

# INTERCHIP DATA COMMUNICATION WITH PHASE FREQUENCY DETECTOR

<sup>1</sup>Vinod Kumar Nirala, <sup>2</sup>Kamal Kishor Joshi,

<sup>1</sup>Assistant Professor Department of ECE JB Institute of Technology

<sup>2</sup>Assistant Professor Department of CSE JB Institute of Technology

**Abstract :** The necessity of high speed clock can be fulfilled in two ways, either a very high frequency clock is directly used or low frequency clock is passed through a clock multiplier to have a high frequency clock. The first method is becomes inadequate when the required speed becomes very high (few gigahertz). The second method might be an alternative for the similar purpose. CMU is used to generate high frequency clock. The DLL based CMU is generally used in the industry to get higher speed clock. At high speed, some of the issues come into the picture strongly to achieve high quality performance. The main constraints appear due to limitations of different building blocks of the DLL. Each different unit has different limitations that affect the overall performances. Phase and frequency detector (PFD) imposes the problem of its resolution. The PFD resolution should be high in order to minimize static phase error that helps in reduction of jitter. It reduction in static phase error increases the closed loop speed. The current mismatch in charge pump is greatly affected due to low resolution of the PFD. Hence we find, high resolution is a stringent requirement to achieve high speed in DLL. Generally PFD is realized by conventional method (D flip flop based) but this project used pre-charge type PFD which offers higher resolution. Actually, PFD is closed loop system so the delay of the circuit becomes an important parameter to determine the speed of the same.

**Index Terms - Circuit, CMU, DLL, Jitter, PFD.**

## 1. INTRODUCTION

To enhance speed and reduce area the better choice is serial data transfer in place of parallel data transfer. It facilitates the high speed performance for the link and minimizes many unwanted effects. It also allows less complexity in design and routing. The serializer might be of higher order to serialize many parallel inputs but in this project 2:1 serializer is used to make higher order serializer. The design of lower order serializer needs fewer transistors so it is less complex. The simulation is also easier for such lower order serializer. By using proper clock frequencies at different stages, these lower order serializers are used to realize higher order serializer [1]. The quality of the signal is determined by different aspects like jitter, reflection and clock skew. These aspects plays major role in determining the reliability and performance of a design requiring high speed. The performance of transmitter not only helps the sender section to send proper data but also the receiver finds it easier to recover the clock and data from the incoming data [2]. Intra chip communication is faster than inter chip or off- chip communication. The signal processing modules used on the chip is even faster than on-chip communication. Basic reason behind such improvement in performance is enhancement in device performance. Bandgap engineering and device scaling basically brought so many desired changes in device properties. All such improvements add to accelerate the speed performance of the signal processing modules. But inter and intra chip communication does not support such a high speed. The major issue is noises, cross talk, limited channel bandwidth etc.

### 1.1 Design of Serializer

2:1 serializer is used to design 8:1 serializer. The basic internal blocks are shown in fig-1. Two flip-flops, one latch and one mux are required to design this. The clock fed to each block is same and also the flip-flops and latches both must be either positive edge triggered or negative edge triggered. The latches are basically providing the delays to the incoming signals. The delay must be half of the clock cycle. Generally, clock starts from the middle of the data input. As rising edge of the clock will select one of the data input and this rising edge is in the middle of the selected data input, then the falling edge of the clock will occur exactly at the falling edge of the second data stream. This will cause a problem of synchronization for the second data stream and the clock. It may be when the falling edge of the clock appears, correct data is unavailable in the second data stream. To avoid such problem of erroneous output, one latch is used. It provides the delay which helps in finding the correct data in the second data stream whenever negative edge of the clock appears. The study includes serializers based on two different concepts-

CMOS based 2:1 Serializer

CML based 2:1 Serializer

CMOS architecture of serializer allows full Swing in the output whereas CML design allows reduced swing. The swing affects the speed of the circuit. The CML logic is faster due to reduced swing but it dissipates the huge power [3]. The reason is being the transistors in the saturation region. There is a path between ground and supply all the time which causes a flow of current continuously and it results in higher power consumption. The CML logic offers less glitches in comparison to the CMOS logic based serializer. Due to having larger swing the delay becomes larger, which causes more glitches [4].

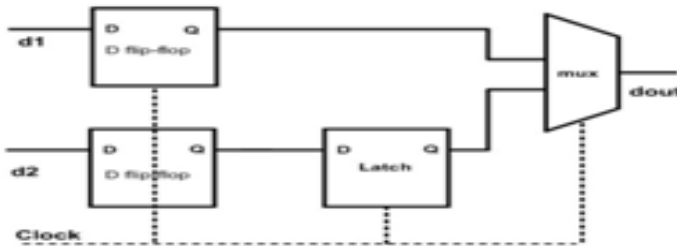


Fig-1: Internal Blocks of 2:1 Serializer

### 1.2 Design of 6b/8b Precoder

Long run length of 0s or 1s creates potential problem for clock recovery from the data. It also causes DC offset in the channel. To avoid such problem precoding is done. It ensures the transition in the data stream if continuous 1s or 0s are present in the original data bits. The precoder designed allows worst case run length of 4 only. For this purpose two extra bits are added at fourth and eighth position respectively. The block diagram is shown in fig-2.

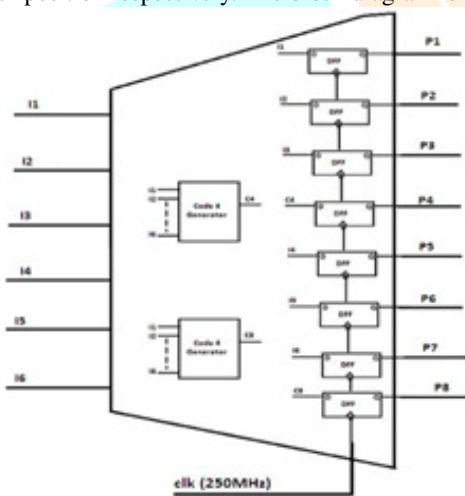


Fig-2: Internal Blocks of 6b/8b Precoder

The output data stream can be represented as I1 I2 I3 C4 I5 I6 I7 C8. C4 and C8 are generated by combinational blocks. As the generation of C4 and C8 provide delays in these codes with respect to actual input streams, it may create problem at higher speed for proper serialization. Hence we need to synchronize these bits. We have used negative edge triggered DFF for this purpose. It will use the same clock which is the input to the CMU. Therefore, final outputs we represented as P1 P2 P3 P4 P5 P6 P7 P8.

### 1.3 Design of C4 and C8 generators

The truth table we used for the design of combination circuit is shown in table-1. The maximum run length possible is 4. C8 is kept very simple. It just inverts the I6 value. It is done because we don't about the next stream. The best choice is to provide a transition with respect to I6 which is shown in the table. Hence we now have to implement the combinational circuits for the generation of these codes. C8 generation is very simple. It just reverses the logic value of I6. So, we concentrate on C4 generation.

$$C4 = (\sim I1)(\sim I2)(\sim I3)(\sim I4) + (\sim I2)(\sim I3)(\sim I4)(\sim I5) + (\sim I3)(\sim I4)(\sim I5)(\sim I6)$$

$$C4 = (\sim I3)(\sim I4)[(\sim I1)(\sim I2) + (\sim I2)(\sim I5) + (\sim I5)(\sim I6)]$$

### 5.2 Truth Table for C4 and C8

I1	I2	I3	C4	I4	I5	I6	C8
0	0	0	1	X	x	x	$\sim(I6)$
x	0	0	1	0	x	x	$\sim(I6)$
x	x	0	1	0	0	x	$\sim(I6)$
x	x	X	1	0	0	0	1

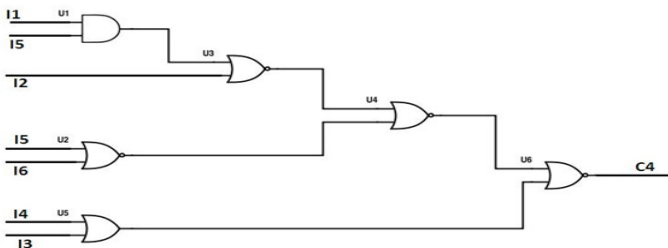
$$C4 = (\sim I3)(\sim I4)[(\sim I2)\{(\sim I1)+(\sim I5)\} + (\sim I5)(\sim I6)]$$

$$C4 = \sim(I3+I4)[(\sim I2)\{(\sim I1.I5)\} + \{(\sim I5+I6)\}]$$

$$C4 = \sim(I3+I4)[\{(\sim I2+I1.I5)\} + \{(\sim I5+I6)\}]$$

$$(\sim C4) = (I3+I4) + \sim[\{(\sim I2+I1.I5)\} + \{(\sim I5+I6)\}]$$

The implementation using the 2 input logic gates are shown in fig-3.

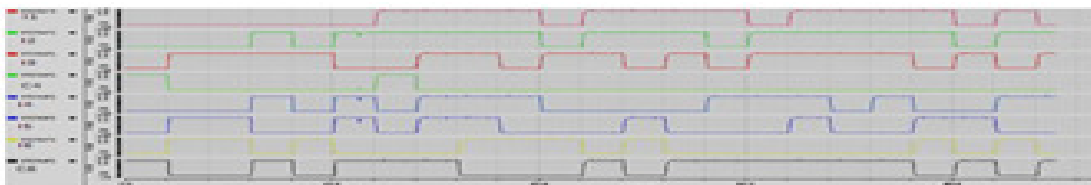


**Fig-3:** Circuit for C4 code generation

As we have seen for data retiming we have used negative edge triggered DFF. We were expecting the total delay due to insertion of combination logic blocks should be less than half of the clock cycle (2 ns). Here clock refers to input clock of CMU. And we got maximum delay of 194ps. Hence the negative edge can be used to synchronize the streams. Now we proceed with the results.

## 2. RESULTS

First we see result of C4 code. All six inputs are random in nature. C4 code is 4<sup>th</sup> from the top. We can see, C4 code does not allow to have run length of 1s or 0s more than 4. C8 is also shown at the bottom.

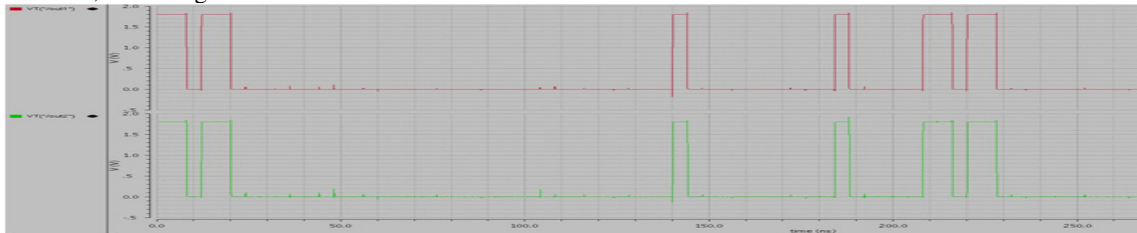


**Fig-4:** C4 codes (4<sup>th</sup> from top) and other input streams.

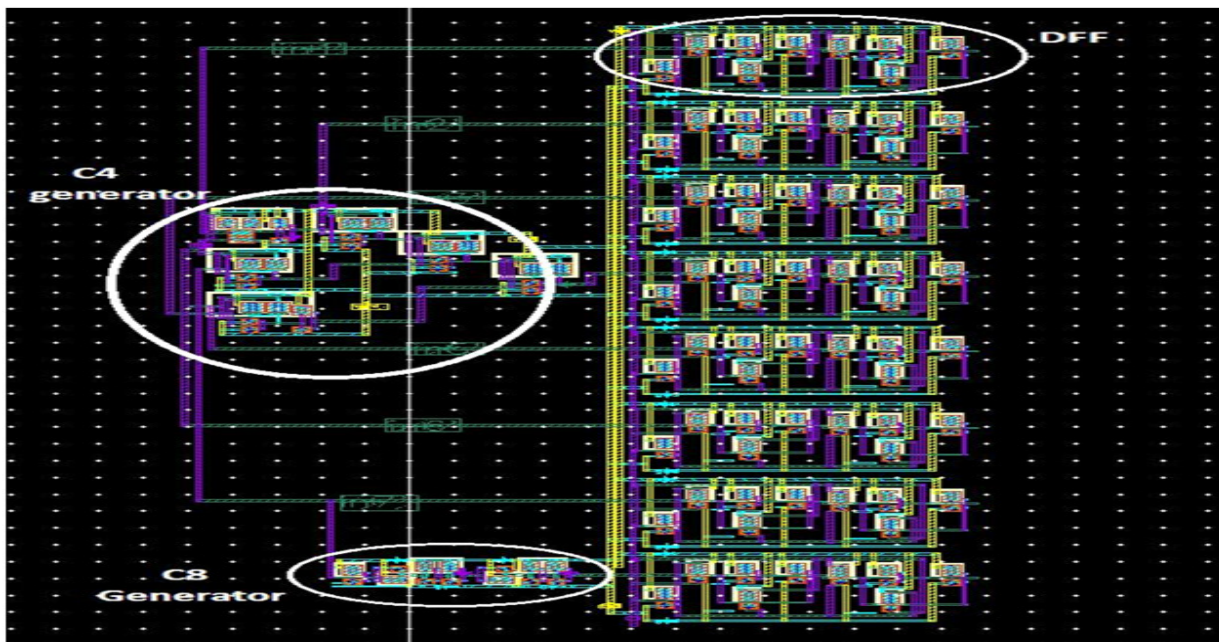
### 2.1 Layout and post layout simulation's results

The layout of C4, C8 and finally overall Precoder are completed. We will present at the last of the chapter. Before that, we see the post layout simulation results along with the schematic's simulation results. Both are in agreement at most of the places. As the operating speed of precoder is not very high (250 Mbps), the post layout simulation did not create much challenges for matching

its responses with the responses of the schematics. During discussion of design we did not discuss about DFF in this chapter. The reason is, it is the same design of DFF we used while designing serializer in chapter 3. As DFF triggers with the negative edge of the clock, we have given inverted clock for the simulation.



**Fig-5:** C4 code: Schematic-result (red) and Post layout simulation-result (green)  
The layout of C4 code generator and complete precoder are shown in fig-6 and 7 respectively.



**Fig-7:** layout of precoder

### 3. CONCLUSIONS

The improvement in device characteristics and signal processing modules can be cashed by increasing the speed of interfacing circuits used between different chips. The serial communication has distinct advantages over parallel communication. Serializers based on CMOS and CML both should be utilized properly to take the advantages of high speed as well as lower power consumption. CMU based on DLL concept is a very useful technique to get high frequency clock. DLL has different components which put stringent limitations on the performance of the CMU. The resolution of PFD must be enhanced to get less static phase error. The main culprit of poor performance of PFD is delay in feedback path. In this regard pre-charge type PFD offers less delay in comparison to conventional PFD. This is why pre-charge type PFD is preferable. Designed “6b to 8b precoder” makes sure that enough transitions in the transmitted data stream so that clock recovery is easier. All the designs are accomplished in Cadence environment. Schematic is done Virtuoso while layout is done in Calibre.

## REFERENCES

- [1]. "Interface IP Survey Release 6" by IPnest released in October 2014.
- [2]. A. Morgenshtein et al., Comparative analysis of serial vs parallel links in NOC, Published in System-on- chip,2004 Proceedings.2004 International Syposium on, pages 185-188,16-18 Nov.2004.
- [3]. Lisha Li, Sripriya Raghavendran and Donald T. Corner, CMOS Current Mode Logic Gates for High-Speed Applications , 12 th NASA Symposium on VLSI Design on VLSI Design , Corner d'Alene, Idaho, USA, oct.4-5, 2005.
- [4]. M. Green and U. Singh, Design of CMOS CML circuits for high speed broadband communicatons , in proc.IEEE Int. Syp. Circuits and Systems, vol. II, May 2003,pp. 204-207.

