Indian J.Sci.Res. 17(2): 204-207, 2018 ISSN: 0976-2876 (Print)
ISSN: 2250-0138(Online)

A SURVEY ON DYNAMIC PARTIAL RECONFIGURATION IN FPGAs

[1]M.Manasa
[1]PG Student, Department of Electronics and Communication Engineering, University College of Engineering, Osmania University, Hyderabad

Abstract- Dynamic partial reconfiguration is the ability to reconfigure some portions on FPGA during its run time. It offers new design space with various benefits, reduce the configuration time and save memory as the partial reconfiguration files are smaller than full ones and also reduce the dynamic power consumption saves the resources. This survey paper focuses on the implementation and the challenges to the system.

Keywords-Dynamic partial econfiguration, FPGA, Dynamic power consumption.

I. Introduction

Field programmable gate arrays (FPGA), functionality can be changed by configuring it with bit stream and it offers Massive parallelism during its operation. Because of its flexibility, it can be adoptable to various applications. Dynamic Partial reconfiguration is the Process of configuring some of the selected blocks on FPGA, while the remaining part continues its operation. i.e., the selected regions on the FPGA will be loaded with new bit streams during run time. This is especially valuable where devices operate in a mission-critical environment that cannot be disrupted while some subsystems are being redefined.Its uses can be best used in the applications where FPGA need to be act as an intelligent device, to change its reaction according to the operating environment. Partial reconfiguration is suitable for designs with many permutations that do not operate simultaneously and hence can share the resources on FPGA. In such systems, one section of FPGA continues to operate, while other section is reconfigured with new functionality. This started from tiny reconfigurable modifications of a CPU, over complex video processing and database acceleration up to software defined radio applications. Common for all these applications is that they benefit from being able to relocate modules to different positions and to pack multiple modules together into a shared reconfigurable region.

Partial reconfiguration provides the designers with the benefits of reduced device count, reduced power and overall reduced cost. When used in conjunction with the dynamic reconfiguration of transceivers, it provides a flexible solution to implement high Bandwidth applications.

Field programmable gate arrays (FPGAs) provide an interesting solution when custom logic is needed for short time to market products. The products embedding FPGA System on chip solutions allow them to be updated once deployed. Recent FPGA architectures, such as Xilinx Virtex Series, allow for partial and dynamic run-time reconfiguration. The FPGA fabric can modify its configuration data at run-time, enabling substitution of specific portions of an implemented hardware design causing the system to be adapted to the needs of the application. Hence reliability, failure-redundancy and run-time adaptively by usage of dynamic partial reconfiguration are introduced that are critical aspects for embedded systems.

As the FPGA comprised of multiple modules, few modules, which are to be dynamically reconfigured are clustered leaving the static modules. We have certain limitations to fully reconfigure the FPGA dynamically, that will be explained in Chapter II, and it also covers basic block diagram, design methodology. Chapter III explains about the challenges faced by the technology, Chapter IV gives details of advantages and applications and Chapter V concludes the topic.

II. Dynamic Partial Reconfiguration

Design on FPGA is implemented as modules, in partial reconfiguration some of the selected modules are reconfigured, while in full reconfiguration all the modules need to be reconfigured but this takes more time to reconfigure and more power consumption compared to partial reconfiguration. It needs more number of bit streams to be loaded and cost also increases.

Below diagram shows the difference between partial and fully reconfigured system's in terms of bit

[1]Corresponding Author

stream files, it is obvious that number of bit stream files to fully reconfigure are more than the partial reconfiguration.

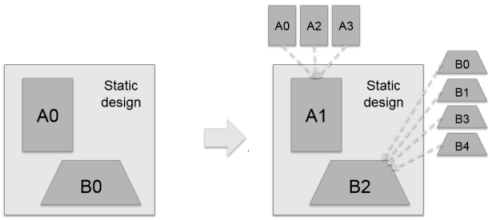

Fig 1: difference between full and partial reconfiguration

Below diagram shows the reconfigurable modules A and B leaving the static modules, these reconfigurable modules will have multiple bit stream files, based on the application requirement appropriate file will be configured into the module.

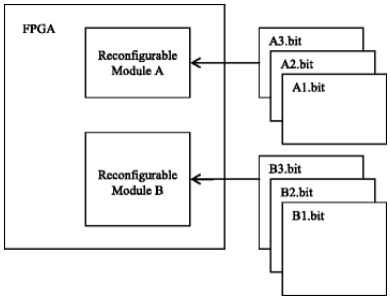

Fig 2: Multiple bit streams of Reconfigurable module

Below diagram shows the block level implementation of Partial reconfiguration,the system is connected to an external Flash memory via a Flash controller which provides read/write access to the external memory. The ICAP module is present to read/write configuration data to/from the devices' configuration memory. Reconfiguration of the dynamic region is managed by a Reconfiguration Controller which can be either external or internal (self-configuration) to the FPGA.

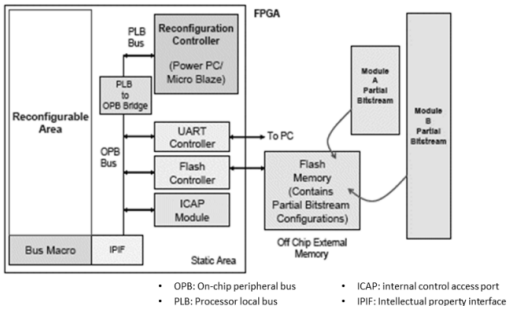

Fig 3: Block diagram of Dynamic Partial reconfiguration

It is responsible for the partial reconfiguration bit stream that must be loaded onto the reconfigurable modular region. However, self-configuration can be carried out through the internal ICAP parallel interface and reduces reconfiguration time as compared to external reconfiguration. For this reason, our reconfiguration controller is present inside the FPGA and communicates to the rest of the system via the Processor Local Bus (PLB). Other peripherals are attached to a slower On-Chip Peripheral Bus (OPB) via the PLB-OPB Bridge. A UART controller is also present to read the initial configuration from an external PC. Self-configuration can be carried out through the internal ICAP parallel interface and reduces reconfiguration time as compared to external re-configuration, controller is present inside the FPGA and communicates to the rest of the system via the Processor Local Bus (PLB). Other peripherals are attached to a slower On-Chip Peripheral Bus (OPB) via the PLB-OPB Bridge. The reconfigurable modules (attached to the bus macros) communicate with the OPB by means of a simplified Intellectual Property Interface (IPIF).

In an SRAM-based FPGA, all user-programmable features are controlled by memory cells that are volatile and must be configured on power-up. These memory cells are known as the *configuration memory*, and they define the look-up table (LUT) equations, signal routing, input/output block (IOB) voltage standards, and all other aspects of the design.

In order to program the configuration memory, instructions for the configuration control logic and data for the configuration memory are provided in the form of a bit stream, which is delivered to the device through the JTAG, Select MAP, serial, or ICAP configuration interface.

In order to implement a partial reconfiguration design successfully, you have to follow a strict design methodology. Here are some guidelines to follow:

- Insert bus macros between modules that need to be swapped out (called partial reconfiguration modules, or PRMs) and the rest of the design (static logic). These are the channels, or ports, through which modules communicate and pass data.
- Follow synthesis guidelines to generate a partially reconfigurable netlist.
- Floorplan the PRMs and cluster all static modules.
- Follow partial reconfiguration specific design rules.

- Run the partial reconfiguration implementation flow.

Initial Xilinx PDR design flows recommended the use of tri-state buffers for communication between dynamic and static parts of the FPGA. They were not very effective leading to a new alternative in the form of predefined routed macros generally termed as bus macros. Bus macros are placed at the edge of boundaries separating dynamic and/or static regions/modules. They allow communication between the different regions of the FPGA. The limitation to this approach was that although signals could pass through the reconfigurable area by use of bus macros, however if a module was being reconfigured, the internal signal route was also re-routed making the parts of the systems on either side of the module isolated from each other. This problem was addressed later on.

Virtex devices support glitchless dynamic reconfiguration: If a configuration bit holds the same value before and after reconfiguration, the resource controlled by that bit does not experience any discontinuity in operation, with the exception of LUT RAM and SRL16 primitives in Virtex-II/Pro devices. This limitation has been removed in the Virtex-4 family.

III. Challenges

Challenges to the dynamic partial reconfiguration can be broadly classified into two categories, there may exist other problems but this classification covers major issues, they are 1.Physical design issues 2.CAD tools

1. Physical design issues

These include the ambiguity on signal values at the interface to the static portion i.e., during reconfiguration of partial region, signal values at the interface may be undetermined. This may affect the entire system operation, static portion on FPGA must be able to handle that.an external enable or disable signal is needed.

The physical locations and the bus width of the interface must be invariant, static portion must not be affected by the floating values on the interface during reconfiguration.

Tristate buffers are used to connect to a shared bus from each partial reconfiguration module, it may reconfigure the selected region at any time, but new FPGA's are not compatible with tristate buffers, now a day's bus macros are using at the interface to handle them.

2 CAD tools

Traditional FPGA CAD tools assume full chip design (similar to ASIC flow). It need smart CAD tools to

- Perform low level compilation for PR
- Provide the needed user abstraction level

During the reconfiguration, need a way to reset or initialize the logic within the partial reconfiguration region, also needs external hardware to handle the clock , it should be in reset mode while reconfiguring and at the same time clock should run in static region.

IV. Advantages and Applications

The major advantage includes resource reuse, i.e., using same hardware multiple applications can be implemented. Design area is reduced. It is also offering low power consumption, flexibility and application portability. Performance will also be increased because of its higher level parallelism, time sliced resource sharing.

Partial reconfiguration is suitable for the designs with many permutations that do not operate simultaneously and hence can share the same resource on FPGA. It can be used in high performance computing systems where frequent algorithm updates are required for different platforms being serviced. Partial reconfiguration supports compression and encryption of the partial bit stream for military applications such as creating enhanced design security applications, where the partial region used to decrypt the incoming bit stream, encrypt a new bit stream, and configure the rest of the device. Partial reconfiguration allows the system to maintain real time links while other modules within the FPGA are being configured.

Applications includes evolvable hardware platforms, modular robotics, Cryptography, telecommunications, networking (exchange of Packet according to traffic), Modulation / frequency / encryption hopping in military radios. In digital signal processing JPEG encoder/decoder systems and Emulator design etc.

V. Conclusion

It is possible to use run time partial reconfiguration to save both power and area, an intelligent run time system is needed to ensure that power is saved and that timing constraints are meet, when using in real time systems. Measuring power consumed during partial reconfiguration aids in determining the design of the run time system and the feasibility of using dynamic partial reconfiguration.

References

[1]. "Dynamic partial reconfiguration on FPGA" Third International Symposium on Intelligent Information Technology Application 2009.

[2] https://www.design-reuse.com/articles/18309/mde-dynamic-reconfiguration-fpga.html

[3] https://www.eetimes.com/document.asp?doc_id

[4] http://www.inf.pucrs.br/~moraes/prototip/artigos /reconf_puc.pdf

[5] A Design Flow for FPGA Partial Dynamic Reconfiguration **DOI:** 10.1109/IMCCC.2012.35

[6] https://en.wikipedia.org/wiki/Reconfigurable_ computing

[7] http://www.fpl2015.org/pdf/sig_papers/dpr/1.pdf