High-Speed Low-Power Approach for Implementation of 8B/10B Encoder for High-Speed Communications

Seyed Moosa Seyed Aalinejad[1*]
1- Department of Electronics, Urmia University of Technology, Urmia, West Azerbaijan, Iran.
Email: seyed_moosa@uut.ac.ir (Corresponding author)

Received: February 2020 Revised: May 2020 Accepted: June 2020

ABSTRACT:
In this paper, the design methodology for a high-speed 8B/10B encoding architecture has been discussed. By means of the new truth table and with the help of Pass-Transistor Logic (PTL), a new structure has been designed in CMOS technology, which shows a superior speed performance. Also, power consumption is optimized because of careful design considerations. These features, along with the simplicity of the employed circuitry are the quality of this work to be repeatedly used in high-speed communication systems. The design process has been explained in detail so that the idea can completely be understood. Moreover, the proposed structure has been demonstrated in the circuit level for better clarification. Post-layout simulation results for TSMC 0.18µm standard CMOS technology depict the correct behavior of the proposed architecture whilst the power consumption is 1.64mW from 1.8v power supply.

KEYWORDS: High-Speed, Low-Power, 8b/10b Coding Scheme, Encoding, CMOS.

1. INTRODUCTION

The rapid growth of CMOS technology has provided the achievement to higher speeds for today's designers. Along with this rise, the development of design techniques smoothes the road for achieving more enhancements in system designs.

One of the challenges on this road is the necessity of lower power dissipation, which lay on the opposite side of speed achievement. As a consequence, it makes the designer consider some trade-offs between these two factors. A good plan to fulfill both of them would be a nice choice for utilization in high-speed, low-power system design.

Clock recovery systems dedicated to telecommunications, constitute one of the demanding structures which need careful design investigations. In order to attain good DC balance, 8b/10b line codes have been employed to prepare enough state changes in these architectures. The circuits which perform the coding and decoding functions are of great importance in modern communication systems.

The literature review demonstrates that the main emphasis in such circuits was put on the reduction of system power. Hence, to obtain a good architecture, the maximum time is spent on the power reduction without affecting the performance by the VLSI engineers. For this purpose, many techniques are used to decrease the power consumption of the design [1].

One of the desired methods for this purpose, which is mostly used in previous works, is clock gating. It is a technique that is employed to control the power consumption of a system by means of digital synchronous circuits. In such systems, the substantial fragment of power dissipation is adjusted by the clock net. By prohibiting the unwanted switching operations, the clock will be disabled in the clock net, and the power dissipations will be reduced considerably [2], [3].

In high-speed communication systems, especially local area networks and computer links, one of the principle concepts for data transmission is the provision of good coding choice along with low-cost transceivers [4]. One of the unique selections for the design of coding schemes in modern data links is the employment of binary codes. Among those binary codes, the one which has no DC component or its DC value is constant (regardless of the data patterns) will be beneficial for fiber optic and electromagnetic wire links [5].

In 1983, Widmer and Franaszek introduced a new coding scheme for high-performance data communications, which has become very popular from its invention time [4]. This algorithm is still continuously used in today's transceivers and in short words and is an inseparable part of transmission protocols [3], [6], [7]. It has also found its applications in other categories such as Serializer/Deserializer design [8], digital signature recognition and etc.

Paper type: Research paper
DOI: https://doi.org/10.29252/mjee.14.3.10
How to cite this paper: S.M. Seyed Aalinejad, **"High-Speed Low-Power Approach for Implementation of 8B/10B Encoder for High-Speed Communications"**, *Majlesi Journal of Electrical Engineering*, Vol.14, No. 3, pp. 81-88, 2020.

In high-speed serial data transmission criterion, the algorithm has found its reputation as an 8b/10b coding scheme. It is because, in the encoding section of the transmitter, the 8-bit input data in parallel mode will be mapped into the 10-bit output stream [9]. A high-speed Serializer will perform the shifting operation of the output bits to the receiver. On the other side, the Deserializer will transmute the captured serial stream into the parallel data. Finally, the decoder converts the data to its original 8-bit string [9].

Reduction of the power consumption in most of the previous designs was the basic factor considered in the design of coding circuitry [3, 10] whilst in [6], the optimization of the code flowing problem has been the center of attention. In this paper, a new high-speed and low-power architecture for the encoding of the 8-bit parallel data to 10-bit outputs is presented, which can be widely used in the high-speed communication systems.

In section 2, the main idea of 8b/10b coding is briefly described. Section 3 is about the design of new architecture. The simulation results, along with the comparative analysis, will be discussed in section 4. Finally, the conclusions are summarized in Section 5.

2. THE 8B/10B CODING

The idiom associated with the line code, which transforms eight bits of data into a stream of 10 bits, has been acknowledged as 8b/10b code in telecommunications. The aim of such coding is to achieve acceptable DC balance along with bounded disparity, which is widely employed in lock recovery systems [9].

The interpretation of such a definition suggests that the difference between the enumerations of ones and zeros in a data stream should never exceed a certain value. This value is two for at least 20 bits of data. Hence, the maximum number of ones and zeros in a row will not overstep the value of five. The main advantage of such coding is the decrement of request for the lower limit of the bandwidth needed for signal transfer.

Hardware realization of the 8b/10b coding scheme could be done by different methods. Depending on the concentration on the design of specific parameters such as hardware requirements, DC balance, and etc., various procedures have been introduced. After the publication of IBM implementation in 1983 by Al Widmer and Peter Franaszek [4], one of the implementations was introduced by K. Odaka for the DAT digital audio recorder in 1984 [11]. Later, Franaszek and Widmer have submitted their proposed architecture [12], and after that, Schouhamer Immink presented an 8b/10b code design for the DCC audio recorder in 1986 [13].

There are also other works over recent years for performance enhancement of the 8b/10b coding scheme [14], [15], [16]. Although there is plenty of improvement in recent works, a comprehensive analysis depicts that the designed structures can further be enhanced by the reconsideration of the conventional truth table.

As the general definition of the 8b/10b scheme indicates and also, as is illustrated in Fig. 1, eight bits of data are transmuted to a 10-bit stream named a symbol or character [4].

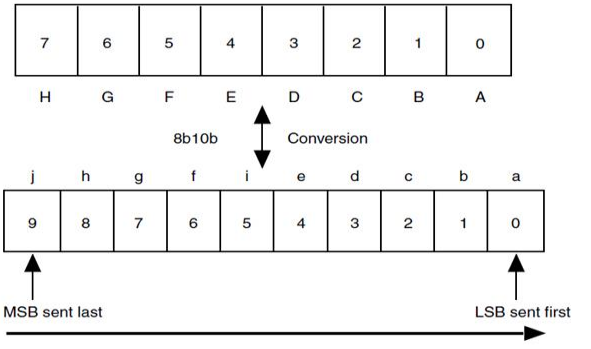

Fig. 1. General representation of 8b/10b coding.

The eight input bits are denoted as A, B, C, D, E, F, G, H. Bit A constitutes the Least Significant Bit (LSB) while bit H defines the Most Significant Bit (MSB). These bits will be divided into two groups:

1- The five-bit group A, B, C, D, E.
2- The three-bit group F, G, H.

The coded bits are known as a, b, c, d, e, i, f, g, h, j in where the order is not alphabetical. These bits are also classified into two groups:

1- The six-bit group a, b, c, d, e, i.
2- The four-bit group f, g, h, j.

The five bits of data in low order of significance are encrypted into a 6-bit group to form the 5b/6b section. On the other hand, the top three bits will be transformed into a 4-bit group to constitute the 3b/4b part.

These code groups are concatenated together to compose the 10-bit symbol that is transmitted on the data link. The data symbols are represented as D.x.y, where x varies from 0 to 31 and y, differs from 0 to 7. It is because the x value corresponds to the five-bit group and the value of y to the three-bit group [4].

Due to the utilization of 10-bit symbols for the encoding of 8-bit data, some of the 1024 codes can likely be eliminated. As a result, a run-length limit of 5 consecutive identical bits can be founded in which the difference between the count of zeros and ones is not more than two. On the other hand, some of the 256 possible 8-bit words can be encoded in two different ways.

With the help of such alternative encodings, the

architecture will have the priority to attain long-term DC-balance in the serial data stream. This feature provides the possibility for the data stream to be transmitted through a channel with a high-pass characteristic environment.

One of the important terms in 8b/10b coding is the Running Disparity (RD), which originates from the difference between the number of ones and zeros in the encoded word. By considering the DC-free nature of the algorithm, such idiom means that the ratio of transmitted ones and zeros is exactly 50%. As a consequence, the difference value is always limited to ±2. Moreover, at the end of each symbol, the difference is either +1 or −1, which is illustrated in Table 1 [17].

Table 1. The associated rules for running disparity.

Previous RD	Disparity of code word	Disparity chosen	Next RD
−1	0	0	−1
−1	±2	+2	+1
+1	0	0	+1
+1	±2	-2	−1

To summarize the definitions pertaining to RD, the following states should be considered:

1- Neutral disparity demonstrates that the number of ones and zeros are identical.
2- The positive disparity shows that we are dealing with more number of ones than the zeros.
3- Negative disparity as the opposite state of positive disparity.

In summary, the 5b/6b and 3b/4b codes will be paired with disparity codes.

Unlike polynomial codes that affect data arrays, this coding operates independently on each byte of the desired data. It guarantees that for every 10 bit of the output data, we have 5 zero bits along with 5 one bits or 4 zero bits and 6 one bits or vice versa. Therefore, we can have a balanced code in output for equal zero and one bits, and by changing the logic value of a bit, the unbalanced code would be achieved. As a result, the unbalanced code can be divided into two groups: 4 zeros and 6 ones or 6 zeros and 4 ones.

Considering the state of the output, the term CRD is being defined, which shows that the last unbalanced code is positive (one) or negative (zero). For the state of 6 ones, the CRD bit becomes logically "1" and for 6 zeros, it will be considered negative. If we assume the positive value for CRD bit, then the output in the other byte will select code with the negative unbalanced state so that the CRD would be removed. Using this mechanism, we will be sure that the transmitted zeros and ones over the fixed period of time will be constant. This means the zero DC value along with the data, which is an advantage of this method. Also, the 0/1/0 conversions will be minimized, which enhances the clock recovery from the data on the transmitter side.

3. THE DESIGNED ARCHITECTURE

To implement the new architecture, just like the previous designs, it is considered that 8b/10b codes will be divided into two smaller sub-blocks during coding and decoding. The division can be done in two forms: Two 4b/5b blocks or a combination of 3b/4b and 5b/6b blocks. The most important point is that the data must be balanced.

To implement the desired architecture, the second form is employed, which is the selected one for previous designs and is shown in Fig. 2.

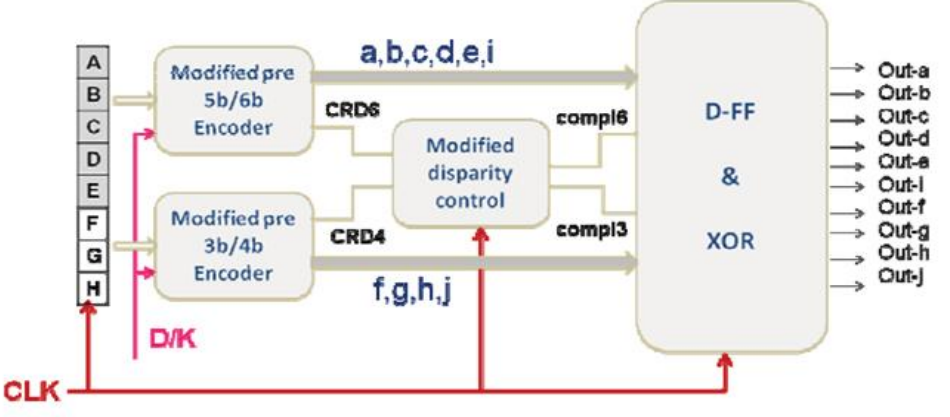

Fig. 2. General scheme for 8b/10b coding.

To achieve high-speed, low-noise and low-power features all together, the truth table has been divided into two separate tables, one for 3b/4b coding and the other for 5b/6b coding. The designed truth table for 3b/4b conversion is illustrated in Table 2.

Table 2. Designed truth table for 3b/4b conversion.

k	H	Adder (F-G)		f	g	h	j
x	0	0		F	1	H	0
x	0	1		F	G	H	1
x	0	2		F	G	H	0
x	1	0		F	G	H	0
x	1	1		F	G	H	0
0	1	2		F	G	H	0
1	1	2		0	G	H	1

According to the truth table, the state of h is equal to the input H. Therefore, simple wiring is needed to connect them to each other, as shown in Fig. 3.

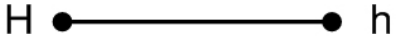

Fig. 3. The relation between H and h.

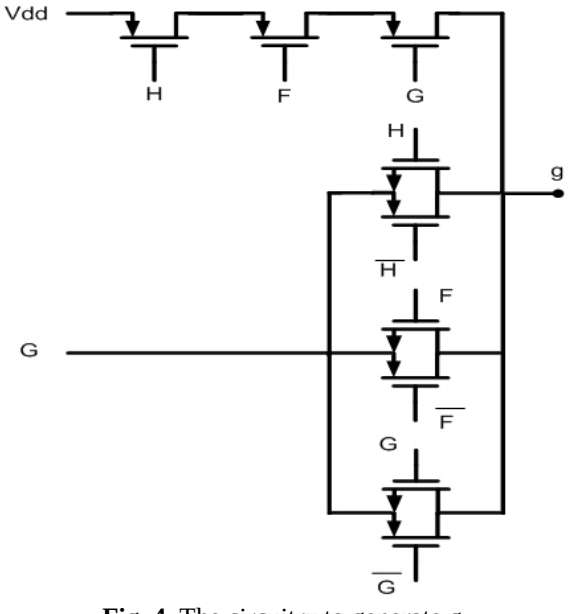

Fig. 4. The circuitry to generate g.

The command k shows that ABCDE is either a control data or a control character. According to the designed truth table, the output g can be defined by the following Boolean relation:

$$g = G + (\bar{H} \cdot \bar{F} \cdot \bar{G}) \quad (1)$$

By means of Eq. (1), the circuitry to obtain g has been illustrated in Fig. 4.

For f and j, we can easily obtain the generating functions as follows:

$$f = (\overline{H \cdot F \cdot G \cdot K}) \cdot F \quad (2)$$

$$j = (H \cdot F \cdot G \cdot k) + (\bar{H} \cdot (\bar{F} \cdot G + F \cdot \bar{G})) \quad (3)$$

In Fig. 5, logic implementation of f and j, is illustrated.

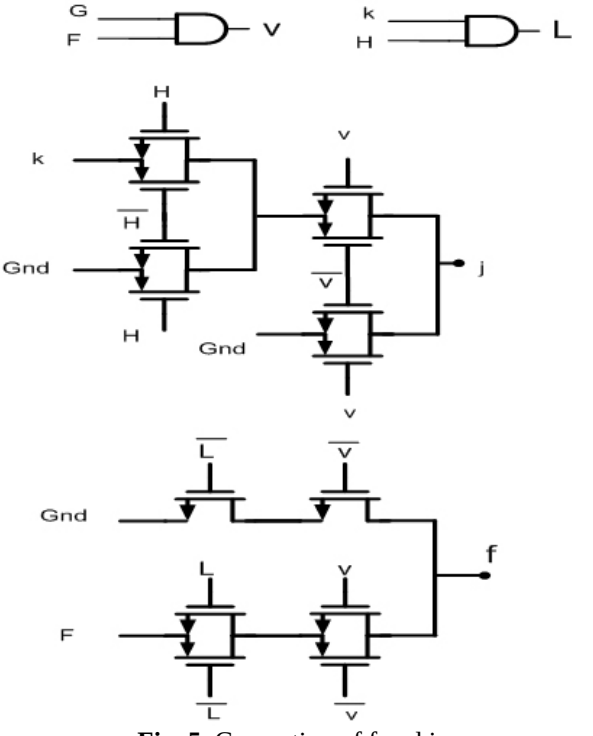

Fig. 5. Generation of f and j.

For the truth table of 5b/6b coding, all 32 possible states for ABCDE inputs are shown in Table 3.

It must be mentioned that in both of the proposed truth tables, the term Adder has been used to indicate the number of input bits where they contain the high logical value.

Again, the command k shows that ABCDE is either a control data or a control character. For normal data, k is equal to zero, while for special character coding, k must be equal to one.

According to the designed truth table and similar to the relation between H and h, the output a is always equal to input A which is shown in Fig. 6.

For the circuit-level realization of the other bits, a 4-2 compressor block is needed. The general architecture of this block is shown in Fig. 7. Although there are five input bits for this structure (A, B, C, D, and C_{in}), for the implementation of the proposed circuit, it is assumed that $C_{\text{in}} = 0$. The three outputs S_1, $Carry$, and C_{out} are utilized to obtain the rest of the

output bits (b, c, d, e, i).

Table 3. Designed truth table for 5b/6b conversion.

k	E	Adder (A-D)		a	b	c	d	e	i
x	0	0		A	1	1	D	0	0
x	0	1		A	B	C	D	1	0
x	0	2		A	B	C	D	0	1
x	0	3		A	B	C	D	0	0
x	0	4		A	0	C	0	0	0
x	1	0		A	1	1	D	1	1
x	1	1…D=0		A	B	C	D	1	1
x	1	1…D=1		A	B	1	D	0	0
1	1	2		A	B	C	D	1	1
0	1	2		A	B	C	D	1	0
x	1	3		A	B	C	0	1	0
x	1	4		A	0	C	0	1	1

Fig. 6. The relation between A and a.

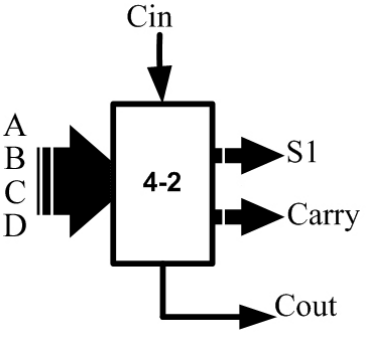

Fig. 7. Block Diagram of a 4-2 Compressor.

The truth table of the compressor is illustrated in Table 4. According to the truth table, if there are the odd number of ones in the inputs, then $S_1 = 1$. for 2 or more ones in the inputs, we have $Carry$ =1 and, if all of the inputs are in the high logic state, then $C_{\text{out}} = 1$.

To implement the compressor and in order to achieve higher speed performance, one of the high-speed architectures reported in [18] or [19] which have less than 2 XOR logic gate-level delays has to be selected.

Table 4. Truth table of 4-2 Compressor.

Adder (A-D)	S_1	Carry	C_{out}
0	0	0	0
1	1	0	0
2	0	1	0
3	1	1	0
4	0	1	1

For this purpose, the 4-2 compressor architecture of [18], which uses Pass-Transistor Logic (PTL) and depicts better speed performance over the circuit of [19], has been employed.

According to the proposed truth table for 5b/6b coding and by means of 4-2 compressor truth table, we will obtain:

$$b = \bar{S_1}.\overline{C_{out}} + B.((S_1 + Cout).\overline{Carry}) \quad (4)$$

$$c = \bar{S_1}.\overline{C_{out}} + C + \bar{C}.D \quad (5)$$

$$d = (\overline{E.S_1.C_{out}}).(\overline{Carry}.D) \quad (6)$$

$$e = E.\left(\overline{D.S_1.\overline{C_{out}}}\right) + \bar{E}.(S_1 * \overline{C_{out}}) \quad (7)$$

$$i = \bar{E}.(\bar{S_1}.C_{\text{out}}.\overline{Carry}) + E.(Carry + k.\bar{S_1}.C_{\text{out}}.\overline{Carry} + S_1.\overline{C_{\text{out}}}.\bar{D}) \quad (8)$$

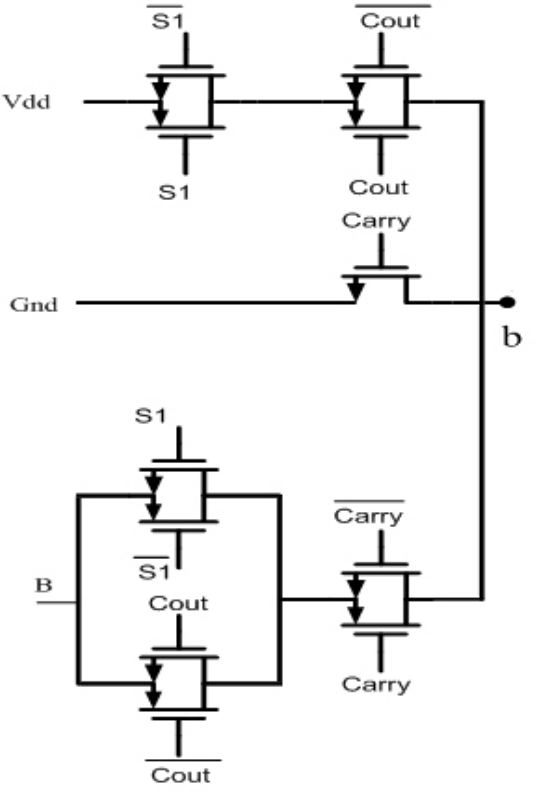

Fig. 8. Proposed circuit for the production of code b.

With the help of the expressions above, the corresponding circuit for the generation of each bit can easily be designed. The scheme pertaining to code b has been illustrated in Fig. 8.

Fig. 9 shows the scheme for the generation of code c, while in Fig. 10, the architecture which produces code d has been illustrated.

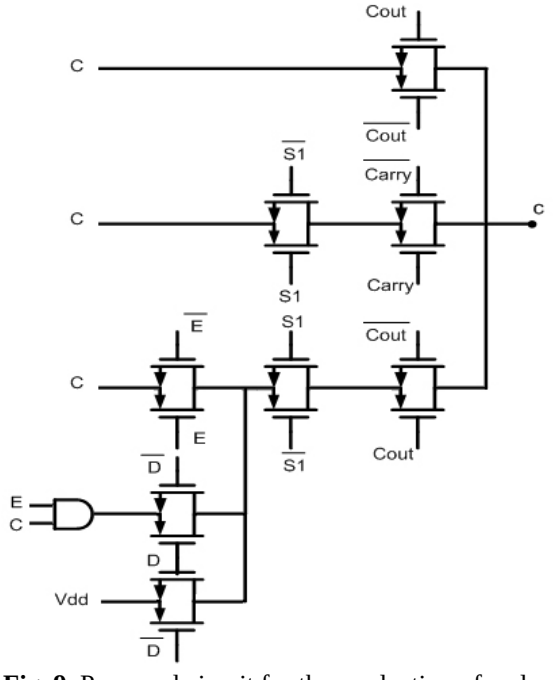

Fig. 9. Proposed circuit for the production of code c.

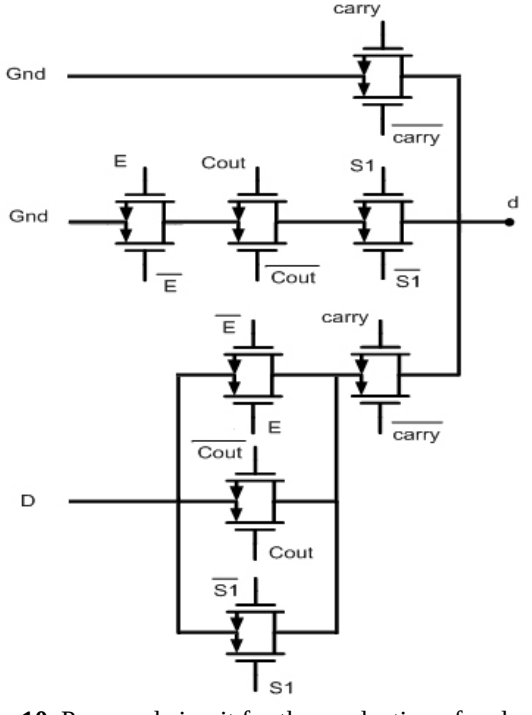

Fig. 10. Proposed circuit for the production of code d.

The proposed circuits pertaining to codes e and i are also demonstrated in Fig. 11, and Fig. 12, respectively.

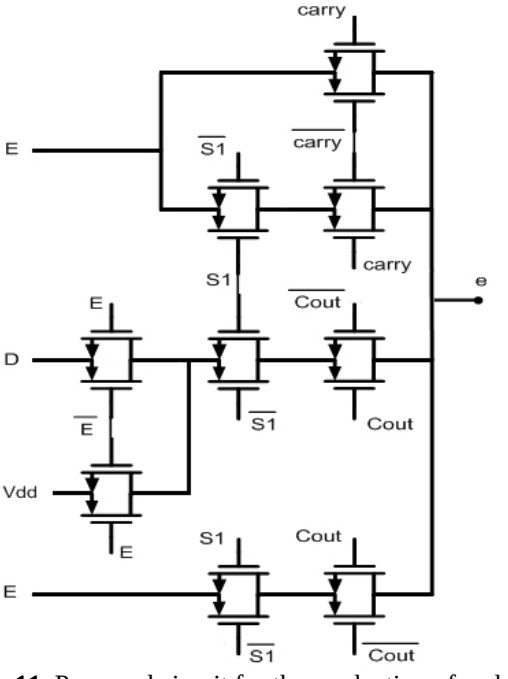

Fig. 11. Proposed circuit for the production of code e.

Investigation of the proposed circuits depicts that the critical path belongs to the generation of code i, where the start point is the input bits of the 4-2 compressor block.

It is then continued by injecting the $Carry$ output to the path which is fed by control signal k. The signal follows its path to the output node by passing through another Transmission Gate (TG), which is controlled by C_{out} signal. Therefore, the propagation latency will be equal to 2.5 XOR logic gates, which is a significant speed improvement compared to the other works.

Also, the total transistor count for the proposed coding scheme is 231, which is substantially smaller than the previously reported works. Because PTL has been utilized for the implementation of the circuits, the Multiplexer (MUX) can be considered as the basis for the gate count. Fig. 13 shows the general circuit-level implementation of a MUX, which is composed of two TGs in parallel.

As Fig. 13 suggests, a combination of 6 transistors can constitute the MUX gate (an inverter is needed to generate the complement state for S). However, in the proposed architecture, there are some gates that are controlled by the same signals. By considering this, the total gate count for the 8b/10b coding scheme will almost be 48. Table 5 summarizes the design specifications for this work.

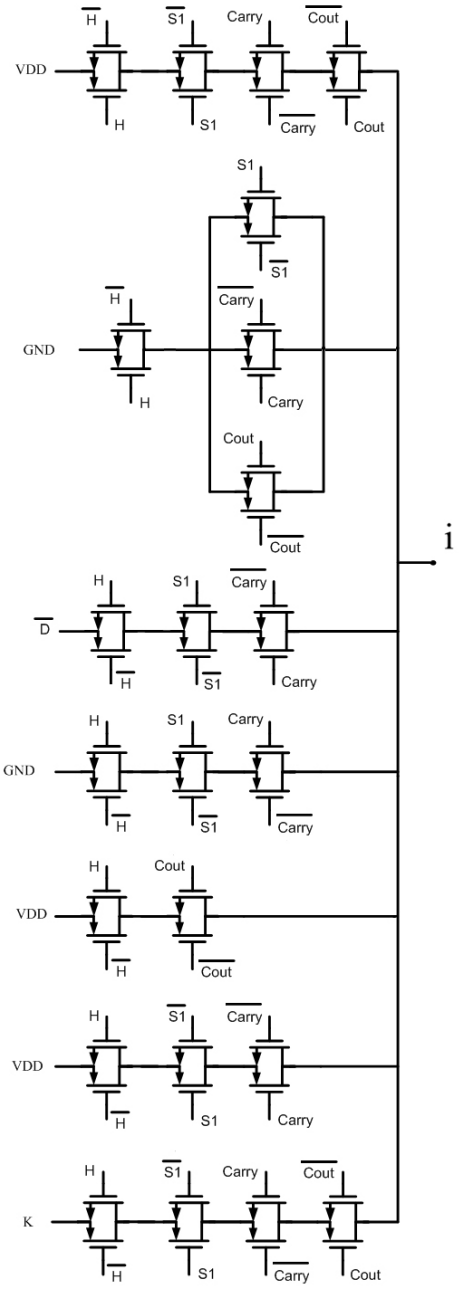

Fig. 12. Proposed circuit for the production of code i.

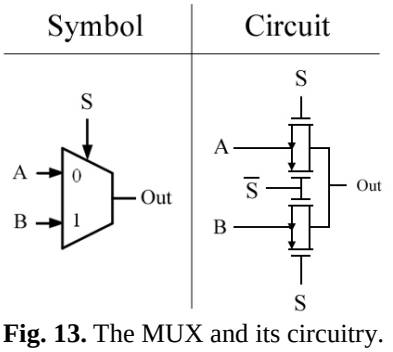

Fig. 13. The MUX and its circuitry.

Table 5. Design specifications for the proposed 8b/10b coding scheme.

Technology	CMOS
Transistor Count	231
Gate-Level Delay (XOR)	2.5 gates
Gate Count	48
Active Area	70µm×30µm

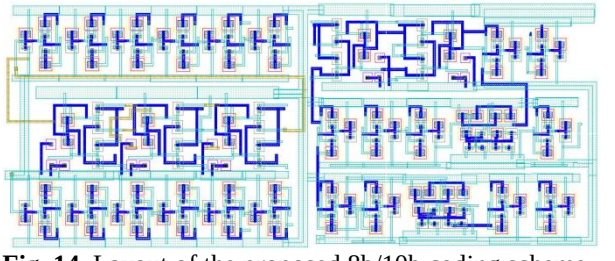

Fig. 14. Layout of the proposed 8b/10b coding scheme.

4. SIMULATION RESULTS AND COMPARISON

To obtain more realistic results, the layout of the proposed scheme has been drawn using TSMC 0.18µm CMOS technology, and the parasitics were extracted in order to be included in the simulation environment. Fig. 14 demonstrates the layout of the designed coding scheme, which occupies an active area of 70µm×30µm on the chip.

Moreover, for a fair comparison between this work and previous architectures, the reported structure of [4] along with newly designed ones discussed in [14], [15] and [16] were redesigned in TSMC 0.18µm standard CMOS technology. Fig. 15 illustrates post-layout simulation results for power and delay comparison (obtained by HSPICE). As the results indicate, the superiority of the proposed scheme is obvious over previous architectures.

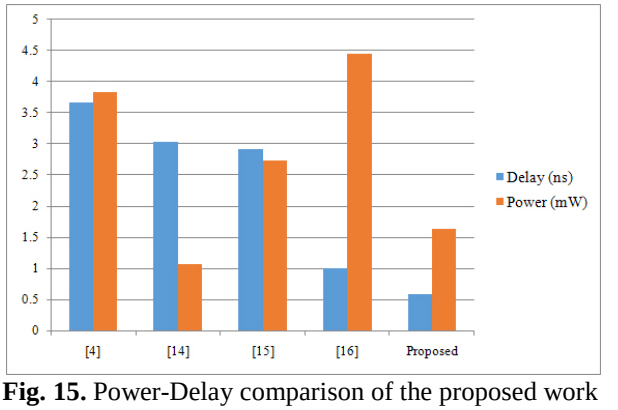

Fig. 15. Power-Delay comparison of the proposed work and previous designs based on TSMC 0.18µm CMOS process.

The only comparable factor of previous works with the designed scheme is the power dissipation of [14]. However, the lower operating frequency and higher active area consumption of this work (due to the number of employed gates), drastically degrades its performance as a competitor.

Finally, Table 6 summarizes the comparative analysis results. It must be mentioned that the operating frequency is defined as the reverse of the value obtained for the critical path delay in simulation results. Since the delay of the proposed scheme was equal to 0.58ns, it can successfully operate at the frequency of 1.7GHz.

Table 6. Comparison between the proposed scheme and previous designs.

	Latency (Clocks)	**Frequency (MHz)**	**Power (mW)**	**Gate Count**	**Technology (µm)**
[4]	0	273	3.83	185	0.18
[14]	5	330	1.06	331	0.18
[15]	0	343	2.74	189	0.18
[16]	0	1000	4.45	---	0.18
Proposed	0	1700	1.64	48	0.18

5. CONCLUSIONS

A novel high-speed architecture has been presented in this paper, which can be widely employed in modern telecommunications. As explained in detail in previous sections, a new truth table was introduced for the 8b/10b encoding scheme, which resulted in the implementation of a low-power high-performance structure. Moreover, the chip size will be small due to the low transistor count of the whole architecture. The aforementioned features, along with the simplicity of the employed circuitry prove the quality of this work to be repeatedly used in high-speed communication systems. Post-layout simulation results for TSMC 0.18µm standard CMOS technology depict the correct behavior of the proposed architecture whilst the power consumption is 1.64mW from 1.8v power supply.

REFERENCES

[1] K. Sahni, K. Rawat, S. Pandey, and Z. Ahmad, "**Low power approach for implementation of 8B/10B encoder and 10B/8B decoder used for high speed communication,"** *2nd International Conference on Emerging Technology Trends in Electronics, Communication and Networking*, Surat, India, 2014.

[2] J. Kathuria, M. Ayoubkhan, and A. Noor, **"A Review of Clock Gating Techniques,"** *MIT International Journal of Electronics and Communication Engineering*, Vol. 1, No. 2, pp. 106-114, 2011.

[3] K. Sahni, K. Rawat, and S. Pandey, **"Power optimization of 8b/10b encoder decoder used for high speed communication,"** *9th International Conference on Industrial and Information Systems (ICIIS)*, Gwalior, India, 2014.

[4] A.X. Widmer, and P.A. Franaszek, **"A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code,"** *IBM Journal of Research and Development*, Vol. 27, No. 5, pp. 440-451, 1983.

[5] Y. Takasaki, M. Tanaka, N. Maeda, K. Yamashita, and K. Nagano, **"Optical Pulse Formats for Fiber Optic Digital Communications,"** *IEEE Transactions on Communications*, Vol. 24, No. 4, pp. 404-43, 1976.

[6] X. Qiaoyu, and L. Huijie, **"8b/10b Encoder Design,"** *The 2nd International Conference on Computer Application and System Modeling*, Paris, France, 2012.

[7] Z. Cai, J. Hao, P.H. Tan, S. Sun, and P.S. Chin, **"Efficient encoding of IEEE 802.11n LDPC codes,"** *Electronics Letters*, Vol. 42, No. 25, pp. 1471-1472, 2006.

[8] M. Maadi, **"An 8b/10b Encoding Serializer/Deserializer (SerDes) Circuit for High Speed Communication Applications Using a DC Balanced, Partitioned-Block, 8b/10b Transmission Code,"** *International Journal of Electronics and Electrical Engineering*, Vol. 3, No. 2, pp. 144-148, 2015.

[9] **"8b/10b Encoder/Decoder,"** *Lattice Semiconductor Reference Design RD1012*, 2015.

[10] G.S. Solanki, R. Agarwal, and S. Sharma, **"Power Optimization of High Speed Pipelined 8B/10B Encoder,"** *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, Vol. 3, No. 7, pp. 14-16, 2013.

[11] K. Odaka, **"Method and apparatus for encoding binary data,"** *U.S. Patent 4456905*, 1984.

[12] P.A. Franaszek, and A.X. Widmer, **"Byte oriented DC balanced (0,4) 8B/10B partitioned block transmission code,"** *U.S. Patent 4486739*, 1984.

[13] K.A.S. Immink, **"Method of transmitting information, encoding device for use in the method, and decoding device for use in the method,"** *U.S. Patent 4620311*, 1986.

[14] **"8B/10B Encoder v5.0"**, *Xilinx, Logicore*, 2004.

[15] Y. Kim, and J. Kang, **"An 8B/10B encoder with a modified coding table,"** *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, 2008.

[16] Q. Wang, S. Hua, and D. Wang, **"A 1.1 GHz 8B/10B encoder and decoder design,"** *Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia)*, 2010.

[17] J. Thatcher, **"Thoughts on Gigabit Ethernet Physical,"** *IBM*, Retrieved 2008-08-17, 1996.

[18] A. Fathi, S. Azizian, K. Hadidi, and A., Khoei, **"A Novel and Very Fast 4-2 Compressor for High Speed Arithmetic Operations,"** *IEICE Transactions on Electronics*, Vol. 95, No. 4, pp. 710–712, 2012.

[19] A. Fathi, S. Azizian, K. Hadidi, A. Khoei, and A. Chegeni, **"CMOS Implementation of a Fast 4-2 Compressor for Parallel Accumulations,"** *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1476–1479, 2012.