FPGA based ASIC Emulator with High Speed Optical Serial Links

Motoki Amagasaki*, Futoshi Murase**, Morihiro Kuga*,
Masahiro Iida* and Toshinori Sueyoshi*

* **Faculty of Advanced Science and Technology, Kumamoto University
2-39-1 Kurokami, Kumamoto, 860-8555 Japan

* {amagasaki, kuga, iida, sueyoshi}@cs.kumamoto-u.ac.jp,

** {murase}@arch.cs.kumamoto-u.ac.jp

ABSTRACT

We propose a multiple FPGA system using high speed optical serial interface built in recent FPGAs and construct ASIC emulator. Although conventional system which uses parallel connection is limited to bandwidth of the number of I/Os, proposed system has no restriction. In this paper, we implement the circuit using Kintex Ultrascale which has optical high speed serial interface and evaluate its performance using five VTR benchmark circuits. As a result, we found that proposed system operated at 21.3MHz. Furthermore, if we use a Virtex Ultrascale+, operating frequency increases up to 29.97MHz.

1. INTRODUCTION

The problem of high manufacturing cost in ASIC (Application Specific Integrated Circuit) becomes remarkable, and the leading role of the implementing medium is shifting to FPGA (Field Programmable Gate Array). In addition to the conventional features of low development cost, FPGA is being improved in performance due to advanced process technology. Virtex UltraScale+ fabricated the 16 nm FinFET process is released in the current commercial device. In this device, it is possible to implement 4 million ASIC gate scale circuit, and the FPGA is a very attractive device in both integration scale and performance.

When implementing a very large circuit such as deep learning or ASIC emulator, it is necessary to implement with multiple FPGAs. Indeed, many studies have studied on multiple FPGA systems[1][2]. When connecting multiple FPGAs in parallel to realize large scale integration, the communication bandwidth (I/O limitation) between FPGAs becomes a problem. The total number of I/Os does not increase dramatically as the process shrinking, so I/O limitation is still crucial problem. Timing skew and connection delay become also critical because a signal across FPGAs has a large delay on the board. In addition, as the size of target application increases, it is predicted that the number of signals across FPGAs increases according to Rent's Law[3]. If the usable I/O number of the FPGA can not satisfy the number of signal across FPGAs, target application can not be implemented.

This work was presented in part at the international symposium on Highly-Efficient Accelerators and Reconfigurable Technologies (HEART2017) Bochum, DE, June 7-9, 2017.

In order to balance between the integration scale and the communication bandwidth, SiP (Silicon in Package) which integrates multiple chips in one package and 2.5 dimensional FPGA which connects dies on the silicon interposer using TSV (Through Silicon Via) are available. Both of them are effective to improve integration scale at the package level. However, when the circuit does not fit within one chip, the problem of communication bandwidth still remains.

On the other hand, FPGA with gigabit transceiver as a hard macro has been released in commercial device. In this case, the data transfer between FPGAs is performed via the high-speed serial interface circuit. For example, Virtex Ultrascale+ XCVU 37P requires a total of 96 high-speed serial interfaces. This means that one-to-one data transfer is possible up to 96 FPGAs. An FPGA system with a high-speed serial interface has a potential to mitigate the problem of the conventional I/O bandwidth and skew between signals, enabling a larger circuit implementation.

Our research group has been studying multiple FPGA system connected by high speed serial interface targeting ASIC emulator. The aim of our research is to realize FPGA based ASIC emulator using a reasonable method. In this paper, we implement the circuit using Kintex Ultrascale which has optical high speed serial interface and evaluate its performance. The remaining part has the following structure. First, we will explain the related research in Section2 and the proposed system in Section3. Section4 evaluates the actual equipment, and Section5 summarizes the future works.

2. RELATED WORKS

This section describes related research. First of all, we describe multiple FPGA system, and next introduce research on ASIC emulator.

2.1 Multiple FPGA system

Multiple FPGA systems are composed of multiple FPGAs. The advantage of this system is to realize large scale circuit simply due to connect FPGAs. However, when FPGAs are arranged on a substrate and are connected in parallel, crosstalk and skew deviation between FPGAs become a problem. On the other hand, SiP and 2.5 dimensional stacking have appeared in recent packaging technology. However, when the circuit scale of the target application does not fit in one package, it is necessary to extend using a plurality of chips. In this case, the bandwidth, crosstalk and clock skew

become a problem.

2.2 ASIC emulator

An ASIC Emulator is an hardware system that emulates the design of ASIC before it is fabricated. The logic verification method of ASIC can be roughly divided into the following three types. (1) emulation using FPGA, (2) simulation using specific LSI dedicated to logic verification, and (3) simulation on a processor using programming languages ??such as Verilog-HDL. (1) and (2) are mainly used in commercial FPGA and (3) is a simulation method using design verification tools such as VCS and Verilog-XL.

Regarding to an FPGA based emulator of (1), many researches[5] have studies how to connect FPGAs in parallel on the board. However, it was difficult to obtain the desired performance due to the problem described in 2.1. In [6], the emulation method using a scalable division mechanism has been proposed. In this case, Information on divided circuits for tens of clock cycles is temporarily stored in the memory on the FPGA. The signals are collectively transmitted to the other divided circuit, expanded on the another FPGA. This method makes it possible to improve the operating frequency. As other research, there is an implementation method using high speed serial link for FPGA communication[4] . Since serial communication has few signal lines, timing skew and crosstalk problems are reduced. Furthermore, this technique can flexibly deal with the number of signals required for communication. However, [4] is limited to the verification of the simulation level using Virtex-6, and actual device evaluation has not been done. Meanwhile, to divide a circuit, [4] uses hMETIS[7] distributed binary form from Minnesota university, so cannot execute fine tuning according to logic cell architecture.

3. PROPOSED ASIC EMULATOR

In this section, we propose FPGA based ASIC emulator system with high speed optical serial link. First, we introduce the development policy of our system and its design flow. Next, circuit partitioning method and structure of our emulation system is described.

3.1 Development policy

Regarding an ASIC emulator using multiple FPGAs, the following problems to be solved exists.

(1) Emulation frequency

When an ASIC design is implemented on multiple FPGAs, we have to consider the circuit delay which passes between FPGAs. Especially when the FPGAs are connected in parallel, timing skew and delay occur and the emulation frequency lowers.

(2) Circuit partitioning

According to Rent's law, as circuit scale increases, the number of signals between FPGAs increases[4]. When using of parallel connection, it is impossible to assign signals more than the number of I/O pins. Of course, formal verification is required in the circuit before and after partition.

(3) Development cost

Commercial emulators are effective solution, but its cost is sometimes very expensive. In addition, to implement on ASIC emulator, we have to modify original circuit design. This modification leads more increases of design cost and complexity.

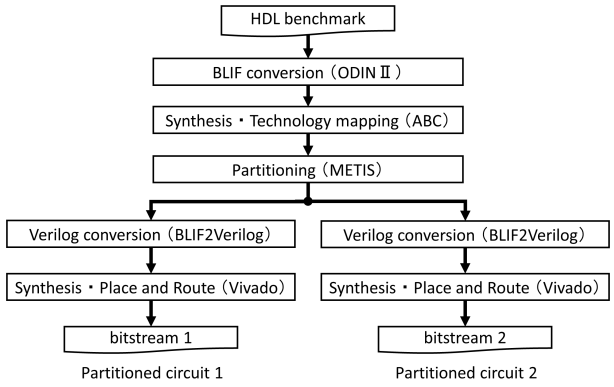

Figure 1: Design flow

In order to mitigate above problems, we propose ASIC emulator which uses high speed optical link and circuit partitioning method. Optical communication is capable of high-speed communication of 10 Gbps or more and long-distance communication as compared with the telecommunications up to Virtex-5. For circuit partitioning, develop METIS[8] based circuit partitioning tool[7]. If we extend circuit scale of ASIC emulator, the system can be realized simply by connecting optical serial ports. Even low priced FPGAs such as Kintex Ultrascale have equipped with as many as 20 optical high-speed serial interfaces and it is easy to extend the scale of the system. In addition, the circuit implementation can only be designed with Vivado Design Suite and some academic tools. There are no needed manual modification of original circuit for implementing on our emulator.

3.2 Design flow

Figure 1 shows the circuit design flow for our system. Applications written in HDL (Hardware Description Language) are elaborated to BLIF format using ODIN II, and logic synthesis and technology mapping are performed by ABC. Next, we use circuit partitioning tool based on METIS[8] to perform area orient partitioning. The obtained circuit is converted to Verilog HDL through BLIF2Verilog. This BLIF2Verilog is format conversion program written in Perl. Next, the divided circuits inputs commercial FPGA design tool (e.g. Xilinx Vivado). In this phase, logic synthesis, functional verification, place and routing are executed. The system operates by downloading the bit stream to target FPGA.

3.3 Partitioning method

The features of the circuit partitioning tool used in this research are as follows.

(1) Unit of partition

Figure 2 shows groups of combinational circuits with FF (Flip Flops) as the Virtex. This is called a logic cone, and METIS treats the unit of logic cone. For this reason, there is no case that combinational logic is partitioned.

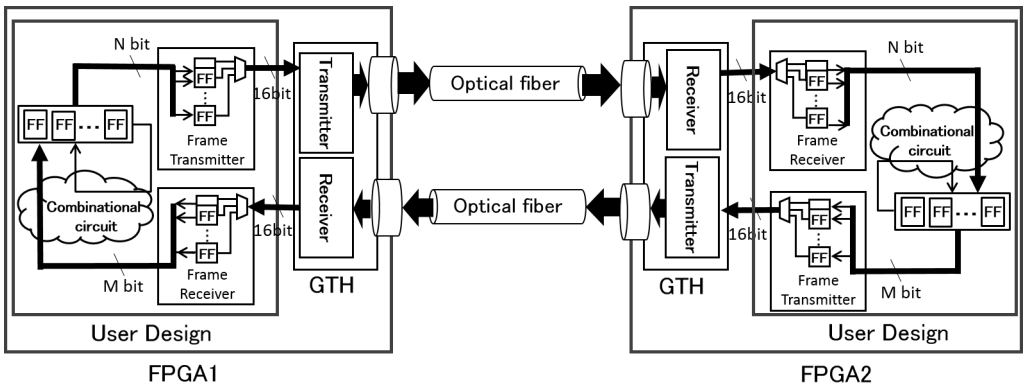

Figure 4: Overview of proposed FPGA based ASIC emulator

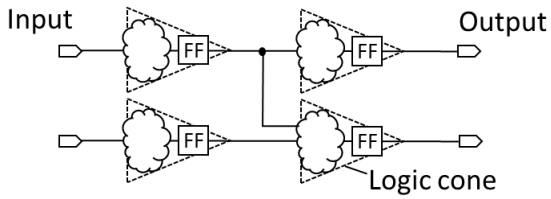

Figure 2: Logic cone

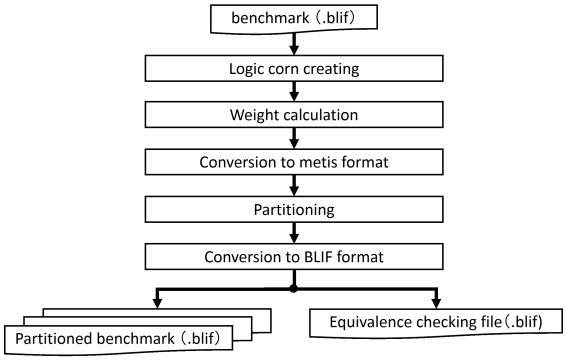

Figure 3: Circuit partitioning flow

(2) Consideration of partition ratio

We modified original METIS so that hardware resources after partitioning become about 1:1 according to the structure of logic cell (e.g. LUT inputs). Then, this tool partitions target application circuit so as to minimize the number of signals across FPGAs as equal area as possible.

(3) Critical path

The proposed system transfers signals through the SerDes (SERializer / DESerializer) circuit. If there are critical path on the signal across FPGAs, there is a possibility that critical path delay increases. Therefore, we have to partition the circuit so that the critical path does not appear on boundary surface between FPGAs. Note that critical path handled here is at the mapping level.

(4) Formal verification

In partitioning step, it is necessary to ensure equivalence of circuits between original and divided circuits. To verify equivalence, this tool generates the circuit merged divided circuits. Then, we can execute formal verification by using ABC.

The circuit partition flow is shown in Fig.3. First, a graph of a logic cone set is generated from a benchmark file of BLIF format. Next, To perform area oriented partitioning, the weights are calculated. Then, the tool divides circuits. Finally, the partitioned circuit and the file for formal verification are output in BLIF format.

3.4 Structure of the system

An outline of the system is shown in Fig.4. In this paper, we use Xilinx Kintex UltraScale FPGA as the target device, and we use optical communication based serial interface called GTH transceiver. This GTH transceiver is power efficient and supports line rates from 500 Mbps to 16.375 Gbps. Each module of GTH is composed of the following blocks.

1. Paratitioned circuit
   For the reason described in 3.3, partitioning tool divides the circuit in units of logic cones with FF as the vertex. The data sent from the frame receive block passes through the combinational circuit and is stored in the FF of each FPGA. Thereafter, the data in the FF is transmitted to the frame transmission block.

2. Frame transmission and receiver block
   The GTH transceiver needs to transfer data in frame units. In the proposed emulator, one frame is set to 16 bits, and the transmission and receiver between the FPGAs is performed on a frame basis. In the transmission block, the output data of the divided circuit is cut out for each frame by a multiplexer and transmitted to the GTH transceiver. In the receiver block, consecutive frame data is received and stored in the internal register. After counting the number of designated frames and receiving all the data in one emulation, the data in the register is output to the another

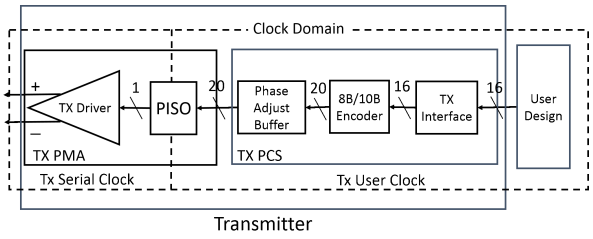

Figure 5: Transmitter

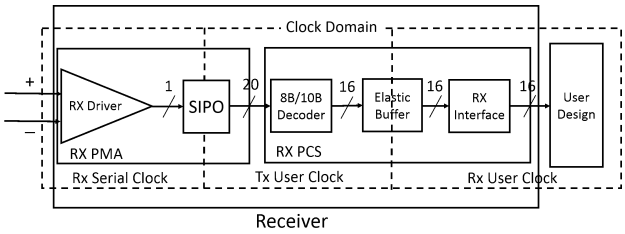

Figure 6: Receiver

circuit.

3. GTH transceiver
   It is a communication module provided as a hard macro block and realizes communication speed up to 16.375 Gbps. This transceiver has a transmitter and a receiver. This transceiver can set communication speed, encoding method and can handle many communication standards. The outline of the transceiver is shown below.

Transmitter

Figure 5 shows the block diagram of the transmitter. The transmitter receives 16 bit data transmitted from the another FPGA by the TX interface. Next, 8B10B encoding is performed on the data. By embedding the clock in the serial data by 8B10B encoding, it is possible to transfer data and clock at the same time. Also, synchronization with the serial communication clock is achieved by the phase synchronization buffer. Thereafter, the transmission data is converted into serial data using PISO (Parallel-In Serial-Out), and the data is transmitted with a differential signal.

Receiver

Figure 6 shows an outline of the receiver. By extracting the clock from transmission data as a differential signal, the receiver synchronizes with the another FPGA. Then, the received data is converted into parallel data using SIPO (Serial-In Parallel-Out), and decoding is performed. An elastic buffer absorbs the difference between the clock of the transmitter and the receiver. After that, it transmits 16 bit data to the user circuit using the RX interface.

3.5 Signal transfer timing

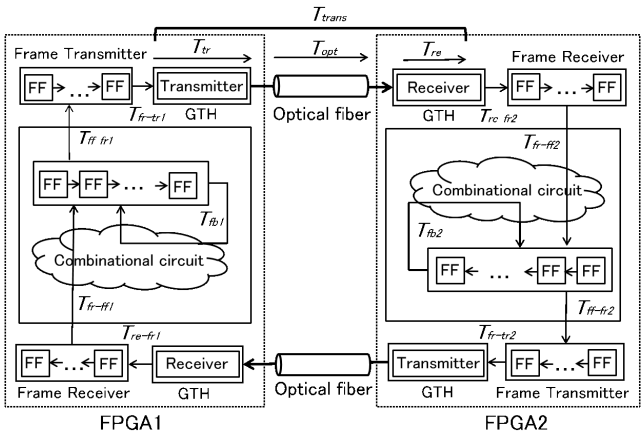

Figure 7: Signal communication of implementiing circuit

In this section, we describe how to calculate system frequency, CLK_{system}, when the system is constructed using two FPGAs. Table 1 shows various parameters for calculating the system clock. The delay of GTH transceiver and optical cable, T_{trans}, is obtained as follows.

$$T_{trans} = T_{tr} + T_{re} + T_{opt} \tag{1}$$

T_{tr} is the transmitter delay, T_{re} is the receiver delay, and T_{opt} is the optical cable delay. The operating frequency of the transceiver is obtained as follows.

$$CLK_{tr} = Linerate/D_{path} \tag{2}$$

Line-rate is the number of transfer bits per second (communication line rate) in high-speed data communication, and D_{path} is the internal data path of the transceiver. Assuming that the number of transfer frames is n and the number of serial links is m, T_{fr} is obtained as follows.

$$T_{fr} = \frac{1}{m} \times \lceil \frac{n}{D_{length}} \rceil \times \frac{1}{CLK_{tr}} \tag{3}$$

D_{length} is the data width of the transceiver interface, and CLK_{tr} is the operating frequency of the transceiver. T_{fr-fr} is the critical path delay from the FF of divided circuit to the frame transmission block. T_{fr-tr} is the critical path delay from the frame transmission block to the transmitter. $T_{fr-ff}$is the critical path delay from the receiver to the FF of divided circuit. T_{re-fr} is the critical path delay from receiver to the frame receive block. The critical path delay between the FFs in the FPGA is T_{fb}, then, these parameters are obtained as follows.

$$T_{ff-fr} = max\{T_{ff-fr1}, T_{ff-fr2}\} \tag{4}$$

$$T_{fr-tr} = max\{T_{fr-tr1}, T_{fr-tr2}\} \tag{5}$$

$$T_{fr-ff} = max\{T_{fr-ff1}, T_{ff-ff2}\} \tag{6}$$

$$T_{re-fr} = max\{T_{re-fr1}, T_{re-fr2}\} \tag{7}$$

$$T_{fb} = max\{T_{fb1}, T_{fb2}\} \tag{8}$$

Table 1: Timing definitions

Parameter	Description
T_{tr}	Transmitter critical path delay
T_{re}	Receiver critical path delay
T_{opt}	Optical fiber communication delay
T_{trans}	The time required for the signal transmission
T_{fb}	Critical path delay between FFs in the FPGA
T_{ff-fr}	Critical path delay from FF of FPGA to frame transmission block
T_{fr-tr}	Critical path delay from frame transmission block to transmitter
T_{fr}	The time it takes to send and receive all of the frame
T_{fr-ff}	Critical path delay from the frame receive block to the FF of FPGA
T_{re-fr}	Critical path delay from receiver to frame receive block

Table 2: Development and evaluation environment

Development board	Kintex UltraScale KCU105 Evaluation Kit
Evaluation device	Kintex XCKU040-2FFVA1156E
Number of I/O	520
Number of Tranceiver	20
Development tool	Vivado Design Suite 2016.2

Table 3: Evaluation conditions

GTH Line rate	6.0 [Gbps]
Bit per frame	16 [bit]
Internal data path	20 [bit]
Number of serial links	1
Reference clock	150 [Mhz]

The total delay, T_{system}, is obtained as follows.

$$T_{system} = max\{T_{ff-fr} + T_{fr-ff}, T_{fb}\} + T_{fr-tr} + T_{trans} + T_{re-fr} + T_{fr} \times 2 \quad (9)$$

$$CLK_{sytem} = 1/T_{system} \quad (10)$$

At the rising edge of CLK_{system}, the transmitter receives signals through optical serial link. When the number of transfer bits n increases, the delay increases by $\lceil (n/D_{path}) \times 1/CLK_{tr} \rceil$.

4. EVALUATION

This section describes evaluation condition and results. We use 5 types of VTR benchmark circuits and implemented them on the proposed ASIC emulator with two FPGAs to evaluate frequency and the transfer latency of the GTH transceiver. As a baseline system, we prepared parallel connected systems. Finally, we discuss the I/O usage rate and the maximum performance of proposed emulator. Note that when using multiple FPGAs, large scale benchmark circuits which cannot be implemented on one FPGA are needed. However, since it is difficult to get such large circuits, we use relatively large five types VPR circuits. Also, it is necessary to consider crosstalk and clock skew in parallel connection, but ignore them in this evaluation.

4.1 Evaluation condition

Table 2 shows evaluation conditions. In this evaluation, we use the Kintex UltraScale KCU 105 evaluation kit from Xilinx. This KCU 105 is equipped with Xilinx FPGA Kintex XCKU 040-2 FFVA 1156E. The number of I/O terminals is 520. In addition, this FPGA has 20 GTH transceivers which are high-speed serial transceivers. Vivado Design Suite 2016.2 is used as a development tool.

Evaluation condition is shown in the Table 3. The maximum line-rate on the specification of the GTH transceiver in the Kintex UltraScale FPGA is 16.375 Gbps, but connector of KCU105 does not correspond to maximum line-rate. So, we set to 6 Gbps as a realistic line-rate. In the proposed emulator, the bit length per frame in the GTH transceiver is designed to be 16 bits, the constraint of frequency is 150 MHz. In this case, T_{trans} was 66.66 ns measured by ILA which is embedded soft core logic analyzer of Xilinx. We compare our system with parallel connected system in terms of system frequency. We assume that delay on the board is 1 ns from [9].

4.2 Evaluation results

Table 4 shows the number of I/Os and the number of transfer frames between two FPGAs, LUT utilization ratio, the delay of each element. The 11th and 12th columns denotes frequency of proposed system and parallel connected system, respectively. If the transfer bits between FPGAs satisfies the I/O constraint, we found that the frequency of the parallel implementation is high. On the other hand, in serial communication, it takes 66.66ns to transfer bits between FPGAs. For example, in “mac2”, the delay excludes transfer time requires 64.53ns. Transfer time obtained a half of external delay.

4.3 Discussion

Based on the results obtained in 4.2, we discuss in terms of I/O utilization of parallel implementation and maximum performance of serial implementation. Table 5 shows the resource utilization and I/O utilization in parallel implementation. We found that I/O utilization is much higher than LUT utilization in parallel system. For example, in ”rs_decoder1”, I/O utilization is very high, 37%, although the resource utilization is less than 1%. If the circuit scale increases, transfer bits between FPGAs increases according to Rent’s Law. Therefore in the parallel connection if transfer bits between FPGAs does not satisfy I/O constrain, implementation is failed. On the other hand, although our system is slower than parallel system, there is no limitation on the number of I/Os. Also, when using parallel connected system, we actually have to consider cross talk and clock skew problem.

The results of table 5 are obtained with line-rate 6Gbps and 1 serial link due to the constraints of evaluation kit. Then, we discuss the case of high-end device which is Virtex Ultrascale+ XCVU13P implementation. Table 6 shows conditions of serial transfer block of both KCU105 and XCVU13P.

Table 4: Emulation frequency (Number of links 1, Linerate 6Gbps)

Circuit	I/O	Frame	LUT utilization [%]	T_{ff-fr} [ns]	T_{fr-ff} [ns]	T_{fr} [ns]	T_{fr-tr} [ns]	T_{re-fr} [ns]	T_{trans} [ns]	Frequency (Serial) [MHz]	Frequency (Parallel) [MHz]
fir_scu_rtl _restructured _for_cmm_exp	149	9	0.51	1.39	3.87	29.97	1.72	3.35	66.66	7.30	217.48
rs_decoder1	193	8	0.78	3.05	6.85	26.60	1.65	2.01	66.66	7.49	198.84
mac2	237	13	5.33	6.90	11.23	43.29	1.22	1.89	66.66	5.73	125.58
des_area	36	2	0.51	3.83	4.23	6.60	1.67	2.45	66.66	11.70	237.52
des_perf	120	8	0.73	1.88	3.61	26.60	1.58	2.31	66.66	9.74	259.67

Table 5: Resource utilization (Parallel)

Circuit	LUT utilization [%]	I/O utilization [%]
fir_scu_rtl _restructured _for_cmm_exp	0.51	28.65
rs_decoder1	0.78	37.11
mac2	5.33	45.57
des_area	0.51	6.92
des_perf	0.73	22.64

Table 6: Conditions of serial transfer

	Device	Serial links	Line rate
Case1	Kintex UltraScale	1	6.00
Case2	KCU105	20	16.37
Case3	Virtex UltraScale+ XCVU13P	128	32.75

The case1 is the value of this evaluation, the case2 and the case3 are the maximum condition of KCU105 and XCVU13P, respectively. Table 7 shows comparison of each condition of Table 6. To improve system frequency, to increase linerate and the number of links are important. In "mac2", frequency of XCVU13P is actually 5.23 times higher than KCU105.

Table 7: Comparison of operating frequency [MHz]

Circuit	Case1	Case2	Case3
fir_scu_rtl _restructured _for_cmm_exp	7.30	27.80	43.20
rs_decoder1	7.49	25.51	37.91
mac2	5.73	21.33	29.97
des_area	11.70	26.44	40.40
des_perf	9.74	28.55	45.02

5. SUMMARY

In this paper, we propose FPGA based ASIC emulator with high speed optical serial link and construct its design flow. Although conventional system which uses parallel connection is limited by bandwidth of the number of I/Os, proposed system has no restriction. We actually construct ASIC emulation system with Kintex Ultrascale FPGA. In the evaluation, we found that proposed system operated reasonable frequency. Then, if we use high end device, frequency is able to more increase.

6. REFERENCES

[1] K.Fleming, M.Adler, M.Pellauer, A.Parashar, Arvind, J.Emer, "Leveraging Latency-Insensitivity to Ease Multiple FPGA," Proc. FPGA, pp175-184, Feb. 2012.

[2] WJ.Fang, A.C.-H. Wu, "A hierarchical functional structuring and partitioning approach for multiple-FPGA implementations," Proc. ICCAD, pp638-643, Oct. 1996.

[3] M.Butts, Logic Emulation and Prototyping, "It's the Interconnect (Rent rules)," RAMP Wrap, Aug 2010.

[4] K.Takahashi, M.Amagasaki, M.Kuga, M.Iida, T.Sueyoshi, "Circuit Partitioning Methods for FPGA-based ASIC Emulator using High-speed Serial Wires", Proc. SASIMI, pp.317-318, Mar. 2012.

[5] Mario P. Vestias, Horacio C. Neto, "Area and performance optimization of a generic network-on-chip architecture," Proc. SBCCI, pp.68-73, Aug. 2006.

[6] H.Nakajo, K.Matsuda, T.Miyoshi, M.Takemoto, S.Funada: "A verification method of a partitioned circuit with a high level synthesis tool based on java language.", Proc. ACOMP, pp.1-9, Nov. 2014.

[7] T.Kajiwara, Q.Zhao, M.Amagasaki, M.Iida, M.Kuga and T.Sueyoshi, "A Novel Three-dimensional FPGA Architecture with High-speed Serial Communication Links," Proc. ICFPT, pp.306-309, Dec. 2014.

[8] University of Minnesota, G.Karypis, "METIS 5.1.x Manual," Mar. 2013.

[9] KC.Nunna, F.Mehdipour, K.Murakami, "3D FPGA versus multiple FPGA system: enhanced parallelism in smaller area," Proc. AusPDC, Vol. 152, pp.37-43, Jan. 2014.