

Complexity and Performance Evaluation of Two Partial Reconfiguration Interfaces on FPGAs: a Case Study¹

Heng Tan, Ronald F. DeMara,
and Anuja J. Thakkar
ECE Department
University of Central Florida
Orlando, FL 32816-2450
demara@mail.ucf.edu

Abdel Ejnoui
Information Technology Dept.
University of South Florida
Lakeland, FL 33803-9807
aejnoui@lakeland.usf.edu

Jason D. Sattler
Space Photonics Inc.
700 Research Center Blvd.
Fayetteville, AR 72701
jsattler@spacephotonics.com

Abstract

In this paper, experiments are conducted in order to quantitatively evaluate the tradeoffs between design complexity and area overhead, reconfiguration flexibility, and reconfiguration latency of two reconfiguration interfaces, i.e. SelectMAP and JTAG. The results show that the SelectMAP interface is highly suitable when reconfiguration latency needs to be kept to a minimum at the expense of a large area overhead and more rigid area control. On the other hand, the JTAG interface is preferable if significant control over the placement of reconfigurable modules in the reconfigurable fabric of the FPGA chip is desired and/or more flexible area management is required. The JTAG design consumes a factor of 3 to 7 times fewer logic resources and a third of the device pins, but can incur reconfiguration latency up to 40 times longer than the SelectMAP design.

1 Introduction

One of the major benefits provided by FPGAs is dynamic reconfiguration ability, which involves altering the programmed design within a SRAM-based Field Programmable Gate Array (FPGA) at run-time. Recently, various applications have considered embracing this capability for performance and efficiency purposes, such as SOC designs and automatic resource defragmentation. This paper addresses basic design considerations when partial reconfiguration techniques are integrated into a reconfigurable system using two major commercially interfaces respectively, namely the JTAG and SelectMAP interfaces. Spatial and temporal performances of each interface are also quantitatively evaluated herein.

2 Prototype Designs

To evaluate the two interfaces, two prototypes with verification modules and communication channels have been developed. for Xilinx Virtex II Pro platforms, supporting SelectMap-based [1] and JTAG-based approaches respectively. Both solutions were based on two Commercial Off The Shelf (COTS) independent hardware units: the Avnet Virtex II Pro development board and a Pentium Host PC. The development board is equipped with the Virtex II Pro XC2VP7 chip. The module based partial reconfiguration flow [2] proposed by Xilinx was also used in designing of these prototypes.

The SelectMap prototype hardware subsystem developed includes two resource subsets, *system resources* and *operational resources*. The system resources are composed of an on-chip PowerPC 405 CPU core, on-chip Block RAMs, and all the onboard peripherals such as the SRAM modules and RS232 interface. The operational resources consist of the actual modules instantiated inside the Virtex II Pro chip. These modules consist of a *fixed subset* and a *reconfigurable subset*. The fixed operational resources includes multiple interfaces to control the external on-board SRAM modules, the RS232 resources, the Block RAMs controllers and a three-segment path including an On-chip Peripheral Bus (OPB), a Processor Local Bus (PLB) and a Bridge Core (PLB2OPB) providing access to the OPB from the PLB. All the remaining resources in the fabric comprise the reconfigurable subset. In order to maintain correct communications with PowerPC through the OPB bus with those reconfigurable modules, simplified *Intellectual Property Interface (IPIF)* and *Bus Macro* structures have been adopted [1]. For testing purposes, the high-level communication channel needs to be bi-directional. The input path begins at the PC which transmits reconfiguration data over the PCI bus to the specific address of an on-board SRAM module, and

¹ This project supported in-part by USAF under contract FA8651-05-C-0221. AAC/PA 01-11-06-014, ref. no. 1267.

then sets the data availability flag. At the same time, the PowerPC polls the flag status, reads data from the addressed SRAM module, and copies data to the hardware component for processing when the data flag is set. On the other hand, the output path data goes with a similar mechanism with a reversed direction.

Because JTAG is a very popular interface, numerous commercial tools are available for testing and debugging purposes. In our JTAG prototype, Chipscope cores [3] from Xilinx have been chosen for verification, which eliminate the requirements for complicated bi-directional communication channels as required with the SelectMAP interface. Therefore, the hardware platform has been greatly simplified. To carry out the reconfiguration operation automatically, an API from Xilinx application note xapp058 is adopted [4]. Instead of using a `.bit` file directly, this API uses the `.xsvf` format. The API reads and interprets the XSVF contents, generates the programming instructions, data, and control signals for the Xilinx devices, and sends them through the JTAG interface. With this API, reconfiguration process can be easily integrated into high-level control logics running on the Host PC.

3 Results and Analysis

The SelectMAP reconfigurable module case study is selected as a 16-bit SECEDED circuit while the JTAG design is an equivalent size 8×8 multiplier. Table I shows the design complexity for configuration and verification. In order to establish the bi-direction communication channel, the design associated with the SelectMAP interface requires eight 32-bit wide IP cores inside the fixed region. On the other hand, the JTAG interface only uses four additional Chipscope cores to carry out the testing with variable data width. Therefore the SelectMAP design consumes a factor of 3 to 7 times more various logic resources in the fabric than JTAG does. These costs can limit the size and area placement flexibility of the reconfigurable modules. Therefore when using SelectMAP interface, large capacity FPGAs, such as the Virtex-II Pro X2VP20 or above, are highly recommended. Furthermore, additional effort is also required for pins assignments and connections with customized bus macros possibly required thereby resulting in an increase in design complexity.

Table I: Resource Utilization

Interface	# of Fixed Modules	# of Pin of Fixed Modules	Reconfigurable module overhead	Slices for Fixed Modules	BRAM for Fixed Modules	TBUF for Fixed Modules	PPC405
SelectMAP	8	77	7 slices	1352	8	352	Y
JTAG	4	25	0	473	0	48	N

Table II: Timing Evaluation

Interface	Full Reconfiguration	Data Communication
SelectMAP	536 msec	123 msec
JTAG	20.3 sec	N/A

As shown in Table II, a single data processing cycle, which starts with sending the data out to the PCI bus and ends with reading the data back from the SRAM after the hardware and PowerPC processing is complete, takes up to 123 msec. On the other hand, in the case of the JTAG interface, the reconfiguration process based on the same size reconfiguration file requires 20.3 seconds to complete. Hence the reconfiguration latency observed is 40 times slower than the SelectMAP interface. This indicates the magnitude of benefit possible using SelectMAP if it is desired to keep reconfiguration latencies low.

4 Conclusion

Experiments show that the reconfiguration speed of the SelectMap interface is almost 40 times faster than the speed of the JTAG interface. However, communication logic and pin assignment are comparatively simpler in the JTAG interface. This simplicity allows higher flexibility in the placement of the reconfigurable modules since the JTAG interface does not consume significant reconfigurable resources within the FPGA chip.

References

- [1] Heng Tan and Ronald F. DeMara, "A Device-Controlled Dynamic Configuration Framework Supporting Heterogeneous Resource Management," in *Proceedings of Engineering of Reconfigurable System and Algorithms (ERSA 05)*, pp. 251 – 254, Las Vegas, NV, U.S.A, June 27 – 30, 2005.
- [2] Xilinx, Inc., *Two Flows for Partial Reconfiguration: Module Based or Difference Based*, v1.1, Nov. 2003.
- [3] Xilinx, Inc., *Chipscope Pro Software User Guide*, v6.3.1, October 2004.
- [4] Xilinx, Inc., *Xilinx In-System Programming using an Embedded Microcontroller*, v3.1, June 2004.