Circuit Partitioning Methods for FPGA-based ASIC Emulator using High-speed Serial Wires

Katsunori Takahashi*, Motoki Amagasaki, Morihiro Kuga,
Masahiro Iida and Toshinori Sueyoshi

Graduate School of Science and Technology, Kumamoto University
Kumamoto, 860-8555, Japan
k_takahashi@arch.cs.kumamoto-u.ac.jp, {amagasaki, kuga, iida, sueyoshi}@cs.kumamoto-u.ac.jp

Abstract— We are studying FPGA-based ASIC emulator via high-speed serial communication. In this emulator, there are restrictions on placement of the FFs on FPGA and we have to reduce replicated logic gates and replicated input terminal when partitioning the cicuit to FPGAs. When the proposed circuit partitioning techniques are compared with hMETIS, it achieved average 56.4% reduction in the technique for suppressing the duplicution of external inputs. In the technique for suppressing the duplicution of nodes, it achieved average 71.8% reduction.

I. Introduction

Conventional FPGA-based ASIC emulators use parallel link for communication between FPGAs. This emulator has three significant problems: low operating frequency, lack of FPGA pins, and wiring congestion of between FPGAs. In order to get over these issues, we study an FPGA-based ASIC emulator with high-speed serial link communication (hereinafter called "ASIC-emulator"). Since the serial link has a significantly higher transfer rate than the parallel link, it can virtualize many wires[1]. However, this emulator has the constraints for circuit partitioning. The flip-flops are allocated on the both cut-planes of partitioned circuits to synchronously run the emulation clock.

A hMETIS, which is a general-purpose partitioning tool for hyper-graph, is convenient to divide the circuit in related work[2]. But the hMETIS doesn't optimize the logic area and the number of input pins. This may induce the increasing of FPGAs which implemented on the emulator, or the degrading of the emulator's operating frequency. In this paper, we present the circuit partitioning technique for the ASIC-emulator using the serial link.

II. Target emulator

Fig.1 shows the overview of preproposed ASIC-emulator. As a basic idea, we assume that the emulator has two FPGAs, and this emulator has a receiver and a transceiver for the serial link. The transceiver and receiver not only communicate state information on FFs, but also perform serialization and deserialization, respectively. Moreover, the clock for communication is different from the emulation clock. The procedure of the emulation is as follows. At first, the external input and 1 cycle clock

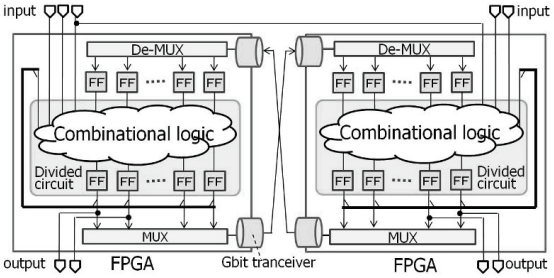

Fig. 1. Overview of the target ASIC-emulator.

input to the emulator. The output data are transmitted to the receiver if all transceiver's FFs held it. The multiplexer (MUX) is required to transmit the fixed-size data to receiver. The demultiplexer (De-MUX) is required to distribute the received data with fixed-size to each FFs. The emulation clock is taken a step when the all data are finished receiving. These processes are repeated every emulation clock. In addition, the two kinds of signals are made from the data transfer clock: the control signals of MUX and De-MUX, and the clock enables of receiver's FFs. Therefore, these signals are not synchronized with the emulation clock. The transceiver's FFs is extracted from the partitioned circuit. In addition, the receiver's FFs are prepared the same number as the transceiver's FFs and are aligned.

III. Circuit partitioning technique

A. Issue for the circuit partitioning

In order to emulate the circuit by using the ASIC-emulator shown in Fig.1, the cut-planes of the partitioned circuits must have FFs. The simple partition compliant to this constraint is achieved with the hMETIS. The hMETIS divide the circuit using hyper-graph. However, the hyper-graph cannot store the information of the external inputs and that of the logic-elements which included in a cone. This incurs the following two issues: the replication of external inputs and the replication of logic elements. The former is induced when the cone which uses the external input is replicated at the circuit partition. The external inputs must synchronously input to the replicated cones. This may decrease the emulation frequency because it needs to keep enough the setup time of FFs of each FPGAs. The latter increases the resource use of FPGAs.

B. Proposed circuit partitioning technique

B.1. Solution for external input replication

This subsection describes the algorithm which suppresses the input replication.

(1) The initial placement of the logic-cones

In the first step, we align the cones to ease the selection of the division point. The initial placement is decided by instance name of FF that defined in HDL file.

(2) Realignment the logic cone

Secondly, we sort the cones by the external inputs to suppress the replication of the external input. As shown in the Fig.1, there are three kinds of the cone input : the external inputs, the inputs via serial link, and the feedback inputs inside an FPGA. It is necessary to pay attention to the input with cone while dividing the circuit in order to suppress the input replication. Therefore, we have to keep connection relations of the input when making cones from the circuit. The sorting procedure is as follows.

1. Count the number of the cones which connect to each external input.
2. Extract the external input in ascending order of the number of connecting modules.
3. Moving the cones connected to each external inputs to the both sides of the row in extracted order.

Fig.2 shows the sorting result. In the process of cone movement, we consider not only the movement but also the suppression of node replication.

(3) Partitioning

Each cone has a different number of logic elements. If we select the partitioning point to the center of the row, the unbalance of the logical elements between the partitioned circuits may occur. Therefore, we move the partitioning point step-by-step from the center depending on the balance of the amount of logic elements.

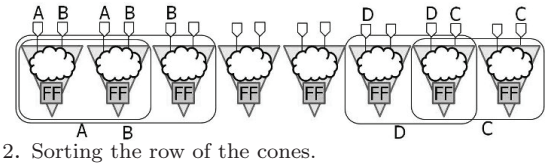

Fig. 2. Sorting the row of the cones.

B.2. Solution for logic cell replication

In order to reduce the replication of logic cells, firstly, we group the cones if they share much number of logic elements. After that, we divide the circuit using the hMETIS. The group of logic elements is handled as a vertex of the hyper-graph. The criterion of merging is the percentage of the logic overlapping between two logic cones, or between a group and a logic cone. In the evaluation, we set it to 70% or more.

IV. Evaluation of Circuit Partitioning techniques

We compared the number of the replicated external inputs and that of the logic elements with hMETIS. Table I shows the benchmark circuits and their characteristics.

TABLE I
BENCHMARK CIRCUITS.

circuit	FF	logic gate	Input
s15850	534	3,448	75
s35932	1,728	12,204	36
s38584	1,426	11,448	39
b18	3,296	92,048	7
b19	6,594	174,157	7
DMA	2,190	19,118	64
DSP	3,607	32,436	63
RISC	7,391	59,974	46

TABLE II
Evaluate the number of replicated external inputs.

benchmark		replicated external inputs			reduction rate[%]	
circuit	inputs	hMETIS	technique1	technique2	technique1	technique2
s15850	75	57	14	32	75.4	43.9
s35932	36	4	4	4	0.0	0.0
s38584	39	26	2	5	92.3	80.8
b18	7	7	2	2	71.4	71.4
b19	7	7	2	5	71.4	28.6
DMA	64	62	25	60	59.7	3.23
DSP	63	29	18	23	37.9	20.7
RISC	46	17	9	13	47.1	23.5
Avg.					56.4	33.9

TABLE III
Evaluate the number of replicated logic elements.

benchmark		replicated nodes			reduction rate[%]	
circuit	nodes	hMETIS	technique1	technique2	technique1	technique2
s15850	3,448	1,882	521	295	72.3	84.3
s35932	12,204	72	3,576	70	-4833.7	2.8
s38584	11,448	1,349	1,208	289	10.5	78.6
b18	92,048	13,905	23,688	702	-70.4	95.0
b19	174,157	20,805	50,846	109	-144.4	99.5
DMA	19,118	2,914	1,601	839	45.1	71.2
DSP	32,436	5,626	11,624	1,256	-106.61	77.7
RISC	59,974	3,370	16,299	1,179	-383.7	65.0
Avg.					-680.5	71.8

(1) The results of replicated external inputs

Table II shows the number of replicated external inputs. The technique 1 described in section III.B.1 reduces by 56.4% of the technique using hMETIS in average, but it cannot reduce the inputs of s35932. This is because s35932 only has external inputs for a cone, or for almost all cones.

(2) The results of replicated logic elements

Table III shows the number of replicated logic elements. The technique 2 described in section III.B.2 reduces by 71.8%. In contrast, there is a case that the unexpected result is obtained in the technique 1 because the circuit such as s35932 has the pre-mentioned external inputs. If we use technique 1, such a case appears because the logic-cone sorting and the suppressing of the reduction of logic elements and external inputs are operated at the same time. Therefore, the circuit with the external inputs like s35932 should use another partition technique.

V. Conclusion

The ASIC-emulator incurs the following two issues when partitioning the circuit: the replication of external inputs and the replication of logic elements. In this paper, we present a study of new circuit partition techniques to solve these issues and compared with hMETIS. As a result, it achieved average 56.4% reduction in the technique for suppressing the reproduction of external inputs. In the technique for suppressing the reproduction of nodes, it achieved average 71.8% reduction. But in the case of former especially, some circuits have the large number of replicated logic elements. In the future, we are going to improve the algorithm based on these results.

References

[1] J.Babb, R.Tessier and A.Agaral, "Virtual wires : Overcoming pin limitations in FPGA-based logic emulators," IEEE Workshop on FCCM, pp.142-151, 1993.

[2] L.Li and C.Tropper, "A design-driven partitioning algorithm for distributed Verilog simulation," Workshop on Principles of Advanced and Distributed Simulation, pp. 211-218, 2007.