processor Using Delay-Error Detection and Correction", ieee journal of solid-state circuits, vol. 41, no. 4, april 2006.

[8]   R.S. Katti, X.Y. Ruan , and H. Khattri," Multiple-Output Low-Power Linear feedback shift register design," IEEE Trans.circuitsSyst.I,Vol.53,No.7,pp-1487-1495,July 2006.

[9]   P. Girard, L.Guiller, C. Landrault, S. Pravossoudovitch and H.J. Wunderlich," A modified clock scheme for a low power BIST test pattern generator," 19th IEEE proc. VLSI test Symp.,CA,pp-306-311,Apr-May 2001.

[10]   S.C. Lei, X.Y.Hou, Z.B.Shao and F. Liang," A class of SIC circuits: Theory and application in BIST design," IEEE trans. circuits syst. II, vol.55, no.2, 2008.

[11]   S.-R. Kuang and J.-P. Wang, "Design of power-efficient configurable booth multiplier,"IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 3, pp. 568–580, Mar. 2010.

[12] Z. Huang and M.D. Ercegovac, "Two-Dimensional Signal Gating for Low-Power Array Multiplier Design," Proc. of IEEE International Symposium on Circuits and Systems, Vol. 1, pp. 489-492, 2002.

[13] J.-S.Wang, C.-N. Kuo, and T.-H. Yang; "Low-power fixed-width array multipliers," International Symposium on Low Power Electronics and Design, pp. 307-312, Aug. 2004.

[14] H. Park and Jr E.E. Swartzlander, "Truncated multiplications with symmetric correction," in Asilomar Conference on Signals, Systems and Computers, ACSSC, 2006.