ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Code Based Compression Techniques in Digital VLSI

Mr.Mohammad Iliyas
Assistant Professor, Dept of ECE and Research Scholar, Nawab Shah Alam Khan College of Engineering and Technology, Hyderabad, India.

Mrs. Farha Anjum
Assistant Professor, Dept of ECE and Research Scholar, Malla Reddy College of Engineering, Hyderabad, India.

Dr.Anil Kumar Sharma
Professor, Dept of ECE, SRU, Alwar, Rajastan, India.

Dr.R.Murali Prasad
Professor Dept of ECE, Vardhaman College of Engineering, Hyderabad, India.

Abstract:
An approach for test pressure is to utilize data pressure codes, for example, measurable coding, Golomb coding and run-length coding to encode the test solid shapes. In these methodologies, the first data is apportioned into images, and afterward every image is relegated with a codeword to frame the encoded data. Each codeword is changed over into the relating image with on-chip decompression equipment. These data pressure codes can be further ordered into four gatherings relying upon whether the extent of the images and code words are settled or variable lengths.

CODE-BASED COMPRESSION:

These are altered contribution to settled yield (FIFO), altered contribution to variable-yield (FIVO), variable-contribution to settled yield (VIFO) and variable-contribution to variable-yield (VIVO) coding strategies. These strategies don't require the basic data about the CUTs and more appropriate for licensed innovation (IP) center construct framework in light of a-chips (SoCs). In FIVO coding, the first test solid shapes are divided into n-bit pieces to frame the images. These images are then encoded utilizing variable-length code words. One type of altered to-variable coding is measurable coding, where the thought is to compute the recurrence of event of the diverse images in the first test solid shapes and make the code words that happen most habitually have less bits and those that happen minimum as often as possible more bits. This minimizes the normal length of a codeword.

A Huffman code portrayed by is an ideal factual code that is demonstrated to give the shortest normal codeword length among all exceptionally decodable altered to-variable length codes. A Huffman code is gotten by developing a Huffman tree. Huffman coding system with altered length of pieces to diminish the test data volume. Nonetheless, it requires complex decoder design to disentangle the huge number of particular squares. That is, full Huffman code that encodes all n-bit images requires a decoder with 2n − 1 states. To address this issue, a particular Huffman code, in which just the k most every now and again happening images are encoded. In specific Huffman coding, an additional piece is included toward the start of each codeword to show regardless of whether it is coded.

The on-chip decoder requires just n + k states since this approach specifically encodes just k images. Additionally, it was demonstrated that a particular Huffman code accomplishes just marginally less pressure than a full Huffman code for a similar image measure while utilizing a much littler decoder. Since the decoder measure becomes just directly with particular Huffman encoding, it is conceivable to utilize a much bigger image estimate, which altogether enhances the adequacy of the code in this manner accomplishing substantially more general pressure. Technique to ad libbing the pressure proportion with a factual code in which the predefined bits in the test blocks are altered in a manner that maximally skews the recurrence dissemination.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Aside from factual coding, there are many settled to-variable coding which abuses the way that most output cuts have a moderately little number of determined bits as depicted. In the event that there are b channels originating from the tester, these methods utilize a variable number of b-bit codeword's to translate the predetermined bits in every output cut. In FIFO coding, the test 3D shapes are divided into n - bit squares to frame the images. These images are then encoded with codeword's that each have b - bits, where b < n: Word reference based pressure methods are case for FIFO coding. In these procedures, every image and codeword separately can be considered as a passage in a lexicon and as a record into the word reference that focuses to the relating image. There are 2n conceivable images and 2b conceivable codeword's, so not really all conceivable images can be in the word reference.

In VIFO coding, the first test blocks are apportioned into variable length images, and the codeword's are each with b-bits long. This VIFO coding technique to encode keeps running of 0s has been talked. The event of keeps running of 0s is expanded by utilizing cyclic sweep design allowing utilization of distinction vector. Mind must be taken while requesting the test solid shape with a specific end goal to augment the quantity of keeps running of 0s in the distinction vector, to enhance the viability of run-length coding. run-length encoding system. In this method, the yield of the output flip-slump is nullified for expanding the run-length of 0s or 1s. For output cells having more than half of bit esteem 1s, Inverters are set in the middle of the sweep cells to get run-length of 0s. They additionally used least move tally (MTC) mapping to decrease the moves amongst bits furthermore to get keeps running of 1s or 0s.

Group	Run-length	Prefix	Tail	Codeword
	0	0	00	000
A_1	1		01	001
	2		10	010
	3		11	011
	4	10	00	1000
A_2	5		01	1001
	6		10	1010
	7		11	1011
	8	110	00	11000
A_3	9		01	11001
	10		10	11010
…	…	…	…..	…..

Table 2.1: Golomb coding scheme for m=4

The test data pressure in light of a run-length coding with a repeating check design was depicted. to improve the viability of the essential run-length coding. In this approach, the present data to be moved is XORed with the past test vector with the proposed patterned sweep design. That is, rather than applying the first test set (TD), a distinction test vector set is connected. The principle favorable position of this plan is that the test vectors can be reordered in a manner that more comparative test vectors come after each other which will expand number of 0s in the distinction vectors. an another kind of factor to-settled code which is based programming based code called LZ77 for test jolt pressure.

In VIVO coding, both the images and codewords have variable length. a variable-to-variable Golomb code for test data pressure. Table 2.1 demonstrates a case of the Golomb coding plan. The codewords are partitioned into gatherings of equivalent size m. The estimation of m is given as m = 2b, where b is the piece measure. The codeword is isolated into two sections: the prefix and the tail. The measure of the gathering prefix is variable while the quantity of bits in the tail is altered. The on-chip decompression equipment for the Golomb code is little bringing about a lower territory overhead.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Group	Run-length of 0s	Prefix	Tail	Codeword
A_1	0	0	0	00
	1		1	01
A_2	2	10	00	1000
	3		01	1001
	4		10	1010
	5		11	1011
A_3	6	110	000	110000
	7		001	110001
	8		010	110010
	9		011	110011
	10		100	110100
	11		101	110101
	12		110	110110
	13		111	110111
...	...	...	...	...

Table 2.2: FDR coding scheme

The encoded test set is composed as far as remainder, multiplier and leftover portion. The utilization of variable length code words allows for productive encoding of longer runs, despite the fact that it requires a synchronization instrument between the tester and the chip. This plan is connected to contrast vector got from the first test vectors. For instance, the encoded grouping relating to 001 00001 0001 00000001 is given as 010 1000 011 1011. This run-length based Golomb code gives wasteful pressure much of the time since every gathering contains a similar number of run-lengths. Later, a recurrence coordinated run-length (FDR) code that has variable-length tails in view of the gathering file. It can be built with the end goal that a shorter run-length can be encoded into a shorter codeword to give better pressure. The FDR code is a variable-to-variable-length code which maps variable-length keeps running of 0s to variable-length code words. In FDR coding, the code words have two sections: the prefix and the tail and they are of equivalent length as appeared in Table 2.2. Accordingly the test data pressure can be more proficient if the keeps running of 0s with shorter run-length are mapped to shorter code words. The FDR is like Golomb code yet the distinction is the variable gathering size.

Group	Run-length of 0s or 1s	Prefix	Tail	Codeword
A_1	0	0	0	00
	1		1	01
A_2	2	10	00	1000
	3		01	1001
	4		10	1010
	5		11	1011
A_3	6	110	000	110000
	7		001	110001
	8		010	110010
	9		011	110011
	10		100	110100
	11		101	110101
	12		110	110110
	13		111	110111
...	...	...	...	...

Table 2.3: ARL coding scheme

For a run-length k, mapping of k is done to a gathering A j where j = [log2(k + 3) − 1]. The FDR code gives an effective test pressure to the test data which has long keeps running of 0s and less 1s. Be that as it may, the test data made out of both keeps running of 1s, so FDR coding was wasteful in accomplishing great pressure. The pressure strategies, for example, rotating FDR (ARL) coding, amplified FDR (EFDR) coding substituting variable-length (AVR) coding rise to run-length coding (ERLC) and changed FDR coding (MFDR) consider both keeps running of 0s and also 1s to shape the code words.

The ARL code is likewise a variable-to-variable length code. Here, the test set T is made out of rotating keeps running of 1s. This coding method considers a substituting paired variable an, and encoding for every run-length is subject to this parameter esteem. In the event that a = 0, the run-length is dealt with as a keep running of 0s. Then again, if a = 1, the run-length is dealt with as a keep running of 1s. The an is transformed after every run is encoded and it continues substituting somewhere around 0 and 1 from there on. The default Introductory estimation of a = 0, that is the information data stream begins with a keep running of 0s.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Group	Run-length	Prefix	Tail	Codeword
A_1	1	0	0	00
	2		1	01
A_2	3	10	00	1000
	4		01	1001
	5		10	1010
	6		11	1011
A_3	7	110	000	110000
	8		001	110001
	9		010	110010
	10		011	110011
	11		100	110100
	12		101	110101
	13		110	110110
	14			

Table 2.4: SAFDR coding scheme

The ARL coding plan is appeared in Table 2.3. Hellebrand and Wusrtenberger (2002) displayed another plan for interchange keeps running of 1s called as the Shifted Alternating Frequency Directed Run-length (SAFDR) coding. This encoding strategy is the enhanced type of the ARL method. Every image is comprised of just 1s or just 0s. The primary piece of encoded data will show the sort of the principal run, that is, 0 or 1, then each codeword will demonstrate the run-length of substitute runs. Since every image has either 0s or 1s alone, there will be no run-length of size 0 and subsequently each codeword is moved to one position higher when contrasted with ARL conspire. Table 2.4 demonstrates the SAFDR coding plan. This aides in accomplishing higher pressure contrasted with ARL coding. Expanded Frequency-Directed Run-length (EFDR) coding was proposed by El-Maleh and Al-Abaji (2002) to ad lib FDR coding. It includes one extra piece in the codeword to code the 1s run-length for boosting the pressure proportion. The FDR code is exceptionally productive for compacting data that has few 1s and long keeps running of 0s yet wasteful for data streams that are made out of both keeps running of 0s and keeps running of 1s. In EFDR, the run of as followed by bit "1" and the keep running of 1s followed by bit "0" are the coded same path as FDR yet including an additional piece toward the start of FDR codeword to demonstrate the kind of run.

Group	Run-length	Prefix	Tail	Codeword	
				runs of 0s	runs of 1s
A_1	0	0	0	000	100
	1		1	001	101
A_2	2	10	00	01000	11000
	3		01	01001	11001
	4		10	01010	11010
	5		11	01011	11011
A_3	6	110	000	0110000	1110000
	7		001	0110001	1110001
	8		010	0110010	1110010
	9		011	0110011	1110011
	10		100	0110100	1110100
	11		101	0110101	1110101
	12		110	0110110	1110110
	13		111	0110111	1110111
.....	.....	.....	.....	.....	.....

Table 2.5: EFDR coding scheme

The EFDR coding plan is appeared in Table 2.5. We can watch that the pressure productivity is most extreme as the run-length increments. For littler run-lengths like 0, 1, 2 or 3, the Golomb, FDR, EFDR furthermore ARL is giving longer codeword than its run-length. Additionally Golomb, FDR, EFDR are for keeps running of 0s (1s) just and if there are keeps running of 1s (0s) then codeword won't be compacted. In the ERLC encoding strategy (Zhan and El-Maleh, 2012), both sorts of runs are viewed as like in EFDR procedure and it is a VIVO coding system. The coding plan has been appeared in Table 2.6. We see that the code for run-length i in ERLC is given to run-length i + 1 in EFDR coding. The novel normal for this approach is that ERLC plot investigates the relationship between two successive runs.

Group	Run-length	Prefix	Tail	Codeword	
				runs of 0s	runs of 1s
A_1	1	0	1	001	101
	2	10	00	01000	11000
A_2	3		01	01001	11001
	4		10	01010	11010
	5		11	01011	11011
	6	110	000	0110000	1110000
A_3	7		001	0110001	1110001
	8		010	0110010	1110010
	9		011	0110011	1110011
	10		100	0110100	1110100
	11		101	0110101	1110101
	12		110	0110110	1110110
	13		111	0110111	1110111
	1	0	1	001	101
.....	.....	.....	.....	.....	.....

Table 2.6: ERLC coding scheme

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

On the off chance that there comparative run-lengths happening continuously, shorter codewords like 000 and 100 are alloted demonstrating the reiteration. This strategy is best valuable when the recurrence of reiteration of codes is high. For getting most extreme move power decrease, the moves must be less. Filling the couldn't care less bits in such an approach to lessen most extreme move will bring about a test design with interchange keeps running of 1s. So in the event that we go for any zero run-length code or a one run-length code the pressure productivity will be less. So it is ideal to pick a pressure plan which packs both keeps running of ones. Likewise from the ERLC encoding strategy we see that by giving a shorter codeword for the monotonous codes, there will be further pressure accomplished.

Symbol	Frequency	Pattern	Huffman	Huffman	Opt. Sel. Huffman
s0	1	0000	11000	00000	10000
s1	1	0001	11001	00001	10001
s2	5	0010	10	110	10
s3	-	0011	-	00011	10011
s4	5	0100	00	10	0
s5	2	0101	111	00101	10101
s6	1	0110	1101	00110	10110
s7	3	0111	010	111	11
s8	2	1000	011	01000	11000
s9	-	1001	-	-	-
s10	-	1010	-	-	-
s11	-	1011	-	-	-
s12	-	1100	-	-	-
s13	-	1101	-	-	-
s14	-	1110	-	-	-
s15	-	1111	-	-	-

Table 2.7: Symbol frequencies and codeword for different Huffman coding

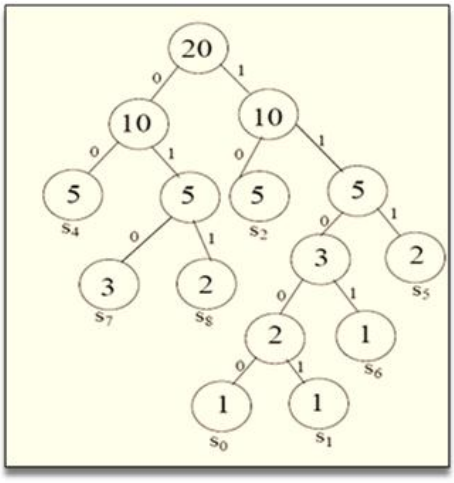

Figure 2.1: Huffman Tree formation

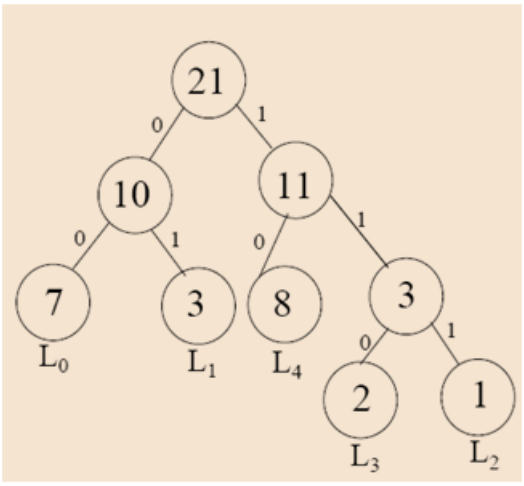

Figure 2.2: VIHC Tree

Table 2.8: Pattern and the codeword for VIHC

Pattern	Occurrence	Codeword
L_0=1	7	00
L_1= 01	3	01
L_2= 001	1	111
L_3= 0001	2	110
L_4= 0000	8	10

Huffman tree for the given test vector is developed as appeared in Figure 2.1. Table 2.7 demonstrates the recurrence of event of the considerable number of images and the relating code words for every image which are gotten from the tree developed. As indicated by the codeword appeared in Table 2.7, the compacted vector is given as 00 1101 10 010 00 10 010 10 011 010 111 00 011 111 11000 11001 which is lessened 55 bits. In Huffman coding, the decoder measure increments exponentially relying upon the image estimate. Gonciari et al. (2003) tended to this issue by the coding plan which utilizes input examples of variable length to the Huffman calculation, called variable-input Huffman coding (VIHC). The VIHC allows a proficient misuse of test sets which display long keeps running of 0's. This alteration in the info design not just impacts the pressure, additionally allows utilizing a parallel decoder utilizing counters.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

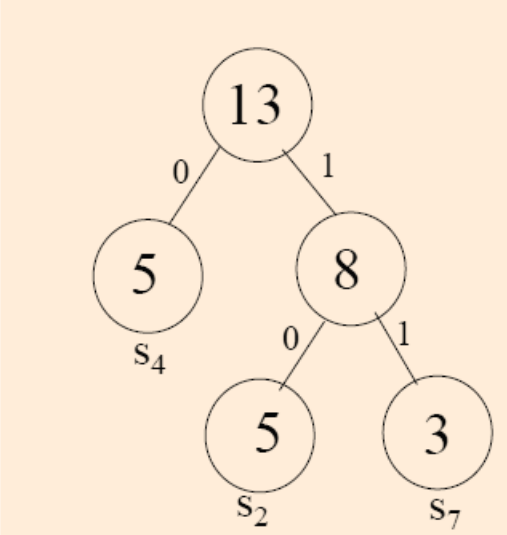

Figure 2.3: Formation of Selective Huffman Tree

This prompts to noteworthy lower range overhead. In VIHC, the gathering size speaks to the greatest worthy number of 0s contained in a keep running of 0s. As per the gathering size the test vectors are parceled into keeps running of 0s of length littler than or equivalent to 4, which are called as examples. These examples are the contribution to the Huffman coding plan, where the quantity of events of the considerable number of examples is resolved and with which the Huffman tree is developed. Table 2.8 demonstrates the examples and the quantity of events of every example and the relating codeword got from the tree developed which is appeared in Figure 2.2. An illustration test vectors is given as 0000 01 0001 1 0000 0001 1 0000 1 01 001 1 0000 1 which is of 56 bits.

The VIHC changes over 56 bits test data into 45 bits encoded data. In particular Huffman code just the most much of the time happening images are encoded and whatever remains of the images are left un-encoded and thus region overhead is decreased further. The particular Huffman tree development is appeared in Figure 2.3, and the test vector is packed to 10 00110 110 111 10 110 111 110 01000 111 00101 10 01000 00101 00000 00001 which is of 69 bits. Kavousianos et al. (2007b) introduce an another ideal system called ideal specific Huffman coding for test data pressure.

This is a superior approach than specific Huffman coding which utilizes extra Huffman codeword just as a part of front of the unencoded data pieces, and the most much of the time happening codewords are alleviated from the additional piece thusly. In this way oftentimes happening codewords with abbreviated codeword overbalances the misfortune from utilizing an additional piece as a part of front of unencoded data squares. As indicated by this, m + 1 Huffman codewords utilized, where m remains for most every now and again happening pieces and 1 remains for the unencoded square. The event recurrence of the codeword is equivalent to the total of the event frequencies of all the unencoded unmistakable pieces. A case test vector is given as 0100 0110 0010 0111 0100 0010 0111 0010 1000 0111 0101 0100 1000 0101 0000 0001 which is of 80 bits is compacted utilizing the codeword appeared as a part of the Table 2.7. This is acquired from the tree which is appeared in Figure 2.3. The test vector is decreased to 0 10110 10 11 0 10 11 10 11000 11 10101 0 11000 10101 10000 10001 which is of 56 bits.

Author's Details:

Mr. Mohammad Iliyas

Received M.Tech. in VLSI SD from JNTUH, and having more than 10 years of experience in both teaching and industry, currently pursuing Ph.D. and working as an Asst. Professor at NSAKCET, Hyderabad.

Mrs. Farha Anjum

Received M.Tech in VLSI SD from JNTUH and having more than 8 years of experience in both teaching and industry, currently pursuing Ph.D. and working as Asst Professor in ECE Dept at MRCE, Hyderabad.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Dr. Anil Kumar Sharma

Received his PhD in 2011 and having more than 30 years of experience in teaching and industry. Currently he is working as Professor in ECE department at SRU, Alwar, Rajasthan.

Dr. R. Murali Prasad

Received his PhD from JNTUA, and having more than 22 years of teaching experience. Currently he is working as Professor at Vardhaman College of Engineering, Hyderabad.