

Platform-based FPGA Architecture: Designing High-Performance and Low-Power Routing Structure for Realizing DSP Applications

K. Siozios, K. Tatas, D. Soudris and A. Thanailakis

VLSI Design and Testing Center, Department of Electrical and Computer Engineering
Democritus University of Thrace, 67100, Xanthi, Greece
{ksiop, ktatas, dsoudris, thanail}@ee.duth.gr

Abstract

The novel design of an efficient FPGA interconnection architecture with multiple Switch Boxes (SB) and hardwired connections for realizing data intensive applications (i.e. DSP applications), is introduced. For that purpose, after exhaustive exploration, we modify the routing architecture through efficient selection of the appropriate switch box with hardwired connections, taking into account the statistical and spatial routing restrictions of DSP applications mapped onto FPGA. More specifically, we propose a new technique for selecting the appropriate combination of switch boxes, depending on the localized performance and power consumption requirements of each specific region of FPGA architecture. In order to perform the mapping, we developed a novel algorithm, which takes into account the modified architectural routing features. This algorithm was implemented within a new tool called EX-VPR. Using a number of DSP applications, extensive comparison study of various combinations of switch boxes in terms of total power consumption, performance, Power \times Delay product prove the effectiveness of the proposed approach.

1. Introduction

FPGA technology offers design flexibility due to its programmability features, which are supported by quite mature design flows [11, 12]. Moreover, an FPGA architecture characteristic changed and improved significantly the last two decades, from a simple homogeneous architecture with logic modules, and horizontal and vertical interconnections to FPGA platforms (e.g. Virtex-4 family), which include except logic and routing, microprocessors, block RAMs etc. The changes/improvements can be also seen in many relevant references [4, 10, 11, 12]. Furthermore, the FPGA architecture changed gradually from homogeneous and regular architecture to a heterogeneous (or piece-wise homogeneous) and irregular (or piece-wise regular). The

platform-based design allows to designer to build a customized FPGA architecture, depending on the application domain requirements. Consequently, selecting appropriate logic modules, memory units, processors etc, a customized solution can be achieved. The platform-based strategy changed the FPGAs role from a “general-purpose” machine to an “application-domain” machine, closing the gap with ASIC solutions. Having in mind the current trend about the design FPGA architecture, we proposed a new software-supported methodology for selecting appropriate interconnection architecture.

More specifically, since general-purpose FPGA provide inefficient solutions in terms of high performance and low-power consumption, researchers started to build new FPGA architectures with specific features, depending on the constraints of the considered application domain, e.g., DSP applications [11, 13], telecom applications [1], embedded processor and high speed connectivity [11]. For example, due to the fact that the addition/accumulation operation appears in data-intensive applications, specific CLB module for optimizing the carry propagation was designed [11]. In [9], a new technique was proposed for eliminating switches in two dimensions, using hardwired junctions between horizontal and vertical segments inside switch boxes.

Due to the fact that about 60% of an FPGA area is consumed by routing resources [2], there is an effort for minimizing this percentage, leading to smaller devices, achieving higher frequencies, and consuming less power. One of the most critical components of the routing is the switch box (SB) that connects the horizontal and vertical routing tracks. A number of different SBs, each of which has its own advantages/disadvantages were proposed [4].

Here, we propose a novel methodology for designing a high-performance and low-power interconnection structure of an island style-based FPGA platform. The essential contribution of this paper is the use of more than one SB. The basic idea behind the novel methodology is to find out the optimal combination of different SBs taking into account the application characteristics. The efficiency of an SB is characterized by analyzing parameters like power dissipation, performance, as well

as the number of required tracks for successful application routing.

Using the appropriate tools and the specific characteristics from a specific application domain (e.g. DSP), the designer can build the appropriate routing architecture for that domain, meeting the input specifications. In other words, the designer can handle the routing components as building blocks of a platform-based design approach.

Taking into consideration that the FPGA resources are not full-utilized, we make an exhaustive exploration with all a number of DSP applications, to find out the optimal combination from three existing SBs (Wilton [4], Universal [4] and Subset [4]), in terms of how many different SBs will compose the FPGA architecture, what type of SBs are the most suitable, and which is the ratio of each one. The possible combinations of SBs were evaluated in terms of power consumption, delay and area. Furthermore, the Xilinx Virtex is the most power aware but it is not so efficient in performance and the required number of routing tracks. On the other hand, the Universal SB tries to minimize the routing channel width, while the Wilton seems to balance all parameters.

2. FPGA Architecture with Multiple Switch Boxes (SBs)

In this section, we discuss the spatial and statistical information of SB connections and their impact on FPGA architecture. We introduce a new method for deriving special maps, each of which describes the number, and the location (spatial) of used transistors of SBs. In order to build these maps, a number of DSP applications, the EX-VPR tool [6], Virtex-like FPGA architecture [3] and Altera Stratix [12] are used. Fig.1 shows the steps of the proposed design methodology for deriving performance and power efficient routing architecture.

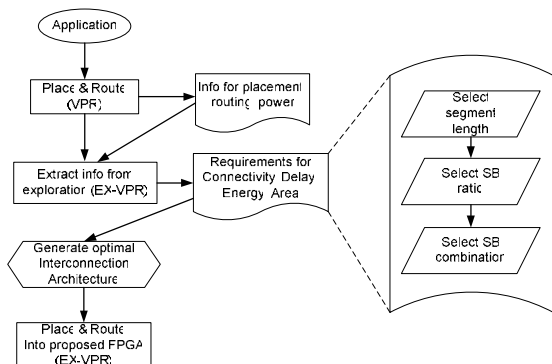


Fig. 1: The proposed methodology

2.1 Spatial Information

The first step of the methodology is to find out the performance, the power consumption, the connectivity and the area of DSP applications. Particularly, by the term connectivity we define the total number of connections, i.e. the “ON” pass-transistors, which take place into a SB. A specific map (or set of curves) can be created for each aforementioned design parameter, which shows the parameter variation across (X,Y)-plane of the whole FPGA device. Fig. 2 presents in normalized manner a picture, which concerns the total connectivity of all the kinds of DSP applications of the whole FPGA. Similar 3-D graphs can be derived for any application-domain specific benchmark or application. It can be seen that the connectivity varies from point to point of FPGA and the number of used pass-transistors (i.e. ‘ON’ connections) decreases gradually from the center of FPGA architecture to the I/O blocks. The connectivity requirement for more tracks in the center of the device, compared to the borders depends on the chosen routing algorithm [5]. The essential message from the 3-D curve of Fig. 2 is the fact that the interconnection resources are not fully-utilized. So, the big challenge which should be tackled by a designer is to increase the hardware utilization. Moreover, the interconnection resources are responsible for the largest portion (>60%) of the power consumption tradeoffs [13]. Consequently, implementing an FPGA with the necessary only hardware resources, significant power consumption and delay reduction can be achieved.

Additionally, a second important conclusion draws from Fig. 2. Although a typical FPGA architecture is a homogeneous and regular one (i.e. just repetition of tile blocks), the actually-used hardware resources provide a non-homogeneous and irregular picture. Ideally, we had to use different interconnection architecture at each (x,y) point of FPGA, which would lead to a totally irregular FPGA architecture increasing, among others, NRE fabrication costs. However, such “extreme” architecture is the optimum solution for implementing applications on FPGA architecture, but apparently it is not a practical and cost-effective implementation. For that purpose, we propose a piecewise-homogeneous and regular FPGA architecture with a few different FPGA regions.

If we define a certain threshold of the connectivity value and project the 3-D diagram to (X,Y) plane of FPGA, we create maps which depict the corresponding for connectivity requirements. Considering connectivity requirements of DSP applications mapped on conventional FPGAs, respectively. The number of distinct regions is based only to the design tradeoffs. By increasing the number of regions, the FPGA becomes more heterogeneous, as it is consisted by more regions. On the other hand, this increase leads to performance

improvement for the device, due to the better routing resources utilization.

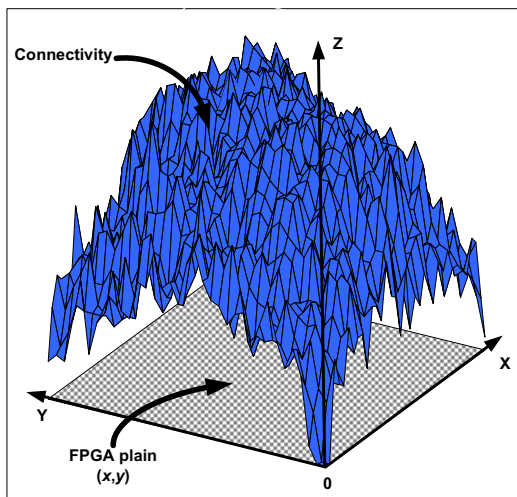


Fig. 2: Overall connectivity across the whole FPGA

The power dissipation is critical issue of an FPGA design process. Since the power consumed in routing is more than 60% of total power of the FPGA device [1], the proposed technique aims at the minimization of this factor. For that purpose, we take into account the SB pass-transistors utilization in the various regions of FPGA map, shown in Fig. 3(a) and (b). Thus, in regions with smaller connectivity (i.e. less transistors) we can use appropriate type of SB with low-power features. The connectivity degree of any (x,y) point of FPGA array is directly related with the power consumption of (x,y) SB location, since less number of active SB connections means less power consumption. Fig. 4(a) and (b) are maps depicting the power dissipation consumed by a single type SB into a homogeneous FPGA device. The power consumption estimations are provided by PowerModel [8]. The introduction of power consumption map is very useful instrument to FPGA device designers to specify the power consumption over each (x,y) point of FPGA device. Determining the “hot spots” locations of FPGA device, the designer can concentrate his/her efforts for power consumption reduction on certain regions only, but not on the whole device. Comparing the connectivity map (Fig. 3(a) and (b)) with the power dissipation map (Fig. 4(a) and (b)), it can be easily concluded that the corresponding maps are similar due to proportional relation between connectivity degree and power consumption.

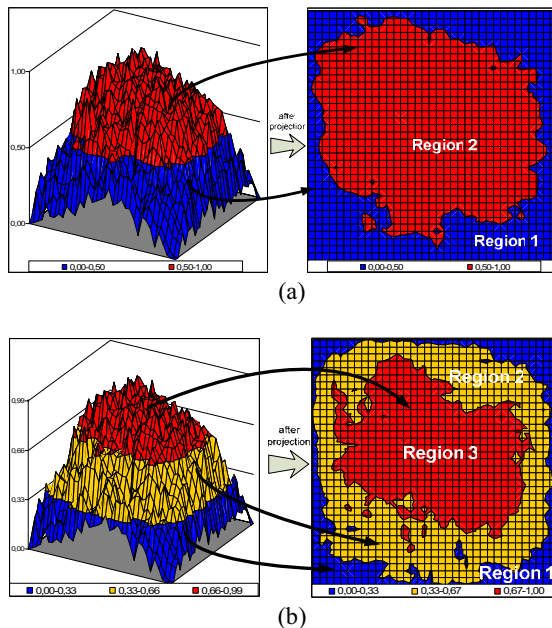


Fig.3: Connectivity requirements for (a) two and (b) three SB regions. Each region shows the percentage of active connections

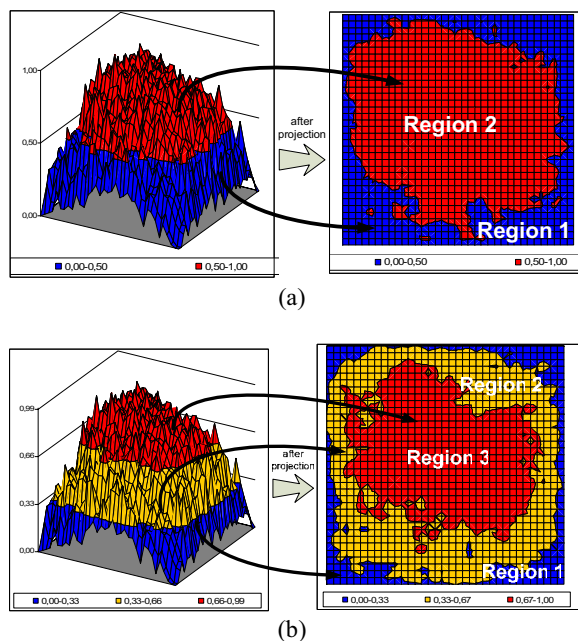


Fig.4: Power dissipation in SBs (a) for two regions, (b) for three regions

Furthermore, increasing the number of distinct SB regions, the designer can identify in more detailed manner the spatial distribution of power consumption and therefore, he/she can choose the most appropriate SB for each region at the expense of the increased heterogeneity of FPGA features. On the other hand, increase of the SB regions has a penalty at the fabrication cost of the device. In this paper, we choose a connectivity factor threshold equals two.

Fig. 5 shows the utilized channel distribution (normalized values) across the whole FPGA device, considering the DSP applications. We can easily infer that the density of the utilized routing tracks is larger at the middle of the device compared to these at bottom and top. Employing the same DSPs, the exploration results show that about of 15% of the available routing tracks in the device periphery, and about of 75% of the available routing tracks in the middle of the FPGA are utilized by the mapped applications.

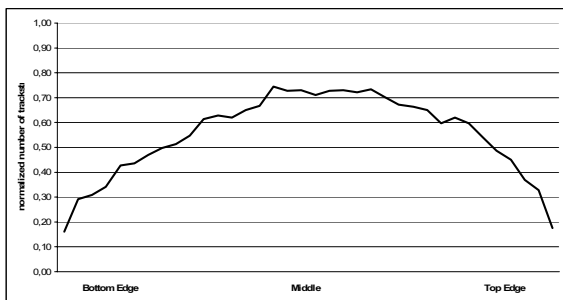


Fig. 5: Channel distribution across the FPGA

We can deduct that as we increase the number of distinct SB regions, the designer can identify in more detailed manner the spatial distribution of power consumption and therefore, he/she can chose the more appropriate SB for each region at the expense of increased heterogeneity of FPGA features. On the other hand, increase of the SB regions has a penalty at the fabrication cost of the device. For this work we choose to use an array with two distinct SB areas, but the designer is able to increase this threshold in order to increase further the power efficiency.

2.2 Connection Pattern Map

The Fig. 3 depicts only the spatial information of certain features of an FPGA. However, they cannot describe the type of used connection pattern, for instance, vertical (|) or horizontal (-) connection. Based on a technique proposed in [9], we derive a connection pattern usage distribution of the whole FPGA (Fig. 6). Furthermore, we specify the probability use of a connection pattern. From

Fig. 6, it can be seen that the horizontal and vertical connections are the most frequently used. It was proven [6] that the horizontal and vertical connections minimize the number of bends of the final routing. This feature was embodied in all academic and commercial placement and routing tools.

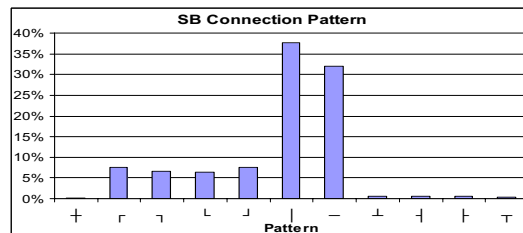


Fig. 6: Statistic approach of connections into SBs

Extracting data from the results after the use of EX-VPR tool for placement and routing of DSP applications, we can build the map of Fig. 7. The map shows spatial information about only the horizontal and vertical connections and it can be considered as a subset of the maps shown in Fig. 3. Moreover, such maps are extremely useful if the designer can use hard-wired connections in his/her FPGA architecture design. The main difference with [9] is the fact that we insert such type of connections *only* for dominant connection patterns, as it is specified by the maps shown in Fig. 7.

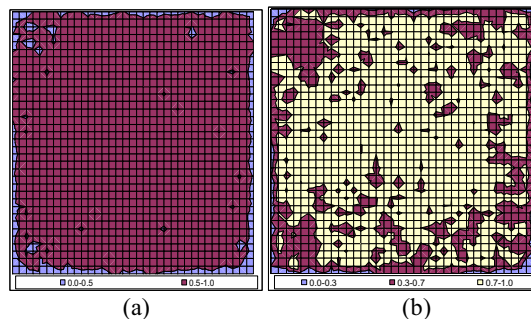


Fig. 7: Distribution of long-line connections (a) for two regions, (b) for three regions

It can be seen that as we approaches to centre of FPGA the channel utilization increases. The combination of spatial and the statistical information of connection patterns allows to designers to substitute the pass-transistors of the most frequently used connection patterns with simple wires (i.e. hard-wired connections). The advantages of hardwired connections have described in [9]. However, there are two main drawbacks comparing with the proposed method; i) the first technique does not take into account any spatial data about the connection patterns and ii) the substitution of

pass-transistors with hard-wired connections in regions with small probability use of the main connection patterns results into increase (redundant) channel width.

2.3 Switch Boxes Combination Exploration

Employing the spatial information regarding with the SBs location the next paragraphs provide detailed data about the selection procedure of the optimal combination of SBs, considering power, delay and area. Moreover, we provide extensive exploration/comparison results for all possible combinations of three different SB devices, namely Wilton, Universal and Subset (of Xilinx XC4000). Assuming two regions with different type of SBs, we explored the entire possible ratio

$$SB_ratio = \frac{\%SB_Type_1}{\%SB_Type_2}$$

where $\%SB_Type_1$ and $\%SB_Type_2$ denote the percentage of used SBs of *Type_1* and *Type_2*, with $\%SB_Type_1 + \%SB_Type_2 = 100\%$. The exploration procedure was done by the EX-VPR tool, which can handle both the above three SBs and user-specified SBs [5].

In order to determine the type and the ratio of SBs which will compose the proposed FPGA architecture, we performed design exploration for four main design parameters: i) the Power×Delay Product (PDP), ii) the performance, iii) the power consumption and iv) the area. Thus, the designer can select the optimal combination of SB, depending on his/her primary optimization goal, e.g. minimal PDP. The values of Fig. 8 to 11 are normalized average values from all DSP applications, as follows:

$$Normalized_PDP_{i,j} = \frac{PDP(SB_comb_i, SB_ratio_{i,j})}{\max_{i,j} \{PDP(SB_comb_i, SB_ratio_{i,j})\}}$$

where the variables SB_comb_i denotes the *i*-th different SBs combination and the *j*-th SB ratio ($SB_ratio_{i,j}$) of the *i*-th SB combination, with $0 \leq Normalized_PDP_{i,j} \leq 1$. Thus, the designer can select the optimal combination of SB, depending on his/her primary optimization goal, e.g. minimal PDP.

Fig.8 shows the exploration results for PDP. It was derived by placing and routing of DSPs into FPGAs with different ratios between the two distinct SB types. The values of the horizontal axis show the percentage of the first SB compared to the second one into the array. For example, the number “21.07” of the curve “Subset-Universal”, shows that the percentage of the Subset SB into the FPGA array is 21.07%, while the Universal SB

occupies the 78.93% of the array. Moreover, having a combination $\{\%SB_Type_1\}$ and $\{\%SB_Type_2\}$, the latter SB type is placed is an orthogonal located in the centre of FPGA, while the SB_Type_1 placed around the orthogonal up to I/O pads.

For the proposed FPGA platform, we choose the combination of “Wilton-Universal” SBs, where the Universal is placed into the centre of the array. From Fig. 8, we can see that PDP is minimized when 38.81% of Wilton SB and 61.19% of Universal SB, are used.

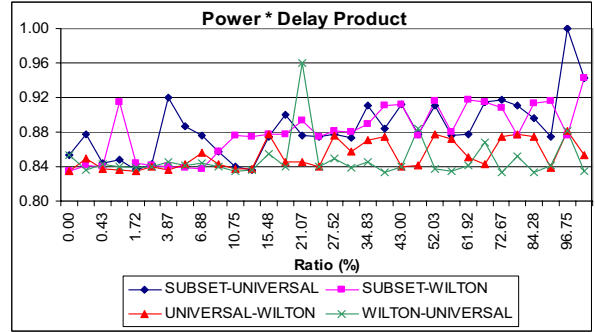


Fig. 8: FPGA Power×Delay product

Performance (i.e. delay) is another critical issue of an FPGA architecture. Fig. 9 shows the delay values for all the possible combinations of SB topologies and ratios. It can be seen that the combination “Wilton-Universal” SBs with ratio 38.8%/61.19% compared both to all other combinations, as well as to the distinct Subset, Wilton and Universal exhibits the smallest delay.

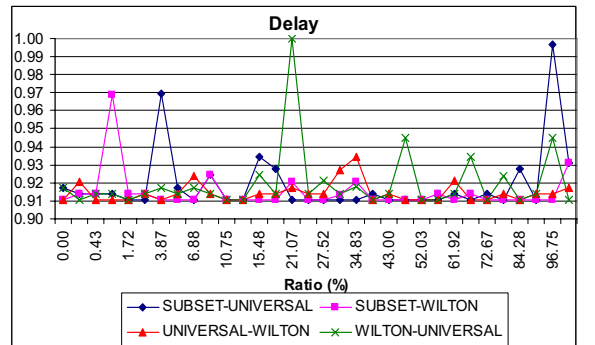


Fig. 9: Delay for the proposed FPGA architecture

Power consumption is a critical parameter that characterizes an FPGA, taking into consideration that an FPGA consumes more power compared to an ASIC implementation. As it is shown in Fig. 10, the combination of Wilton and Universal SBs, with the ratio of 38.8%/61.19%, is the most power efficient architecture, compared to the other SB combinations and ratios.

The last parameter where we did exploration is the FPGA area. Given a certain SB, there is a minimal channel width value, W_{min} , for performing the routing process. However, the use of different SBs means different W_{min} 's, due to their own specific routing capabilities. This may result into channel length increase or decrease, depending on the specific SB combinations. In Fig. 11, it can be seen that the SB combination "Universal-Wilton" provides the most area-efficient solution and the corresponding SB ratio is 20% (i.e. 20% Universal SBs and 80% Wilton). Comparing the optimal-area solution with the optimal EDP solution, it is concluded that the latter one requires area increase by 10%.

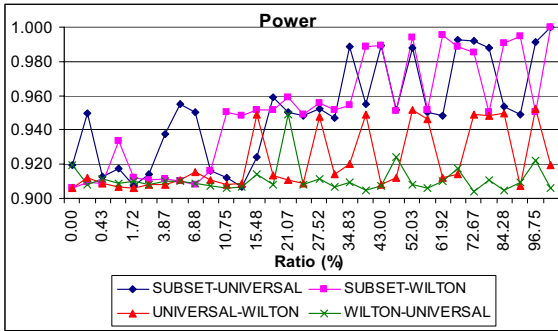


Fig. 10: Power dissipation

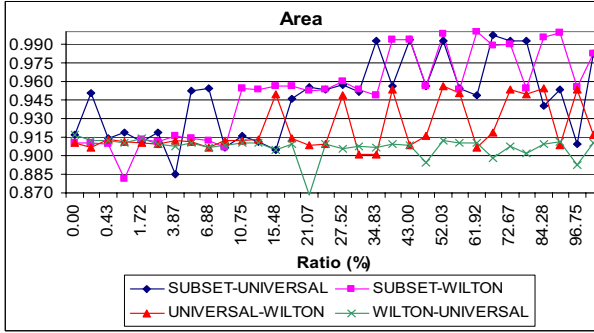


Fig. 11: Area estimation

The aforementioned exploration procedure for two regions (i.e. combination of two different SBs) can be also applied for larger number of regions, given the available different SBs. More specifically, we applied the same exploration procedure for three regions. Due to lack of space, we cannot provide the corresponding curves for PDP, power, performance and area. However, we found that the combination "Subset-Wilton-Universal" of 35%, 50% and 15%, respectively, provides the optimal EDP results, which is better than the corresponding EDP for two regions. It should be stressed that increase of regions number implies more heterogeneity FPGA device and

thus, increased fabrication cost, which might be acceptable for large volumes only. The primary goal of the proposed methodology is to prove that combination of different properly-chosen SBs results into performance and power consumption.

3. Hard-wired Methodology

Beside the SB selection that described at Section 2, there is the possibility to maximize the device performance, while minimizing the power and area requirements. By applying a modified approach of hardwired connections, proposed in [9], we remove pass-transistors from SBs and replace them with wires. The selection of which transistors will be removed is based on the exploration results shown in Fig. 3 and Fig. 6.

As we described at Section 2.2, we choose to replace only transistors that form the horizontal (-) and vertical (|) connections at the SBs. A critical issue that affects the device efficiency is the length of the interconnect wire. Even though in this work we choose a uniform distribution of routing tracks, it is possible the designer to have regions into the FPGA where the channel width is wider. This feature will improve the device performance, as the regions with wider channel will be used for the centre of the FPGA (where there is more need for connectivity resources). In order to implement the hardwired methodology, first we have to define how many transistors will be replaced with wires. In other words, the task here is to define the length of the interconnect wires (i.e. how many CLBs each wire spans) that will be used for application routing into the FPGA.

4. Efficient Routing Procedure

4.1. EX-VPR Tool

In order to support the design space exploration procedure, we used the EX-VPR tool [5], enhanced with the option for supporting the placement and routing of applications in FPGAs with multiple SBs. The tool is based on VPR [6], and it is part of the MEANDER framework [7]. Extensive description of the whole design flow can be found in [5]. Here, we describe only the extended features of the EX-VPR tool that affects the architecture exploration.

The EX-VPR tool was extended by adding a silicon area model that estimates the area of the device in um^2 , assuming STM 0.18 μm technology. Another very important extension is the addition of user-defined full-custom switch boxes, while the original version of VPR supported only three types of switch boxes namely Subset (similar to the one used in Xilinx XC4000 devices), Wilton and Universal. This feature is possible as the EX-

VPR handles devices with switch boxes where the acceptable connections among routing tracks are defined by the designer. For demonstration purposes besides the three existing switch boxes to VPR we have implemented four additional switch boxes [5]. In addition to that, the EX-VPR has the ability of integrating IP cores. This feature allows the user to reserve a part inside the FPGA with specific (x,y) coordinates for placement of IP modules (e.g. CPUs, memories). The main advantage is the fact that the designer can realize onto FPGA architecture a composite system and therefore, he/she can perform rapid prototyping of a new design. The power consumption of an FPGA is calculated by the extended version of PowerModel [8], which takes in mind the new FPGA components (new SBs, IP cores, etc).

Table 1 shows a qualitative comparison among VPR [6] and EX-VPR. The (✓) symbol indicates that the corresponding feature is available in the design framework, while the (✗) symbol indicates that the specific feature is not supported.

Table1: Qualitative comparison among VPR and EX-VPR

Feature	VPR [10]	EX-VPR
Placement	✓	✓
Routing	✓	✓
Bitstream Generation	✗	✓
Supported Switch Boxes (SBs)	<ul style="list-style-type: none"> ○ Subset ○ Wilton ○ Universal 	<ul style="list-style-type: none"> ○ Subset ○ Wilton ○ Universal ○ User specified
Support multiple SB simultaneously	✗	✓
IP core	✗	✓
Power Estimation	✓	✓
Timing info (sec)	✓	✓
Silicon Area (um ²)	✗	✓
Application specific FPGA design	✗	✓
Bitstream generation	✗	✓
Partial device Programming	✗	✓
Run-time device Programming	✗	✓
GUI	✓	✓
Graphical architecture description	✗	✓
Run through HTTP	✗	✓

Table 1 show that the EX-VPR provides the VPR features, while it also provides the flexibility for full-custom switch box definition, the IP handling option, the

silicon area calculation, and finally the remote access to it. The remote access to EX-VPR allows the user to run the tool without having them installed in his/her own computer. It is evident that the EX-VPR is the most complete academic placement and routing tool for architecture level exploration, and is at least in terms of provided features comparable with commercial tools.

4.2 Routing Algorithm

Below, we describe the algorithm, which performs the placement of multiple SBs, the routing of MCNC applications in the new FPGA architecture and the placement of hard-wired connections (if it is required), is shown in Fig. 12. This algorithm was realized by EX-VPR tool, which will be discussed at the next paragraph.

Input:
 Technology mapped netlist, Architecture file
 Designer constraints:
 SB_num ← number of distinct SB regions
 SB_type ← SB types
 SB_(x,y) ← co-ordinates (x,y) of each SB

Initialization:
 Initialize SB matrix (n_x,n_y,W) to unknown
 /* n_x,n_y are the co-ordinates (x,y) of each SB
 W is the routing channel width */

Algorithm:
 for i = 1 to n_x do { // run for x-axis
 for j = 1 to n_y do { // run for y-axis
 for p=1 to SB_num do { // run for all the SB regions
 place_sb(x,y) ← true
 /* The place_sb(x,y) function choose the right SB type for the specific (x,y) based on the SB_(x,y) designer constraints. */
 end for
 for source = 1 to W do // run for all the routing tracks
 // map track-source to track-destination
 W_{destination} ← SB_map(W_{source})
 update_routing_graph ← true
 /* The function update_routing_graph modifies the initial routing graph in order to handle the characteristics of the new SB types, taking in mind the number of different SBs, as well as the number of SB regions*/
 } } }
 find_routing(SB_num, SB_type, SB_(x,y)); // routing
 if (enable_hardwired = True)
 {
 /* Hard-wired some connections into the FPGA */
 find_fpga_critical_area ← True
 /* Find out the area (x1,y1)-(x2,y2) where we will apply the hardwire technique */
 replace_transistor(horizontal,wire)
 replace_transistor(vertical,wire)

```

update_routing_graph ← True
/* Replace the horizontal and vertical placed transistors
inside SBs with wires and updates the routing graph */
recomputed_delay_for_nets ← true
find_out_critical_path ← true
recomputed_power_consumption ← true
}
end if

```

Fig. 12: Algorithm for SB placement into proposed architecture and Hard-wired connections

4.3 EX-VPR Routing Flow

Fig. 13 shows the way that EX-VPR tool used to realize applications into the proposed novel FPGA architectures. The tool has five distinct steps. First of all, the application is mapped and placed onto the available CLBs of the reconfigurable array. The dimension, as well as the CLB's architecture is defined into the FPGA architecture file [14]. The second step involves the SB selection, based on the application requirements. As it has been mentioned above, each SB is characterized in terms of delay, power and area requirements. By defining the application constraints, the tool finds out the number of different SBs that will be used in FPGA simultaneously, as well as the ratio of them into the array. At the third phase of the tool execution, the application is routed into the mixed SB device. This is done by finding out the region that each SB type should be placed, and then routing the application into this mixed-interconnect FPGA architecture. The next step is an optional feature that lets the designer to make some hardwired connections into the SBs, in order to minimize the application delay and power consumption. This way that the hardwired connections take place was examined briefly into Section 3. Finally, EX-VPR produces the bitstream file that configures the FPGA device with the application, taking in mind all the features of the proposed architecture.

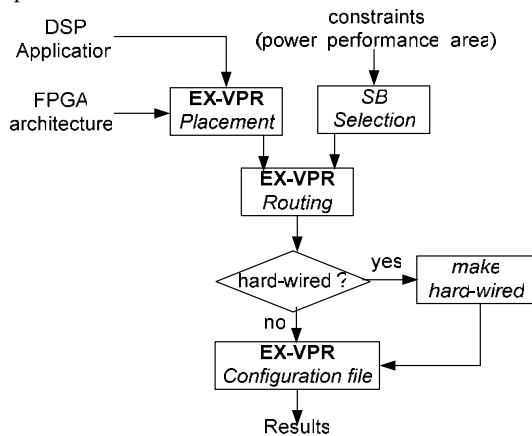


Fig. 13: EX-VPR routing procedure

The aforementioned exploration procedure for two regions (i.e. combination of two different SBs) can be also applied for larger number of regions, given the available different SBs. Specifically we applied the same exploration procedure for three regions. Due to lack of space, we cannot provide the corresponding curves for PDP, power, performance and area. However, we found that the combination “Subset-Wilton-Universal” of 34.06% 47.84% and 18.1%, respectively, provides the optimal PDP results, which is better than the corresponding PDP for two regions. It should be stressed that increase of regions number implies more heterogeneity FPGA device and thus, increased fabrication cost, which might be acceptable for large volumes only.

5. Experimental Results

The proposed interconnection architecture was implemented and evaluated by a number of well-known DSP applications. Among them are encryption, filters, ALUs and data manipulation. Table 2 shows the results for power consumption and delay for (i) Virtex-like FPGA architecture with Subset, Wilton and Universal SB, (ii) the Altera Stratix FPGA and (iii) the proposed architecture with multiple SBs.

The results for the single SB architecture are measured with VPR [6] tool, while proposed architecture uses EX-VPR tool. All the applications are placed and routed into the optimal array and channel width that the tools are reported. Since our primary goal is the design of both high performance and low-power FPGA architecture, we chose the optimal PDP value from the exploration results (Fig. 5).

It can be seen that the proposed method achieved significant reduction of 30% (average value) in delay and reasonable power gain of 15% (average value) compared to VPR-based platforms. Also, it is worse than existing commercial FPGAs in delay. The reason arises from the fact that the STRATIX platforms use more mature design tools and include special purpose components (ALUs, RAM, etc.). It should be stressed that we achieved to design a high performance FPGA, without any negative impact on power, although high performance circuit means high switching activity and eventually increased power. On the other hand, the new methodology requires about 15% wider channel than Wilton and the same width for Subset.

Table 2: Comparison results between the proposed FPGA architecture with multiple-SBs and single SB FPGA architecture in terms of delay and power consumption

DSP application	Subset		Wilton		Universal		Altera Stratix		Multiple SBs Architecture	
	delay $\times 10^{-8}$	power $\times 10^{-3}$	delay $\times 10^{-8}$	power $\times 10^{-3}$	delay $\times 10^{-8}$	power $\times 10^{-3}$	delay $\times 10^{-8}$	power $\times 10^{-3}$	delay $\times 10^{-8}$	power $\times 10^{-3}$
alu4	3.02	30.7	5.03	18.5	3.61	25.5	1.93	187.5	2.72	23.62
barcode	1.33	5.38	1.33	5.41	1.33	5.47	0.63	187.5	0.93	5.14
bigkey	2.68	58.4	2.72	57.7	2.52	62.1	needs more I/O		1.84	56.43
clma	7.82	87.0	1.50	66.7	8.20	83.3	2.68	187.5	3.76	75.05
cordic	2.15	7.49	3.54	45.2	2.52	6.37	2.11	187.5	1.92	6.83
decod	38.5	1.56	39.9	15.6	38.9	1.59	1.07	187.5	needs more resources	
des	5.37	33.9	4.96	36.9	5.23	35.2	needs more I/O		3.63	33.56
diffeq	2.50	21.0	5.12	10.2	2.56	20.7	1.47	187.5	2.38	16.43
dsip	2.83	47.2	2.98	44.7	3.64	36.7	needs more I/O		2.21	40.72
ellip	1.33	5.38	1.33	5.41	1.33	5.47	0.64	187.5	0.93	5.45
fft_256	85.3	4.77	94.8	4.31	85.7	4.77	0.5	187.5	62.02	4.38
gcd	1.33	5.38	1.33	5.41	1.33	5.47	0.64	187.5	0.93	5.15
mac32	0.11	54.8	0.11	54.8	0.11	53.5	needs more I/O		0.1	51.65
mult32a	4.56	1.24	4.58	1.25	4.58	1.25	1.66	187.5	3.22	1.23
phase_decoder	1.26	7.19	1.31	6.94	1.45	6.24	0.76	187.5	0.94	6.48
rot	1.82	19.9	2.14	16.9	1.86	19.7	2.05	187.5	1.35	17.76
traffic	70.6	2.54	70.2	2.60	70.2	2.56	0.35	187.5	50.2	2.56

Table 3: Power×Delay Product for a number of different architectures

Benchmark	Subset	Wilton	Universal	Altera Stratix	Multi SBs	Multi SBs + Hardwired
alu4	9.27E-10	9.31E-10	9.21E-10	3.62E-09	6.42E-10	4.73E-10
barcode	7.16E-11	7.22E-11	7.28E-11	1.18E-09	4.78E-11	4.01E-11
bigkey	1.57E-09	1.57E-09	1.56E-09	-	1.04E-09	8.34E-10
clma	6.80E-09	1.01E-09	6.83E-09	5.03E-09	2.82E-09	1.02E-09
cordic	1.61E-10	1.62E-09	1.61E-10	3.96E-09	1.31E-10	9.34E-11
decod	6.04E-10	6.22E-09	6.19E-10	2.01E-09	-	5.01E-10
des	1.82E-09	1.83E-09	1.84E-09	-	1.22E-09	8.64E-10
diffeq	5.25E-10	5.22E-10	5.31E-10	2.76E-09	3.91E-10	1.21E-10
dsip	1.34E-09	1.33E-09	1.34E-09	-	9.01E-10	7.83E-10
ellip	7.16E-11	7.20E-11	7.28E-11	1.28E-09	5.07E-11	2.04E-11
fft_256	4.06E-09	4.09E-09	4.09E-09	9.38E-10	2.72E-09	1.03E-09
gcd	7.15E-11	7.23E-11	7.28E-11	1.24E-09	4.79E-11	2.34E-11
mac32	6.0E-11	6.03E-11	5.89E-11	-	5.17E-11	3.90E-11
mult32a	5.65E-11	5.73E-11	5.73E-11	3.11E-09	3.96E-11	1.87E-11
phase_decoder	9.05E-11	9.09E-11	9.05E-11	1.43E-09	6.09E-11	4.53E-11
rot	3.62E-10	3.62E-10	3.66E-10	3.84E-09	2.34E-10	1.70E-10
traffic	1.79E-09	1.83E-09	1.82E-09	6.56E-10	1.29E-09	8.93E-10

Table 3 shows the Power×Delay product for the DSP applications for a number of different architectures. The last column (named “Heterogeneous + Hardwired”) refers to the proposed interconnection architecture if we apply and the feature of making some connections (as we see in the previous Section). From the results, we can see that if we apply the hardwired methodology of selected SB transistors, we achieve reduction in both power and delay of the mapped application.

The gain of the proposed methodology for high-speed interconnection compared to existing well-established architectures is shown in Table 4. As we can see, our interconnection scheme achieves about 55% better results in Power×Delay product compared to Subset, Wilton and Universal architectures, while it is about 75% better than Altera Stratix. The reason for this is the power consumption of the Altera’s devices, which is bigger compared to academic architectures. Finally, the proposed architecture with multiple SBs, could be improved about 40% if we apply the hardwired feature in selective transistors of the SB, as we see from the last column of Table 4.

Table 4: Gains in Power×Delay for the proposed interconnect architecture (Heterogeneous SB+Hardwired) compared to other architectures

Benchmark	vs. Subset (%)	vs. Wilton (%)	vs. Universal (%)	vs. Stratix (%)	vs. Multi-SB (%)
alu4	48.98	49.16	48.61	86.92	26.37
barcode	43.96	44.26	44.88	96.60	16.11
bigkey	46.71	46.86	46.70	-	19.67
Clma	85.00	-1.94	85.0	79.70	63.85
cordic	42.00	94.16	41.81	97.63	28.77
decod	16.58	91.95	18.99	75.02	-
Des	52.53	52.79	53.06	-	29.07
diffeq	76.95	76.83	77.16	95.60	69.05
Dsip	41.38	41.21	41.38	-	12.99
Ellip	71.49	71.64	71.95	98.31	59.75
fft_256	74.68	74.79	74.80	-9.86	62.08
Gcd	67.29	67.47	67.83	98.05	51.14
Mac32	35.30	35.30	33.72	-	24.49
Mult32a	66.92	67.33	67.33	99.3	52.78
phase decoder	49.99	50.17	49.93	96.82	25.63
rot	53.06	52.99	53.60	95.57	29.09
traffic	50.20	51.07	50.30	-36.07	30.51
Average:	54.29	56.82	54.54	74.90	37.58

Acknowledgements

This work was partially supported by the project IST-34793-AMDREL and the project PENED '03, which are funded by the European Commission and the GSRT of Ministry of Development.

6. Conclusions

A novel FPGA interconnection methodology for high speed and power efficient island-style FPGA architectures was presented. The main contribution of our work is the SB placement into the FPGA according to the spatial information about connections. The comparisons in terms of performance and power give promising results. Finally, the designer can apply the methodology for any type of SB, given the application domain requirements.

References

- [1] T. Miyazaki et. al, “PROTEUS-Lite Project: Dedicated to Developing a Telecommunication-oriented FPGA and its Applications”, IEEE Trans. on Very Large Scale Integration (VLSI) Systems, Vol. 8, No. 4, pp. 401-414, Aug. 2000
- [2] Dehon A, “Balancing interconnect and computation in a reconfigurable computing array (or, why you don’t really want 100% LUT utilization), ACM/SIGDA Int. Symp. on FPGAs, pp. 69-78, 1999
- [3] Deliverable Report “Survey of existing fine-grain reconfigurable hardware platforms,” AMDREL project, IST-2001-3437
- [4] G. Lemieux and D. Lewis, “Design of Interconnection Networks for Programmable Logic”, Kluwer Academic Publishers, 2004
- [5] K. Siozios, et.al. “An Integrated Framework for Architecture Level Exploration of Reconfigurable Platform”, FPL2005, pp. 658-661.
- [6] V. Betz, J. Rose and A. Marquardt, “Architecture and CAD for Deep-Submicron FPGAs”, Kluwer Academic Publishers, 1999.
- [7] <http://vlsi.ee.duth.gr/amdrel:8081>
- [8] K. Poon, A. Yan, S. Wilton, “A Flexible Power Model for FPGAs”, FPL2002, pp.312–321, France,2002
- [9] S. Sivaswamy, et. al., HARP: Hardwired Routing Pattern FPGAs, in proc. of Int. Symp. FPGA, Feb. 20–22, 2005, Monterey, USA
- [10] K. Compton and S. Hauck, “Reconfigurable Computing: A Survey of Systems and Software”, ACM Computing Surveys, 2002, pp171–210
- [11] <http://www.xilinx.com>
- [12] <http://www.altera.com>
- [13] K. Leijten-Nowak et.al, “An FPGA Architecture with Enhanced Datapath Functionality”, FPGA’03, California, USA, pp. 195-204
- [14] http://vlsi.ee.duth.gr:8081/help/DUTYS_manual.pdf