The Stratix™ Routing and Logic Architecture

David Lewis*, Vaughn Betz*, David Jefferson, Andy Lee, Chris Lane, Paul Leventis*,
Sandy Marquardt*, Cameron McClintock, Bruce Pedersen, Giles Powell,
Srinivas Reddy, Chris Wysocki, Richard Cliff, and Jonathan Rose*

Altera Corporation, 101 Innovation Drive, San Jose, CA 95134
(*) Altera Toronto Technology Centre, 151 Bloor St W., Toronto, Ont, Canada M5S 1S4

Abstract

This paper describes the Altera Stratix logic and routing architecture. The primary goals of the architecture were to achieve high performance and logic density. We give an overview of the entire device, and then focus on the logic and routing architecture. The Stratix logic architecture is based on a cluster of ten 4-input LUTs and its routing consists of staggered routing lines. We describe the development of the routing architecture, including its directional bias, its direct-drive routing which reduces both area and delay. The logic array block and logic cell design is also described, and new routing structures with in the logic array block, and logic element features are described.

Categories and Subject Descriptors

B.3 [**Integrated Circuits**]

1. INTRODUCTION

The primary goals for Stratix were to achieve high performance and density in a 9 layer metal 0.13µm process. The targeted logic capacity range was approximately 10,000 logic elements (LEs = 4-input LUTs + flip-flops + carries) to 100,000 LEs. In addition to the logic capacity and speed, a range of memory block sizes were to be supported to address the needs of the wide variety of application demands. To provide support for signal processing, a dedicated DSP block was required with support for common signal processing functions. Although these will be described briefly, the focus of this paper will be on the development of the routing architecture, and novel features in the logic element.

Altera devices comprise an array of logic array blocks (LABs) interconnected by rows and columns of routing wires. The LAB architecture naturally creates a tall and narrow layout. This paper shows how this imposes a directional bias on the routing fabric. Further, the larger drivers required for the longer column wires due to the LAB height, coupled with the lower amount of available space for vertical metal also tends to force a horizontal bias in the routing architecture. One focus of this paper is to evaluate the amount of directional bias that is optimum, and how far it can be altered from optimum to adjust for layout constraints.

A second aspect of the routing architecture is the choice of electrical switch. Both speed and the increasing threshold voltage effects in deep sub-micron VLSI also encourage the use of rail to rail routing voltage swings, which avoid signal degradation due to pass transistors. The evaluation of the interaction between switching topology and performance for directly driven routing wires will also be described.

Some novel aspects of the logic element will also be described. The Stratix logic element is an evolutionary development from previous Altera architectures, notably the a mixture of features in the Apex logic cell as well as the Mercury logic cell, but contains several new features to improve density and speed, as well as support new functionality.

Section 2 of the paper gives a brief overview of the Stratix architecture and describes the range of resources available on it. Section 3 describes more detailed development of the routing architecture, with particular focus on the effects of the LAB physical aspect ratio on directionally biased routing architectures, as well as the use of direct drive routing structures. Section 4 describes some novel features of the logic element, and Section 5 concludes.

2. OVERVIEW OF THE STRATIX ARCHITECTURE

An overview of the resources available in a Stratix die is shown in Figure 1. The basic logic resources of Stratix are contained in the Logic Array Blocks (LABs), which will be described further in Sections 3 and 4. Stratix also includes three different memory blocks to support the wide range of applications that are intended for implementation. First, the smallest M512 blocks contain 576 bits and can be organized as 512x1, 256x2, 128x4, 64x9, or 32x9. These efficiently support small buffers, FIFOs, and can also be used as shift registers. Second, the M4K blocks contain 4608 bits and can be organized in widths from 1 to 36 bits. The M4K blocks also support two fully independent ports, including two simultaneous reads or writes. In dual port mode, the width is restricted to 18 bits. The third type of memory, called MegaRAMs, contain 576K bits and are capable of accesses up to 144 bits wide. These offer large bit count to support applications such as large buffers in network applications. Due to the large physical size of the MegaRAMs, they occupy a large physical extent in the FPGA core. Stratix is the first FPGA to include three distinct memory sizes to support diverse applications.

FPGA'03, February 23-25, 2003, Monterey, California, USA.

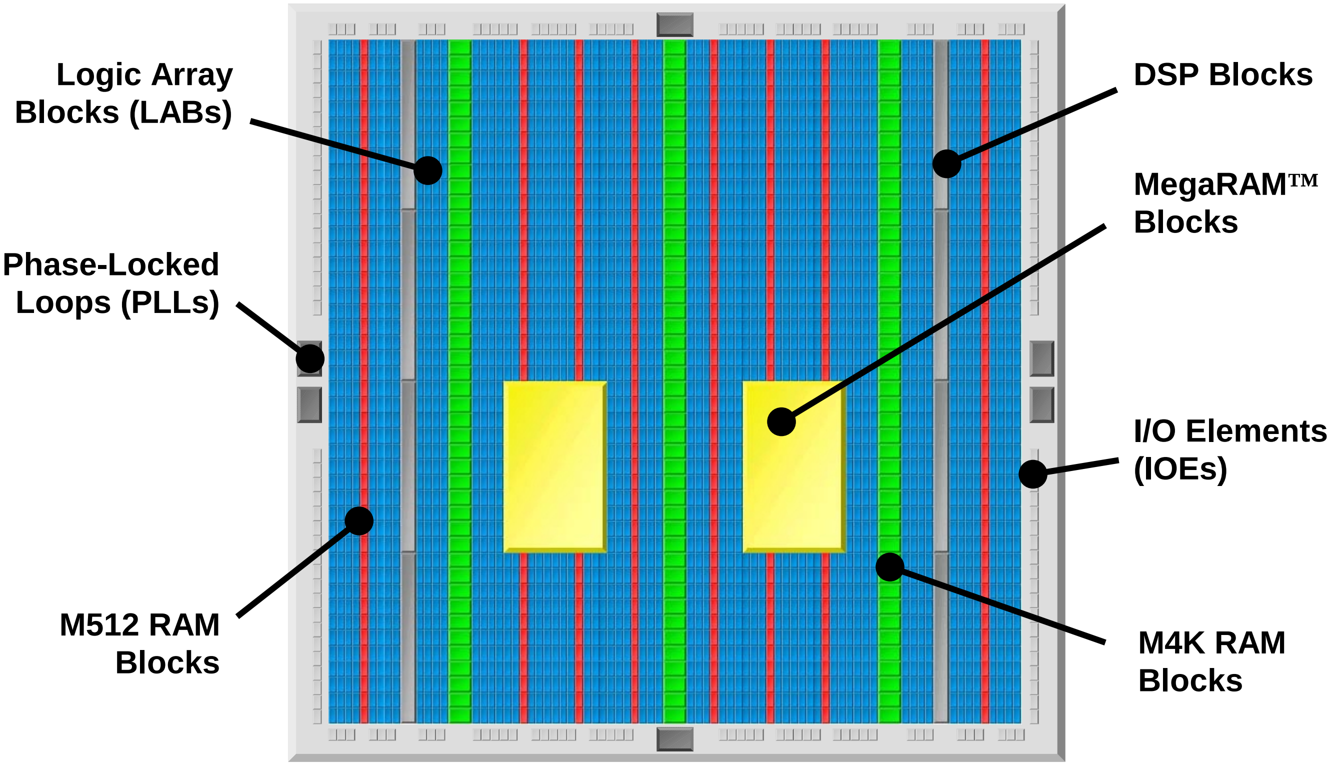


Figure 1 Overview of Stratix Die

Stratix also introduces DSP cores for signal processing applications. The DSP blocks comprise a number of multipliers and adders. These can be configured in various widths to support multiply-add operations ranging from 9x9 to 36x36, and including a wide range of operations from multiplication only, to sum of products, and complex arithmetic multiplication. Internal connectivity in the DSP blocks also supports pipeline registers for common DSP building blocks to support applications such as digital filters.

3. Routing Architecture

This section describes the development of the routing architecture for Stratix. It first describes the experimental infrastructure used to evaluate various architectural choices. Next, we describe layout implications on track count in the channels. Finally an evaluation of the area and delay impact of directly driven (non-tristated) routing wires, and the details of the routing architecture are presented.

3.1 Experimental Infrastructure

The routing architecture was developed using the Altera FPGA Modeling Toolkit (FMT.) Although a detailed description of the FMT is beyond the scope of this paper, a brief description of the tool and methodology will be given. Based on the VPR packing, placement and routing tool [1][10], FMT extends the architectural exploration infrastructure to deal with complexity of modern commercial FPGA architectures. FMT is significantly extended beyound academic VPR to describe a wide variety of physical, logical, electrical, and timing properties of a production architecture. The architecture file required to describe the Stratix architecture is approximately 8,000 lines in length

Given an architecture file, the FMT is used to perform packing, placement and routing experiments on a set of proprietary benchmark circuits. Similar to the methodology described in [1][8], the most common mode of architectural evaluation is to perform placement and routing on a set of circuits using a binary search on the channel width to determine the minimum routable channel width. The array size is chosen to be the minimum that will accomodate the particular circuit, subject to constraints on the ratio of the number of rows to columns. The placement and routing is then repeated using the minimum channel width increased by a constant factor, typically 20%, to approximate the typical use of a FPGA with some excess routing available. After the final placement and routing, the FMT outputs the channel width required to route the circuit and its critical path delay. FMT also uses the detailed transistor level models of the routing switches to compute the silicon area required for the routing fabric and outputs it as well as an estimate of total die area. Using this, the area and delay of a set of FPGA architectures can be compared. The placement and routing algorithms use the same basic algorithms as production Altera software, so the FMT can provide a good prediction of the performance of prototype architectures.

3.2 LAB Architecture and Directional Bias

A large number of Altera devices have used a LAB-based architecture, including the Flex 6000 [6], Flex 8000, Flex 10K, Apex 20K [4,6], and Mercury [2] architectures. Stratix continues to use a LAB-based architecture, but with substantial changes in the routing fabric.

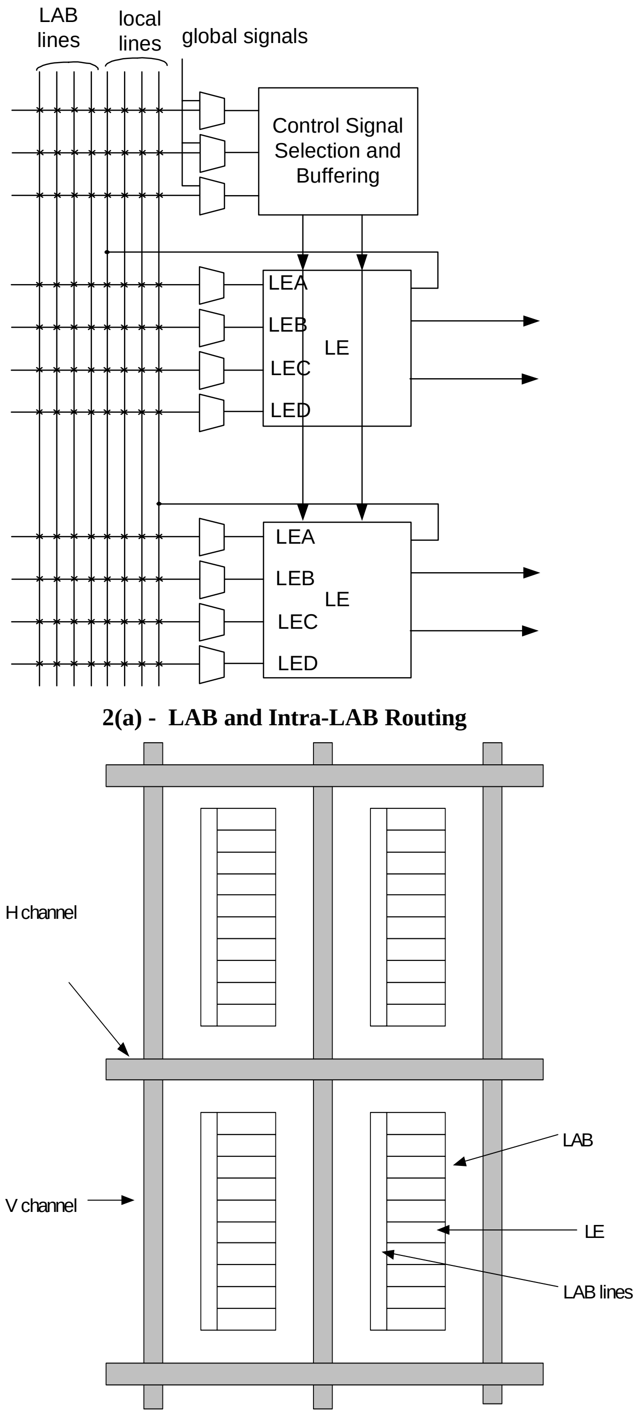
A LAB-based routing architecture consists of a tightly connected block of logic elements connected to a less connected routing fabric. Stratix has two levels of hierarchy of routing resources. The lowest level of the architecture is a logic element (LE) which comprises a single 4-input LUT and flip-flop. The details of the Stratix LE will be described in Section 4. The first level of routing hierarchy is formed by a collection of 10 LEs which are grouped into a logic array block (LAB). These have connectivity between the local routing wires within the LAB. Figure 2(a) shows an overview of a LAB. These routing wires consist of LAB lines which route signals external to the LE to the input pins of the LEs, and local lines, which route the outputs of the LEs to inputs of LEs within the same LAB. The second level is formed by a collection of rows and columns of routing wires to connect the inputs and outputs of the LABs, shown in Figure 2(b). The rows and columns will be referred to as H or V wires (horizontal and vertical) for brevity in this paper.

The goal of the Stratix routing architecture was to provide a high performance routing fabric for the FPGA. This needed to support not only the LABs, but IOs and a variety of memory resources and DSP blocks. Previous Altera architecture have used a hierarchical long-line architecture, in which all the wires in a row or column have aligned starting points and end points. This is effective in a hierarchical partitioning CAD flow as used in previous architectures. The Stratix router's use of negotiated congestion routing algorithms does not achieve benefits from the hard boundaries imposed by hierarchical routing architectures. Consequently an early decision was to focus on architectures with staggered wires, in which the start and end points of wires are staggered uniformly through the routing channels. Staggered wires have also been used in other previous commercial architectures, including the XC4000 [11] and Virtex [12].

A feature that Stratix shares with previous Altera architectures is a directionally biased routing architecture, with substantially different compositions of the row and columns of routing resources. This occurs due to two reasons, both related to the layout aspect ratio of the LAB.

The first consequence of a LAB oriented architecture is the added cost of connections to vertical routing wires. This arises from the larger drivers required for the longer vertical metal trace per LAB distance, as well as the metal width and space required for the wire. This added cost causes a desire to reduce the number of vertical wires compared to the number of horizontal wires. Although the actual transistor area required to implement the routing fabric is modeled in the FMT, it does not model metal density and therefore we resort to more approximate models of cost to evaluate the effect of metal density.

The second issue is a shift in the routing fabric's demand for wire due to the ratio of the number of rows and columns of logic, which is referred to as the logical aspect ratio. In a completely symmetrical architecture using a square LAB and square die there would be no reason to prefer horizontal wires over vertical wires and the ideal ratio of vertical to horizontal wires would be 1:1. When the ratio of rows and columns changes, the routing will prefer one orientation more than the other.



2(a) - LAB and Intra-LAB Routing

2(b) - Global Routing Structure

Figure 2 – Local and Global Routing Structures

Figure 4 illustrates the concept of the physical aspect ratio of the LAB, the physical ratio of the die, and the logical ratio of the die. The physical aspect ratios are the ratios of the actual physical height to width of the layout. For practical chips, the physical aspect ratio of the die tends to be near 1. The logical aspect ratio of the die is the ratio of the number of rows to the number of columns, and is the ratio of the physical aspect ratio of the die to the physical aspect ratio of the LAB. When the logical aspect ratio of the LAB is changed, the routing demand between H and V wiresalso changes.

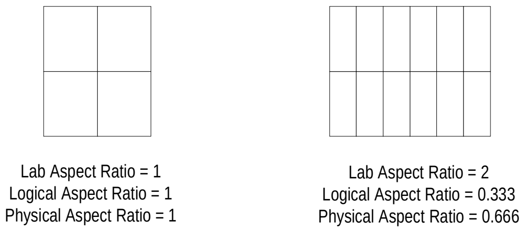

Lab Aspect Ratio = 1
Logical Aspect Ratio = 1
Physical Aspect Ratio = 1
Lab Aspect Ratio = 2
Logical Aspect Ratio = 0.333
Physical Aspect Ratio = 0.666

Figure 4 - Aspect Ratios

This has been investigated previously in [7], however, our analysis includes the physical layout aspects of the LAB and an approximate model of wire cost. Increasing the number of columns compared to rows reduces the logical aspect ratio, and the routing tends to prefer more horizontal wires compared to vertical wires. Because the LAB architecture tends to form a layout that is taller than it is wide, this effect can be exploited to reduce the number of relatively expensive vertical routing wires required. To explore this, two experiments were run. To evaluate the effect of aspect ratios on delay, a set of combinations of LAB aspect ratios from 0.5:1 to 2:1 were used. With a fixed die aspect ratio of 1:1 this implies logical aspect ratios from 2:1 to 0.5:1. For each of the aspect ratios, a set of channel width ratios from 0.5:1 to 2:1 was used. Figure 5 shows the geometric mean critical path delay for all of these combinations. It can be seen that for any aspect ratio, there is a channel width aspect ratio that achieves near minimum delay. Comparing the minimum delay points for each LAB aspect ratio, it can also be seen that the best channel width ratio for each of the skewed aspect ratios is less than a 1% impact on delay.

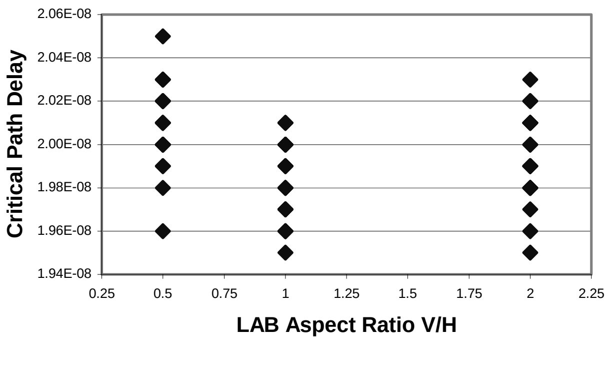


Figure 5 - Critical Path Delays vs. LAB aspect ratio

The optimal channel width expressed in terms of total routing wire demand is displayed in Figure 6. This plots the average number of routing wires per channel (average of horizontal and vertical channels) for various vertical to horizontal routing wire ratios. The plots show the results for three different LAB aspect ratios. These show that a 2:1 change in the aspect ratio causes a proportionally much smaller change in the optimal ratio of vertical to horizontal routing wires. A 2:1 change in LAB aspect ratio causes only approximately 20% change in the optimal ratio of vertical to horizontal routing wires. Thus, this effect appears to be able to only contribute a relatively small shift in the vertical to horizontal wire ratio. If the cost of V wires is substantially higher than the cost of H wires, this could lead to a substantial increase in the overall die area.

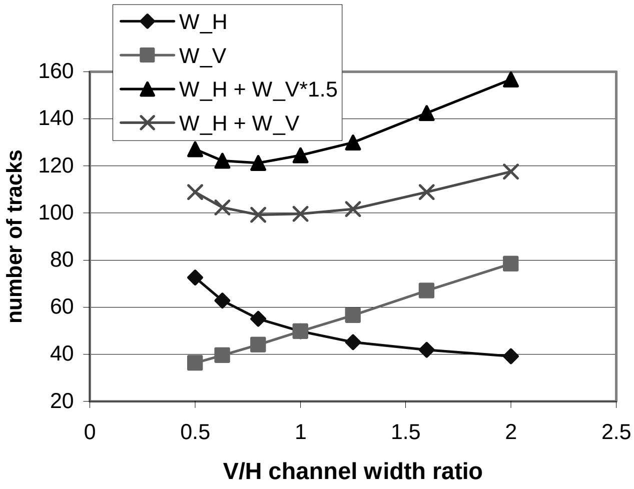


Figure 6. Effect of channel width ratio on total Routing Demand, for various aspect ratios

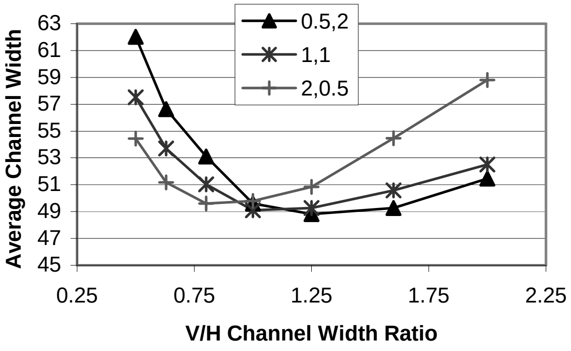


Figure 7 - Channel widths vs. V/H ratio for 2:1 physical aspect ratio, showing 50% larger cost for V wire shifts optimal V/H ratio slightly lower

Further study shows that the total wire count is relatively insensitive to the W_V/W_H across a broad range. Figure 7 shows the values of W_H, W_V as well as the total W_H + W_V * 1.5. The last of these approximates the total cost of routing if a V wire costs 50% more than an H wire. This is a significant penalty that is not directly related to the actual cost, but intended to be illustrative of the effect of larger costs for V wires that may arise due to the limited metal space in the vertical direction. The minimum of the discrete set of data points plotted shows that the minimum cost remains with W_V/W_H = 0.8, but the curve is clearly shifted to the left compared to an equal cost for V and H wires approximated by W_H + W_V. The increase in estimated cost for W_V/W_H = 0.63 is now less than 1% compared to W_V/W_H = 0.8, so choosing any W_V/W_H in the range 0.63 to 0.8 appears reasonable. This makes a narrower V channel feasible, which helps reduce wire demand in the reduced space available for a LAB with physical aspect ratio of 2:1.Thus it is possible to balance metal constraints by trading H and V channel widths to a reasonably broad range.

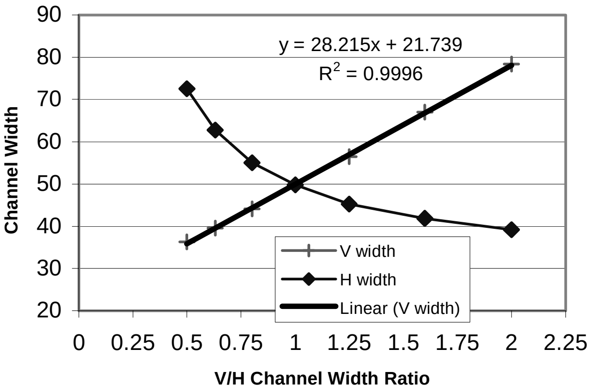


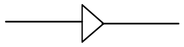
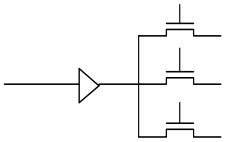
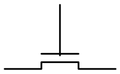
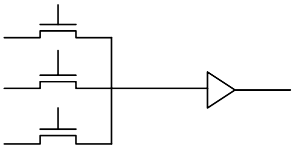
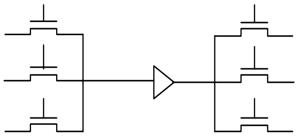
Figure 8 - H and V channel widths obey hyperbolic relationship.

3.3 Detailed Routing Architecture

A primary contributor to overall FPGA performance is the choice of routing switches. FMT can model routing switches containing any combination of input muxing pass transistors, buffers, and output demux transistors. There are 8 combinations of these features, of which 5 unique types are electrically feasible, as shown in Table 1. These are (a) buffers (b) buffers and output pass transistors (c) pass transistors (d) direct drive muxes (e) pass transistor muxes with buffers and output demuxes. PMOS pullups (not shown) are used to restore levels after the input multiplexers driving buffers. Buffers do not have any ability to perform programmable routing but may be present in the architecture at various places to repower signals. Thus the prime aspect to evaluate is the choice of the five possible routing switches.

As transistor dimensions decrease in the 0.13um range and below, transistor Vt becomes an increasing fraction of Vdd. This makes circuits with NMOS-only pass transistors in the routing fabric less attractive due to signal deterioration. Further, it is desirable to use the relatively small transistors on the input side of a routing mux to perform switching, rather than the large transistors required to drive the routing wires. This led to an evaluation of the possibility of using direct drive muxes for some of the routing to achieve high speed and lower area for each routing switch. The potential disadvantage with the use direct drive switches is the restriction that each routing switch is always enabled, so each routing wire can only be driven from a single source. It was expected that the use of direct drive switches would cause an increase in the number of routing wires required, since each wire can only be driven from a single point, reducing the routing flexibility of the wires. Also, an increase in routing area was also anticipated for architectures with a large number of direct drive muxes in the routing fabric due to the decreasing routing flexibility.

Table 1 - Routing Switch Topologies

Input Pass Trans?	Buffer?	Output Pass Trans?	Name	Circuit Diagram
N	Y	N	buffer	
N	Y	Y	buffered switch	
Y	N	N	pass transistor	
Y	Y	N	direct drive mux	
Y	Y	Y	mux-demux	

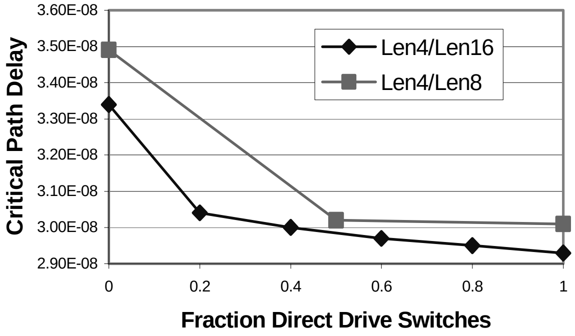


Figure 9 - Average critical path delay for varying fraction of direct-drive routing wires.

Figure 9 shows the effect of using direct drive routing for two routing architectures containing 50% length 4 and 50% length 16, and 50% length 4 and 50% length 8 routing wires. The base architecture uses 50% pass transistors and 50% buffered switches in both architectures. As some of the routing wires are replaced with direct driven wires, the delay decreases as is expected. It appears that only a relatively small fraction of wires need to be direct drive to achieve good speed, although the data set on the length 4 and length 8 architecture is sparser due to CPU time limitations.

The second aspect is the evaluation of the expected increase in routing area due to the use of direct driven wires. Because a directly driven wire can only be driven from one location, it was expected that their use would cause a reduction in the effective utilization of these wires, and hence an increase in channel width and routing area. Contrary to expectations, the use of direct drive routing switches results in a monotonic decrease in routing area as more direct drive switches are used. Figures 10 and 11 illustrate this. For the length-4 and length-8 architecture, the required channel width is approximately constant. For the length-4 and length-16 architecture, there is an increase in channel width due to the fact that single drive point forms a more restrictive routing constraint on the length 16 wires. Although the channel width increases, the total routing area decreases as wide routing transistors are replaced by small ones on the inputs of the routing muxes. Thus in either of the two combinations of wire lengths, 100% direct drive muxes leads to the fastest and smallest architecture.

Figure 12 shows another unexpected benefit of direct drive routing architectures. In our experiments, the final routing is performed on a channel width 20% larger than the minimum required for routing. Figure 13 shows the final critical path delay as the number of excess tracks is varied from the minimum possible to 20% more than the minimum. This clearly shows that additional tracks are required to reduce routing stress and achieve good performance for the pass/buff architecture. In contrast, the direct drive architecture performs well with the minimum channel width and has little benefit from adding extra wires. Although the direct drive architecture requires more wires to route, for any channel width that does route successfully, it achieves much lower critical path delay. Thus the direct drive routing architecture is more tolerant of routing stress and allows a smaller channel width to achieve a given performance.

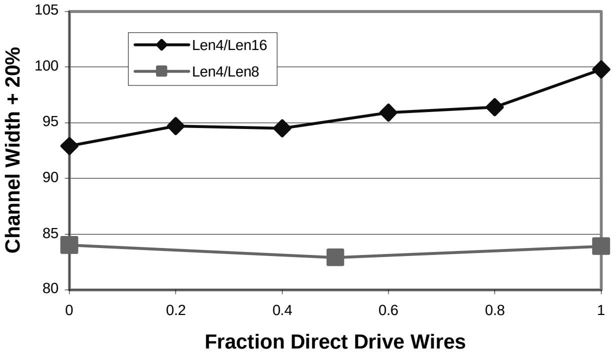


Figure 10. Channel Width required for varying fraction of direct-drive wires.

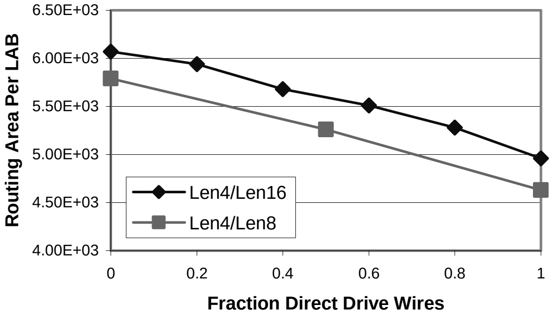


Figure 11. Total routing area required for varying fraction of direct-drive wires.

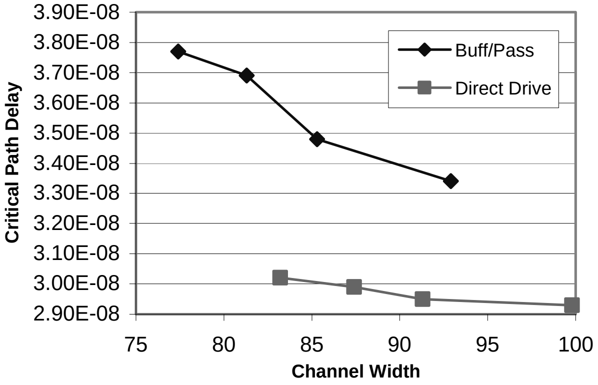


Figure 12. Effect of excess channel width; direct-drive architecture requires fewer wires to achieveoptimal performance. Data points are Wmin+0,5,10,20%

3.4 Routing Wire Mix

The FMT was used to select the overall choice of wire lengths for Stratix. Between the time the above results were generated and the time at the experiments to select the mix of wire lengths were performed, the timing extraction in FMT was improved to more accurately model physical location of the LABs. We also changed the process model to 0.13um.

A set of experiments was performed to determine the best mix of routing wires for area and delay. Figure 13 shows a comparison of a mix of length 4 and 8 wires compared to a mix of length 4 and 16 wires. In both architectures, all of the routing switches are direct drive muxes. The fraction of the wires that are length 4 is varied in both cases. Figure 14 plots the average critical path delay as the fraction of length 4 wires is varied. In all cases, the mix of length 4 and length 8 is faster. The architecture with all length 16 is substantially slower than architecture with all length 8 when no length 4 wires are present.

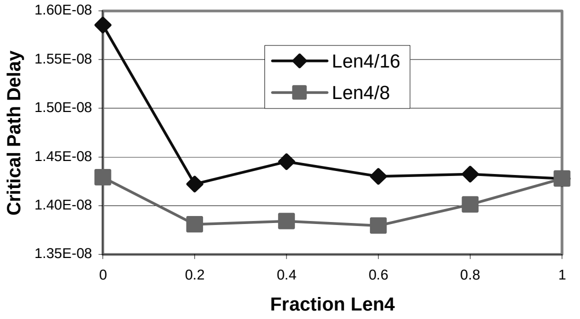


Figure 13. Comparison of routing architectures with mix of length 4/8 wires and length 4/16 wires.

Figure 14 shows the amount of routing area required for the two different mixes of wire lengths. Because average connection length is short, area decreases monotonically with an increasing fraction of length 4 wires. Plotting the area vs. delay for the architecture in Figure 15 shows that the two points corresponding to 60% or 80% length 4 look attractive, and anything with less than 60% length 4 is unattractive. Compared to the study on wire length mix in [8], athough Stratix uses direct drive routing muxes exclusively, as well LABs containing 10 LEs in contrast to the 4 used in [8], it is interesting to note that a mix of predominantly length 4 and some length 8 wires give the best tradeoff between speed and area.

Further experiments on smaller discrete sets of choices of wires including various combinations of fast wires led to the use of a length 4 wire and fast length 8 wire as the primary routing fabric. The length 4 wire is minimum width and space, while the length 8 wire is double width and space.

Additionally, a small number of long distance routing resources were added, using length 24 wires in the horizontal direction and length 16 wires in the vertical direction. The length 16 and 24 wires are on thick metal layers and have wider pitch and space. These wires are designed to provide nearly optimal delay for a given distance. The physical aspect ratio of the LAB makes an optimal V wire shorter in logical LABs compared to an optimal length H wire.

In order to reduce delay and reduce layout difficulty, the length 16 and 24 wires only make connections to other routing resources in a small number of locations along their length, typically every 4 LABs, reducing the number of deep via stacks and electrical loading on these wires. This makes it necessary to use a hierarchical routing structure. Length 16 and 24 wires can only be driven by length 4 wires, and can only drive length 4 wires. Although this entails some delay penalty to drive on and off the length 16/24 network, long distance connections can be made at early optimal speed, and corner to corner delay on the largest die is under 6ns.

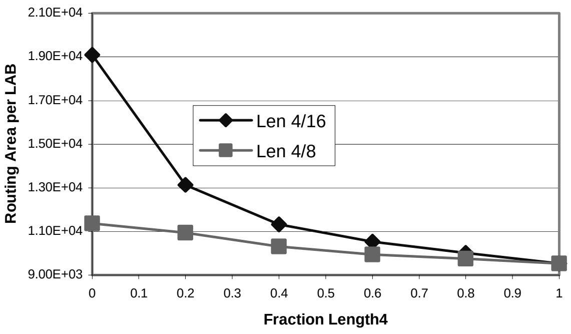


Figure 14. Routing area comparison for length 4/8 and 4/16 architectures.

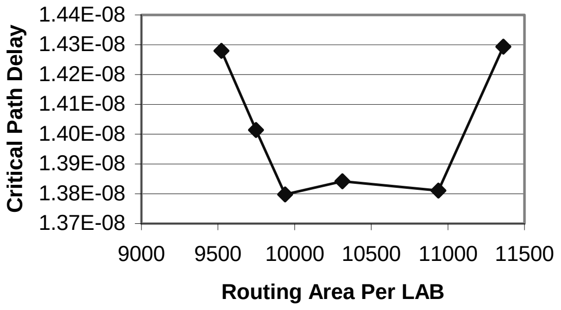


Figure 15. Delay vs. Routing area for length 4/8 architecture.

4. LOGIC BLOCK ARCHITECTURE

The Stratix LAB contains several new features compared to previous Altera architectures to improve both density and performance. These include both intra-LAB routing as well as modifications the LE functionality.

4.1 Intra-LAB Connectivity

As shown earlier in Figure 2 the Altera LAB comprises some number of LAB lines, local lines, and a number of LEs. The LAB lines contain signals input from the general purpose routing fabric. There is one local line per LE, each of which is driven by an output of an LE. Each LE is has identical input connection topology to the LAB lines and local lines. In all previous LAB oriented architectures, the basic LE has a 4-input LUT and a flip flop, with the FF either the LUT output or FF output being able to drive the routing, and the FF loading either from the LUT output or sharing a signal with one of the LUT inputs.

A control signal block routes infrequently used signals with specific LE functionality to the LEs. These signals include synchronous control signals such as clocks, clock enables, and synchronous and asynchronous loads and clears.

Previous Altera architectures allowed each LE to select each LE input from any of the LAB or local lines. This requires a fully populated mux with a fanin equal to the total number of LAB and local lines, with substantial area cost. To reduce the area of the muxes required to perform this selection, Stratix is the first Altera FPGA which has a regular structure in the LAB level connectivity, but a depopulated connectivity. In Stratix each of the LE input pins can only access half of the LAB and local lines, forming a 50% connectivity. Although the use of intra-cluster depopulation has previously been reported [5] in the context of random-logic circuits with a fully permutable LUT, the constraints of layout and various pin restrictions lead to a different connectivity than suggested by [5]. With a 50% population such that each LAB line connects to two of the four LE input pins, 6 different connectivities are possible for the LAB and local lines, with some of the lines connecting to the AB,AC,AD,BC,BD, and CD pins of LEs respectively. Alternatively, a simple depopulation with only two different connectivities, such as AB and CD respectively is possible.

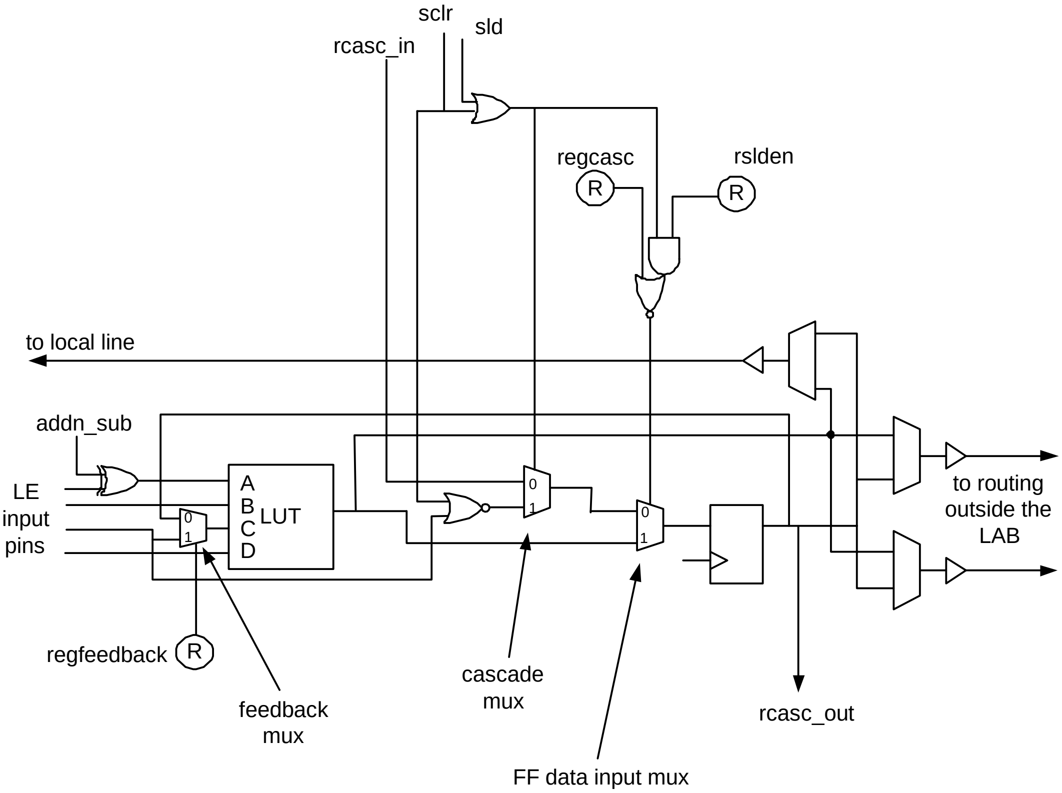


Figure 16. Logical View of Stratix LE functionality.

An important aspect of the LAB line population pattern is that all LEs have identical connectivity to the input pins of the LAB. This means that all LEs are permutable, so that the output of an LE can be routed to any of the available routing resources provided for that LAB, without consideration of the routing to the input. This provides valuable routing flexibility by letting the router assign the location of an LE in the LAB. It is not known if this is also true for previous architectures that use partial connectivity between the general routing and the individual LEs [12].

The Stratix LE makes neither of these depopulation schemes desirable. Experiments with the simple depopulation proved to have poor routability due to conflicts for LE pins on nets with multiple fan outs. The 6-way depopulation is also not attractive due to logical properties of the Stratix LE. In particular, when used in arithmetic mode, only the A and B pins of the LE are required, making the CD lines useless. This is an especially severe constraint, as arithmetic logic in the LAB produces the highest demand for input signals. Therefore Stratix adopts a 4-way depopulation with 25% of the LAB and local lines connected to the AC,BC,AD, and BD pins of the LEs respectively. Experimental evaluation of this structure showed that approximately 7% area reduction could be achieved with less than 1% degradation in performance.

4.2 LE Architecture

Figure 16 shows an overview of the logical functionality of the LE. Similarly to previous LEs in the most recent Apex [4] and Mercury [2] architectures, the essentials of the structure are a 4-LUT which can drive the output of the LE, or be used as the data input of a flip-flop (FF). The FF can also be programmably configured to load its data from the C input of the LE, either unconditionally, or under the control of the SLD signal. LAB-wide SCLR and SLD signals can be used to synchronously clear or load the FF. Asynchronous signals are not shown.

The first of the new features of the Stratix LE is a register feedback mode. In this mode the output of the FF directly drives one of the LUT inputs. In this mode regfeedback configuration bit is set, causing the reg feedback mux to select the output of the FF as the input to the LUT. The data input of the FF is then taken from the corresponding LE input. This allows a FF on a critical net to directly drive its fanout using high speed routing inside the LE instead of using general purpose routing.

A second enhancement is the use of a register cascade feature. A substantial fraction of FFs have another FF for both their fanin and fanout, forming a shift register. Implementing these in LEs requires a mux from a LAB or local line to provide the input data to FF, stealing one from a LUT and requiring that each such FF be paired with a 3-LUT. The register cascade feature adds a 2:1 mux in front of the FF data input, allowing the output of the FF in the adjacent LE to be used as its data. This allows shift registers to be implemented without using any additional routing resources to connect the FF inputs and outputs. When the regcasc configuration bit is set, the FF will be loaded from the register cascade input from the previous LE. Note that the synchronous control logic is operative, so the assertion of SLD or SCLR will cause the FF to be loaded or cleared. The configuration RSLDEN enables the operation of the synchronous control signals in the LE.

Finally, an XOR gate on the input of the LE allows a LAB-wide addn_sub signal to be used to dynamically select between addition and subtraction in the arithmetic modes of the LE. This allows a collection of LEs in the same LAB to use a common signal to select between addition and subtraction.

The Stratix LE also supports features adopted from the Mercury architecture [2], which include the use of a fast LE level cascade, and 5-bit wide block carry select. In contrast to previous Altera architectures, the carry chain is oriented vertically to avoid interruptions due to the frequent vertical stripes of the various RAMs and DSP units.

5. CONCLUSIONS

This paper has described the development of a high performance routing architecture for FPGAs using the Altera LAB architecture. It has shown how physical aspect ratio affects the demand and cost for horizontal and vertical connections, but the overall die cost can be reduced by skewing the ratio of horizontal and vertical channel widths, leading to a directionally biased routing architecture. It also shows that despite the reduced number of locations that can drive a wire using direct drive muxes, there is an overall decrease in both area and delay by using direct drive muxes as the only routing switch in the device. Further, direct drive mux routing architectures can route circuits using near-minimum number of tracks with only a small delay impact due to the increased routing stress.

The Stratix FPGA is also the first Altera FPGA to use a partially populated LAB to LE connectivity to reduce area and delay. The exact choice of routing patterns used here is chosen to include constraints on LE pin usage that arise due to arithmetic modes of the LE. Other features of the LE such as register cascade and quick feedback improve density and performance by allowing registers and LEs to be combined in flexible ways without using extra routing resources.

6. ACKNOWLEDGMENTS

The authors would like to thank Mike Hutton for his review and comments on the paper as well as help in editing.

7. REFERENCES

[1] V. Betz, J. Rose, and A. Marquardt, “Architecture and CAD for Deep-Submicron FPGAs”, Kluwer Academic Publishers, 1999

[2] M. Hutton et al, “Interconnect Enhancements for a High-Speed PLD Architecture”, *Proc FPGA-00*, pp 3-10

[3] R. Cliff et al, “A Next Generation Architecture Optimized for High Density System Level Integration”, *Proc. CICC 99*, pp 175-178

[4] M. Hutton, K. Adibsamii, and A. Leaver, “Timing Driven Placement for Hierarchical Programmable Logic Devices”, *Proc. FPGA-01*, pp3-11

[5] G. Lemieux and D. Lewis, “Using Sparse Crossbars within LUT Clusters”, *Proc. FPGA-01*, pp 59-68

[6] K. Veenstra et al, “Optimizations for a Highly Cost-Efficient Programmable Logic Architecture”, *Proc FPGA-98*, pp 20-24

[7] V. Betz and J. Rose, “Directional Bias and Non-Uniformity in FPGA Global Routing Architectures*”, Proc ICCAD-96*, pp 652-659

[8] V. Betz and J. Rose, “FPGA Routing Architecture: Segmentation and Buffering to Optimize Speed and Density”, *Proc FPGA-99*, pp 59-68

[9] V. Betz and J. Rose, “Effect of the Prefabricated Routing Track Distribution on FPGA Area-Efficiency”, *IEEE Trans. VLSI*, Sept 1998, pp 445-456

[10] V. Betz and J. Rose, “Automatic Generation of FPGA Routing Architectures from High-Level Descriptions”, *Proc FPGA-00*, pp 175-184.

[11] H. Hsieh et al, “Third Generation Architecture Boosts Speed and Density of Field-Programmable Gate Arrays”, *Proc CICC* 1990 pp 31.2.1-31.2.7.

[12] Xilinx Inc., “Virtex-II Platform Handbook”, 2000