VOL. 12, NO. 21, NOVEMBER 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

THREE-CODING TEST COMPRESSION TECHNIQUE FOR SOC BASED DESIGN

Chakrapani K., Saravanan and S., Muthaiah R.
School of Computing, SASTRA University, Thirumalaisamudram, Thanjavur, Tamil Nadu, India
E-Mail: sarn@core.sastra.edu

ABSTRACT

Testing a system-on-chip (SoC) result in serious challenges due to the growth of volume and test power consumption of test data sets. Several compression techniques for test data came into light for reducing volume of the test data and its scan power. This paper mainly concentrates on increasing the compression ratio and power consumption for scan vectors through new three coding compression technique. This encoding scheme includes block merging along with three types of coding techniques to obtain efficient compression of scan test sets. This paper review the impact of compressed test data on consumption of power and test application time are also considerable. The actual test data is decoded with Simple decoder architecture. Experimental Results on ISCAS89 benchmark circuit shows the effective ratio of proposed method compared with prior works.

Keywords: three coding, test data compression, test compatibility, test power, test application time.

1. INTRODUCTION

Test application time and large volume of the test data are the demanding problems in testing over system on a chip (SoC). SoC has multiple intellectual properties (IP) core embedded in chip circuit becomes more complex with large volume of test sets. Several factors like test data power, test data volume and testing time determine the complexity. Each of these blocks must be trained with larger test patterns that are pre-computed. The SoC having large volume of test data surpasses the storage capacity of Automatic Test Equipment (ATE), and even increase more test power and high time required for testing.

This paper focuses on optimal selective count compatible based run length, which achieves maximum compression with less test cost. They used 10 code run length along with selective count method. Block segment concept is used in inter and intar merging technique.

Test power is another factor which causes dynamic power dissipation for test patterns, which is reduced due to switching activity between the test data patterns, while testing the chip on SoC having power consumption is necessary since excessive power can cause high power consumption which damages the Circuit under Test (CUT) to be tested.

Optimal selective count compatible based run length, which achieves maximum compression with less test cost. They used 10 code run length along with selective count method. Block segment concept is used in inter and intar merging technique [1].

The thermal based test data compression by using dictionary based coding method. Minimized transition count in scan chain is used in this method. They proved less compression in Thermal based don't care filling and higher temperature produced in compression based don't care filling [2]. Based on LFSR compression ratio along with low power consumption of test patterns, its Shift power and capture power is discussed with Compatible block code. They also worked to resolve the conflict bit which occurs between low power filling and LFSR seed encoding [3].

Using Dynamic reseeding of LFSR method is proposed. Combinational decompression is employed on LFSR input without the need of phase shifter / state skipping module. This method promisses huge test coverage, reduces hardware resources and less test application time [4]. Another one method is flexible Run length code to achieve higher compression ratio and shorten test pattern application time. The internal 2-n PRL codes the reference segment divided into 2n patterns inside the single segment. Thus multiple segments are compatible with N runs of N-PRL [5].

More Power is consumed in test mode rather than in normal mode. The switching activity causes the power consumption in test patterns [6]. On minimizing the switching activity test power can be reduced. The bits with unknown values are filled with zeros and ones, such that the power is calculated by using weighted transition metric (WTM) [7, 8].Time for test application is high where the volume of data is large, which leads to have more test cost. It is determined by transfer of test patterns of ATE memory frequency. The large volume of test data is reduced to gain low test application time [9]. The limitations of ATE memory is overwhelmed by reducing test application time for SoC circuits.

Many compression techniques for test data are proposed for the problems having testing time and power by abusing the unknown bits. The pre-computed test data reduces actual number of bits used for storage in ATE memory [6, 7, 13, 17]. The compressed is to be decompressed using decompress or architecture to apply in Circuit Under Test (CUT) for the test sets. The test power, test data volume and test application time is reduced by test compression method. Linear decompression, broadcast code based and scan based schemes are some of the schemes reduces that volume of test data.

This paper categorizes the other sections as, the works are related to the method proposed in section 2, proposed compression method to validate the compressing data in section 3, the process of decompression is shown in section 4, the test power is shown in section 5, the

Experimental results for the above proposed method are shown in section 6 and ultimately in section 7 shows the conclusion for the work.

2. RELATED WORKS

In the process of reducing the compressed test data several compression techniques are used.Linear decompression shifts data linearly using LFSR reseeding and XOR operations for encoding the test data [3] by automatic test pattern generation (ATPG). And here the broadcast scan based schemes shifts the data from single scan chains to multiple scan chains. It reduces the decompression area of the test data. The linear and broad scan schemes low area overhead. In Code based schemes the test data is partitioned and encoded with special symbols. The symbols are then replaced with code word to obtain the compressed data [2]. Normally the precomputed test data is higher than the compressed data. Larger the test data more memory of Automatic Test Equipment (ATE) is required, resulting in reduced compression ratio. The compressed data not only reduces ATE memory but also gives low test power and lessens test application time.

Golomb codes of group size with variable length to a run of 0's. Golomb-codes are mainly of two different parts named prefix and tail. Use run of 0's as input pattern test sets for greater compression.Compression and decompression based on Golomb codesfits for compressing the precomputed test sets of System On aChip (SoC). The main advantages are, it is a less data compression and gives lower area [4]. Data compression technique has test independent methods which runs with 1's and 0's. FDR code [5] is a run length code with variable to variable thatrunswith 0's. It has prefix and tail with same size of bit to form code word which are assigned with frequency of variable length. The changing of inputs gives various results. The EFDR code (extended FDR code) is also similar to FDR but runs with both 1's and 0's [6]. To determine run length of 1's extra bit 0 is replaced with 1 at starting of code word formation and the test power is high.

A unique approach to minimize the test data power and volume[6] is obtained by TRP to decrease data volume, power andtime required for the test pattern. It is based run length code to increase compression ratio and mapping the unknown bits to zeros and ones in reducing the scan average power and test application time. The paper [7] shows the concurrently encoded data for test data compression and to reduce time for the chip under testing.The independent compression techniques, BM [9] technique merging the compatible consecutive blocks of precomputed test data and filling the blocks consisting all 0's and1's to compress the data. The BM-8C [10] is another independent technique that encodes every merged block and one or many consecutive and compatible blocks for converting each of the merged block to a relative code-word according to eight properties. It is row-column reduction routine to decrease test data volume.

The compression method 9C [8] splits the symbols and finds whether the two halves were specified with 0's and 1's then encodes the data with nine symbols. The test non independent techniques SHC [15] which encodes the data according to frequency occurred. The variable length input huffmann code also encodes as frequency occurred for the test patterns. Shift in power was focused by paper [18] with threshold method of unspecified test patterns. Appropriating indexing and encoding is in this method for reducing switch activity of the test patterns.

Low-power scan operation [1] achieved by compression test vectors. The compressed data will be saved in the memory of ATE. Again the data is recovered by the on-chip decoder. Mostly it doesn't need the structural information of the embedded cores.Low power selective compression different stages of test data is formed then applying the code word method to get compression ratio. The data is taken linear and splitting the test data patterns process to take long time. Then binary values are used to map the test sets, to determine testing power[18].Coloring algorithm method is used in paper [19] which promises for low power test pattern compaction.XOR network based scan power reduction was identified in [20] travelling salesman problem is used to find out the least number of test pattern switching.

2.1 Novelty of this proposed work

Test data compression is one of the major issue need to be considered in modern VLSI testing. Better and efficient Run length code is proposed in this work. This method focuses on efficient three coding run length compression technique along with block merging method. This proposed encoding method promisses more compression ratio, better test application time and low power consumption than compared with previous work. Experimental results also evident that proposed decoder architecture consumes less area than earlier proposed methods.

3. PROPOSED COMPRESSION METHOD

The proposed compression method is related to partitioning of the total test data into different blocks of bits sized b where b is multiples of 2, and merges the partitioned, consecutive and compatible blocks. These blocks are said to be compatible only if corresponding bits are same or one bit be X (don't care bit). Merging all consecutive compatible blocks into one merged block with count to represent how many blocks merged [10]. Encoding the partitioned test data with proposed three properties half-compact, half-invert and uncompact. The merged block is said to be half-compact only if first half of the block is compatible with next half of the block. If the merged block is said to be half-invert only if first half of the block is inverse compatible with remaininghalf part of the block. The merged block is said to be un-compact when it doesn't have both compatible and inverse compatible and it is mixed of 0, 1 and X (don't care bit). The Considering an example of the test data 10XX0X1X1X1X0X0X11XX is divided into two blocks of bit size as 10 and is formed as 10XX0X1X1X and 1X0X0X11XX the formed blocks are apparently two consecutive and compatible blocks and they are merged

into a single block as 100X0X111X with merged count 2. After merging these all consecutive compatible blocks, BM_8C [10] has used the 8 properties to encode the test data to achieve 68.14% of average compression ratio. The proposed method used three properties to encode the test data and achieved 69.9% of compression ratio. This increases the 1.76% of compression ratio and reduced the uncompressed merged test data. Compression achieved for the above test set is generated by Mintest [11] for six large ISCAS89 benchmark circuits.

Table-1, as described in [10] shows the encoding of test data as divide into five groups B0-B4 for the merging of consecutive compatible blocks. MC column provides the merged count for the merged blocks and having prefix and tail to form the codeword to encode the test data. If MC belongs to first group (MC = 1) that indicates it doesn't merged with any block. If the merged count is in between 8 to 15 ($8 \leq MC \leq 15$)then MC belongs to fourth group A3 indicates that number of bits used. The group B2 has prefix 110 to indicate the count of merged block that liesin between 4 to 7, and group gives the utilization of bits to add in codeword. For each and every group the proposed compression method uses three properties as shown in column types and tail is formed according to the prefix and types to encode the test data.

Table-1. Encoding table for proposed compression technique.

Group	MC	Prefix	Types	Tail	Code word
B0	1	0	half-compact	1 + block/2	0 + 1 + block/2
			half-invert	0 + block/2	0 + 0 + block/2
			uncompact	block	0 + block
B1	2-3	10	half-compact	1 + block/2	10 + 1 bit + 1 + block/2
			half-invert	0 + block/2	10 + 1 bit + 0 + block/2
			uncompact	block	10 + 1 bit + block
B2	4-7	110	half-compact	1 + block/2	110 + 2 bits + 1 + block/2
			half-invert	0 + block/2	110 + 2 bits + 0 + block/2
			uncompact	block	110 + 2 bits + block
B3	8-15	1110	half-compact	1 + block/2	1110 + 3 bits + 1 + block/2
			half-invert	0 + block/2	1110 + 3 bits + 0 + block/2
			uncompact	block	1110 + 3 bits + block
B4	16-31	1111	half-compact	1 + block/2	1111 + 4 bits + 1 + block/2
			half-invert	0 + block/2	1111 + 4 bits + 0 + block/2
			uncompact	block	1111 + 4 bits + block

Table-1, as described in [10] shows the encoding of test data as divide into five groups B0-B4 for the merging of consecutive compatible blocks. MC column provides the merged count for the merged blocks and having prefix and tail to form the codeword to encode the test data. If MC belongs to first group (MC = 1) that indicates it doesn't merged with any block. If the merged count is in between 8 to 15 ($8 \leq MC \leq 15$)then MC belongs to fourth group A3 indicates that number of bits used. The group B2 has prefix 110 to indicate the count of merged block that liesin between 4 to 7, and group gives the utilization of bits to add in codeword. For each and every group the proposed compression method uses three properties as shown in column types and tail is formed according to the prefix and types to encode the test data.

Table-2, shown the example of test data encoding with block size 10 and data is formed into blocks from C1 to C10 as unmerged blocks. The blocks C1-C2, C3-C7 and C8-C10 are consecutive compatible blocks then the blocks are merged to form merged block, MC column represent the count for merged block. Type column represents which type of data is formed and the compressed codeword is formed. The last column length indicates total bits formed from codeword then total length gives the compressed test data. Here the second merged block count is 5 and first half of the block is inverse to second half so it forms half-invert block having codeword 110 01 0 011X0. The codeword having prefix 110 to indicate the count of merged block that lies in between 4 to 7, 01 represent the binary representation of MC, 0 represents the type of the block half-invert and 011X0 represents first half of inverse merged block. The encoding is similar for other blocks of data. The proposed method compress the total test data containing 100 bits to 33 bits having compression ratio 67%.

Table-2. Proposed encoding example.

Blocks	Unmerged Blocks	Merged Blocks	MC	Type	Cdeword	Length
C1	10XX1 1XXX1	10X11 10XX1	2	half-compact	10 0 1 10X11	9
C2	1XX1X X0XX1					
C3	0X1X0 10XX1	011X0 100X1	5	half-invert	110 01 0 011X0	11
C4	X11XX XX0XX					
C5	0XXX0 X0XXX					
C6	X11X0 XXXXX					
C7	X1XXX 1X0X1					
C8	110X0X XX0X	11010 1XX01	3	un-compact	10 1 110101XX01	13
C9	X1XX0 1XXX1					
C10	1X01X 1XX01					

4. DECODING METHOD

A decoding architecture is formed for the proposed compression method to decode the original test from the compressed data to perform design under test (DUT) as shown in Figure-1. The design is scanned fully having single out. The compressed data having codewords can be stored in memory of ATE then sending one to next bit for testing over Data_in (D_in). The decoder has finite state machine (FSM) having clk (clock) as input is considering as ATE clock and clk_dut is the input clock for testing the circuit. The En_out is control signal for scan out data which are generated from FSM. The decoding process have some blocks of decoding such as decoder, register, counter, array of multiplexers and buffer. These are described briefly to decode the encoded data.

The decoder has input clk and the D_in comes from register counter, enables the decoder data through FSM and D_out is fed to register counter which has data reset for the merged count then reg_D is used to store values and reset using data_rst containing from FSM, the decoder has n bit data and when enabling en_dec will decreasing reg_D for every clk cycle. The counter is used for block lengths of data provides with clk_dut is reset to b bit blocks. Several states of FSM is controlled with counter which the codewords are recycled.

The architecture has array of multiplexers for blocks of data and every mux has 0, 1 and Data_in as inputs and sel signal is used for selecting the two control bits. Any one of the signal is selecting output of FSM, if the data has half-compact then the first half of the data remains same and leaves to output to buffer.

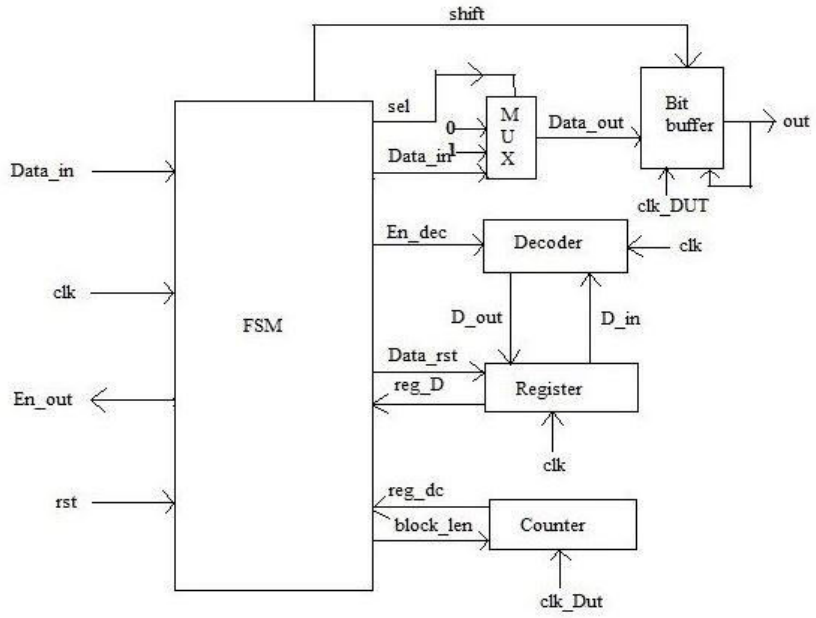

Figure-1. Decompression architecture for compressed three coding.

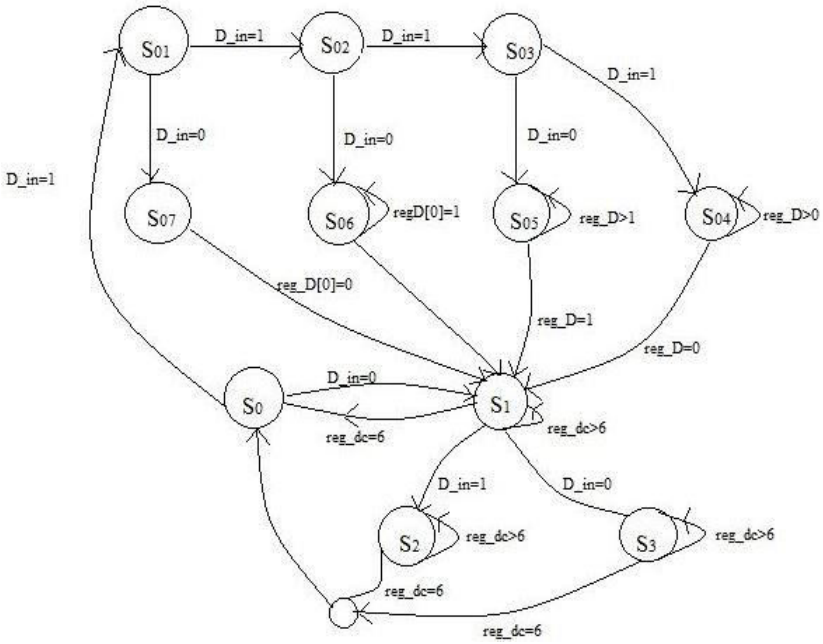

Figure-2. Finite state machine for decompression.

To shift the bits buffer is used as bit shift register to store output data from mux and them to out data. The data is comes out when En_out and shift signals are set to 1. Then output of scan decompressed data is send through (scan_out) out port and the data is back to the buffer.

In Figure-2 shows the FSM for the decompression architecture which is used to identify the codewords and to activate the other signals that are controlling data sending bits. Let us considering the merged type is half-compact the first half of the block of decompressing data is selected and produces the same data of d0 - d (b/2-1). If the type is half-invert it inverse second half of the data and select value for d (b/2) - d (b-1). The sel signal is selected to determine every half of the block. The FSM has out 0, 1 and data_in to generate the types of merged data. The FSM has the states S0, S1, S2 and S3. It has S01 to S07 states to check the prefix bit and fed the next bits to state S1. The blocks merged stores in the states S04 to S07. Considering S01 state if D_in = 0 it goes to S07 similarly if D_in = 1 it goes to S02. The process will continue for further states and moves to store the values. The states S2, S3 and S1 represent the type of code it activates. S2 activates half-compact, S3 activates half-invert and S1 activates un-compact only when tail starting bit is not occurred. The signals are further loaded depends on the occurrence of merged count. The signals have reset the block length of data.

5. POWER ESTIMATION

The compressed test data on power consumption during testing is determined. This paper shows how the test power is consumed by filling unknown bits with appropriate binary values for the merged data.In SOC consumed power is addressed into two types named as dynamic and static power. When transitions are occur from one to zero and zero to one the dynamic power is taken into account. While the process of power consuming leakage power caused by static power which is not considered in this paper. On using the Weighted Transition Metric (WTM) [18] test power consumption is estimated. The WTM models for the power consumption not only depends on the transitions occurred for test patterns, it also depends on various positions. Weighted transition metric is mainly associated to switching activity during scan chain operation for circuit under test. WTM for the scan vector is applied by considering scan length of the circuit. Let us consider a scan test vector $V_p = V_{p,1}, V_{p,2}, V_{p,3}, \ldots V_{p,l}$ and length of the test vector is l. WTM for test patterns is calculated as,

$$WTM_p = \sum_{q=1}^{l-1} (V_{p,q} \oplus V_{p,q+1}) * (l - q) \quad (1)$$

Considering an example of the test data 10X01X101X010XX1 is divided into two blocks of bit size as 10 and is formed as 10X01X10 and 1X010XX1The divided blocks containing X is filled as 10001110 and 11010001 and applying the Weighted Transition Metric to calculate power. This is caused by switching activity between bits transferred. The data having the length 8, the power is 12 for first value and 16 for the next value, this is achieved from equation (1).

6. EXPERIMENTAL RESULT

The experiments that are conducted on six large ISCAS89 benchmark circuits have achieved best compression ratio. The proposed three coding technique gives the better compression for the Mintest [12] test sets. The other code based schemes are considered to compare with proposed method to show better results.

The proposed three code compression technique provides various blocks of compression as exhibited in

Table-3. The 2nd column in Table-3 contains total number of test data bits. The next column block size has 1 to 9 columns shown the blocks of different sizes and the last column shown best compression ratio among all blocks. The percentage of compression ratio (CR) is calculated as,

$$CR\ \% = \frac{(TD-TE)}{TE} * 100 \quad (2)$$

Table-3. Compression ratio for different block size.

Circuits	Total data	Block size									Best Compression Ratio
		4	6	8	10	12	14	16	18	20	
s5378	23754	56.31	60.56	59.67	59.39	59.76	58.24	57.91	58.0	57.29	60.56
s9234	39273	54.68	57.05	58.13	58.37	56.22	56.45	54.11	52.99	53.07	58.37
s13207	165200	87.22	87.96	88.02	88.28	87.57	86.53	85.99	86.96	86.46	88.28
s15850	76986	73.31	74.64	75.08	74.84	73.63	74.08	73.02	71.82	71.71	75.08
s38417	164736	62.16	64.02	62.06	60.83	60.08	59.93	58.79	57.69	55.50	64.02
s38584	199104	70.72	72.72	73.0	73.07	72.16	71.93	71.01	70.68	69.88	73.07

In above equation the total test data is represented with TD and total compressed test data is represented with TE. The proposed three code compression ratio is compared with other existing methods like Golomb [4], FDR [5], VIHC [14], SHC [15], 9C [8], EFDR [6], BM [9] and BM_8C [10] and the best compression ratio for benchmark circuits is shown in Table-4. The average compression ratio is provided in last row of Table-4. Considering circuit s13207 and comparing the proposed with FDR and VIHC achieves 6.98 % and 4.8 % high compression ratio respectively. The higher compression leads to less testing time and low test power.

Table-4. Comparison of compression ratio with other code based techniques.

Circuits	Golomb [4]	FDR [5]	VIHC [14]	SHC [15]	9C [8]	EFDR [6]	BM [9]	BM_8C [10]	Proposed
s5378	37.11	48.02	51.52	55.10	51.64	53.67	54.98	58.56	60.56
s9234	45.25	43.59	47.25	54.20	50.91	48.66	51.19	57.49	58.37
s13207	79.74	81.30	83.51	77.00	82.31	82.49	84.89	87.52	88.28
s15850	62.82	66.22	67.94	66.00	66.38	68.66	69.49	73.69	75.08
s38417	28.37	43.26	53.56	59.00	60.63	62.02	59.39	59.92	64.02
s38584	57.17	60.91	62.28	64.10	65.53	64.28	66.86	71.66	73.07
Average	51.74	57.22	61.02	62.57	62.90	63.30	64.47	68.14	69.90

The decompression is performed for the proposed encoded test data for ISCAS89 benchmark circuits in HDL languages. The percentage of area overhead decompression is calculated as,

$$\text{AOD}\ \% = \frac{\text{Decompressor Area}}{\text{Benchmark circuits Area}} * 100 \quad (3)$$

On achieving the high compression ratio the area of decompressor causes more cost, considering the circuit s9234 and comparing with [10] the area increases up to 1.5 % and for the circuit s38417 it reduces to 0.8 %. Table 5 show the proposed three code area overhead with other existing methods as Golomb [4], FDR [5], VIHC [14], SHC [15], 9C [8], BM [9] and BM_8C [10]. The decompression area costs more so can use the other compression techniques for low area overhead.

Table-5. Comparison of area overhead decompression.

Circuits	Golomb [4]	FDR [5]	VIHC [14]	SHC [15]	9C [8]	BM [9]	BM_8C [10]	Proposed
s5378	4.0	7.8	5.8	16.0	8.2	12.8	16.5	18.3
s9234	3.2	5.9	4.6	13.0	6.2	9.7	12.7	14.2
s13207	4.1	3.5	2.2	5.7	3.7	5.8	6.6	7.4
s15850	2.0	3.6	2.3	6.5	3.8	5.9	7.1	6.2
s38417	0.5	1.4	0.7	2.0	1.5	2.3	2.8	2.0
s38584	0.7	1.5	0.7	2.0	1.6	2.3	2.9	2.1

Now test power is analyzed for the circuits using (WTM), the total and average power is considered for the unmerged and merged data. If total set of test data contains N vectors $V_1, V_2, V_3, \ldots V_N$ then the total power is obtained by summing the all the vectors 1 to N. The average power is obtained by dividing the total power by total number of test sets. The total power and average power and is calculated as,

$$Power_{total} = \sum_{p=1}^{N_v} \sum_{q=1}^{l-1} (V_{p,q} \oplus V_{p,q+1}) * (l - q) \quad (4)$$

$$Power_{average} = \frac{\sum_{i=1}^{N_v} \sum_{j=1}^{l-1} (V_{p,q} \oplus V_{p,q+1}) * (l-q)}{N} \quad (5)$$

Table-6. Comparison of total and average power.

Circuits	Total Data (T_D)	Unmerged Data		Merged Data	
		Total Power	Average Power	Total Power	Average Power
s9234	39273	629314	3957	615624	3871
s13207	165200	1825366	7734	1576235	6678
s15850	76986	1702737	13513	1726197	13699
s38417	164736	11636595	117541	11710145	118284
s38584	199104	11649193	85655	11521618	84717
Average	--	5488641	45680	5429963	45449

Table-6 represents the comparison between the total power and average power for unmerged and merged data for the circuits for proposed three code technique. The average of average power is reduced to 231 units after compressing the test data leads to get low average power. The total power is also reduced for test data.

Table-7 represents the comparison results of average power for the proposed three code compression technique with existing methods such as Mintest [12], DC_BT [18], FDR [5], EFDR [6], ERLC [19]. The three coding technique gives better average power on comparing to existing methods. The average power achieves up to 83.3 % on comparing with the Mintest test sets.

Table-7. Comparison of average power with existing methods.

CIRCUITS	MINTEST [12]	DC_BT [18]	FDR [5]	EFDR [6]	ERLC [19]	Proposed
s9234	14630	8132	5692	3469	3500	3871
s13207	122031	17809	12416	8016	8115	6678
s15850	90899	24850	20742	13394	13450	13699
s38417	601840	578450	172665	117834	120775	118284
s38584	535875	108050	136634	89138	89356	84717
Average	273055	147458	69630	46370	47039	45449

The test application time (TAT) is described for the proposed three code compression technique to calculate the upper bound for the compressed data as shown in [1, 16]. The testing time is analyzed for output data has a single scan chain. The frequency considered for ATE frequency is clock (clk) frequency and scan frequency is out frequency, where $f_{clk} < f_{out}$. The frequency for the scan out is $f_{out} = \alpha f_{clk}$. The value of α (alpha) is greater than 1 and it is the powers of 2 to synchronize the values. The TAT is calculated for different values of alpha. The upper bound is calculated by considering the T_{shift} time required to shift the compressed test data and T_{decode} time required to decode the compressed test data. So, the TAT for scan out is given as,

$$TAT_{out} \leq T_{shift} + T_{decode} \quad (6)$$

The data is shifting to tester frequency, the time required to shift the compressed data is given as,

$$T_{shift} = \frac{TE}{f_{clk}} \quad (7)$$

where TE is the compressed test data. Similarly to decode the data time is required is given as,

$$T_{decode} = \frac{TD}{f_{out}} \quad (8)$$

Hence, the test application time to find upper bound for the total test data and is given as,

$$TAT_{out} \leq \frac{TE}{f_{clk}} + \frac{TD}{f_{out}} \quad (9)$$

$$TAT_{out} \leq \frac{TE}{f_{clk}} + \frac{TD}{\alpha f_{clk}} \quad (10)$$

Table-8 represents the comparison of test application time for upper bound with other existing methods like Golomb [4], FDR [5] and TRP [1]. Considering the circuit s13207 with α = 16 for FDR and comparing with proposed method gives 28 % of time reduction while testing.

Table-8. Comparison of test application time for upper bound.

Circuits	F_{clk} (MHz)	α	Upper Bound (ms)			PROPOSED
			Golomb[4]	FDR[5]	TRP[1]	
S5378	20	4	1.016	0.914	0.881	0.765
		8	0.868	0.765	0.733	0.616
		16	0.794	0.691	0.658	0.542
S9234	20	4	1.565	1.598	1.571	1.308
		8	1.320	1.362	1.326	1.062
		16	1.197	1.230	1.203	0.940
S13207	20	4	3.738	3.609	3.697	3.033
		8	2.705	2.577	2.664	2.005
		16	2.189	2.060	2.148	1.484
S15850	20	4	2.393	2.262	2.277	1.921
		8	1.912	1.781	1.796	1.440
		16	1.617	1.540	1.555	1.199
S38417	20	4	7.959	6.732	5.308	5.022
		8	6.929	5.702	4.278	3.993
		16	6.414	5.188	3.763	3.478
S38584	20	4	6.752	6.379	6.357	5.169
		8	5.508	5.135	5.113	3.925
		16	4.886	4.512	4.490	3.303

7. CONCLUSIONS

This paper presents three coding compression techniques to lessen the test power, test application time and to improve compression ratio for SoC based design. Three types of symbols are used to encode the precomputed test data to achieve 69.9 % of compression ratio for ISCAS89 benchmark circuits. The test power is reduced through switching activity transition of bit filling with 0's and 1's using WTM. The average power is reduced up to 83.3 % when compared existing methods. The test application time is made to reduce with increasing of compression ratio. The area of decompression cause more cost.

REFERENCES

[1] Vohra H. & Singh A. 2016. Optimal Selective Count Compatible Runlength Encoding for SOC Test Data Compression. Journal of Electronic Testing. 32(6): 735-747.

[2] Karmakar R. & Chattopadhyay S. 2017. Temperature and data size trade-off in dictionary based test data compression. Integration, the VLSI Journal. 57: 20-33.

[3] Tian C. H. E. N., Dandan S. H. E. N., Xin Y. I., Liang H., Xiaoqing W. E. N. & Wei W. A. N. G. 2016. Reseeding-Oriented Test Power Reduction for Linear-Decompression-Based Test Compression Architectures. IEICE TRANSACTIONS on Information and Systems. 99(11): 2672-2681.

[4] Yuan H., Ju Z., Sun X., Guo K. & Wang X. 2016. Test Data Compression for System-on-chip using Flexible Runs-aware PRL Coding. Journal of Electronic Testing. 32(5): 639-647.

[5] Chandra A & Chakrabarty K. 2003. Test data compression and test resource partitioning for system-on-a-chip using frequency-directed run-length (FDR) codes. IEEE Transactions on Computers. 52(8): 1076-1088.

[6] Zorian Y., Marinissen E. J. & Dey S. 1998, October. Testing embedded-core based system chips. In: Test Conference, 1998. Proceedings. International (pp. 130-143). IEEE.

[7] Touba N. A. 2006. Survey of test vector compression techniques. IEEE Design & Test of Computers. 23(4): 294-303.

[8] Sismanoglou P. & Nikolos D. 2011, December. Test data compression based on the reuse of parts of the dictionary entries. In: Electronics, Circuits and Systems (ICECS), 2011 18th IEEE International Conference on (pp. 538-541). IEEE.

[9] Lu S. K. & Huang Y. C. 2012. Improving Reusability of Test Symbols for Test Data Compression. Journal of Information Science and Engineering. 28(2): 351-364.

[10] Yu Y., Yang Z. & Peng X. 2012, May. Test data compression based on variable prefix dual-run-length code. In: Instrumentation and Measurement

Technology Conference (I2MTC), 2012 IEEE International (pp. 2537-2542). IEEE.

[11] Lee L. J., Tseng W. D., Lin R. B. & Chang C. H. 2012. $2^ {n} $ Pattern Run-Length for Test Data Compression. IEEE transactions on computer-aided design of integrated circuits and systems. 31(4): 644-648.

[12] Kavousianos X., Kalligeros E. & Nikolos D. 2007. Optimal selective Huffman coding for test-data compression. IEEE transactions on computers. 56(8).

[13] Gonciari P. T., Al-Hashimi B. M. & Nicolici N. 2003. Variable-length input Huffman coding for system-on-a-chip test. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 22(6): 783-796.

[14] El-Maleh A. H. 2008. Test data compression for system-on-a-chip using extended frequency-directed run-length code. IET Computers & Digital Techniques. 2(3): 155-163.

[15] Chandra A. & Chakrabarty K. 2003. A unified approach to reduce SOC test data volume, scan power and testing time. IEEE transactions on computer-aided design of integrated circuits and systems. 22(3): 352-363.

[16] Mehta U., Dasgupta K. & Devashrayee N. 2011. Suitability of various low-power testing techniques for IP core-based SoC: a survey. VLSI Design, 2011, 4.

[17] Hellebrand S., Liang H. G. & Wunderlich H. J. 2001. A mixed mode BIST scheme based on reseeding of folding counters. Journal of Electronic Testing. 17(3-4): 341-349.

[18] Gopal P. R. & Saravanan S. 2015. Low power estimation on test compression technique for SoC based design. Indian Journal of Science and Technology. 8(14): 1.

[19] Saravanan S. & Upadhyay H. N. 2012. Achieving low power test pattern by efficient compaction method for SoC design. Journal of Artificial Intelligence. 5(4): 244.

[20] Saravanan S, Sowmiya G, Sai RV. 2013 Design and analysis of reduced test power in scan based design. International Journal of Engineering and Technology. 5:692-5.