A 6.25 Gbps CMOS 10 B/8 B decoder with pipelined architecture*

Zhang Xiaowei(张小伟) and Hu Qingsheng(胡庆生)†

Institute of RF- & OE-ICs, Southeast University, Nanjing 210096, China

Abstract: A fully pipelined 10 B/8 B decoder is presented with shorter critical path than before, and so its speed is improved greatly. Based on the proposed architecture, a 10 B/8 B decoder is implemented based on standard cells in 0.18 μm CMOS technology with a core area of 375 × 375 μm^2. Measurement results show that the decoder works well and its speed can be up to 6.25 Gbps. At a 1.8 V power supply, the total power consumption is 21.6 mW during 6.25 Gbps operation and the peak-to-peak jitter in the eye diagram is 177.8 ps.

Key words: SerDes; 10 B/8 B decoder; pipelined; high-speed
DOI: 10.1088/1674-4926/32/4/045009 **EEACC:** 2750D

1. Introduction

High speed serializer/deserializers (SerDes) are now more and more widely used in communication systems for serial interconnections. A basic SerDes is made up of two functional blocks: the parallel-to-serial converter and the serial-to-parallel converter. In order to transfer data over a high-speed serial interface, data are encoded before transmission and decoded upon reception[2]. As a line code, 8 B/10 B encoder maps 8-bit bytes to 10-bit symbols to achieve DC-balance and bounded disparity, and provides enough state changes to allow reasonable clock recovery at the same time[1].

In this paper, a high speed 10 B/8 B decoder is designed for a 6.25 Gbps SerDes in a 0.18 μm CMOS process. Using the conventional decoder architecture proposed by IBM[1], however, the speed is limited because the logic depth is higher[4]. To overcome this drawback, pipelined architecture, logic optimization and other methods are investigated for ASIC or FPGA implementation[3, 4]. In this paper, a pipelined decoder to reduce the delay of the critical path is proposed, and as a result the operating frequency of the 10 B/8 B decoder is increased greatly.

2. 10 B/8 B decoder detailed operation

2.1. Conventional 10 B/8 B decoder architecture

Based on the encoding table in Ref. [1], in the encoder, 8-bit data are encoded into 10-bit that contain an equal number of 0 s and 1 s. In addition, the code is built so that no more than five consecutive 0 s and 1 s are ever transmitted. Some of the individual 10-bit symbols have an equal number of 1 s and 0 s, while others have either four 1 s and six 0 s, or six 1 s and four 0 s. In the latter case, the disparity between 1 s and 0 s is used as an input to the next 10-bit symbols to reverse the disparity and maintain an overall balanced stream.

A 10 B/8 B decoder decodes 10-bit symbols into 8-bit and an accompanying bit K. The decoder contains four main blocks[7]: 6 B/5 B and 4 B/3 B decoder, error checking block and disparity generation block, as shown in Fig. 1. The input 10-bit symbols dataIn[9:0] are *abcdeifghj*, where a is the LSB and j is the MSB. The decoder works by decoding the bits *abcdei* into the digits *ABCDE* in a 6 B/5 B decoding block and the bits *fghj* into *FGH* in a 4 B/3 B block. Finally, *ABCDE* and *FGH* compose output data *ABCDEFGH* (dataOut[7:0]), where A is the lowest and H the highest. As to bit K, it is decoded as other digits *ABCDEFGH* in the design.

In the decoder, two other blocks for error checking and disparity generation are used to detect the errors coming from the input codes and give some error flags.

Similar to the encoding scheme, Tables 1 and 2 give part of the 6 B/5 B and 4 B/3 B decoding table[1], respectively, where a–j are the encoded 10-bit input and A–H are the 8-bit decoding output. For decoding, the i- and j-bit are dropped and some of the remaining bits are complemented as indicated by bold 0 and 1 entries[1].

From Table 1, the decoding expression[3] of A(dataOut[0]) can be obtained, and the corresponding schematic diagram is shown in Fig. 2[4, 5].

Let *aeqb* = (*a* & *b*) | (!*a* & !*b*),

ceqd = (*c* & *d*) | (!*c* & !*d*).

Then we can get

*p*22 = (*a* & *b* & !*c* & !*d*) | (*c* &*d*& !*a*& !*b*) | (!*aeqb* & !*ceqd*),

*p*13 =(!*aeqb* & !*c* & !*d*) | (!*ceqd* & !*a* & !*b*),

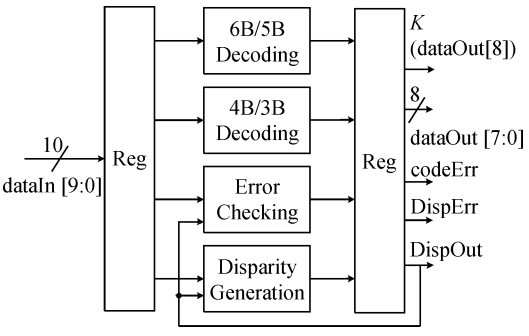

Fig. 1. Block diagram of a conventional 10 B/8 B decoder.

* Project supported by the National High Technology Research and Development Program of China (No. 2006AA01Z239).
† Corresponding author. Email: qshu@seu.edu.cn
Received 3 July 2010, revised manuscript received 2 December 2010

Table 1. 6 B/5 B decoding.

Name	*abcdei*	*ABCDE*	*K*
D.0	011000	**00**000	0
D.0	100111	**00**000	0
D.1	100010	1000**0**	0
D.1	011101	**1000**0	0
D/K.23	111010	11101	X
D/K.23	000101	**11101**	X
K.28	001111	00111	1
K.28	110000	**00111**	1
D.31	101011	1**1**1**1**1	0
D.31	010100	**1**1**1**1**1**	0

Table 2. 4 B/3 B decoding.

Name	*fghj*	*FGH*	*K*
D/K.x.0	0100	0**0**0	x
D/K.x.0	1011	**0**0**0**	x
D/K.x.1	1001	100	x
K.28.1	0110	**100**	1
D/K.x.3	1100	110	x
D/K.x.3	0011	**110**	x
D.x.7	1110	111	0
D.x.7	0001	**111**	0

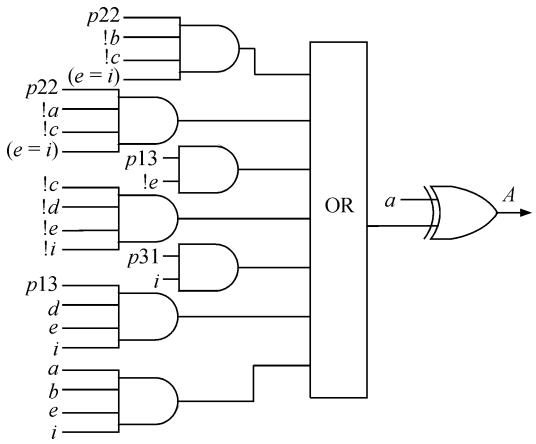

Fig. 2. Decoding schematic diagram of *A* (dataOut[0]).

$p31 = (\ !aeqb\ \&\ c\ \&\ d) \mid (\ !ceqd\ \&\ a\ \&\ b)$.

The decoding logic for *BCDE* and *FGH* is similar to that of *A*. It can be seen that the decoding logic consists of a huge combinational logic that has an important effect on the decoder's speed.

During transmission, code errors and running disparity errors may be caused by noise or other impairments[1]. As to the error checking, it must detect two kinds of error: illegal characters violating the encoding rule and disparity errors violating the polarity rule. The violations of coding rules are[1]:

$a = b = c = d$,

$P13\ \&\ !e\ \&\ !i$, $P31\ \&\ e\ \&\ i$,

$f = g = h = j, e = i = f = g = h$,

$i \neq e = g = h = j$,

$(e = i \neq g = h = j)\ \&\ !\,(c = d = e)$,

$!P31\ \&\ e\ \&\ !i\ \&\ !g\ \&\ !h\ \&\ !j$,

$!P13\ \&\ !e\ \&\ i\ \&\ g\ \&\ h\ \&\ j$.

The disparity error checking includes[1] (1) disparity of all 6 B and 4 B sub-blocks must be 0, +2, or –2; (2) nonzero disparity blocks must alternate in polarity; and (3) D.7 and D/K.x.3 must follow the disparity rules.

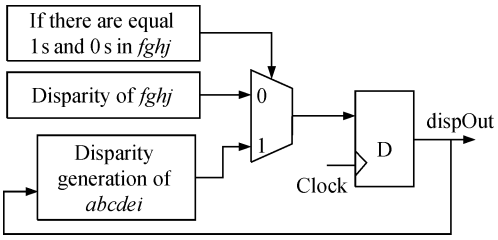

Fig. 3. Disparity generation block diagram.

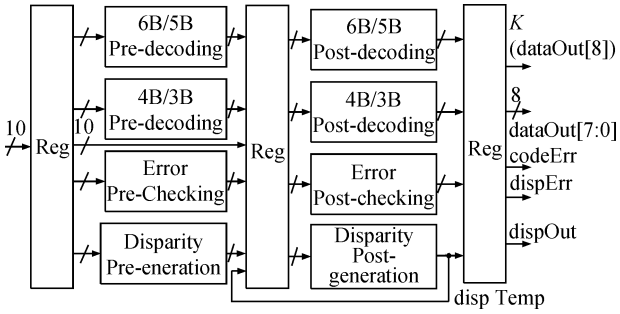

Fig. 4. Block diagram of the proposed decoder.

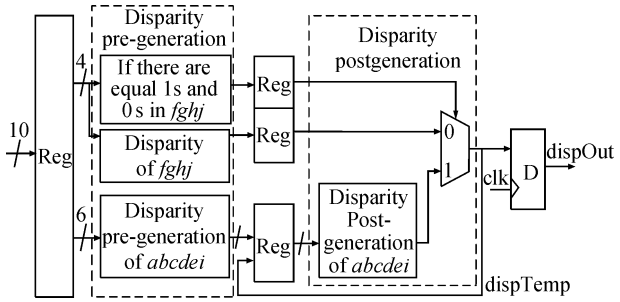

Fig. 5. Pipelined disparity generation block.

To detect the above errors, a disparity generation block is needed to compute the current disparity and feed back the result for the next disparity generation and error checking (see Fig. 3). If there are equal 1 s and 0 s in *abcdei*, the previous disparity output will be sent to the multiplex, and otherwise the disparity of *abcdei* will be sent to the multiplex. Also, if there are equal 1 s and 0 s in *fghj*, the disparity of *abcdei* will be sent out directly, otherwise the disparity of *fghj* will be sent out as dispOut[6].

This conventional architecture is simple for implementation; however, it cannot work at high speeds, such as 6.25 Gbps. For example, from the synthesis result we know that the delay of the critical path is greater than 1.65 ns, which is far more than that desired. This is because the larger combinational circuits, e.g. 6 B/5 B block, 4 B/3 B block, and disparity generation block limit the speed.

2.2. Pipelined 10 B/8 B decoder architecture

To overcome the speed limitation of conventional architecture, an improved 10 B/8 B decoder is developed. Shown as

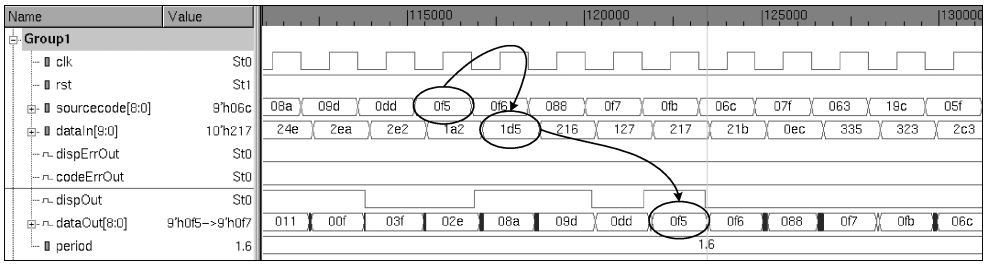

Fig. 6. Gate-level simulation results after placement and route.

Fig. 4, the decoder consists of two stages: the pre-decoding and the post-decoding. For the first stage, it contains 6 B/5 B and 4 B/3 B pre-decoding, error pre-checking and disparity pre-generation blocks. In the second stage, post-decoding blocks of 6 B/5 B and 4 B/3 B, as well as other two post-processing blocks are included. Between the two stages are pipeline registers.

Ideally, pipeline registers should divide a big combinational logic into two small logics that have nearly identical delays in order to make the circuit speed as fast as possible. However, it is difficult to find the accuracy position that can divide the logic into two identical parts with the same delay due to complex factors. By analyzing the logic stages of the critical path, combining with the simulation, registers are inserted into the proper locations in some critical paths first, then inserted into other non-critical paths. Below we will discuss the disparity generation block as an example.

In Fig. 3, the critical path of the disparity generation block starts from *b* (dataIn[1]) through *p*13 to dispOut, and its delay is about 1.5 ns longer than the delay desired. So the timing of *abcdei* disparity generation should be improved with pipelined architecture, as shown in Fig. 5. We can see that the disparity generation block is divided into two sub-blocks: the disparity pre-generation and the disparity post-generation.

Since the block of *fghi* disparity generation is not in the critical path, we just need to insert the registers to meet the timing. For the block of *abcdei* disparity generation, however, a group of pipelined registers are inserted near *p*13 to divide it into two stages. The former stage performs some computation, and part of them, such as *p*22, *p*13 and *p*31, are shared with other blocks, including 6 B/5 B pre-decoding block and 4 B/3 B pre-decoding block. The second stage receives the signal dispTemp coming back from the multiplex and the result of the first stage to generate the next disparity. As a result, the delays of the two critical paths are reduced from 1.5 to 0.87 ns and 0.84 ns separately, resulting in the delay being reduced greatly.

Using this pipelined architecture, the combinational circuits are slimmed and the critical path is shortened. Synthesis results show that the delay of the first stage's critical path is 0.9 ns and that of the second stage's is 0.88 ns, so the two stages are balanced and the critical path's delay is reduced from 1.65 to 0.9 ns, illustrating that the speed of the proposed decoder is improved significantly.

The only expense of this architecture is the small increase in area and power because some registers should be inserted between two stages. To trade off the performance from the view of speed, area and power, we should design the pipeline scheme properly.

Table 3. Synthesis constraints.

Property in DC	Value
Clock period (ns)	1.2
Clock uncertainty (ns)	0.04
Don't touch network	clock
False path	reset_n, bypass
Load (pF)	0.08
Input delay (ns)	0.6
Output delay (ns)	0.5
Maximum area	0

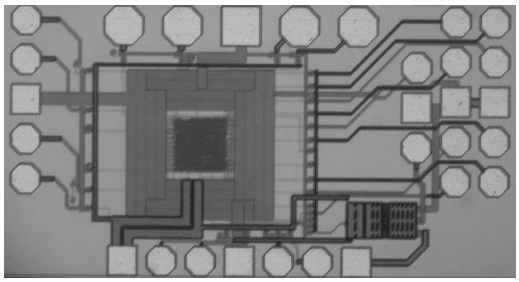

Fig. 7. Photograph of 10 B/8 B decoder .

3. Implementation of 10 B/8 B decoder

The specific design of this decoder is based on the TSMC 0.18 μm standard cell library. This is a pad-limited design and a timing-driven strategy is used to improve the delay performance. Table 3 shows the main synthesis constraints in the design. After the placement and route, actual interconnect delays are back-annotated and the design is re-synthesized for more accurate timing analysis[8].

Figure 6 gives the gate-level simulation result after placement and route, in which an SDF (Standard Delay Format) file is used to back-annotate the delay information. As shown in Fig. 6, 0x1D5 is decoded into 0x0F5, which is the same as the source code and the disparity output is generated correctly. The decoding delay is 2 clock cycles.

4. Measurement results

The 10 B/8 B decoder has been fabricated using TSMC 0.18 μm 1P6M CMOS technology, and its photograph is shown in Fig. 7 with a core area of 375 × 375 μm^2.

Measurements are carried out on-wafer. All of the input

Table 4. Performance comparison of 10 B/8 B decoders.

Parameter	Cadence[9]	Fukaishi[10]	ASICS.ws[11]	This work
Technology	TSMC 0.18 μm	0.25 μm	UMC 0.18 μm	TSMC 0.18 μm
Area	540 NAND equivalents + 21FFs	Unknown	Unknown	721 NAND equivalents + 81FFs
Data rate (Gbps)	1.25	5.0	3.0	6.25

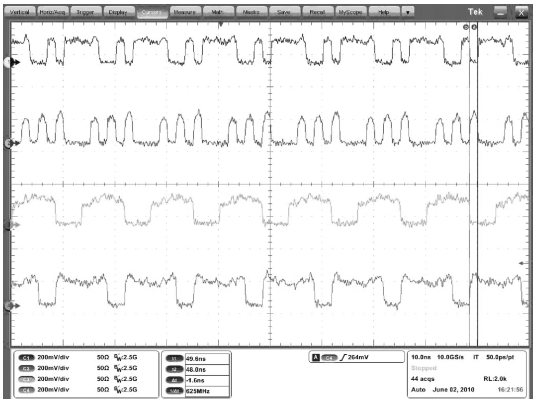

Fig. 8. Measurement results of data output.

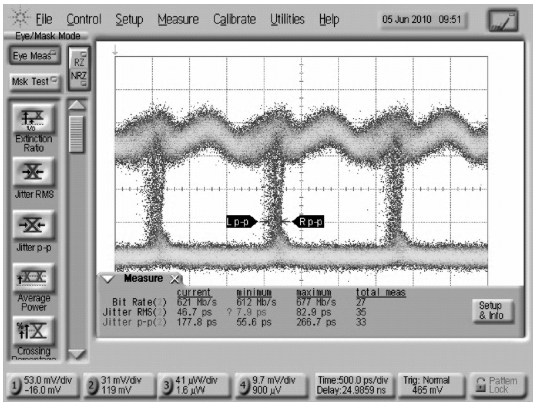

Fig. 9. Eye diagram of *A* (dataOut[0]) @ 6.25 Gbps.

signals are added from three sides except the right side, and output signals are tested from the left side with three probe rows. Figure 8 gives the measurement results of dataOut[0], dataOut[1], dataOut[4], and dataOut[5] (from top to bottom) at the data rate of 6.25 Gbps.

Figure 9 shows the eye diagram of signal dataOut[0]. Its output voltage range is 119 mV (peak to peak) under 50 Ω termination. We can see that the eye is open enough at the speed of 6.25 Gbps, with a peak-to-peak jitter of 177.8 ps and RMS jitter of 54.6 ps. At the same speed, the measurement result of power consumption is 21.6 mW under a power supply of 1.8 V.

Table 4 compares the performance of our work with that of three IP cores, in which 10 B/8 B decoders are implemented with CMOS technology. Among the three decoders, only the decoder in Ref. [10] is pipelined, others are not. From Table 4 we can see that although the proposed design has a slightly larger area, its speed is much faster compared to Ref. [9]. The decoder of Fukaishi[10] used in a receiver chip set for ultra-high-resolution digital displays is pipelined and implemented in 0.25 μm technology with a maximum data rate of 5 Gbps. The decoder of ASICS.ws, which is used in the SATA controller, has a maximum data rate of 3 Gbps[11]. So, we can say that our proposed 10 B/8 B decoder achieves an overall performance improvement with the pipelined structure.

5. Conclusions

A fully pipelined high-speed 10 B/8 B decoder is designed and implemented in TSMC 0.18 μm CMOS technology with a core area of 375× 375 μm^2. The measurement results show that this decoder works well at 6.25 Gbps with a power consumption of 21.6 mW in a 1.8 V power supply. In the future, this decoder will be integrated into a high speed SerDes.

References

[1] Widmer A X, Franaszek P A. A DC-balanced, partitioned-block, 8 B/10 B transmission code. IBM Journal of Research and Development, 1983, 27(5): 440
[2] http://en.wikipedia.org/wiki/8B10B_encoding
[3] Zhao Wenhu, Wang Zhigong, Fei Ruixia, et al. Study on 8 B/10 B encoding and decoding method based on logic analysis for fiber communication. Journal of Circuits and Systems, 2003, 8(2): 48
[4] Franaszek P A, Widmer A X. Byte oriented DC balanced (0, 4) 8 B/10 B partitioned block transmission code. US Patent, No. 4486739, Dec. 4, 1984 (Expired now)
[5] http://asics.chuckbenz.com/
[6] Wen Long, Fan Xiaoya. 8 B/10 B decoder design. Sci Technol Eng, 2007, 7(18): 4611
[7] Lattice Semiconductor Corporation. 8 B/10 B encoder/decoder. Reference Design RD1012, Nov. 2002
[8] He Xiaohu, Hu Qingsheng, Xiao Jie. An example of back-end design for ASIC in deep submicron technology. China Integrated Circuit, 2006, 8: 37
[9] 8-bit/10-bit decoder (DEC8B10B) technical data sheet. Cadence. Feb 2007
[10] Fukaishi M, Nakamura K, Heiuchi H, et al. A 20-Gb/s CMOS multichannel transmitter and receiver chip set for ultra-high-resolution digital displays. IEEE J Solid-State Circuits, 2000, 35(11): 1611
[11] Serial ATA I/II Device Controller IP Core. ASICS.ws