Improving FPGA Routing Architectures Using Architecture and CAD Interactions

Benjamin Tseng, Jonathan Rose, Stephen Brown

Department of Electrical Engineering, University of Toronto, Ontario, Canada

Abstract

This paper examines the interactions between the CAD tools that are used to configure an FPGA's routing resources and the design of the routing architecture itself. Such an understanding is used to determine where to reduce the number of routing switches in the FPGA while maintaining routability. Experiments are used to study a switch block that was previously thought to have unacceptably low flexibility [7]. We show that the performance of this switch block can be improved by adapting the global router to require less flexibility in the architecture, and by careful placement of physical pins on the logic blocks. Also, it is demonstrated that the fewest routing switches are required when each logical pin appears on only one side of the logic cell rather than two or more. [1]

1 Introduction

Field-Programmable Gate Arrays (FPGAs) provide an innovative approach to Application Specific Integrated Circuit (ASIC) implementation that reduces both turnaround time and manufacturing cost. FPGAs suffer from lower logic density and speed compared to mask-programmed arrays MPGAs because the programmable routing switches (such as pass transistors [6], antifuses [1, 4] or EPROM transistors [12]) take up more space and have higher resistance and capacitance than simple metal wires. This paper addresses these drawbacks by investigating techniques to reduce the number of switches needed in an FPGA. The basic strategy used is to 'tune' the architecture and the CAD tools so that they are able to work together more effectively.

1.1 Routing Architecture Model

An FPGA can be modeled as a two-dimensional array of logic cells interconnected by vertical and horizontal routing channels, as illustrated in Figure 1. Figure 2 shows a sample unit VLSI layout block that can be repeated to form the FPGA in Figure 1. This unit block, called an FPGA tile, illustrates the three major parts in the model: the Logic (L), Connection (C), and Switch (S) blocks. The L blocks house the combinational and sequential logic that form the functionality of a circuit. For the logic cell in this experiment, we have adopted the four-input lookup table and D flip-flop that [8] suggests is a good choice in terms of logic density. There is a total of seven logical pins on this block. The number of logic block sides on which each logical pin physically appears is determined by an architectural parameter called T. Figures 3a and 3b illustrate the $T = 4$ case and the $T = 1$ case, respectively. In the figure, the logical pins are numbered from 0 to 6. For $T > 1$, each logical pin appears on more than one side. When two or more physical logic cell pins correspond to one logical pin, the physical pins are said to be *electrically equivalent*. The number of logical pins per logic cell is set by the parameter P, where $P = 7$ for the FPGA model used here.

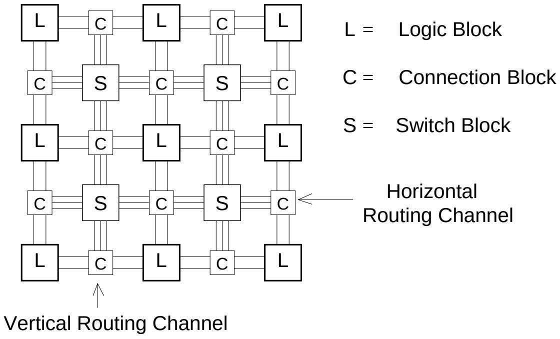


Figure 1: The FPGA Model

The C and S blocks illustrated in Figures 1 and 2 make up the routing architecture of an FPGA. The C blocks are used to connect the L block pins to the routing channels via programmable switches. In Figure 2, each of the switches in the C blocks is shown by an **X**. The *flexibility* of a C block is set by a parameter called F_C, which defines the number of tracks in the adjacent channel that each logic cell pin can connect to. In the figure, each logic cell pin connects to two tracks, and so F_C=2. For maxi-

[1] This work was supported by MICRONET, an NSERC Summer Research Fellowship, and a grant from Bell-Northern Research.

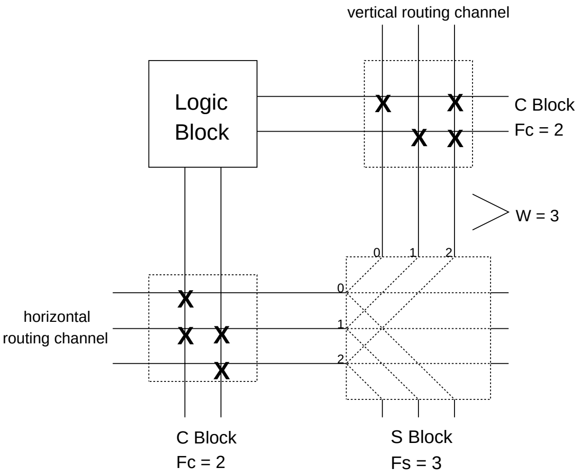
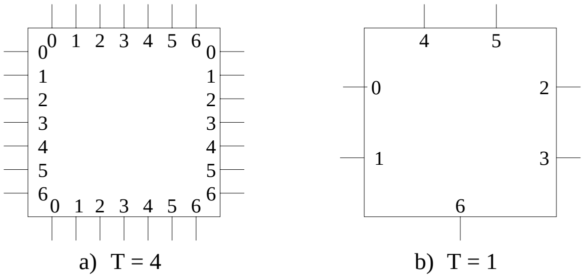


Figure 2: An FPGA Tile



a) T = 4

b) T = 1

Figure 3: Definition of T

mum flexibility in the C block, each logic cell pin would be switchable to all of the W wiring tracks, where W is the number of tracks in the channel.

The S blocks in Figure 1 connect wiring segments in one channel segment to those in another. We define the flexibility of an S block, F_S, to be the number of tracks that an incoming wire can connect to on the three other sides. The S block in Figure 2 shows an example in which a signal entering the S block on wiring segment number 0 on the left side can connect to one wiring segment on each of the three other sides, which implies $F_S = 3$. Although not shown, the other wires are similarly connected. For maximum S block flexibility, each incoming wire would be switchable to all of the outgoing wires on the other sides (i.e. $F_{Smax} = 3 \times W$).

1.2 Motivation

A recent study [7] explored the relationship between the routability of an FPGA and the flexibility of its interconnecting structures. Here, routability is defined to be the percentage of a circuit's connections that can be successfully completed by the CAD tools, and the flexibility of the interconnecting structures are set by the F_C, F_S, W, and T parameters. In general, increased flexibility increases routability but will also increase the number of routing switches. The goal then, is to devise an FPGA architecture and associated routing tools that require minimum flexibility while maintaining 100% routability.

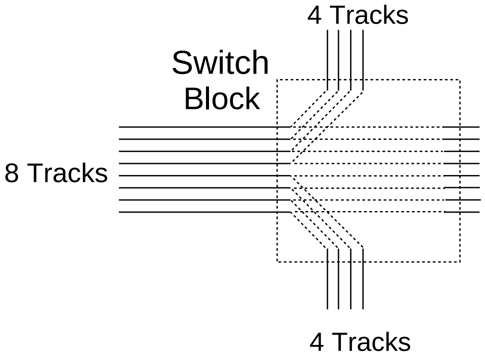


Figure 4: Switch Block Routing Bends

The study concluded that a high flexibility in the C blocks is required for good routability, but a relatively low flexibility is sufficient in the S blocks. As a specific example, with $F_S = 3$ and $F_C = W$, 100% routing completion was possible for a set of benchmark circuits, while requiring only an average of 0.8 tracks above the theoretical minimum value as determined by the global router.

In that study, although a less flexible S block, $F_S = 2$, also achieved 100% routing completion, successful routing required an average of 9 tracks above the theoretical minimum value and so many more routing switches. The increase in the number of wiring tracks was caused by the reduction in the number of available paths for making a connection. Figure 4 shows why this occurs when the S block has a flexibility of two ($F_S = 2$). Here, each incoming signal has a programmable switch straight through an S block, and can *either* bend left or right, but not both. Observe that, for any global route that bends as it travels through this S block, the number of tracks that actually make the bend is one half of the total (W). For this reason, every bend in the global route of a connection cuts the number of usable paths in half leading to a low routability for this kind of S block. Thus, to improve routability, we must increase the number of available paths, which can be accomplished in two ways: increase the number of tracks per channel, W, or decrease the number of bends.

Since an increase in W leads to greater chip area for the FPGA, it is much more attractive to reduce the number of bends in the global routes. The remainder of this paper describes a number of techniques that are used to reduce the number of routing bends. These techniques yield a low enough value of W to make $F_S = 2$ feasible.

2 Experimental Procedure

We use an experimental approach to analyzing FPGA architectures. We implemented a set of circuits in FPGAs with different routing architectures. By 'implement' we mean the synthesis of a circuit into an architecture using a set of CAD tools.

2.1 Basic Procedure

For each circuit, the following implementation is performed:

1. The circuit is technology mapped into a network of L blocks. This is done using an early version of the Chortle program [5].

2. Using a placement program [10], the logic cells are placed using the min-cut placement algorithm.

3. The placed circuit is passed on to **PGAroute** [9] for global routing. This process assigns each connection to a specific set of channels. Hence, the global router determines the maximum channel density of the circuit, which is the minimum number of tracks needed in the FPGA to route the circuit. The router has several options which affect the number of bends it will use as described in Section 2.2 below.

4. The channel path assigned by the global router and the routing architecture (defined by F_C, F_S, W, and T) are fed into the detailed router (**CGE** [3]) for final routing. The detailed router assigns specific wiring segments and determines which switches to turn on in the FPGA for a given connection.

In the above procedure, the mapping and placement are performed once for each circuit, but global and detailed routing are performed several times, as the experimental parameters are varied. The output of the procedure is the total number of tracks, and the number of routing switches needed for each circuit, for the architecture specified by the given parameters (F_S, F_C, W, and T).

2.2 Bend Reduction

The number of bends in a global route for a connection can be reduced in several ways. In this section, we present one technique based on the routing algorithm, and one related to the routing architecture.

2.2.1 Global Router Bend Optimization

In the **PGAroute** global routing algorithm [9], we have modified the internal cost function to penalize the switch block routing bends. The primary consideration of the original cost function was to minimize channel density. The new cost function incorporates a new component to favour a route with fewer routing bends as its main priority. Note that since congestion is now the secondary cost function, the maximum channel density will likely increase. This effect is discussed further in Section 3.1.

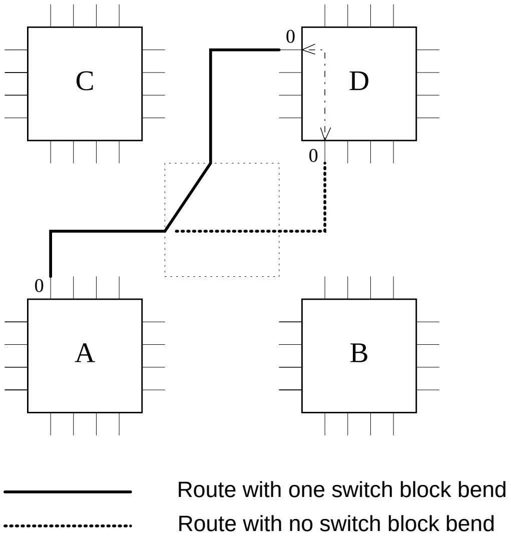

C
D
A
B
0
0
0
Route with one switch block bend
Route with no switch block bend

Figure 5: Physical Pin Placement

2.2.2 Logic Block Pin Placement

In studying various failures in detailed routing, it became apparent that careful positioning of the physical pins can help in reducing switch block bends. Figure 5 illustrates this. Assume that it is necessary to connect pin 0 on block A to pin 0 on block D. If pin 0 on block D is only available on the left side of the block, then the global route shown by the solid line must be used and one switch block bend is needed. However, if pin 0 is available on the bottom of the block D, then the dotted global route can be used and no switch block bends are needed.

Providing a global router with alternatives for a physical pin (such as pin 0 on block D) can be achieved using two concepts:

- Electrical Equivalence: this occurs when a physical pin appears on more than one side of the logic block ($T > 1$). With this property, each logical pin becomes more accessible. A global router is free to substitute one electrically equivalent pin for another. It should choose the physical pin that results in the fewest switch block bends in the route, while minimizing channel density.

- Functional Equivalence: a group of logical pins are called functionally equivalent when they can perform

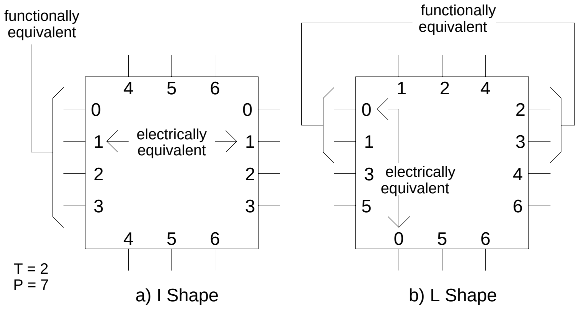


Figure 6: Physical Pin Placement: $T = 2$

the identical set of logic functions. Such pins are interchangeable at the global routing step, so that the router can choose any unassigned pin that is functionally equivalent, depending on which one would give the lowest routing cost. The logic block assumed for this study contains four functionally equivalent pins since the inputs to a lookup table are functionally equivalent.

Figure 6 shows two possible pin placements when each logical pin appears on two sides of a logic block ($T = 2$). Figure 6a shows the 'I' shape pin placement. Here, the functionally equivalent pins are located on the same side of the logic block, and the electrically equivalent pins are all opposite one another. With this arrangement, each pin is available on only two sides of the block, which means that a global router is likely to need bends to implement some connections.

Figure 6b shows an alternative pin placement known as the 'L' shape. Here, the functionally and the electrically equivalent pins are distributed evenly across so that each logical pin is accessible on all four sides of the logic block. For example, to access an input to the logic block, the global router can pick pin 0, which is accessible from the left and the bottom sides. Alternatively, the router can make use of the functional equivalence property and pick one from pins 1, 2, or 3, which are available on the other two sides. With this scheme, the global router can pick the pin on the side that would result in the fewest switch block bends. One can view this modification as moving the right-angled bends to the inside of the logic cell where there is no performance penalty in making the turn.

A similar revision in physical pin placement is made when $T = 1$, meaning that there is no electrically equivalent pins. Even though there is no electrical equivalence in the case $T = 1$, by invoking functional equivalence, the same advantages described for $T = 2$ can be gained.

3 Results

The effect of S block routing bend reduction on routability of FPGAs was evaluated using five benchmark circuits from four sources - Bell-Northern Research, Zymos, and two different designers at the University of Toronto. The number of blocks and the number of connections for each circuit is given in Table 1.

Circuit	Num Blocks	Num Conn	Source	Type
busCntl	109	392	UT D1	Bus Cntl
dma	224	771	UT D2	DMA Cntl
Ebnr	362	1257	BNR	Logic/Data
dramFsm	401	1422	UT D1	State Mach.
z03	586	2135	Zymos	8-bit Mult

Table 1: Experimental Circuit Characteristics

The experimental results are evaluated using two figures of merits:

1. The number of programmable switches required to implement a circuit. Since switches cost in area and speed, this is an important architectural measure. The switch count is measured per FPGA tile that includes one logic cell surrounded by two C blocks and one S block. The number of routing switches for the C block and the S block are described by Equations 1 and 2 respectively:

$$\#\ of\ Switches\ in\ C\ Block = \frac{1}{2} \times T \times P \times F_C \quad (1)$$

$$\#\ of\ Switches\ in\ S\ Block\ = 2 \times F_S \times W \quad (2)$$

2. The actual number of tracks per channel, W, required by the detailed router is an important figure of merit. For C and S blocks with flexibilities less than 100%, W tends to be higher than the maximum routing channel density, as set by the global router. A good architecture would minimize this difference.

3.1 Experiments

The aim of this experiment is to show how the architectural and CAD adjustments described in Section 2.1 can make the $F_S = 2$ a reasonable S block architecture as measured by switch and track counts. Each circuit is implemented eight times, by using all of the combinations of the following options: with or without the bend reduction in the global router, with the I shape pins or the L shape, and finally with $T = 1$ or $T = 2$.

Table 2 shows the effectiveness of the bend reduction techniques. By invoking bend reduction, the total number of bends in the five circuits has been reduced to less than half of its original value. This bend reduction was achieved with little or no increase in maximum channel density.

Total Number of Switch Block Bends in Five Circuits	
Without Bend Reduction	With Bend Reduction
4900	2164

Table 2: Routing Bends Reduction Over Five Circuits

Fs = 2; P = 7						
T	Bend Reduc.	Pin Plmt	Channel Density	W	Fc	Switch Per Tile
2	No	I	11	19	18	329
2	Yes	I	12	16	15	276
2	No	L	9	18	17	270
2	Yes	L	11	12	11	206
1	No	I	10	23	22	241
1	Yes	I	11	21	19	213
1	No	L	10	22	21	236
1	Yes	L	11	16	15	171

Table 3: Summary of Results for $T = 2$ and $T = 1$

Table 3 gives the experimental results for the different scenarios when averaged over the five circuits. The first column indicates the T value used. *Bend Reduc.* indicates whether the global router bend reduction algorithm is applied. This algorithm involves the use of functional equivalence and the revised cost function to penalize routing bends. *Pin Plmt* indicates if the I or the L physical pin placement was used. *Channel Density* reports the maximum routing channel density as determined by the global router. W gives the minimum number of wiring tracks required for the detailed router to achieve 100% routability. F_C is the minimum routable C box flexibility. *Switch Per Tile* indicates the number of switches in a tile for the given values of F_C, F_S, W, T and P.

Table 3 shows that the switching requirement can be drastically reduced by invoking the bend reduction cost function and using the L shape pin placement. The significance of the results in Table 3 will be analyzed in detail in the following sections.

3.2 Architectural Conclusions

In this section, we draw several conclusions about routing architectures from the experimental data.

3.2.1 $F_S = 2$ versus $F_S = 3$

Table 3 illustrates that S block flexibility of two ($F_S = 2$) can be made plausible from the track and switch count points of view. When bend reduction is considered, the channel density tends to increase as expected, but only slightly. This increase is caused by the change in the priority of the global router cost function from minimizing channel density to minimizing S block routing bends. However, the reduction in the number of bends means that the routing channel required substantially fewer tracks to provide the same number of usable paths. The average number of excess tracks required in the case of $F_S = 2$ and $T = 2$ has been reduced from 8.6 to 1.4. Similar results are obtained when each logical pin physically appears on only one side (i.e. $T = 1$). In this case, the average number of excess tracks required drops from 13.2 to 5.2.

Also, in applying these techniques to the circuits, for $T = 2$, the reduction in switch count averaged 37% (from 329 to 206) for the five circuits. Correspondingly for $T = 1$, there is a 29% reduction in switches (from 241 to 171).

Results Averaged Over Five Circuit			
Global Router	Switch Block Flexibility	Switch Count	W
Without Bend Reduction	$F_S = 2$	329	19
With Bend Reduction	$F_S = 2$	206	12
	$F_S = 3$	181	11

Table 4: $F_S = 2$ versus $F_S = 3$ with $T = 2$

Given the effectiveness of S block bend reduction in reducing the switching requirement for $F_S = 2$, the result can now be compared to that of $F_S = 3$, a more flexible S block.

Table 4 presents a summary of the switching requirement of the two S block flexibilities. The first two rows repeat the results for $F_S = 2$. The last row shows the results for $F_S = 3$, a more flexible S block where each wire can be switched to all three outgoing directions. Note that these results show slight improvement over those presented in the previous study [7] because it was found that functional equivalence also helps in reducing switching requirements for higher S block flexibilities. Comparing these results to the improved $F_S = 2$, the switch count and the track count results are very similar. However, it is apparent that $F_S = 3$ is still slightly superior to $F_S = 2$ since the former always required fewer switches.

3.2.2 Logic Block Pin Placement: L versus I

Results Averaged Over Five Circuits		
Switch Block Flexibility	Pin Placement	Switch Count per FPGA Tile
$F_S = 2$	I	276
$F_S = 2$	L	206
$F_S = 3$	I	201
$F_S = 3$	L	184

Table 5: Pin Placement: 'I' shape versus 'L' shape

For all of the circuits, the simple change of pin placement from the 'I' shape to 'L' shape reduces the required number of switches in an FPGA. Table 5 compares two logic block pin placements for two S block flexibilities: $F_S = 2$ and $F_S = 3$. For $F_S = 2$, changing the pin placement from 'I' shape to 'L' shape reduces the number of switches required to successfully route the circuits by 25%, from 276 to 206 switches per tile. Similarly for $F_S = 3$, the switching requirement is reduced by more than 8%, from 201 to

184 switches per tile when using the 'L' shape. Thus, the 'L' pin placement is superior in terms of the number of switches per tile. As explained in Section 2.2.2, the reason that the 'L' shape pin placement prevails is that it gives better access to the logical pins.

3.2.3 $T = 1$ versus $T = 2$

Results Averaged Over Five Circuits				
Measure	$F_S = 2$		$F_S = 3$	
	T = 1	T = 2	T = 1	T = 2
Switch Count	171	206	137	181
Track Count	16	11	12	11

Table 6: $T = 1$ versus $T = 2$

The experimental procedure was also used to answer the question of how many sides each logical pin should appear on. This number is defined by the parameter T and directly relates to the electrical equivalence property.

Table 6 compares the result for the two different values of T, $T = 1$ and $T = 2$. The comparison is made based on switch and wiring track requirements. For each figure of merit, the two T values are contrasted under two S block flexibilities, $F_S = 2$ and $F_S = 3$. The results are averaged over the five circuits. Note that the same value of T is used for each logic block pin.

The results indicate that one physical pin per logical pin ($T = 1$) requires fewer switches per tile than $T = 2$. For $F_S = 2$, in going from $T = 2$ to $T = 1$, there is a 17% reduction in switch count, from an average of 206 to 171 switches. Similarly, the $F_S = 3$ case results in a 24% reduction in switch count, from an average of 181 to 137 switches. In terms of the track count, $T = 1$ tends to need more tracks per wiring channel than $T = 2$. This is especially true for $F_S = 2$, where an average of five more tracks are required over $T = 2$. However, the switch reduction in the $F_S = 3$ case does not occur at the expense of many additional tracks above the channel density as determined by the global router.

Overall, the combination of $T = 1$ and $F_S = 3$ is the best of the considered options because it requires the fewest switches to achieve 100% routing completion.

4 Conclusions

This paper has examined the effect of routing bend reduction on FPGA routability. It has shown that by reducing the number of bends in the global route for a given connection, the number of wiring tracks (W) required for successful routing decreased substantially for the case of $F_S = 2$. This directly leads to a reduction in the total number of routing switches required for 100% routing completion. Furthermore, it has shown that an improved physical pin placement ('L' shape) contributes to reducing channel density and routing bends. This modification works because it makes the logical pins more accessible from all sides of the logic block. These techniques make a less flexible switch block ($F_S = 2$) comparable in switching requirement with a more flexible ($F_S = 3$) switch block. Also, the circuits can be routed with $F_S = 2$ without a drastic increase in the number of excess tracks. However, based on the total number of routing switches, $F_S = 3$ remains a better choice than $F_S = 2$. Finally, experimental data shows that each logic cell pin should appear on only one side of the logic cell.

Overall, the results in this paper have reinforced the idea that through careful consideration of the interactions between architecture and algorithms, one can achieve substantial improvement in architecture performance.

References

[1] M. Ahrens et. al, "An FPGA Family Optimized for High Densities and Reduced Routing Delay," Proc. 1990 Custom Integrated Circuits Conf., May 1990, pp. 31.5.1 - 31.5.4.

[2] W. Carter et. al, "A User Programmable Reconfigurable Gate Array," Proc. 1986 CICC, May 1986, pp. 233-235.

[3] S. Brown et. al, "A Detailed Router for Field-Programmable Gate Arrays," Proc. ICCAD 90, Nov 1990, pp.382-385.

[4] K. El-Ayat et. al, "A CMOS Electrically Configurable Gate Array," International Solid State Circuits Conf. Digest of Technical Papers, Feb. 1988.

[5] R. Francis et. al, "Chortle-crf: Fast Technology Mapping for Lookup Table-Based FPGAs," Proc. 28th DAC, June 1991, pp. 613-619.

[6] H. Hsieh et. al, "Third-generation architecture boosts speed and density of field-programmable gate arrays," Proc. 1990 CICC, May 1990, pp. 31.2.1-31.2.7

[7] J. Rose, S. Brown, "Flexibility of Interconnection Structures for Field-Programmable Gate Arrays," IEEE Journal of Solid-State Circuits, Vol. 26, No. 3, Mar. 1991, pp. 277-282.

[8] J. Rose et. al, "Architectures of Field-Programmable Gate Arrays: The Effect of Logic Block Functionality on Area Efficiency," IEEE Journal of Solid-State Circuits, Vol. 25, No. 5, Oct. 1990, pp. 1217-1225.

[9] J. Rose, "Parallel Global Routing for Standard Cells," IEEE Transactions on CAD, Vol. 9, No. 9, Sept. 1990, pp. 1085-1095.

[10] J. Rose et. al, "ALTOR: An automatic standard cell layout program," Proc. Can. Conf. VLSI, Nov. 1985, pp. 168-173.

[11] S. Singh et. al, "Optimization of Field Programmable Gate Array Logic Block Architecture for Speed," Proc. CICC 91, May 1991, pp. 6.1.1 - 6.1.6.

[12] S. Wong et. al, "A 5000-Gate CMOS EPLD with Multiple Logic and Interconnect Arrays," Proc. 1989 CICC, May 1989, pp. 5.8.1 - 5.8.4.