

Architectural Support for Runtime 2D Partial Reconfiguration

Fei Wang and Jack Jean
Department of Computer Science and Engineering
Wright State University, Dayton, OH 45435, U.S.A.
{wang.29, jack.jean}@wright.edu

Abstract: *Traditional FPGA architectures can potentially allow the dynamic swap in and out of hardware tasks through 2D partial reconfiguration. A segmented bus structure is proposed to be superimposed on such architectures. Its purpose is to support connections between on-chip hardware tasks, or between hardware tasks and chip IO pads. The required area overhead is also estimated.*

Keywords: Reconfigurable Computing, FPGA, Partial Reconfiguration

1. Introduction

Consider a 2-dimensional partially reconfigurable FPGA chip that allows the dynamic swap in and out of hardware tasks or IP (intellectual property) circuits. The feasibility of such a chip improves as FPGA chip capacity increases.

When a new hardware circuit is swapped in, it should be connected to either chip I/O pads or other hardware circuits already on the chip. In this paper, such connections are called IP-IO and inter-IP connections, respectively. For example, the new circuit may be an image compression circuit that needs to take inputs from an on-chip memory controller circuit via inter-IP connections and outputs to an off-chip network controller chip via IP-IO connections.

Previous works on operating systems of such machines normally aim to manage on-chip resources efficiently and concentrate on the runtime task placement and scheduling [1] [5]. There is usually no consideration about inter-IP or IP-IO connections. Such connections on commercial FPGA chips require time-consuming routing algorithms that may not be practical for runtime systems. In [4], packet switching was proposed for dynamic multi-tasking using a static torus

network. The segmented bus in this paper is a dynamic network that uses circuit switching.

This paper proposes a new partially reconfigurable FPGA architecture that has two distinct properties:

1. The minimum partial reconfiguration unit is a square cluster, which contains $N \times N$ CLBs (Configurable Logic Blocks). Compared to the non-square unit used in Xilinx Virtex-II chips, i.e., several columns of CLBs as a unit, square clusters can improve chip utilization [6]. A large circuit can be mapped onto more than one cluster.
2. Segmented buses are introduced around clusters so as to support inter-IP and IP-IO communications. These buses are used solely for runtime support. They are not used for “intra-IP” routing which is still supported by traditional FPGA routing resources. At runtime only the segmented buses are to be routed, while the intra-IP communications using the traditional FPGA routing resources are not disturbed. This arrangement is to shorten the runtime routing time.

Section 2 of the paper describes the architecture. Section 3 gives the area cost. Section 4 concludes the paper.

2. Chip Architecture

2.1 CLB Model and CLB Block

The CLB model suggested by Betz in [2] is assumed. In that model a CLB is composed of four flip-flops and four four-input look-up tables. Such a CLB is equivalent to 1,678 minimum width transistors in terms of area. The model is close to that of Xilinx Virtex chips.

A *CLB block* is defined as a CLB plus routing resources associated with it [2]. Since routing resources may even compose 90% of chip area [3], it is reasonable to assume that the area of a CLB block is at least five times as big as that of a CLB itself.

2.2 Segmented Bus

The proposed segmented bus and its accessories are extra resources superimposed on traditional FPGA chips. The bus consists of resources covering two different types of areas:

1. *Backbone network*: this type of resources covers the area between clusters.
2. *Local network*: this type covers the area within clusters.

2.2.1 Backbone Network and Cluster Block

As shown in Figure 1, the backbone network contains crossbars and bus switch blocks. In addition along cluster boundary, there are tri-state buffers that can be used to insulate the cluster from the bus. The bus switch block is dedicated to relaying bus segments and changing bus routing directions. A *cluster block* is composed of the cluster itself, the two crossbars with tri-state buffers, and the bus switch block at its lower and right corner.

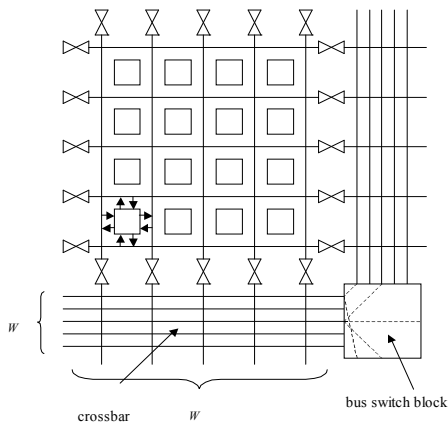


Figure 1. A Cluster Block

In Figure 1, a cluster is connected to the backbone network via W wires going through tri-state buffers. At the crossbar side, these W wires are bi-directional. While at the cluster side, each wire bifurcates into separated incoming/outgoing

bus wires, and they are all unidirectional. This is illustrated in Figure 2.

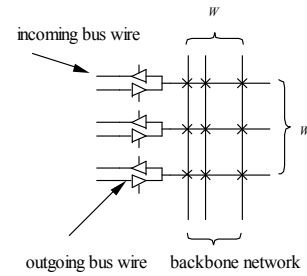


Figure 2. Tri-state Buffers

Assume that there are k incoming bus wires and k outgoing bus wires in each routing channel between neighboring CLBs and there are only $0.5*k$ incoming (outgoing) bus wires at the boundary of each cluster. If a cluster has $N*N$ CLBs, then $W = k*N$. (Let k be called the bus wire density.)

2.2.2 Local Network

In this paper, outgoing bus wires at north, east, west and south sides of a CLB are distinguished using labels $N[0..k-1]$, $E[0..k-1]$, $W[0..k-1]$ and $S[0..k-1]$, respectively. Incoming bus wires are similarly labeled. Bus wires between neighboring CLBs as part of the segmented bus are called the *local network*. CLBs are connected to the local network through output and input connection blocks.

Output Connection Block: Figure 3 indicates how CLB output signals are connected to the local network. One specific path is used to illustrate such connections. Each CLB has four output signals, labeled as f , g , h , and i . One of them goes through the top multiplexer, a turned-on tri-state buffer (marked with a square), a turned-on switch transistor (marked in square), and reaches one of the outgoing bus wires at south side, i.e., $S[0]$ (in bold dash line).

Input Connection Block: Connections between CLBs and incoming bus wires are shown in Figure 4. Each CLB has four look-up tables, each associated with four 14:1 multiplexers. To accommodate two incoming bus wires the 14:1 multiplexer should be expanded into a 16:1 one. Corresponding to the two additional inputs, two NOMS transistors with the minimum channel width are added. Therefore for each CLB, 32 extra transistors are needed.

Each CLB is associated with four groups (NEWS) of k incoming bus wires each. At one time only one out of them are selected via two 4:1 multiplexer. Hence at most two out of k signals can be selected to feed into the CLB. Incoming bus wires and wires associated with those 4:1 multiplexers compose crossbars, again with NMOS switch transistors with minimum channel at intersection points.

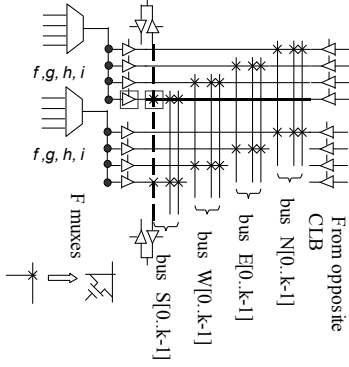


Figure 3. CLB Outputs to the Local Network

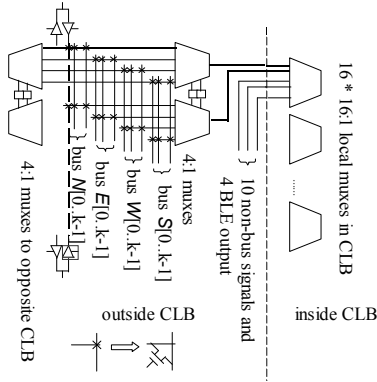


Figure 4. CLB Inputs from the Local Network

3. Area Cost

In this section, without performing physical layout, the area associated with the segmented bus is calculated using notations and values as summarized in Table 1 following Betz's conventions [2]. In the table the unit for the "Size" column is in terms of the number of minimum NMOS transistors while areas for multiplexers and tri-state buffers include memory cells associated with selection lines and control lines.

Table 1: Notations and Values Used

Notation	Definition	Size
A_{mcel}	one-bit memory cell	6
$A_{mux(H)}$	H :1 mutiplexer	
$A_{mux(2)}$	2:1 mutiplexer	8
$A_{mux(3)}$	3:1 mutiplexer	16
$A_{mux(4)}$	4:1 mutiplexer	18
A_{3st}	tri-state buffer (using a 1-bit memory cell)	20
A_{3st_bnd}	tri-state buffer at cluster boundary (using a 2-bit memory cell)	31

The area cost depends on a variety of parameters, such as k , N and popularities of crossbars, P_{CRS_BAR} . For a crossbar, the popularity is the percentage of intersection points that are deployed with switch transistors. Three different types of crossbars are used. They are associated with the backbone network, CLB input connection blocks and CLB output connection blocks.

3.1 Total Segmented Bus Area

Suppose the chip has totally $m*m$ clusters. There are $2*m$ backbone network crossbars at top and left boundary that are not associated with any cluster block. The total segmented bus area is as follows.

$$A_{CHIP} = m*m*A_{CLUST} + 2*m*A_{CRS_BAR}$$

where A_{CLUST} is the extra area per cluster block and A_{CRS_BAR} is the extra area per crossbar. And

$$A_{CRS_BAR} = 7*W*W*P_{CRS_BAR}$$

$$A_{CLUST} = N*N*A_{CLB} + 2*A_{CRS_BAR} + 12*W*A_{3st} + 8*W*A_{3st_bnd}.$$

Here A_{CLB} is the extra area per CLB which is due to the local network. That is,

$$A_{CLB} = 0.5*(k*4*F + k*4*F*A_{mcel}) * P_{OUT} + F*4*A_{3st} + F*A_{mux(5-F)} + 0.5*(k*4 + k*4*A_{mcel}) * P_{IN} + 2*A_{mux(4)} - 2*A_{mcel} + 32$$

where P_{OUT} (P_{IN}) is the popularity of the output (input) connection block, and F is the number of output signals that can reach the local network. (F is less than four.)

The area A_{CHIP} is expressed in the number of transistors. It can be converted into the equivalent

number of CLB blocks since a CLB block contains 8,390 (=1,678*5) transistors.

3.2. Area Overhead Examples

Figure 5 shows the segmented bus area overhead as the cluster size N changes. The overhead is the percentage of the segmented bus area over the total chip area.

The segmented bus area is based on the following assumptions: $N \leq 14$, $k \leq 8$, $F=2$, and all popularities P are equal to one. For a specific bus wire density k and a fixed cluster size N , the number of clusters a chip has (i.e., m^2) is chosen so that the resulting total chip area is closest to 120×120 CLB blocks, close to a high-end Xilinx Virtex chip. (The value of m falls in the range between $\lfloor \frac{120}{N+1} \rfloor$ and $\lceil \frac{120}{N} \rceil$.)

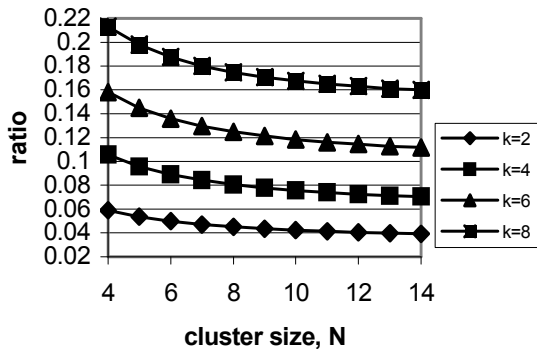


Figure 5. Bus Overhead versus N ($P=1$)

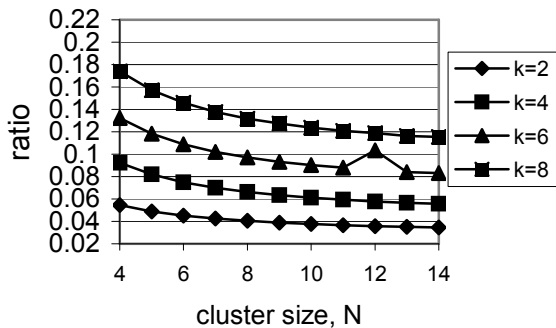


Figure 6. Bus Overhead versus N ($P=0.5$)

From Figure 5, it can be found that

- (1) The smaller the cluster size (N) is, the bigger the segmented bus overhead is.
- (2) The addition of two wires in each channel (k) incurs no more than 5% area increase.

- (3) The proposed segmented bus is not very expensive. For cases of $k \leq 4$, bus overhead is mostly less than 10%. With $k=6$, the bus overhead in most situations is in the range of 11% ~ 15%; with $k=8$, 16% ~ 21%.

In Figure 6 it is assumed that all those crossbars are only 50% populated. As a result, the bus area overhead has been reduced. This is obtained at the expense of losing some routing flexibility.

4. Conclusions

A segmented bus structure is proposed to be superimposed on an FPGA architecture so as to support on-chip inter-IP or IO-IP connections under a runtime 2D partial reconfiguration environment. The area overhead is calculated and the bus is found not very expensive. To quantify the benefits in a runtime environment, the prototype of an operating system is being developed based on this new chip architecture. Dynamic IP circuit placement and routing are part of the efforts.

References

- [1] K. Bazargan, R. Kastner, and M. Sarrafzadeh, "Fast Template Placement for Reconfigurable Computing Systems", IEEE Design and Test of Computers, vol. 17, no. 1, pp. 68-83, 2000.
- [2] V. Betz, Jonathan Rose and Alexander Marquardt, "Architecture and CAD for Deep-Submicron FPGAs", Kluwer Academic Publishers, 1999.
- [3] Andre Dehon, "Reconfigurable Architectures for General-Purpose Computing", Ph.D. thesis, MIT, 1996.
- [4] Theodore Marescaux, Andrei Bartic etc., "Interconnection Networks Enable Fine-Grain Dynamic Multi-tasking on FPGAs", FPL02, Montpellier, France, pp. 795-805, Sep. 2002.
- [5] Christoph Steiger, Herber Walder and Macro Platzner, "Operating System for Reconfigurable Embedded Platforms: Online Scheduling of Real-Time Tasks", IEEE Transactions on Computers, Vol. 53, No.11, pp. 1393-1407, Nov. 2004.
- [6] Fei Wang, Jack Jean and Shuxia Sun, "Aspect Ratio Effects on Reconfigurable Computing", ERSA05, Las Vegas, Nevada, USA, June 2005.