Performance Evaluation of Dynamic Partial Reconfiguration on FPGA by Hardware Time Division Execution

Satoshi Yonemitsu and Akira Yamawaki*

Kyushu Institute of Technology, Kitakyushu, 804-8550 Japan

*Corresponding Author: yama@ecs.kyutech.jp

Abstract

In recent years, the development of mobile terminals such as smartphone is remarkable. However, along with its evolution, there is a problem of increasing the cost and the power consumption by increasing the chip size of system-on-chip (SoC) in terminal. One of approaches tackling this problem is to take advantage of Dynamic Partial Reconfiguration (DPR) on reconfigurable devices. In DPR, large-scale circuits can be time-divisionally executed on a small die in SoC. In other words, cost reduction and power saving can be achieved with the minimum necessary chip size. In this research, we attempt to show a performance merit of DPR by performing a time division execution of a large scale circuit on small portion in a reconfigurable device, Field Programmable Gate Array (FPGA). In addition, we try to estimate how much cost can be reduced in the current general FPGA by using DPR.

Keywords: FPGA, Dynamic Partial Reconfiguration, Time division execution, Performance, Cost.

1. Introduction

In recent years, the development of mobile terminals such as smartphones is remarkable. It maintains a small size, but has many functions. They can be realized by the evolution of System-on-chip (SoC) in the terminal. However, with the development of its performance, increase in chip cost and power consumption due to large scale of chips in the terminal has become a problem now.

One way to solve the problem is Dynamic Partial Reconfiguration (DPR) using reconfigurable devices. Although there are large number of prior research using DPR technology for the data processing applications ⁽¹⁻³⁾, In DPR, large-scale circuits can be time-divisionally executed

with small-scale chips. That is, cost reduction and power saving, can be achieved with the minimum chip size required. Regarding the increase in power consumption which is one of the tasks, an attempt was made to reduce power consumption using DPR and approximate values were calculated ⁽⁴⁾.

Therefore, in this research, we study the cost reduction. First, we will examine the performance by parallel processing of one task. This is because meaning can be found in intensive processing if the processing speed is doubled when the processing circuit is doubled. Next, we compare the performance of two tasks when parallel dedicated hardware is used and the case where time division is performed using DPR in one circuit part. Finally, we estimate how much cost reduction could be achieved as a result of this research based on current standard market price of FPGA.

2. Time division processing and parallel processing of hardware

Fig 1. An ideal parallel hardware

Fig 2. An ideal time division hardware

2.1 Parallel processing of ideal hardware

An ideal parallel hardware configuration is shown in figure 1. In this figure, dedicated hardware corresponding to the process exists, and each hardware has a dedicated interface with the memory, so it is possible to read and write the memory in parallel. Also, since each hardware can operate independently, the performance as hardware is high.

2.2 Time division processing of ideal hardware

Ideal time division hardware is shown in Figure 2. Since this hardware is small in scale, large-scale circuits can't be mounted at the same time. In addition, since the interface with the memory exists only in one piece of hardware, the throughput becomes low. However, by mounting and processing dedicated circuits on a time-division basis, necessary resources are reduced, so that cost and power consumption can be kept low.

2.3 Comparison between parallel and time division

Figure 3 shows a comparison between parallel

Fig 3. Comparison between parallel and time division

Fig 4. Outline of the parallel processing

processing and time division processing and time division processing in ideal hardware. First, the processing performance is compared. In parallel processing, each dedicated hardware can perform memory access, so it has high performance. In the time division processing becomes sequential, and the processing performance becomes low. Next compare the costs. When parallel processing is required, all the circuits necessary for processing must be mounted at the same time, so the cost is high. On the other hand, the time division processing is low in cost because the scale of the device itself does not increase even if the content of processing increases. Finally, we compare the power consumption. Since it can be considered to depend on the scale and capacity of hardware, it is higher in parallel processing and lower in time division processing.

3. Basic study on parallel processing performance

3.1 Overview of parallel processing evaluation

In this chapter, we evaluate the parallel processing performance of hardware for one task. This task is image processing, and in this research we use our own grayscale circuit. The outline of the evaluation method is shown in Figure 4. First, one grayscale circuit is implemented and the processing performance of one image is measured with a counter created inside the hardware. Next, two grayscale circuit are implemented, one image is processed by two circuits, and its performance is measured with a counter. In addition, the same applies to the case of four grayscale circuits. By this method, it is evaluated whether parallelization of circuits in small scale hardware is effective for high speed processing.

Tab 1. Processing time in parallel processing

Circuits\Size	256*256	960*640	1280*720	1920*1080
1	524314	4915227	7309667	16393664
2	516123	4838429	7257637	16329626
4	511342	4800989	7176227	16174000

3.2 Evaluation of parallel processing performance

Table 1 shows the number of implemented grayscale circuits, the size of processed images, and processing time. The processing time indicates how many 10 ns clocks were measured in the processing. For example, when processing an image with a size of 256*256 with one grayscale circuit, the processing time is 524,314, that is, 5,243,140 ns, which is about 5.24 ms.

It can be seen from Table 1 that the decrease in processing time due to the number of processing circuits is small. In other words, as shown in Figure 5 (a) and (b), it means parallelization does not hold the hypothesis that the processing time is simply halved. Therefore, in small-scale hardware, speeding up by parallel processing is ineffective, and it became clear that the effect corresponding to the resource to be used can't be obtained. A consideration on this cause is shown in Figure 6. In small-scale hardware,

Fig 5. Comparison between parallel and time division

Fig 6. Consideration of processing time

Fig 7. Outline of the time division processing
memory access can't parallelize, so it spends a lot of time. Since it is only a short image processing time in the parallelized part, it is considered that no significant change in processing time is seen as a result.

4. Evaluation of time division processing

4.1 Overview of time division processing evaluation

In this chapter, we evaluate the time division processing performance of hardware for two tasks. The outline of the evaluation method is shown in Figure 7.

First, time division processing will be described. In time division processing, one grayscale circuit is used. For two tasks, one grayscale circuit is used for time division processing. After processing task 1, switch to circuit for processing task 2 using DPR on FPGA. Then, task 2 is processed and the total processing time is taken as the result. Next, parallel processing will be described. In parallel processing, dedicated hardware for task 1 and task 2 is implemented. Each circuit is activated simultaneously with the start of the measurement, the image is read from the memory, processing is performed, and measurement is ended at the stage where it is output to the memory.

4.2 Processing performance result

Table 2 shows the processing time in the case of time division processing for two tasks in each image size. In addition, the ratio of Actual processing, DPR, and Memory access at the total processing time is shown in Figure 8 (a) (b) (c), and (d). In table 2, the DPR time is almost constant

Tab 2. Processing time of each process in time division

Type\Size	256*256	960*640	1280*720	1920*1020
Actual Processing	16384	153600	230400	518400
DPR	1823506	1813603	1682247	1765600
Memory access	1032218	9676827	14515227	32289976
Total	2872109	11644031	16427874	34573977

Fig 8. Ratio of processing time in time division irrespective of the image size, and it can be seen that as the image size increases, the actual processing time and the memory access time increase. Figure 8 indicates in the processing of a small image such as 256*256, the ratio of the DPR in the total processing time becomes high, but as the image size increases, the ratio of the DPR in the total processing time is low.

Table 3 shows the processing time when two tasks of each image size are processed in parallel by two dedicated hardware. The condition in this case is the same as the one doubling the task of parallel processing in two grayscale circuits in Table 1, and it is simply about twice the processing time in each image size.

4.3 Comparison of processing performance

The results of time division processing and parallel processing verified in this chapter are summarized in Table 4. The figure uses the result of image size of 1920*1080. The number of processing circuits is one in the time division processing and two in the parallel processing. Since the Actual processing time becomes shorter by the number of circuits being implemented, parallel processing is earlier. Memory access can't be parallelized, and there is no temporal difference between time division processing and parallel processing. In time division processing, it takes time to switch circuits by executing DPR. As a result,

Tab 3. Processing time in parallel processing

Size	256*256	960*640	1280*720	1920*1020
Processing time	1032331	9676830	14515227	32659228

Tab 4. Summary of processing performance

Picture size 1920*1080		
Type	Time division	Parallel
Processing circuits	1	2
Actual processing	518400	259200
Memory access	32289976	32400028
DPR	1765600	0
Total	34573977	32659228

parallel processing was able to process 1914749 clocks, or about 19.1ms earlier. However, since only one grayscale circuit is implemented at the same time in time division processing, it is substantially half the size of parallel processing. In other words, it means that processing can be performed at almost the same speed with half the circuit scale, and time division processing using DPR can contribute to cost reduction.

5. Estimation of cost reduction

In this research, we verified the cost reduction and power saving by time division processing of large scale circuit with small scale circuit using reconfigurable device. In this chapter, we will estimate the cost reduction of the time division processing shown in this research based on the current market price of FPGA. An estimate of cost reduction by time division processing in the verification result is shown in Figure 9. In this figure, reference was made to the circuit size and price of Xilinx Spartan-6 LX series. The horizontal axis of the graph is the total number of slices indicating the size of the FPGA and vertical axis shows the relative cost with the minimum scale FPGA being 1. For example, the numerical values 600 and 1 of the device A in the figure indicate that the device has a cost 1 with a slice number of 600. Table 4 shows that the time division processing and the parallel processing have almost

Fig 9. Estimation of cost reduction

the same performance in the processing of the full high vision image of 1920*1080, and in addition, the time division processing can be performed on half the scale of the parallel processing. When this is applied to a graph, this means that large-scale processing that had been done with device B with the number of slices 23000 and cost 19.8 can be performed with almost the same processing performance using device C with slices 11700 and cost 10.1. Its cost reduction rate is about 49%, and it can be said that time division processing using DPR is effective for cost reduction.

6. Conclusion

In this research, time division and parallel processing in ideal hardware are compared in terms of performance, cost and power consumption, and a basic study on performance when one task is processed in parallel by multiple processing circuits has been carried out. Furthermore, the effect of time division processing with a small scale device is shown by comparing the case where the two tasks are time divisionally processed using the DPR of the FPGA and the case where it is processed in parallel by two dedicated hardware. We estimated this result based on the market price of the current FPGA and showed that cost reduction of up to about 49% can be achieved.

References

- (1) Seyyed Mahdi Najmabadi, Zhe Wang, Yousef Baroud, and Sven Simon : "Online Bandwidth Reduction Using Dynamic Partial Reconfiguration", Proceedings of the 24th IEEE International Symposium on Field Programmable Custom Computing Machines, pp. 168-171, 2016
- (2) Gianluca Durelli, Fabrizio Spada, Riccardo Cattaneo, Christian Pilato, Danilo Pau, and Macro D. Santambrogio : "Adaptive Raytracing Implementation using Partial Dynamic Reconfiguration", Proceedings of 2014 IEEE 28th International Parallel & Distributed Processing Symposium Workshops, pp. 236-242, 2014
- (3) Gayathri R Prabhu, Bibin Johnson, and Sheeba Rani J : "FPGA based Scalable Fixed Point QRD core using Dynamic Partial Reconfiguration", Proceedings of 2015 28th International Conference on VLSI Design and 2015 14th International Conference on Embedded Systems, pp.345-350, 2015
- (4) Satoshi Yonemitsu, Sota Kono, Akira Yamawaki :

"Power Saving Effect of Dynamic Partial Reconfiguration on FPGA" Proceedings of The 5th International Conference on Industrial Application Engineering 2017, pp. 165-170, 2017