Using Bus-Based Connections to Improve Field-Programmable Gate Array Density for Implementing Datapath Circuits

Andy Ye and Jonathan Rose
The Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto
10 King’s College Road, Toronto, Ontario, Canada M5S 3G4
(416) 978-1652, (416) 978-6992
{yeandy, jayar}@eecg.utoronto.ca

ABSTRACT

As the logic capacity of Field-Programmable Gate Arrays (FPGAs) increases, they are being increasingly used to implement large arithmetic-intensive applications, which often contain a large proportion of datapath circuits. Since datapath circuits usually consist of regularly structured components (called bit-slices) which are connected together by regularly structured signals (called buses), it is possible to utilize datapath regularity in order to achieve significant area savings through FPGA architectural innovations. This paper describes such an FPGA routing architecture, called the multi-bit routing architecture, which employs bus-based connections in order to exploit datapath regularity. It is experimentally shown that, comparing to conventional FPGA routing architectures, the multi-bit routing architecture can achieve 14% routing area reduction for implementing datapath circuits, which represents an overall FPGA area savings of 10%. This paper also empirically determines the best values of several important architectural parameters for the new routing architecture including the most area efficient granularity values and the most area efficient proportion of bus-based connections.

Categories and Subject Descriptors

B.7.0 [**Integrated Circuits**]

General Terms

Design, Experimentation, Measurement, Performance

Keywords

FPGA Architecture, Datapath Regularity, Area Efficiency, Routing Architecture, Reconfigurable Fabric

1. INTRODUCTION

Field-Programmable Gate Arrays (FPGAs) that process multiple bits of data at a time are an alternative architectural approach for implementing datapath circuits on reconfigurable hardware that presents new opportunities for exploiting datapath regularity. In particular, multi-bit processing increases the number of signals that can be grouped and routed as buses and, consequently, can be effectively used to reduce the number of programmable routing connections in an FPGA. This reduction in programmable connections can lead to substantial decreases in the number of routing switches and result in significant increases in FPGA routing density. Called multi-bit FPGAs, the detailed implementation of these devices often consists of logic blocks that can process several bits of data at a time and routing resources that connect these logic blocks together. By processing wide data formats, these FPGAs become especially efficient at implementing large arithmetic-intensive datapath circuits including computer graphics, multimedia, digital signal processing, and Internet routing applications.

Several multi-bit FPGA architectures have been proposed in the past [1]–[12] with a wide range of routing architecture designs; and, in this work, we focus on the problem of incorporating bus-based connections into segmented-style routing resources [13]. In particular, we propose a specific routing architecture, called the multi-bit routing architecture, and empirically evaluate its area efficiency. The result obtained is highly relevant to the current FPGA research due to the fact that many state-of-the-art commercial FPGAs (including the Altera Flex, Stratix, and Cyclone series [14] and Xilinx 5200, Virtex, and Spartan families [15] of FPGAs) use similarly styled routing resources and, with their ever-increasing logic capacity, commercial FPGAs are being increasingly used to implement large datapath-intensive applications.

It is essential to have a set of automated design tools to make the effective use of multi-bit architectures. A set of datapath-oriented CAD tools, including synthesis, packing, placement, and routing tools, have been developed at the University of Toronto; and in this paper, these tools are used to investigate the area efficiency of the multi-bit routing architecture by experimentally determining the best granularity values and the best amount of bus-based connections for the proposed routing architecture. Extensive research [16] [17] [18] [19] [20] [21] has been conducted in the past in order to

FPGA'05, February 20-22, 2005, Monterey, California, USA.
Copyright 2005 ACM 1-59593-029-9/05/0002...$5.00.

Logic Block	Vertical Routing Channel
Horizontal Routing Channel	Switch Block

Figure 1: An FPGA Tile

determine the best structures for various conventional FPGA routing architectures. These studies have shown the importance of routing architecture on the overall area-efficiency of FPGAs. None of the studies, however, have explored bus-based connections, which require the preservation of datapath regularity (all these studies use conventional CAD algorithms, which destroy the regularity of datapath circuits and essentially turn datapath into finite state machine-like netlists of “randomly” connected logic gates). In this study, a set of datapath-oriented algorithms [22] [23] [24] is used, which preserve a great amount of user-specified regularity. The preserved regularity, in turn, is used in the investigation of the area efficiency of the proposed routing architecture.

An earlier version of this work [25] explores the design space of bus-based connections in a limited way by using several simplifying architectural assumptions. In this work, we perform a much more extensive search of the entire architectural design space to much accurately determine the most area-efficient values of several important architectural parameters including the best granularity of the bus-based connections. An analytical analysis is also performed to determine the maximum theoretical benefit of bus-based connections on FPGA routing.

The rest of this paper is organized as follows: Section 2 motivates the multi-bit routing architecture design by describing the advantages of the bus-based routing connections in implementing datapath circuits; Section 3 presents the multi-bit routing architecture in detail. Section 4 presents the experimental results on the area efficiency of the proposed routing architecture; and Section 5 gives concluding remarks.

2. BUS-BASED ROUTING CONNECTIONS

The multi-bit routing architecture that we will explore is designed as a tile-based FPGA architecture and is structurally similar to the segmented-style FPGA tiles, which were described in detail in [13]. As shown in Figure 1, a segmented-style tile consists of a logic block, two routing channels, and a switch block. The logic block is designed as a generalized version of the logic blocks used by the Altera FLEX 8K and FLEX 10K series of FPGAs, and is assumed to contain N output pins and I input pins. The routing resources (consisting of the routing channels and the switch block), on the other hand, are similar to the Xilinx 5200, Virtex, and Spartan families of FPGAs; and each routing channel is assumed to contain W routing tracks.

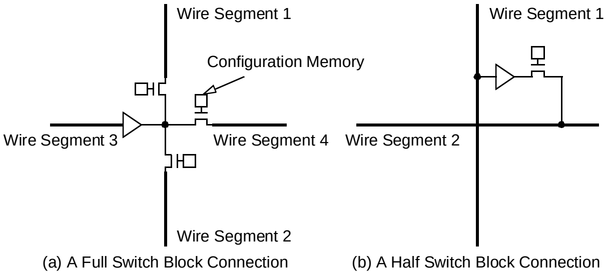


Figure 2: Full and Half Switch Block Connections

Various resources in the tile are connected together through four types of programmable routing connections, including the input connections, the output connections, the full switch block connections, and the half switch block connections. The logic block input pins are connected to the routing tracks through the input connections; and each pin has $\lceil F_{\text{ci}} \times W \rceil$ connections, where F_{ci} represents the fraction of tracks in a routing channel that the pin is connected to. Overall, the total number of input connections per tile, C_{input}, is equal to $\lceil F_{\text{ci}} \times W \rceil \times I$. Similarly, logic block output pins are connected to the routing tracks through the output connections; and the total number of output connections per tile, C_{output}, is equal to $\lceil F_{\text{co}} \times W \rceil \times N$, where F_{co} is equal to the fraction of tracks in a routing channel that an output pin is connected to.

Each routing track in a routing channel is composed of a series of wire segments; and each segment extends across L logic blocks. As shown in Figure 2, a wire segment drives several other wire segments at its ends and at a selected set of internal locations. The set of end connections for the segment (including the driving buffer) as shown in Figure 2(a) are called a full switch block connection; and the total number of full switch block connections per tile, C_{full}, is a function of the topology of the switch blocks. For the most commonly used switch block topology — the disjoint topology [26] — C_{full} is equal to $4\left\lfloor \frac{W}{L} \right\rfloor$. Each internal connection of the segment (also including the driving buffer) as shown in Figure 2(b) is called a half switch block connection, since it requires fewer switches to implement. The number of half switch block connections per tile, C_{half}, also depends on the switch block topology; and, for the disjoint topology, C_{half} is equal to $2\left(W - \left\lfloor \frac{W}{L} \right\rfloor\right)$.

Note that C_{input}, C_{output}, C_{full}, and C_{half} are all monotonically increasing functions of W; and their values as functions of W are summarized in column 2, 3, 4, and 5 of Table 1, respectively. For the table, the calculations are based on a typical FPGA tile with $N = 4$, $I = 10$, $F_{\text{ci}} = 0.5$, $F_{\text{co}} = 0.25$, $L = 2$, and the disjoint switch block topology.

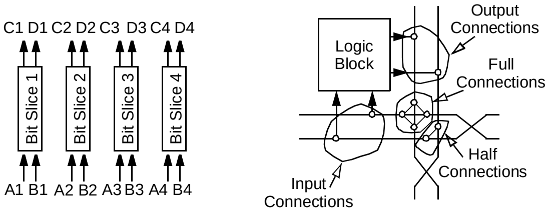
As in previous studies [13], in this paper, the active area (the area consumed by transistors), A, is used to estimate the overall area consumed by various FPGA routing resources; and this area is measured as the number of minimum width transistors based on the following formula:

$$A = \sum_{\text{All Trans.}} \left(0.5 + \frac{\text{Drive Strength of the Current Trans.}}{2 \times \text{Drive Strength of Min. Width Trans.}}\right) \quad (1)$$

Based on the formula, Table 1 also lists the total active area consumed by the FPGA tiles and the total area consumed by their rout-

Table 1: Programmable Routing Connections

W	C_{input}	C_{output}	C_{full}	C_{half}	A_{FPGA}	A_{conn}	$\frac{A_{\text{conn}}}{A_{\text{FPGA}}}$
1	10	4	2	1	2018	190	9.4%
2	10	4	4	2	2140	293	14%
4	20	4	8	4	2463	578	23%
6	30	8	12	6	2823	902	32%
8	40	8	16	8	3086	1128	37%
10	50	12	20	10	3447	1452	42%
12	60	12	24	12	3710	1678	45%
16	80	16	32	16	4274	2168	51%
20	**100**	**20**	**40**	**20**	**4898**	**2718**	**55%**
24	**120**	**24**	**48**	**24**	**5462**	**3207**	**59%**
28	**140**	**28**	**56**	**28**	**6025**	**3697**	**61%**
32	**160**	**32**	**64**	**32**	**6589**	**4187**	**64%**
40	**200**	**40**	**80**	**40**	**7777**	**5227**	**67%**
48	240	48	96	48	8904	6206	70%
64	320	64	128	64	11160	8165	73%
80	400	80	160	80	13475	10185	76%
96	480	96	192	96	15730	12144	77%
128	640	128	256	128	20240	16062	79%



(a) A Datapath Circuit

(b) A Conventional FPGA Tile

Figure 3: An Implementation Example

ing connections in column 6 and 7 respectively. The total connection area as a percentage of the total FPGA area is shown in column 8. As shown, like the connection count, the connection area as a percentage of the total FPGA area increases with increasing W. For small channel widths, the programmable routing connections consist of 10% to 20% of the total FPGA area; for large channel widths, on the other hand, the connection area consists of over 70% of the total FPGA area. Most importantly, for the channel width of 20 to 40, which are the typical track counts for the given architecture parameters (i.e. the number of tracks needed to make the circuits route), the programmable routing connections consume a substantial amount (between 55% to 67%) of the total FPGA area. (Note that for the active area calculations, all transistors in the tile are properly sized using the methodology outlined in [13].)

The large amount of area consumed by the routing connections motivates the multi-bit routing architecture design, which uses more sparse bus-based connections in place of the denser, conventional, bit-based connections. The area advantage of the bus-based connections over bit-based connections is best illustrated through an example. Consider implementing the simple 4 bit-slice datapath circuit, shown in Figure 3(a), using copies of a conventional FPGA tile, shown in Figure 3(b); and also assume that each logic block (will be called the conventional logic block in the remainder of this paper) is just large enough to accommodate a single bit-slice of the circuit. Figure 4 shows the conventional FPGA implementation; and, as shown, one needs a minimum of four FPGA tiles with four routing tracks per channel in order to implement the logic and transport the input and output signals of the circuit; and overall these four tiles consume 16 input connections, 16 output connections, 32 full switch block connections, and 16 half switch block connections.

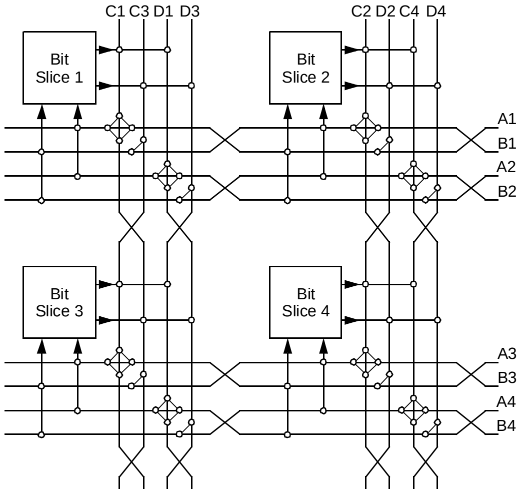


Figure 4: The Conventional Implementation

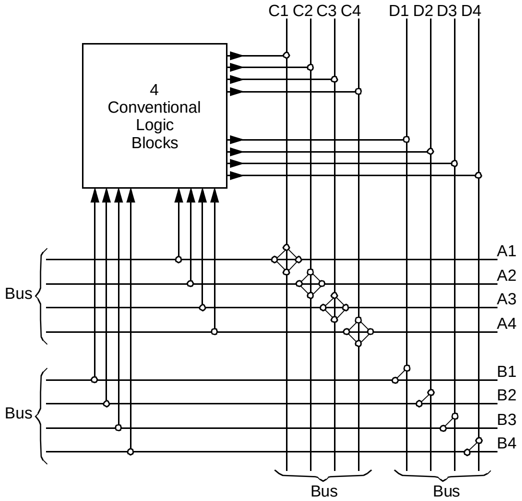


Figure 5: The Multi-Bit FPGA Implementation

Using bus-based connections, on the other hand, one can create a larger FPGA tile that encompasses the entire 4-bit wide circuit by quadrupling the logic capacity of the logic block to create a *multi-bit logic block* (by grouping 4 conventional logic blocks together) and substituting each individual wire in the original tile by a 4-bit wide *routing bus*. Overall, two buses per channel is needed in order to implement the same circuit; and the resulting architecture, shown in Figure 5, contains only 8 input connections, 8 output connections, 16 full switch block connections, and 8 half switch block connections — which is equivalent to an overall reduction of 50% for each type of routing connection. Note that this reduction is due to the fact that routing resources are grouped into buses and programmable connections are only provided for resources that have the same bit positions in their buses.

In general, the basic building block of an FPGA tile with bus-based routing connections is an M-bit wide multi-bit logic block, where M is called the *granularity* (defined as the number of conventional logic blocks that a multi-bit logic block contains) of the tile. This logic block should have the same logic capacity as M conventional logic blocks; and its corresponding routing resources should be constructed out of M-bit wide buses. This means that each routing channel should contain W_B routing buses, where each bus will contain M routing tracks. The input and output pins of the multi-bit logic block should also be grouped into M-bit wide buses (called input buses and output buses respectively); and all the buses are connected together in the same way that individual wires are connected together in a conventional FPGA. So in terms of bus-based connections, F_{ci} and F_{co}, represent the fraction of routing buses in a routing channel that each input bus and output bus are connected to, respectively; L represents the number of multi-bit logic blocks that each wire segment extends across; and the switch blocks of the bus-based connections are created by applying conventional switch block topologies to routing buses instead of individual routing tracks.

Using these architecture parameters, one can generalize the above example to a set of *ideal datapath circuits,* which can be best modeled as networks of M-bit wide *datapath components* that are randomly connected by M-bit wide buses. (Each datapath component is defined to be M bit-slices.) In general, if a datapath component and its associated signals can be implemented by a set of conventional FPGA tiles arranged in a $H \times H$ square with W tracks per channel, the same circuit can also be implemented by a single FPGA tile with bus-based connections. Specifically, the granularity of the tile, M, should be equal to H^2; and the logic capacity of the multi-bit logic block should be H^2 times of the logic capacity of a conventional FPGA logic block. Finally, each routing channel of the tile should contain W_B M-bit wide routing buses where $W_B = \frac{W \times H}{M} = \frac{W}{\sqrt{M}}$. (Note that it is assumed that the formulas for M and W_B apply equally well for both the integer and the fractional values of H.)

Column 3 of Table 2 shows the average active area required to implement a single FPGA tile with bus-based connections for a variety of granularity values; and column 5 lists the area required to implement the corresponding conventional FPGA tiles. These calculations are based on the following architectural parameters: $F_{ci} = 0.5$, $F_{co} = 0.25$, and $L = 2$. We also assume disjoint switch blocks, and $I = 10$ and $N = 4$ for the conventional FPGA logic blocks. Column 6 shows the area reduction of the bus-based connections over the bit-based connections. As shown, FPGA tiles with bus-based connections consume significantly less implementation area as their conventional counter parts; and this area reduction increases with increasing M. For example, for $M = 2$ the area reduction is around 20%; and, for $M = 32$, the area reduction can be as much as 70%.

Table 2: Multi-Bit Routing and Area Reduction

M	Bus-Based		Bit-Based		$\frac{A_{bus}}{A_{bit}}$
	W_B	A_{bus}	$W = \lfloor W_B \times \sqrt{M} \rfloor$	A_{bit}	
2	10	6402	14	8022	80%
4	10	12312	20	19590	63%
8	10	24134	28	48203	50%
12	10	35954	34	83400	43%
16	10	47776	40	124428	38%
20	10	59596	44	166810	36%
24	10	71416	48	213703	33%
28	10	83238	52	265107	31%
32	10	95058	56	321021	30%

Note that the above comparison assumes the same parameter values, including F_{ci}, F_{co}, and L, for both types of FPGA tiles; and it did not experimentally search for the best parameter values for either the bit-based or the bus-based tiles. The comparison also only considers tiles with either purely bit-based or purely bus-based connections; and the bus-based connections can lose area-efficient when they are used to implement circuits that are less regular than the ideal datapath model (circuits containing singular signals, narrower buses, or connected signals that are from different bit positions of buses). To implement these circuits well, one needs an architecture that can efficiently accommodate irregularities as well as highly utilize datapath regularity.

To address the above issues, the remainder of this paper outlines the detailed structure of a routing architecture, called the multi-bit routing architecture, which encompasses both bus-based connections and conventional bit-based connections. The area efficiency of the proposed architecture is then empirically studied using a set of automated CAD tools to determine the best granularity values and the most appropriate amount of bus-based connections that the architecture should contain. Finally, the architecture is compared against the conventional FPGA routing architectures for area efficiency based on a set of the most area-efficient parameter values that are experimentally determined for both architectures.

3. MULTI-BIT ROUTING ARCHITECTURE

In order to accommodate irregularity, the multi-bit routing architecture contains both bus-based and bit-based connections. The overall structure of the architecture is shown in Figure 6, which consists of a two-dimensional array of multi-bit logic blocks interconnected by horizontal and vertical routing channels. Each logic block contains M logic clusters, which is constructed out of a series of Basic Logic Elements (BLEs); and each BLE consists of a Look-Up Table (LUT) and a flip-flop. (Note that the exact structures of the BLEs and the logic clusters are defined in detail in [13] [27].) Each routing channel contains two types of routing tracks — the bus-based tracks (grouped into M-bit wide routing buses), which only use bus-based connections, and the bit-based tracks, which always use bit-based connections.

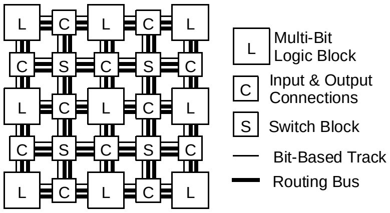


Figure 6: The Multi-Bit Routing Architecture

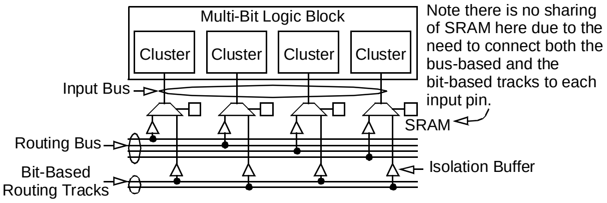

Multi-Bit Logic Block
Cluster
Cluster
Cluster
Cluster
Note there is no sharing of SRAM here due to the need to connect both the bus-based and the bit-based tracks to each input pin.
Input Bus
SRAM
Routing Bus
Bit-Based Routing Tracks
Isolation Buffer

Figure 7: Input Connections ($M = 4$)

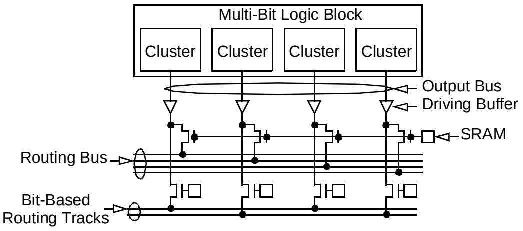
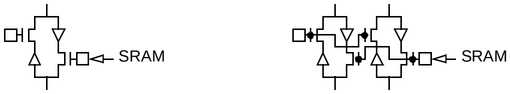


Figure 8: Output Connections ($M = 4$)



(a) Bit-Based Bi-Directional & Buffered (b) Bus-Based Bi-Directional & Buffered (*M*=2)

Figure 9: Routing Switches

The detailed implementations of the input connections, the output connections, and typical examples of the routing switches used in the switch blocks are shown in Figure 7, Figure 8, and Figure 9, respectively. Note that, except the input connections, configuration memory is shared in all bus-based connections in order to further reduce the implementation area. For the input connections, on the other hand, the most area efficient topology, which shares the input multiplexers between the two types of connections, is used instead of sharing the configuration memory.

The set of potential FPGA architectures can be an extremely large design space for any given FPGA architectural definition. For example, 19 architectural parameters are needed in order to completely define the multi-bit routing architecture and its associated multi-bit logic blocks. The comparable conventional architecture discussed in Section 2, on the other hand, can be characterized using fewer design parameters since it is completely bit-based. Nevertheless, 12 architectural parameters are still needed. (Note that each logic block of the conventional architecture is assumed to be a logic cluster.)

This combination of parameters creates a design space that is too large to be explored completely. This study uses a more intelligent exploration strategy where many of these parameters are set to known good values from previous FPGA studies. Care is also taken in the parameter selection process to ensure a fair comparison between the multi-bit architecture and the conventional architecture. In the remainder of this section, we first define each of the architectural parameters and then justify their settings for both the multi-bit and the conventional routing architectures.

3.1 A Summary of Architectural Parameters

Table 3: Multi-Bit Architectural Parameters

Class.	Symb.	Conv. FPGA	Definition
Routing Capacity Parameters	M	no	the granularity of the architecture
	W_f	yes	the number of bit-based tracks per channel
	W_c	no	the number of bus-based tracks per channel (equal to $W_B \times M$)
Logic Block Parameters	K	yes	LUT size — the number of inputs that a LUT has
	N	yes	the number of outputs per logic cluster (Each logic cluster output is directly connected to a unique logic block output; and logic block outputs are grouped into N output buses.)
	I	yes	the number of inputs per logic cluster (Each logic cluster input is directly connected to a unique logic block input; and logic block inputs are grouped into I input buses)
	T_p	yes	the topology of the physical placement of the logic block inputs and outputs
Connection Parameters	F_{cif}	yes	the number of bit-based tracks that a logic block input connects to as a percentage of W_f
	F_{cic}	no	the number of routing buses that an input bus connects to as a percentage of $\frac{W_c}{M}$
	F_{cof}	yes	the number of bit-based tracks that a logic block output connects to as a percentage of W_f
	F_{coc}	no	the number of routing buses that an output bus connects to as a percentage of $\frac{W_c}{M}$
	T_i	yes	the topology of the physical placement of isolation buffers
Switch Block Parameters	T_s	yes	the switch block topology
	F_{sf}	yes	the bit-flexibility of the switch blocks — the number of other tracks that a bit-based track connects to in a switch block
	F_{sc}	no	the bus-flexibility of the switch blocks — the number of other buses that a routing bus connects to in a switch block
	L_f	yes	the segment length of a bit-based track
	L_c	no	the segment length of a bus-based track
	S_f	yes	the type of bit-based routing switches that connect bit-based routing tracks to each other in a switch block
	S_c	no	the type of bus-based routing switches that connect routing buses to each other in a switch block

The 19 architectural parameters that completely define the multi-bit routing architecture are listed in Table 3. The first column shows the classification of the parameters; and column 2 lists the symbol for each. As shown, the parameters are classified into four categories. First, the routing capacity category, contains three members, including M, W_f, and W_c. They characterize the structure of a routing channel. The second category contains four members, including K, N, I, and T_p. Along with M, they completely define the structure of the multi-bit logic blocks. The third and the fourth category, the connection and switch block parameters, contain five and seven members each; and they define the structures of the input and output connections and the switch blocks, respectively.

The 12 parameters that describe the conventional routing architecture is a subset of the 19 parameters listed in Table 3, where column 3 of the table indicates if a given architectural parameter also can be used to describe the conventional architecture. Finally the parameter definition is given in column 4.

3.2 Parameter Values

Two sets of parameter settings are used in this study with one for the investigation of the best granularity and the best proportion of bus-based connections and the other for the comparison of the area efficiency between the multi-bit and the conventional routing architectures. These settings are summarized in Table 4, where the classification and the symbol of each parameter are listed in column 1 and 2, respectively. As shown by column 3, all 19 parameters are involved in the investigation of granularity and proportion, where M, W_f, and W_c are the output (dependent) variables of the investigation. K (LUT size) is set to be 4 since it has been shown to be one of the most efficient LUT sizes [28] [29] and it is also used in many commercial FPGAs [14] [15]. N and I are set to be 4 and 10 respectively since this combination was shown to be one of the most efficient by [27] and is used in many previous FPGA studies [13] [19] [29] [30] [31] [32]. The setting of T_p and T_i are discussed in more detail latter on in the section.

For the remaining parameters, both F_{cic} and F_{cif} are set to be 0.50; and F_{coc} and F_{cof} are set to be 0.25. These values were found to generate good area results by the study done in [13] for bit-based resources. The disjoint switch block topology [26] with F_{sf} and F_{sc} set to be three is used for both the bit-based and the bus-based connections since this is one of the most efficient and widely used topologies for many academic architectures and several commercial ones. A fully buffered global routing architecture is also assumed — all switches in the switch blocks are buffered switches — since buffered switches are widely used in many commercial FPGAs [14] [15] [33]. The track length is set to be two for both types of routing tracks since the track length of two along with the cluster size of four was found to generate good area results in [13] for conventional FPGAs.

Comparing the area efficiency of the multi-bit architecture against the conventional architecture requires the definition of two sets of independent architectural parameters. One set describes the multi-bit architecture and is shown in column 4. The other set describes the comparable conventional architecture and is shown in column 5. For the multi-bit architecture, W_f and W_c are the output (dependent) variables of the investigation. M is set to be 4 since it is shown to be one of the most area efficient granularity values by the granularity investigation. K is again set to be 4.

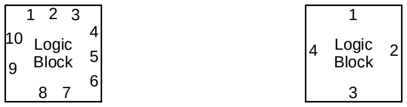
Table 4: Values for Architectural Parameters

Class.	Param.	Granularity & Proportion	Area Efficiency	
			Multi-Bit	Conventional
Routing Capacity Parameters	M	variable	4	n.a.
	W_f	variable	variable	variable
	W_c	variable	variable	n.a.
Logic Block Parameters	K	4	4	4
	N	4	4	4
	I	10	10	10
	T_p	see Sec. 3.2	see Sec. 3.2	see Sec. 3.2
Connection Parameters	F_{cif}	0.5	best	best
	F_{cic}	0.5	equal to F_{cif}	n.a.
	F_{cof}	0.25	best	best
	F_{coc}	0.25	equal to F_{coc}	n.a.
	T_i	see Sec. 3.2	see Sec. 3.2	see Sec. 3.2
Switch Block Parameters	T_s	disjoint	disjoint	disjoint
	F_{sf}	3	3	3
	F_{sc}	3	3	n.a.
	L_f	2	best	best
	L_c	2	equal to L_f	n.a.
	S_f	bi-directional buffered	bi-directional buffered	bi-directional buffered
	S_c	bi-directional buffered	bi-directional buffered	n.a.

The rest of the parameters can be classified into two groups — one group, including T_p, T_i, T_s, F_{sf}, F_{sc}, S_f, and S_c, describes the topological features of the architecture and the other group, including N, I, F_{cif}, F_{cic}, F_{cof}, F_{coc}, L_f, and L_c consists of single numerical values. The topological parameters are set to be the same values as the ones used in the investigation of granularity and proportion. Two of the numerical parameters, N and I, describe the multi-bit logic block structure; and they are also set to be the same values as the ones used to address granularity and proportion.

The remaining numerical parameters describe the multi-bit routing architecture. As shown in Table 4, it is assumed that the architectural parameters that describe the bus-based resources are always equal to their corresponding parameters that describe the bit-based resources. Since the routing resources usually consume the majority of FPGA area, these bit-based parameters, including F_{cif}, F_{cof}, and L_f, are systematically searched to ensure that the best possible area results are obtained for the multi-bit routing architecture. These experimentally determined values will be presented in detail in Section 6.

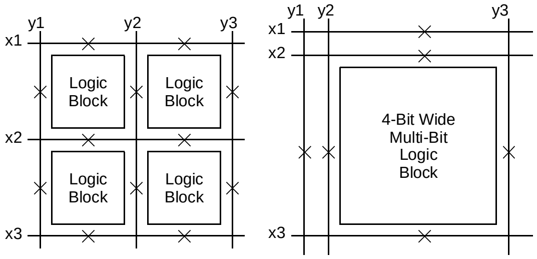
Similarly, the corresponding numerical parameters of the conventional architecture are also systematically searched to find a set of values that generate the best area; and all other parameters are set to be the most area efficient values based on the results of the previous studies.



(a) Physical Placement of 10 Logic Block Input Pins or Input Buses

(b) Physical Placement of 4 Logic Block Output Pins or Output Buses

Figure 10: T_p for $N = 4$ and $I = 10$



(a) Conventional Routing

(b) Multi-Bit Routing

Figure 11: Isolation Buffer Topology

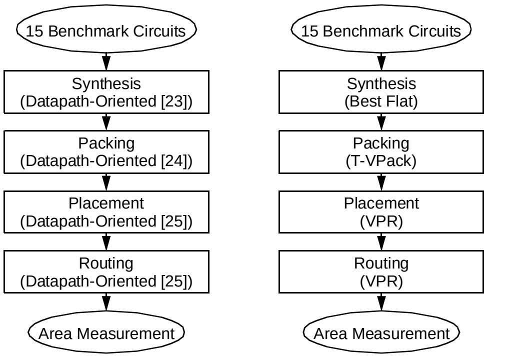
$\boldsymbol{T_p}$ For the conventional FPGA, the logic block inputs and outputs are assumed to be uniformly distributed around the perimeter of each logic block [13]. This distribution topology takes the advantage of the logical equivalency among the cluster inputs or outputs [13]. An example of the distribution topology is shown in Figure 10. Here each number represents either a cluster input or a cluster output.

The multi-bit architecture uses a similar distribution topology. However, instead of uniformly distributing input or output pins, the input buses and the output buses are uniformly distributed. For the multi-bit routing architecture, each number in Figure 10 represents an input/output bus instead of an individual input/output pin. Again this uniform distribution topology takes the advantage of the logical equivalency among input buses or output buses.

$\boldsymbol{T_i}$ The function of the isolation buffers is to electrically isolate the routing tracks from the input connections. For the conventional routing architecture, each routing track has one isolation buffer for each logic block position that it passes [13]. An example is shown in Figure 11(a) where an X indicates the presence of an isolation buffer. In the figure, the conventional FPGA consists of 4 conventional logic blocks, 3 horizontal routing channels, and 3 vertical routing channels. There is one routing track in each channel. In total, there are 12 isolation buffers in the figure. In general, the total number of isolation buffers, C, in a conventional architecture can be determined by the following formula:

$$C = W\times(X\times(Y+1)+(X+1)\times Y)\,, \qquad (2)$$

where W is the number of routing tracks in each routing channel, X is the number of rows of clusters, and Y is the number of columns of clusters.



(a) CAD Flow for the Multi-Bit Architecture

(b) CAD Flow for the Conventional Architecture

Figure 12: CAD Flows

For the multi-bit routing architecture, electrically, it is also sufficient to place only one isolation buffer for every multi-bit logic block position that a routing track passes. However, this topology gives the multi-bit routing architecture an unfair area advantage since it needs only half of the isolation buffers as compared with an equivalent conventional architecture. This unfairness is illustrated by Figure 11(b), which is a transformation of Figure 11(a) by grouping four conventional logic blocks into a multi-bit logic block. As shown, only 6 isolation buffers are needed for the new architecture. In general, the total number of isolation buffers, C', needed in the multi-bit routing architecture is determined by the formula:

$$C' = W\times\left(\frac{X\times(Y+1)}{\lceil\sqrt{M}\rceil}+\frac{(X+1)\times Y}{\lceil\sqrt{M}\rceil}\right) = \frac{C}{\lceil\sqrt{M}\rceil} \qquad (3)$$

Since isolation buffers do not influence the overall routability of the FPGAs, extra isolation buffers are added to the multi-bit architecture to cancel this unfair advantage. For the multi-bit architecture shown in Figure 11(b), two isolation buffers, instead of one, are counted for every multi-bit logic block position that a track passes. (Note that the adjusted isolation buffer placement slightly disadvantages the multi-bit architecture since all routing channels in Figure 11(b) should contain two tracks.)

4. EXPERIMENTAL RESULTS

The multi-bit routing architecture has been used to implement several benchmark circuits using the datapath-oriented CAD flow shown in Figure 12(a). We developed a set of fifteen benchmark circuits by extracting pieces from the Pico-Java processor from SUN Microsystems [34], which cover all the major datapath components of the processor. These circuits are synthesized into LUTs using the EMC datapath-oriented synthesis process as described in [22], which preserves the regularity of datapath circuits while attempting to minimize area. Table 5 gives the name, size (number of BLEs) of each circuit for each synthesis granularity value (here the synthesis granularity is defined to be the maximum datapath width that is preserved by the synthesis process and is an input to the synthesis).

The synthesized circuits are then packed into multi-bit logic blocks using the CNG datapath-oriented packing algorithm as described in [23]. The algorithm packs adjacent bit-slices into a series of logic blocks; and the packed circuits are then placed using a placement algorithm modified from the VPR placer [13] as described in [24]. This datapath-oriented placer moves each multi-bit logic block as a single unit if it contains adjacent bit-slices. Otherwise, each logic cluster is optimized individually. The placed circuits are then routed using the CGR datapath-oriented router as described in [24], which is modified for the efficient use of bus-based routing resources. Using a set of specially designed cost functions, the router tries to balance the use of the bit-based and the bus-based

Table 5: Experimental Circuits

Circuits	#BLEs Obtained at Each Synthesis Granularity					
	1	2	4	8	12	16
code_seq_dp	362	364	364	364	364	364
dcu_dpath	958	962	966	974	982	974
ex_dpath	2823	2747	2649	2719	2947	2955
exponent	467	517	517	539	567	565
icu_dpath	3254	3237	3245	3245	3273	3277
imdr_dpath	1286	1268	1255	1286	1288	1283
incmod	870	862	867	940	948	1005
mantissa_dp	912	919	942	966	971	982
multmod_dp	1602	1636	1634	1636	1636	1636
pipe_dpath	452	499	452	503	503	501
prils_dp	363	396	393	385	385	393
rsadd_dp	350	314	313	305	305	305
smu_dpath	561	557	557	560	563	561
ucode_dat	1264	1273	1304	1278	1282	1286
ucode_reg	78	80	82	86	86	94

routing resources based on routing congestion and the goal of timing optimization.

Figure 12(b) shows the CAD flow used for implementing the same circuits on the comparable conventional routing architecture. For this flow the best available flat synthesis results are used instead of the regularity preserving datapath synthesis. The T-VPack algorithm is then used for packing; and the VPR tools [13] are used for placement and routing.

The area results are measured at the end of the CAD flows. For each multi-bit architecture, a set of experiments was performed by repeatedly invoking the CGR router over a range of values for W_c and W_f. For each invocation a fixed value of W_c is first chosen in increments of M starting from 0 bus-based tracks per channel. Then the router is instructed to search for the minimum number of additional bit-based routing tracks, W_f, that is needed in order to successfully route each circuit. The resulting architectures are then classified into fixed percentile ranges based on the percentage of bus-based routing tracks in the routing channels.

Within each percentile range, the minimum area obtainable for each circuit is first recorded, and then averaged across the benchmarks using arithmetic averaging. Note that since arithmetic averaging is used, the results presented here contain a higher percentage of contribution from the larger benchmark circuits; and the results on granularity, proportion, and area are presented in turn.

4.1 Effect of Granularity on Area

Figure 13 plots the average area required to implement the benchmark circuits against the granularity value, M, for the multi-bit architecture. Here the granularity is shown on the x-axis; and the area is shown on the y-axis. The plot contains two curves. The top curve represents the best area obtainable by a set of multi-bit architectures that contain no bus-based tracks. The bottom curve represents the most area efficient multi-bit architectures containing bus-based tracks. The percentile ranges of bus-based tracks used by the architectures are labeled beside each data point on the bottom curve.

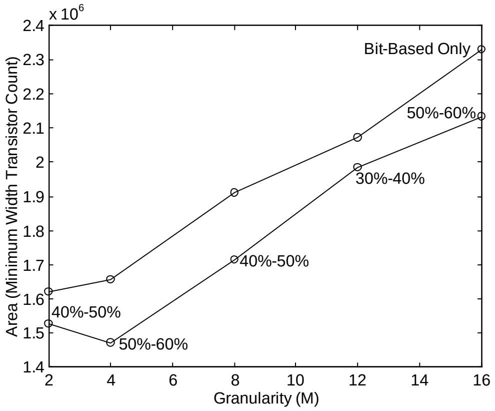


Figure 13: Area vs. Granularity

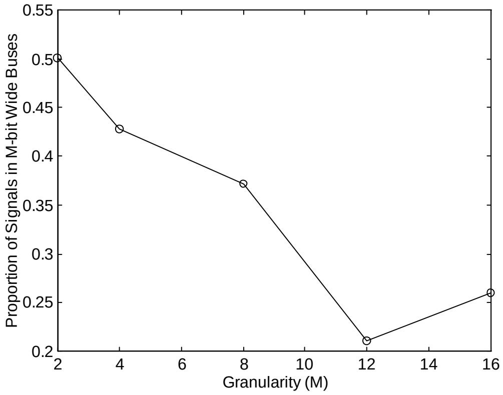


Figure 14: Proportion of Signals in M-Bit Wide Buses

As shown, for all granularity values, the bus-based routing tracks can be used to increase the area efficiency of the multi-bit architecture. For example, when M is equal to 2, the best architecture with bus-based tracks is 5.6% smaller than the architecture with no bus-based tracks; and when M is equal to 4, the best architecture with bus-based tracks is 11% smaller. Overall, the most area efficient multi-bit architectures have a granularity value of 4 and have a bus-based track count that is between 50% to 60% of the total number of routing tracks per channel. This is significantly different from the best granularity values for implementing the ideal datapath model, where, as discussed in Section 2, higher granularity values always result in higher area efficiency. The difference is due to the fact that, unlike the ideal model, a significant percentage of signals in the benchmark circuits cannot be grouped into M-bit wide buses; and, as shown by Figure 14, with increasing M, the percentage of signals that can be grouped into M-bit wide buses decreases nearly monotonically across all benchmark circuits.

4.2 Effect of Proportion of Buses on Area

Figure 15 plots the average area required to implement the benchmark circuits against the percentile ranges for the bus-based routing tracks. In the figure, the x-axis represents the percentage of bus-based tracks per channel; and there are eight percentile ranges

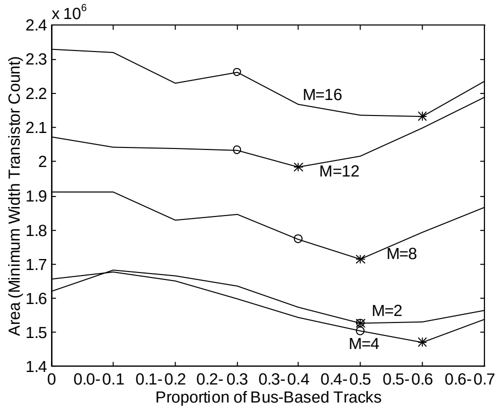


Figure 15: Area vs. Proportion

on the axis, including (0%, 0%], (0%, 10%], (10%, 20%], (20%, 30%], (30%, 40%], (40%, 50%], (50%, 60%], and (60%, 70%]. The y-axis represents the area. There are 5 curves in the figure. Each curve represents a set of multi-bit architectures with a fixed granularity value. An "*" marks the location of the minimum area on each curve. As shown, the most area efficient proportion of bus-based tracks remains within the range of 30% to 60% for all granularity values.

The best proportion values also closely correlate to P, the percentage of signals that can be grouped into M-bit wide buses after packing, since these signals can be most efficiently routed through the bus-based routing tracks. In the figure, the percentile range that P resides in is marked by an "o" for each granularity value; and the graph shows that, with the exception of $M = 16$, the most area efficient percentile range is always less than 1 percentile range away from P.

4.3 Multi-Bit Versus Conventional Routing

To compare the multi-bit routing architecture against the conventional architecture, two sets of experiments were performed. The first set determines the most area efficient values for the numerical parameters described in Section 3. These values are then used in the second set of experiments to measure area.

The best values for three numerical architectural parameters, including F_{cof}, F_{cif}, and L_f, were systematically searched using a divide-and-conquer approach to reduce the number of searches required to explore the three-dimensional design space. To further reduce the number of searches, only the multi-bit architectures that contain no bus-based tracks are used for F_{cof} and F_{cif}. It is assumed that the results are equally applicable across all variations of the multi-bit architectures; and each is described in turn.

$\boldsymbol{F_{\text{cof}}}$ As it is described in [13], for the conventional FPGA architecture, the most area efficient values for F_{cof} is equal to $\frac{1}{N}$. Using the same set of experiments, it is found that the same formula applies to the multi-bit architecture containing no bus-based routing tracks.

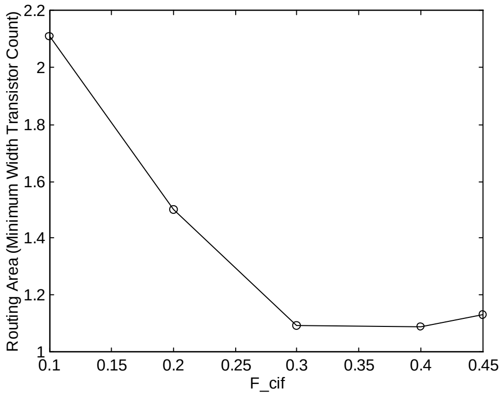


Figure 16: Routing Area vs. F_{cif}

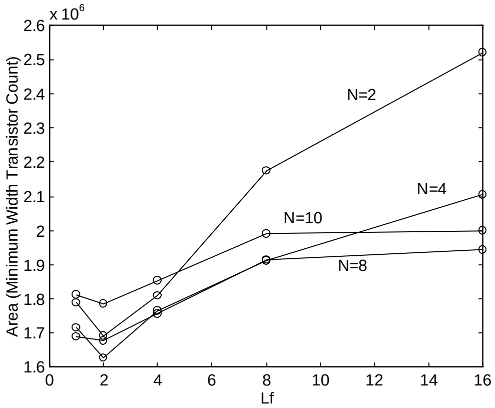


Figure 17: Area vs. L_f

$\boldsymbol{F_{\text{cif}}}$ Figure 16 plots routing area against F_{cif} for the multi-bit routing architecture. As shown, the best routing area occurs when F_{cif} is equal to 0.4. Interestingly the best F_{cif} value for the conventional routing architecture is also equal to 0.4.

$\boldsymbol{L_f}$ Figure 17 is a plot of the average area required to implement the benchmark circuits versus the track length, L_f. It is assumed that 50% of the tracks in the multi-bit architecture are bus-based; and L_c is always equal to L_f. In the figure, the x-axis represents L_f, which ranges from 1 to 16. The y-axis represents the area. There are 4 curves in the figure; and each represents a unique cluster size, N, including 2, 4, 8, and 10. For these cluster sizes, I is set to be 4, 10, 18, and 22, respectively. These values of I are shown to generate good area results for their corresponding cluster sizes [13]. As shown, the cluster size of 4 and the track length of 2 are the best architectural choices for the multi-bit routing architecture. Furthermore, the track length of 2 is always the most area efficient value across all cluster sizes. Finally, using the same experiment, the best L_f value for the conventional routing architecture is determined also to be 2.

Figure 18 plots the total area versus the percentage of tracks that are bus-based for the multi-bit architecture. As shown, when there

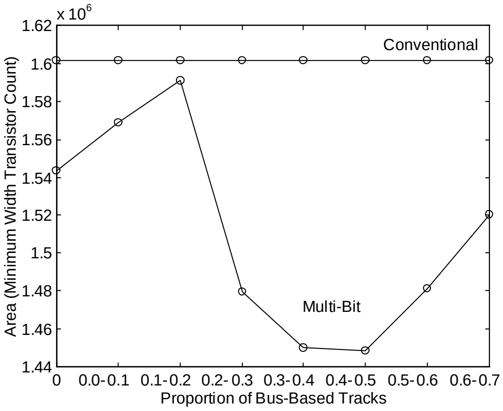


Figure 18: Multi-Bit and Conv. Implementation Area

are only a small percentage of bus-based routing tracks (0%–20%), the implementation area of the benchmarks actually increases with the increasing bus-based track count; and there are two main causes of this increase. First when there are only few bus-based tracks, not all logic block input pins can be connected to all logic block output pins through these tracks. This limitation dramatically reduces the usefulness of the bus-based connections, hence resulting in the increases in area. A secondary cause is that when the bus-based tracks are added to the routing fabric, routing resources are differentiated into two types — the bus-based connections and the bit-based connections. This differentiation reduces the routing flexibility of the routing fabric and also accounts for the rise in area.

As the number of bus-based tracks is increased to the 20%–30% range, enough logic block pins can be connected to each other through the bus-based connections, and the benefit of bus-based routing starts to outweigh the decreases in flexibility. As a result, the total area required to implement the benchmark circuits decreases until it reaches the minimum at the 40%–50% range. When the number of bus-based tracks is further increased, the number of bus-based connections provided by the architecture starts to exceed what is actually required by the circuits. As a result, the router is forced to use bus-based tracks to route individual signals, which reduces the area efficiency of the multi-bit architecture past the 50% point.

Overall, the best area for the multi-bit architecture is achieved when bus-based tracks account for 40% to 50% of the total number of routing tracks. At this point, the benchmark circuits use 6% less area as compared to the multi-bit architectures that contain no bus-based tracks. It is interesting to note that even though 90% of the LUTs in the benchmark circuits belong to four-bit wide datapath components, only 40% to 50% of the tracks should be bus-based. This is because many datapath components are not only connected by buses but also by a substantial amount of non-bus control signals (only forty-eight percent of the signals in the benchmarks can be grouped into 4-bit wide buses); and as a result even highly regular circuits still might need a substantial amount of bit-based tracks.

Comparing to the conventional architecture, all multi-bit architectural variations performed better. Even with no bus-based routing tracks, the multi-bit architecture is 3.6% smaller due to the more efficient datapath-oriented placement and routing. Overall the best multi-bit architecture is 9.6% smaller than the best standard architecture, which represents a routing area reduction of over 14%.

Finally, Table 6 lists the per-circuit-behavior of the benchmark circuits. As shown, for the 40% to 50% percentile range, 12 circuits (representing 79% of the total benchmark area) require less area to implement when compared with the conventional FPGA architecture. When compared with multi-bit FPGA architectures that contain no bus-based tracks, 7 out of 15 circuits, representing 72% of the total benchmark area, require less area to implement.

Table 6: Per Circuit Routing Area (10e5)

Circuits	**Conventional FPGA**	**Multi-Bit FPGA**	
		0% Bus-Based	**40%–50% Bus-Based**
code_seq_dp	3.22	2.66	2.96
dcu_dpath	9.02	8.81	7.24
ex_dpath	38.5	34.6	26.9
exponent	4.79	5.40	5.30
icu_dpath	42.0	40.7	34.2
imdr_dpath	14.5	12.9	11.8
incmod	6.39	7.27	7.73
mantissa_dp	11.4	11.6	10.5
multmod_dp	15.6	14.7	17.9
pipe_dpath	3.35	2.71	3.03
prils_dp	3.07	2.54	2.91
rsadd_dp	2.60	2.09	2.14
smu_dpath	4.36	4.16	4.26
ucode_dat	13.3	11.6	10.5
ucode_reg	0.668	0.565	0.633

5. CONCLUSIONS

This paper has explored the relationship between the bus-based connections and the area efficiency of multi-bit FPGAs. To this end, an analytical analysis of the bus-based connections is first presented, which assumes an ideal datapath model. To account for irregularity that typically present in realistic datapath circuits, the paper then proposes a routing architecture, called the multi-bit routing architecture, that incorporates both bus-based connections and bit-based connections. A set of experiments was then performed using the architecture and a set of datapath-oriented CAD tools. The principle conclusions of this empirical study are that the granularity value of 4 gives the best area result for the multi-bit routing architecture. Furthermore, to achieve the best area, 40% to 50% of the total number of routing tracks should be bus-based despite the fact that, in the benchmark circuits, over 90% of LUTs are in regular datapath components. Finally the best multi-bit architecture is 9.6% smaller than the best conventional architecture, which represents an overall routing area savings of over 14%.

6. REFERENCES

[1] D. Chen and J. Rabaey, “A Reconfigurable Multiprocessor IC for Rapid Prototyping of Algorithmic-Specific High-Speed DSP Data Paths,” *JSSC*, Dec. 1992, pp.1895–1904.

[2] A. Yeung and J. Rabaey, “A Reconfigurable Data Driven Multi-Processor Architecture for Rapid Prototyping of High Throughput DSP Algorithms,” *HICCS,* Jan. 1993, pp.169–178.

[3] R. Bittner and P. Athanas, “Wormhole Run-time Reconfiguration,” *FPGA,* Feb. 1997, pp.79–85.

[4] D. Cherepacha and D. Lewis, “DP-FPGA: An FPGA Architecture Optimized for Datapaths,” *VLSI Design,* 1996, pp.329–343.

[5] C. Ebeling, et. al, “RaPiD — Reconfigurable Pipelined Datapath,” *FPL,* Aug. 1996, pp. 237–241.

[6] J. Hauser and J. Wawrzynek, “Garp: A MIPS Processor with a Reconfigurable Coprocessor,” *FCCM,* Apr. 1997, pp.24–33.

[7] E. Waingold, et. al, “Baring It All to Software: Raw Machines,” *IEEE Computer,* Sep. 1997, pp.86–93.

[8] T. Miyamori and K. Olukotun, “A Quantitative Analysis of Reconfigurable Coprocessors for Multimedia Applications,” *FCCM,* Apr. 1998, pp.2–11.

[9] A. Marshall, et. al, “A reconfigurable arithmetic array for multimedia applications,” *FPGA,* Feb. 1999, pp.135–143.

[10] A. Alsolaim, et. al, “Architecture and Application of a Dynamically Reconfigurable Hardware Array for Future Mobile Communication Systems,” *FCCM,* Apr. 2000, pp.205–214.

[11] S. Goldstein, et. al, “PipeRench: a reconfigurable architecture and compiler,” *IEEE Computer,* Apr. 2000, pp.70–77.

[12] K. Leijten-Nowak and J. van Meerbergen, “An FPGA architecture with enhanced datapath functionality,” *FPGA,* Feb. 2003, pp.195–204.

[13] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs,* Feb. 1999, Kluwer Academic Publishers.

[14] *Altera Documentation Library,* Sep. 2004, Altera Corporation.

[15] *Xilinx Data Sheets,* 2004, Xilinx Inc.

[16] J. Rose and S. Brown, “Flexibility of Interconnection Structures for Field-Programmable Gate Arrays,” *JSSC,* Mar. 1991, pp.277–282.

[17] S. Brown, G. Lemieux, and M. Khellah, “Segmented Routing for Speed-Performance and Routability in Field-Programmable Gate Arrays,” *J-VLSI,* 1996, pp.275–291.

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

[19] V. Betz and J. Rose, “FPGA Routing Architecture: Segmentation and Buffering to Optimize Speed and Density,” *FPGA,* Feb. 1999, pp.59–68.

[20] M. Sheng and J. Rose, “Mixing Buffers and Pass Transistors in FPGA Routing Architectures,” *FPGA,* Feb. 2001, pp.75–84.

[21] G. Lemieux and D. Lewis, “Circuit Design of FPGA Routing Switches,” *FPGA,* Feb. 2002, pp.19–28.

[22] A. Ye, J. Rose, and D. Lewis, “Synthesizing Datapath Circuits for FPGAs with Emphasis on Area Minimization,” *FPT,* Dec. 2002, pp.219–227.

[23] A. Ye and J. Rose, “Using Multi-Bit Logic Blocks and Automated Packing to Improve Field-Programmable Gate Array Density for Implementing Datapath Circuits,” *FPT,* Dec. 2004, pp.129–136.

[24] A. Ye, “Field-Programmable Gate Array Architectures and Algorithms Optimized for Implementing Datapath Circuits,” *Ph.D. Thesis,* Jun. 2004, University of Toronto (http://www.eecg.toronto.edu/~jayar/pubs/theses/Ye/AndyYe.pdf).

[25] A. Ye, J. Rose, and D. Lewis, “Architecture of Datapath-Oriented Coarse-Grain Logic and Routing for FPGAs,” *CICC,* Sep. 2003, pp.61–64.

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

[27] V. Betz and J. Rose, “How Much Logic Should Go in an FPGA Logic Block?,” *IEEE Design and Test Magazine,* Spring 1998, pp.10–15.

[28] J. Rose, R. Francis, D. Lewis, and P. Chow, “Architecture of Field-Programmable Gate Arrays: The Effect of Logic Block Functionality on Area Efficiency,” *JSSC,* Oct. 1990, pp.1217–1225.

[29] E. Ahmed and J. Rose, “The Effect of LUT and Cluster Size on Deep-Submicron FPGA Performance and Density,” *Trans. on VLSI,* Mar. 2004, pp.288–298.

[30] G. Lemieux and D. Lewis, “Using Sparse Crossbars within LUT Clusters,” *FPGA,* Feb. 2001, pp.59–68.

[31] A. Marquardt, V. Betz, and J. Rose, “Timing-Driven Placement for FPGAs,” *FPGA,* Feb. 2000, pp.203–213.

[32] A. Marquardt, V. Betz, and J. Rose, “Speed and Area Tradeoffs in Cluster-Based FPGA Architectures,” *Trans. on VLSI,* Feb. 2000, pp.84–93.

[33] P. Leventis, et al, “Cyclone: A Low-Cost, High-Performance FPGA,” *CICC,* Nov. 2003, pp.49–52.

[34] *Pico-Java Processor Design Documentation,* Sun Microsystems, 1999.