A Design and Performance Evaluation of Dynamically Self-Reconfigurable System Based on Virtex5

Liang Zhang[1a] , PeiYi Shen[1b] , YuXin Cai[1c] , JingWen Liu[2d],Juan Song[2e],LuKui Zhi[2f],LuoBing Dong[2g]

[1]Software Engineering,Xidian University,Xi'an, Shaanxi, China

Software Engineering,Science and Technology, Xidian University,Public Security Department

[a]liangzhang@xidian.edu.cn,[b]pyshen@xidian.edu.cn,[c]caiyuxin@stu.xidian.edu.cn

[d]liujingwen010@163.com,[e]h.uchuan2008@163.com, [f]sxsgatkjc@163.com,[g]dongluobing@163.com

Keywords-FPGA; dynamic partial reconfiguration; Virtex-5; time-consuming model

Abstract: Facing with limitation of hardware resources and increasingly requirements of processing ability, reconfigurable computing technology has become an inevitable trend in compute-intensive system. Dynamic and Partial Reconfiguration (DPR) is a special feature embedded in Field Programmable Gate Arrays (FPGAs), giving designer the ability to reconfigure a certain portion of the FPGA during run-time without influencing the other parts. This paper makes an analysis of time consumption of FPGA-based dynamic reconfiguration and proposes a piecewise reconfigurable system time-consuming model for Xilinx Virtex-5, and then analyzes and obtains the time-consuming parameters of each stage. In addition, an entire time-consuming calculation formula of reconfiguration process is introduced by combining all stages formulas. The experiments results indicate that calculation formula error rate of the reconfiguration time-consuming model is less than 10%, and which can verify the accuracy of the formula.

Introduction

Dynamic reconfiguration technology refers to making dynamic configuration for FPGA logic functions in the real time operation of system. The technology can be divided into dynamic global or partial reconfiguration according to the scope of the reconstructed hardware structure. Dynamic global reconfiguration means that we need to reconfigure the entire FPGA chip during remodeling, while dynamic partial reconfiguration refers to that partial FPGA resources can be dynamic reconfigured in runtime.

In DPR's research, an important direction is mainly focused on implementing and performing the time-consuming estimation of the DPR system. Precise time-consuming model can help user to control reconfigurable model in reconfiguration system, and improve system's robustness. In recent years, many researchers have been working the DPR model estimation for different FPGA system, and analyzing the key parameters which affect the DPR model.

Papadimitriou[6]et.al. presented a general framework for measuring the reconfiguration time of the FPGA from the perspective of system. Xilinx Virtex-II was used as the hardware platform, and Compact Flash(CF) storage media was used to store global or partial configuration file. Internal Configuration Access Port(ICAP) is used as reconfiguration port. In addition, they made use of customer reconfiguration controller based on OPB bus to analyze and test the reconfiguration time consumption.

Claus[8] et.al. introduced a mult-platform controller allowing for maximum dynamic partial reconfiguration throughput for Virtex-II and Virtex-4. They use two methods to measure reconfiguration time. The first one uses a hardware counter built into the PLB ICAP controller. The second method is to use an interrupt which can notify the CPU that the reconfiguration process has completed.

LiuClaus[11] et.al. discussed the multiple ICAP design architectures and thoroughly investigated their performance with measurement for different size of partial bitstream based on Xilinx PR

technology and ICAP primitive in Virtex-4 FPGA. They also did experiments to investigate the performance of various design structures about DMA_HWICAP and MST_HWICAP.

Although there have been an enormous amount of studies about DPR model, there is no research of reconfigurable system time-consuming model for Xilinx Virtex-5 series, and the reconfiguration time estimation accuracy of existing models are not high. Therefore, on the basis of these studies, this paper proposes a piecewise and higher accuracy performance assessment model and calculation formula for the DPR system based on Virtex-5 LX110T hardware. By analyzing the parameters of time consumption with each stage, an entire DPR model estimation expression is obtained, and it is also proven that the error rate is an order of magnitude reduction compared with previous results.

The rest of the paper is organized as follows. Section 2 presents a common set of performance evaluation model for DPR Soc self-reconfigurable system. Section 3 obtains each part formula of the assessment model and derives the time-consuming calculation formula which error rate is within the threshold range. Section 4 makes qualitative and quantitative analysis of the formula. Section 5 summarizes the full text.

PERFORMANCE EVALUATION MODEL

This section proposes a common performance evaluation model for self-reconfigurable systems of Virtex-5 FPGA family. The generic structure of the Virtex-5 Pro FPGA is shown in Fig.1, which consists of soft-core processor Microblaze(MB), that is used for reconfiguration control and management, the high-bandwidth processor local bus(PLB), the dedicated reconfiguration controller XPSHWICAP based on PLB bus, the storage media Compact Flash(CF) which is used for storing partial reconfigurable file and SysACE controller which used for controlling access to CF external memory. In reconfiguration process, when users want to load the configuration file of reconfiguration, the MB processor will call SysACE controller to access the partial configuration files which are stored in CF. Then the configuration information will be written to XPSHWICAP's FIFO registers. Finally, the reconfiguration data will be transferred into the configuration space of FPGA through the ICAP port.

Reconfiguration time(RT) measurements are mainly concentrated on three stages, The detailed description is shown in Fig.1.

- CF-MB is the access time of data being transferred from the CF memory card to a local BRAM within MB processor cache;
- MB-ICAP is the processing time of data being restructured and the writing time of data being transferred from local BRAM MB cache to ICAP device FIFO;
- ICAP-CM is the configuration time of data being transferred from the write FIFO memory to the FPGA through the ICAP port;

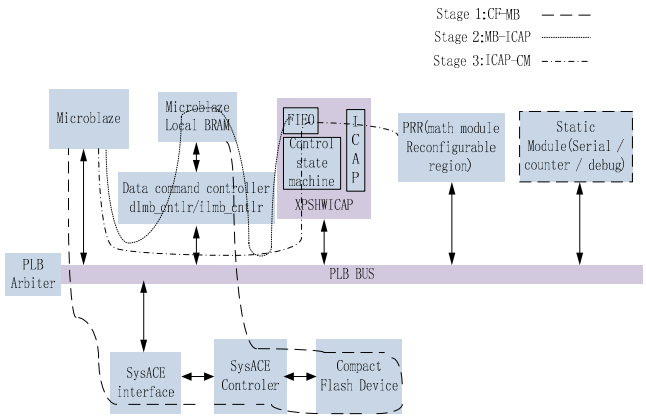

Figure 1. the reconfiguration time-consuming model of DPR SoC system

In addition to those three stages, taking into account some special application scenarios of DPR SoC system, there are some other time-consuming stages which should be considered, such as the analysis time of configuration code, the time of initializing and starting reconfigurable device, the

time of processor sending instruction to XPSHWICAP or OPBHWICAP controller, the copy time of configuration data being transferred from user space into the kernel, these kinds of time are classified as additional time overhead RT_{addition}, and we treat them as constants, and then the RT time computational formula of DPR SoC self-reconfigurable system can be obtained, and it is shown as (1).

$$\text{RT} = RT_{\text{CF-MB}} + RT_{MB-ICAP} + RT_{ICAP-MB} + RT_{addition} \quad (1)$$

RECONFIGURATION TIME MODEL PIECEWISE DELAYS MEASUREMENT

Based on other DPR research[6][8][11], four key parameters which influence DPR system reconfiguration can be extracted, the parameter settings of ICAP configuration port, the design of reconfigurable controller, the selection of memory type, bus types and their settings. This section focuses on making more comprehensive analysis and expansion of parameters library which affects the reconfiguration performance, while tries to derive the time-consuming calculation formula of different stages and ensure that the error rate is within the range of tolerable threshold values.

A. SysACE controller bandwidth measurement

Some DPR researches which are mentioned in section 1 shows that storage medium has an important influence on the reconfiguration throughput of DPR SoC system[5][6], in the particular cases of storage medium(CF), this section focuses on analyzing and calculating the quantitative influence of SysACE access efficiency (bandwidth) on the CF-MB phase time-consuming, the measurement system is shown in Fig.2.

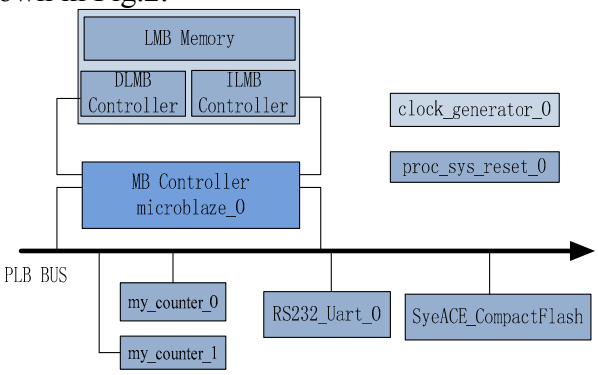

Figure 2. the hardware system architecture of SysACE bandwidth measurement

In Fig.2, custom counter core is used as the hardware timer. At the same time, the bandwidth of the controller mainly depends on the efficiency of SysACE read function, so the time measurement of software application is only executed on this API. The clock frequency is 100MHZ, and the measured time-consuming data about reading different size of files in CF card is shown in Table1.

Based on the data of Table 1, we conduct an intuitive statistics. It can be concluded that the time of reading file presents the stability linear relationship with size of the file. The theoretical value of bandwidth in Table1 is corresponds to the slope of the line in Fig.3, they are all maintained at 0.63 MB/s constant value around.

The average value of $\overline{\text{Bandwidth}}$ is used as a reference bandwidth of SysACE module, and we can obtain $\overline{\text{Bandwidth}} = 0.63418\ MB\ /\ s$ based on the data of Table 1. Therefore, when the length of configuration file is L, we can further obtain the time-consuming formula of CF-MB stage, as the following formula (2) shows.

$$\text{RT}_{\text{CF-MB}} = \text{L} * 10^{-3} / 0.63418 \quad (2)$$

TABLE I. MEASURENMENT DATA OF READING DIFFERENT SIZE OF FILES

Size of file(bytes)	Clock period	Time(s)	Bandwidth(MB/s)
8502454	1284697577	12.85	0.630350195
10634408	1589834609	12.89	0.635220126
9790335	1479949647	14.80	0.630405405
6714651	1012053941	10.12	0.632411067
7247302	1072404234	10.72	0.633656716
14070571	2111296088	21.11	0.634770251
63644956	9587934592	92.88	0.632105977
4547133	684125570	6.84	0.634502924

In (2), *bytes* is used as unit of L, *MB / s* is used as unit of the average bandwidth of SysACE controller module, and thus we can infer that the unit of $RT_{\text{CF-MB}}$ is milliseconds.

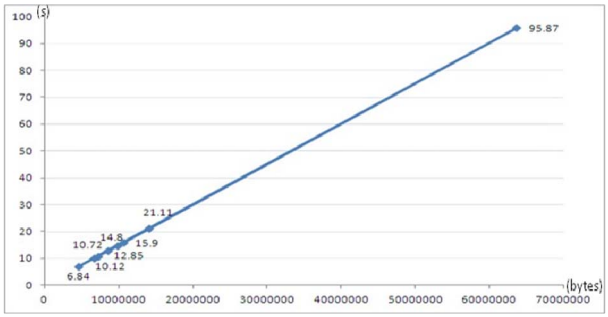

Figure 3. the intuitive charts of time-consuming measurements

B. *XPSHWICAP design and time overhead measurements*

XPSHWICAP is a dedicated reconfiguration controller based on PLB bus, and it gradually replaces OPBHWICAP controller based on OPB bus on account of its performance advantage. Its influence on the time-consuming stage is concentrated on the MB-ICAP process, and it has a direct effect on $^{\text{RT}}_{\text{MB-ICAP}}$. In order to obtain accurate time-consuming formula, this section designs and implements the application of XPSHWICAP driven design by setting the single-pass amount of data transmission that is written into XPSHWICAP's register. The part code of XPSHWICAP driven design is shown in Table 2.

In code design, the processor writes cycle_num*sizeof(word) size data to XPSHWICAP FIFO each time. The cycle_num user interface is provided by our design approach, users can dynamically adjust the amount of data which XHwIcap_DeviceWrite function transmits depending on the practical application.

To measure the cost time of MB-ICAP stage, under the premise of XPSHWICAP's FIFO depth being 1024*32(bit), we set different values about the amount of data transmission in XHwIcap_DeviceWrite function, the test results are shown in Table 3. From Table 3, we can see that with the index rises of each transmission count, the time overhead of MB-ICAP phase exhibits decreased tendency.

TABLE II. THE PART CODE OF XPSHWICAP DRIVEN DESIGN

```
Xuint word[1024];
int cycle_num,index_cmd;
for (i=0; i<BitstreamLength; i+=4)
{ for(index_cmd=0;index_cmd<cycle_num;index_cmd++)
  {  i+=4;
    if (i<BitstreamLength)
    { data <- systemACE_Buffer[0..7];
     word[index_cmd] <- data[0..3]; }
   else
     word[1]=0x0;
  }
}
XHwIcap_DeviceWrite(&HwIcap, word, cycle_num);
```

TABLE III. SINGLE-PASS AMOUNT OF TRANSMISSION AND TIME OVERHEAD MEASUREMENT TABLE

XHwIcap_DeviceWrite() amount of data transmission	Time overhead (ms)	write FIFO(bit)
2word	599040	1024*32
4word	357120	1024*32
8word	236160	1024*32
16word	175680	1024*32
32word	148320	1024*32
64word	134640	1024*32
128word	127800	1024*32
256word	127144	1024*32
512word	130848	1024*32
1024word	130332	1024*32

We will take the time-consuming data that is measured by hardware timer in Table 3 as the valid data, and draw fitting curve by Matlab. Then the quantitative relationship between API transmission data volume and time consumption of MB-ICAP stage can be obtained, and is shown in Fig.4. $RT_{MB\text{-}ICAP}$ stands for the time consumption of MB-ICAP stage, N stands for single-pass amount of data transmission in *XHwIcap_DeviceWrite* function, L stands for the size of configuration file. Thus the formula of $RT_{MB\text{-}ICAP}$ can be approximately obtained and shown in (3). In (3), *bytes* is the unit of L, *ms* is the unit of time overhead.

$$RT_{\text{MB-ICAP}} = \frac{8.398 * \exp(-0.3195 * N) * L}{24576} + \frac{1.47 * \exp(-0.0001801 * N) * L}{24576} \quad (3)$$

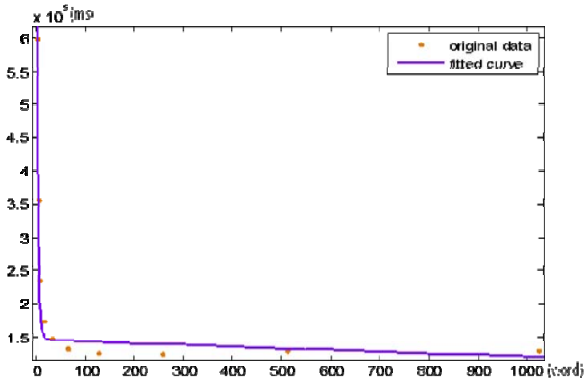

Figure 4. the fitting curve between amount of data transmission and MB-ICAP time consumption

C. *Time-consuming calculation formula derivation*

In our performance evaluation model, it is easy to obtain the time measurement data of CF-MB and MB-ICAP stages by means of hardware timer or software timestamp. Whereas, the time-consuming data magnitude of ICAP-CM stage is too small, and also tends to a constant value according to device itself, it is difficult to find workable measurement solution. To simplify the time-consuming model, we use the empirical data of manufacturer to replace the time overhead of ICAP-CM stage. To trigger ICAP port configuration operation for FPGA, Virtex devices need to fill 2KB data in ICAP at one time, the average time overhead of this process is 0.02526ms.When the size of configuration file is L bytes, the time overhead of $^{RT_{ICAP\text{-}CM}}$ stage satisfies the following formula(4).

$$RT_{ICAP\text{-}CM} = \frac{0.02526 * L}{2048} \tag{4}$$

We respectively plug (2), (3) and (4) in (1), and then the whole time-consuming (5) can be obtained, meanwhile the final reconfiguration time (6) can be gained by further simplifying the calculation (5).

$$RT = \frac{L*10^{-3}}{0.63418} + \frac{8.398*\exp(-0.3195*N)*L}{24576} + \frac{1.47*\exp(-0.0001801*N)*L}{24576} + \frac{0.02526*L}{2048} + RT_{addition} \tag{5}$$

$$RT = L*\{0.00159 + 0.000342*\exp(-0.3195*N) + 0.0000598*\exp(-0.0001801*N)\} + RT_{addition} \tag{6}$$

(6)

Finally, after getting the whole RT time-consuming calculation expression of reconfiguration operation, we can deduce the formula of reconfiguration throughput by using (7) between ARTP and RT, ARTP is Actual Reconfiguration Throughput, BS is the size of partial configuration file, the ARTP expression is shown in (8).

$$ARTP = \frac{BS}{RT} \tag{7}$$

$$ARTP = \frac{L}{0.00159 + 0.000342*\exp(-0.3195*N) + 0.0000598*\exp(-0.0001801*N) + \frac{L}{RT_{addition}}} \tag{8}$$

The time consumption of all stages in DPR SoC system uses *ms* as unit, meanwhile the size of configuration file uses *bytes* as unit, so reconfiguration throughput *ARTP* uses *Kbytes/s* as the unit of measurement.

Verification and analysis of time-consuming model

Section 3 deduces the accuracy formula of performance evaluation model, this section will make an analysis of qualitative and quantitative for the formula.

D. Qualitative verification and analysis of formula

This section will analyze and verify the correctness of (6) from the perspective of qualitative, and the reference model that this section uses is the one which Kyprianos Papadimiriou[12] proposed. Their model also uses CF as storage medium of partial and global configuration file, the difference is that processor which is used for reconfiguration management is higher performance PowerPC hardware processor, the whole hardware architecture of DPR Soc system is greatly match with this paper's design. In Kyprianos Papadimiriou performance evaluation model, the global reconfiguration time is divided into three stages: CF - PPC, PPC, and ICAP - CM stage. And the scale drawing of piecewise delays about their evaluation model is shown in Fig.5.

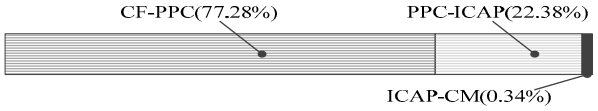

Figure 5. the distribution ratio of different stages about reference model

Fig.5 is proportional distribution of reconfiguration time overhead. First, it validates the correctness of our derivation which ICAP-CM stage brings quite small impact on FPGA refresh and time consumption, In other words, $^{RT_{ICAP\text{-}CM}}$ reflects the whole time-consuming accuracy of DPR reconfiguration system and has weak effect on global.

Second, we respectively calculate time overhead of reconfiguration stages by (2), (3) and (4). In calculation process, the size of reconfiguration file is 24576 bytes and the single-pass amount of data transmission in XHwIcap_DeviceWrite function is 2 words. Meanwhile, because there is no additional overhead time in our DPR system, assuming $^{RT_{addition}}$ =0, the time-consuming calculation results of the different stages are shown as follows.

$$RT_{CF-MB} = 38.75\,ms,\ RT_{MB-ICAP} = 5.9021\,ms,\ RT_{ICAP-CM} = 0.3031\,ms.$$

Thus we can obtain time-consuming ratio:

$$RT_{CF-MB}\% = 86.05\%,\ RT_{\text{MB-ICAP}}\% = 13.28\%,\ RT_{\text{ICAP-CM}}\% = 0.67\%.$$

The proportional distribution of model which this paper proposed is shown in Fig.6. We make comparison with Fig.5, and it can be seen that it is closer to Kyprianos Papadimiriou's reference model, and the existing difference is mainly caused by efficiency of PPC hard-core and MB soft-core processor accessing to the CF peripheral. Thus it verifies the feasibility of time-consuming calculation formula of various stages and (6) from the perspective of qualitative.

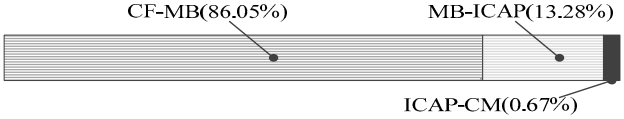

Figure 6. the distribution ratio of reconfiguration stages about this paper

Whether it is Kyprianos Papadimiriou's time-consuming scale drawing or our, which all reflect one phenomenon that the overhead of reconfigurable systems is mainly concentrated in the first stage CF-MB. In other words, in order to significantly improve the throughput of DPR system, it is important to change the storage medium access efficiency and bandwidth. The important work of this paper focus on reducing the time-consuming of MB-ICAP phase by optimizing API access efficiency of XPSHWICAP hardcore, if we can replace CF storage medium which this paper chose to DDR, the overall overhead of DPR system can be reduced fully in theory, and then we can obtain theoretical optimal reconfiguration throughput.

E. Verification and analysis of quantitative formulas

In section 4.1, we verify the inferred formula of this paper from the qualitative relationship, this section will analyze and determine the correctness of formula from the perspective of quantitative by obtaining the relative error between reconfiguration throughput theoretical values and real measured values.

The reconfiguration throughput theoretical values can be obtained from (2) ~ (4), (8). This paper uses 24576 bytes size of reconfigurable file, when the amount of API data transmission is 2 words, the results about overhead of various stages and the global reconfiguration throughput of DPR system are summarized as follows.

$$RT_{CF-MB} = 38.75\ ms$$

$$RT_{MB-ICAP} = 5.9021\ ms$$

$$RT_{ICAP-CM} = 0.3031\ ms$$

So the value of $ARTP_{\text{calc}}$ is 546.653Kbytes/s.

In order to verify theoretical throughput reliability, we will calculate relative error by the following (9).

$$\text{ERROR}\% = \frac{|\ ARTP_{calc} - ARTP_{actual}\ |}{ARTP_{actual}} \qquad (9)$$

To get real measured values of reconfiguration throughput, this paper measures reconfiguration time aiming at the DPR system of Fig.1, and reconfigures custom math module by the use of multiplication, addition and non-functional partial reconfiguration bitstream file(mult.bit, adder.bit and blank.bit). Meanwhile, the reconfiguration time overhead of CF-MB and MB-ICAP stages are measured, the results are shown in Table 4.

TABLE IV. THE RECONFIGURATION TIME CONSUMPTION OF CF-MB AND MB-ICAP PHASES

Reconfigurable file	File size (bytes)	CF-MB (10^{-5}ms)	MB-ICAP (10^{-5}ms)
mult.bit	24576	4159298	596160
	24576	4165320	596160
	24576	4167295	596160
adder.bit	24576	4138756	596160
	24576	4139193	596160
	24576	4140136	596160
blank.bit	24576	3962590	596160
	24576	3965279	596160
	24576	3963654	596160

After acquiring the theoretical value and the real measured value of reconfiguration throughput, we can calculate relative error between them by combining with (9). The calculation process is shown as following.

$$ARTP_{\text{actual}} = \frac{24576}{(41.59298 + 5.96160)} = 516.796\ \text{(Kbytes/s)}$$

$$ERROR\ \ \% = \frac{|\ 546\ .653\ -\ 516\ .796\ |}{516\ .796} = 5\ .78\ \%$$

We will compare the relative error of this paper with the other references[6][8][11], and the results are shown in Table 5. Compared with the performance evaluation of reconfiguration system in other schemes, our design improves measurement precision by one order of magnitude, and has higher reliability.

Conclusion

A performance evaluation model and time-consuming calculation formula of DPR system based on Virtex-5 FPGA are proposed in this paper. We have verified the relative error of reconfiguration time-consuming calculation formula which this paper proposed is closer to 5.78%

The performance evaluation model can be ported to any reconfigurable platform with embedded processor and the external memory for storing the partial bitstreams, such as platform that incorporate the hardcore PowerPC or the softcore Microblaze MB and the ICAP port, and contain a CF and Sys ACE controller[13].

Moreover, we have provided detail data concerning different sizes of reconfiguration bitstreams loaded from a CF memory and time-consuming data concerning the different amount of data transmission in MB-ICAP stage. In addition, the relative error between real measured value and theoretical value of this paper is compared with other reference research, and it is proven that the accuracy of performance evaluation model which this paper presents is higher.

TABLE V. THE RECONFIGURATION TIME CONSUMPTION OF CF-MB AND MB-ICAP PHASES

References	Storage Medium	File size(bytes)	$ARTP_{calc}$ (KB/s)	$ARTP_{actual}$ (KB/s)	ERROR %
Liu[11](2009)	DDR2	79900	970.84	589.23	39.3%
Liu[11](2009)	DDR2	75900	15180	9730.8	32.9%
Claus[8](2008)	DDR	70500	12589.3	4659.6	62.9%
Papadimitriou[6] (2010)	CF	14600	252.60	144.4	42.8%
This design	CF	24576	546.65	516.8	5.78%

Acknowledgment

This project is supported by NSFC Grant(No.61305109, No.61072105), by 863 Program(2013AA014601), and by Shaanxi Scientific research plan(2013K06-09).

References

[1] Fong R. J., Harper S. J., Athanas P. M. A Versatile Framework for FPGA File Updates: An Application of Partial Self-Reconfiguration. In Proceedings of the IEEE International Workshop on Rapid System Prototyping (RSP), 2003, 117-123.

[2] Griese B., Vonnahme E., Porrmann M., Ruckert U., Hardware Support for Dynamic Reconfiguration in Reconfigurable SoC Architectures. In Proceedings of the International Workshop on Field Programmable Logic and Applications (FPL). 2004, 842-846.

[3] Gelado I., Morancho E., Navarro N., Experimental Support for Reconfigurable Application-Specific Accelerators in Proceedings of the Workshop on the Interaction between Operating Systems and Computer Architecture (WIOSCA), in conjuction with the International Symposium on Computer Architecture (ISCA). 2006, 50-57.

[4] Delahaye J. P., Palicot J., Moy C., Leray P., Partial Reconfiguration of FPGAs for Dynamical Reconfiguration of a Software Radio Platform. In Mobile and Wireless Communications Summit, 16th IST, 2007, 1-5.

[5] Papadimitriou K., Anyfantis A., Dollas A., Methodology and Experimental Setup for the Determination of System-level Dynamic Reconfiguration Overhead. In Proceedings of the IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM). 2007, 335-336.

[6] Papadimitriou K., Anyfantis A., Dollas A., An Effective Framework to Evaluate Dynamic Partial Reconfiguration in FPGA Systems. IEEE Transactions on Instrumentation and Measurement, 2010, 1642-1651.

[7] Claus C., Muller F. H., Zeppenfeld J., Stechele W., A New Framework to Accelerate Virtex-II Pro Dynamic Partial Self-Reconfiguration. In Proceedings of the IEEE International Parallel and Distributed Processing Symposium (IPDPS). 2007, 1-7.

[8] Claus C., Zhang B., Stechele W., Braun L., Hubner M., Becker J., A Multi-Platform Controller Allowing for Maximum Dynamic Partial Reconfiguration Throughput. In Proceedings of the IEEE International Conference on Field Programmable Logic and Applica-tions (FPL). 2008, 535-538.

[9] French M., Anderson E., Kang D. I., Autonomous System on a Chip Adaptation through Partial Runtime Reconfiguration. In Proceedings of the IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM). 2008, 77-86.

[10] Manet P., Maufroid D., Tosi L., Gailliard G., Mulertt O., An Evaluation of Dynamic Partial Reconfiguration for Signal and Image Processing in Professional Electronics Applications. EURASIP Journal on Embedded Systems, 2008.

[11] Liu M., Kuehn W., Lu Z., Jantsch A., Run-Time Partial Reconfiguration Speed Investigation and Architectural Design Space Exploration. In Proceedings of the IEEE International Workshop on Field Programmable Logic and Applications (FPL). 2009, 498-502.

[12] Kyprianos Papadimiriou, Apostolos Dollas. etc. Performance of Partial Reconfiguration in FPGA Systems: A Survey and a Cost Model. ACM Joural. 16-24.

[13] Digilent Inc., 2008. [Online]. Available: http://www.digilentinc.com

[14] XILINX INC., Partial Reconfiguration User Guide. UG702 (v13.1) March 1, 2011.

[15] Chun-Hsian Huang, Pao-Ann Hsiung, Hardware Resource Virtualization for Dynamically Partially Reconfigurable Systems. IEEE Embedded Systems Letters, Vol. 1, No. 1, May 2009.

[16] Ciaron Murphy, Virtual Hardware Using Dynamic Reconfigurable Field Programmable Gate Arrays.Engineering Development Centre (EDC), GERI, Liverpool John Moores University, UK.