Design and Analysis of Segmented Routing Channels for Row-Based FPGA's

Massoud Pedram, *Member, IEEE,* Bahman S. Nobandegani, and Bryan T. Preas, *Member, IEEE*

***Abstract*— FPGA's combine the logic integration benefits of custom VLSI with the design, production, and time-to-market advantages of standard logic IC's. The Actel family of FPGA's exemplifies the row-based FPGA model. Rows of logic cells interspersed with routing channels have given this family of FPGA devices the flavor of traditional channeled gate arrays or standard cells. However, unlike the conventional standard cell design, the FPGA routing channels contain predefined wiring segments of various lengths that are interconnected using antifuses. This paper develops analytical models that permit the design of FPGA channel architecture and the analysis of the routability of row-based FPGA devices based on a generic characterization of the row-based FPGA routing algorithms. In particular, it demonstrates that using probabilistic models for the origination point and length of connections, an FPGA with properly designed segment length and distribution can be nearly as efficient as a mask-programmable channel (in terms of the number of required tracks). Experimental results corroborate this prediction. This paper does not address specifics of the routing algorithms, but investigates the design of the channel segmentation architecture (i.e, various lengths and patterns of segments and connections among these segments) in order to increase the probability of successful routing.**

I. Introduction

A. Motivation

FIELD Programmable Gate Arrays (FPGA's) are ASIC's that can be configured by the user to perform specific functions. They combine the logic integration benefits of custom VLSI with the design, production, and time-to-market advantages of standard logic IC's. Unlike gate arrays, standard cells, and custom IC's, programmable logic devices do not require non-recurring costs or custom factory fabrication. This permits the designers to define the logic functions of the circuit and revise these functions as necessary at a low cost.

The latest wave of FPGA's combines capabilities that until now were unique to gate arrays, and design benefits will now be found only in the programmable-array logic. These devices split broadly into two groups with different levels of granularity—that is, the size of their basic electronic structures varies. Some chips have large logic structures; the Logic Cell Array from Xilinx Inc., consists of I/O blocks surrounding configurable logic blocks that include Flip-flops as well as combinational logic. Other chips resemble conventional gate arrays in their granularity; the Act 1 and Act 2 FPGA's from Actel Corp., comprises a matrix of logic modules, with the rows separated by wiring channels. The Actel architecture is based on an antifuse technology where logic gates and their interconnections are programmed by shorting wire segments in prescribed locations.

The Actel family of FPGA's exemplifies the standard cell-based FPGA model. The Actel implementation combines the advantage of mask programmable gate arrays with the flexibility of user programming. Rows of logic cells interspersed with routing channels have given the Actel architecture the flavor of traditional channeled gate arrays or standard cells. Unlike traditional gate arrays, the routing channels are not empty areas where customized metallization can be performed. Instead, they contain predefined wiring segments of various lengths. These wiring segments are interconnected using two-terminal programmable elements, called antifuses, that represent one-time programmable elements comprising a diffusion layer placed over polysilicon with a dielectric between them. The antifuse plays the role of a via in this FPGA. There are two types of antifuses: horizontal and vertical. Horizontal fuses are used to realize longer segments by linking two adjacent horizontal segments within a channel. Vertical or cross fuses establish the link between a horizontal segment and an intersecting vertical segment that connects the routing segment on the channel to the pin on the logic block. The actual mechanism to establish a link is to apply a high voltage to the antifuse to break down the dielectric and make a contact between the diffusion and the polysilicon.

Routing interconnections for the FPGA's involves determining which segments to use for which nets. The segmented channel routing problem can be formulated as assigning segments to nets so as to complete all connections within the given routing resources (tracks and fuses). The foremost objective is to achieve 100% routing completion. Another objective is to minimize the number of fuses traversed by each net so as to optimize the timing characteristics.

The channel segmentation design problem is to design an arrangement of segments in routing channels that is as efficient as a non-segmented (mask programmed) routing channel in terms of number of tracks required. There is a substantial need for studying the channel segmentation design problem for row-based FPGA's. Routing for FPGA's is performed by programming fuses that implement the switch elements. These switches have high resistance and capacitance. The number of

Manuscript received January 7, 1994; revised July 19, 1994. This work was supported in part by NSF's Research Initiation Award under contract No. MIP-9211668 and by Xerox, PARC. This paper was recommended by Associate Editor S. Goto.

M. Pedram and B. S. Nobandegani are with the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles CA 90089 USA.

B. T. Preas is with the Computer Science Laboratory, Xerox Palo Alto Research Center, Palo Alto, CA 94304 USA.

IEEE Log Number 9405721.

such switches used in routing a net should be minimized as high resistance and capacitance lead to slower response time for the net and, hence, performance degradation. At the same time, fewer switches imply reduced routability. This tradeoff between performance and routability should be carefully considered as part of the channel segmentation design problem.

Dealing with row-based FPGA's inherently involves the problem of estimating the number of necessary tracks, where tracks are straight sections of wire that span the entire width (or length) of a routing channel and can be composed of a number of wire segments of different lengths. The limited size of the chip constrains the number of available tracks, and this can severely affect the reconfigurability of the logic blocks; so it is extremely important to have accurate estimates of the number of necessary tracks in order to decide how many and what size devices should be used to implement a given logic circuit. In addition, the delay contributed by the programmable interconnect is much higher than that of regular interconnect, and thus the interconnection topology (length and pattern of the switch elements) in FPGA devices strongly influences the performance of the circuit.

In this paper, we describe a generic solution to the segmentation design problem that can be tuned to any particular distribution of the connections. This is particularly important if we take into consideration the fact that various types of circuits should be mapped into different channel architectures in order to achieve higher routability. Our analytic derivations rely on a procedural characterization of the segmented routing algorithms and hence can be tuned as new algorithms replace the older generation.

B. Prior Work

The mask-programmable interconnect provides complete freedom for assigning a connection to any routing track while occupying only a portion of the track equal to the length of the connection plus the design rule spacing. The routing estimation and analysis problems for the customized interconnect have been researched for many years. We give a short review of the important works in this area.

Sastry and Parker [10] modeled interconnections as independent two-point wires covering an average length and derived expressions for channel widths, probability of routing completion, and wire lengths. They showed that wire length distribution has the form of a Weibull distribution with location and shape parameters. Kurdahi and Parker [7] assumed *birth* of a wire at pin slot i and length of a wire l are independent random variables with probabilities $p_B(i)$ and $p_L(l)$. They used uniform distribution for $p_B(i)$ and geometric distribution for $p_L(l)$. Based on these assumptions, the required routing area is estimated.

Sechen [11] presented an interconnection length estimator for sea-of-gates layout style. For each size of net, the half perimeter of the smallest rectangle enclosing all pins on the net is computed by assuming that a sample cell is placed randomly within a square array of area equal to the average number of its neighboring cells. Various scenarios and a look up table are used to determine all possible arrangements of cells that establish a given bounding box. Total interconnection length is then computed by summing (over all nets) the half perimeter lengths of the rectangles enclosing pins on the nets. Chen and Bushnell [3] introduced an area estimator for random placement with the assumption that wires do not share tracks. Pedram and Preas [8] presented formulas for calculating the total interconnection length and the chip height for standard cell layouts based on the notion of net neighborhood populations (which captures the competition among nets for routing resources). Their wire length model relies on knowledge of the actual design processes (placement and routing) and physical structures.

The newest generation of Xilinx devices (XC4000) has provided much more routing flexibility, and hence the routability analysis problem for these devices is not that much different from that of the channeled gate arrays. A routing estimation procedure for the previous generation (XC3000) is given in [1]. We therefore focus on the Actel architecture.

Traditional channel routing cannot be directly applied to row-based FPGA's since it does not take into account the restricted nature of the wiring sources. The following conceptual differences can be recognized. While in channel routing two nets can be assigned the same track if their horizontal spans do not overlap, in case of segmented channel routing each net has to be assigned to a different segment, making the channel density a function of the particular distribution of segment lengths. In gate arrays, the inputs are available simultaneously in two adjacent channels. In FPGA's, the inputs are accessible from either the above or below channel; but not both. This eliminates the vertical constraints since at most one terminal enters the channel at any given column. More details on the architecture for row-based FPGA's and the segmented channel model can be found in [4] and [6], respectively.

New routing estimation and analysis procedures are needed to capture the above differences. This is a very recent development and research is being conducted at different places. In the following, we review some of the relevant work.

A special case of the routing estimation for the segmented channels is addressed by El Gamal *et al.* [5], who showed that (in their own words) with judiciously chosen segment lengths, the channel width needed to achieve high probability of routing completion is not much greater than that for a comparable size gate array. Their assumptions are that the connection lengths are selected independently with geometrically distributed lengths and that the origination points have uniform distribution. They introduced a segmentation model in which the channel is partitioned into several regions and each region consists of tracks of equal-length segments, but segment length is varied uniformly across the regions. Each region is allocated certain number of tracks, however the segments are not arranged in a uniform way but in staggered fashion (see Section II for details). This model is referred to as the **staggered non-uniform segmentation model**.

Burman *et al.* [2] use the staggered non-uniform segmentation model. To optimize the use of this model, they propose a routing scheme based on the assignments of the nets to the appropriate track by delay computation and delay-matching techniques. They determine the three parameters for this model (that is, the number of regions, the number of tracks; and the offset factor) by performing empirical analysis on several standard benchmarks.

Zhu *et al.* [12] describe a few shortcomings of the staggered non-uniform segmentation model and make suggestions as to how to rearrange the segment lengths in order to avoid excessive waste of segments allocated to nets in each length range and minimize the number of track types, and how to scale up or down the number of tracks set aside for nets within a specific range. They also describe an algorithm that takes an arbitrary net distribution and an integer M as inputs and generates a segmented channel that is suitable or M-segment channel routing.

C. Overview and Organization of the Paper

In a mask-programmed design, the routing channel has complete flexibility (within the design rules and routing constraints). As a result, routers for mask-programmed designs can often complete routing using only a few tracks more than the channel density. In contrast, routing for FPGA channels that include predefined sgements is much more constrained as the routing can only by mapped to existing segments. In this paper, we show how to design segmented channel architectures where the number of required tracks is only a few more than that required by the mask-programmed router.

We develop analytical models that enable us to design efficient segmented channel architecture (in the sense described above) as well as to study the routability problem. We adopt the staggered non-uniform segmentation model. Given the probability density functions for the origination point and length of the connections within the segmented channel, we calculate the expected number of segmented tracks of type k[1] and hence the total number of segmented tracks. If the channel segmentation architecture is not fixed, we can use this information to allocate the desired number of tracks of type k. Alternatively, if the channel architecture is fixed and the total number of tracks T is known, we can use this information to predict the probability of routing completion, and if successful routing is possible, the number and types of tracks used. We have derived expressions for generic one- and two-segment routing procedures. Our approach is general and can be applied to other (performance-driven) segmented routing schemes. Finally, we present closed-form expressions for the above parameters assuming specific (but prevalent) density functions for the signal origination point and length. Our models deal with one- and two-segment routing that can be generalized to M-segment routing as well.

The remainder of the paper is organized as follows. Section II contains our detailed analysis and derivations for one- and two-segment routing paradigms. Section III specializes the general results to special cases. Section IV presents the results of our implementations. Based on our derivations, we show that up to 100% of channels generated according to our scheme are successfully routed with random input data. Section V presents some discussions and the ongoing work.

[1] A segmented track of type k contains segments of length equal to some function of k.

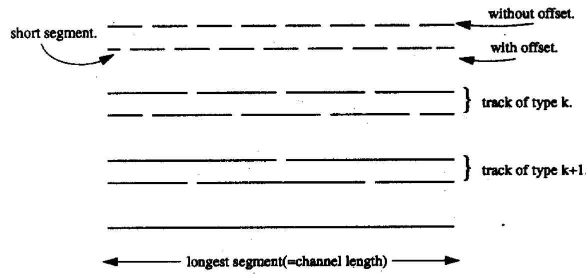


Fig. 1. The FPGA routing channel.

II. Determination of the Segmented Track Requirements

The architecture for row-based FPGA's is described in [4] and the segmented channel model is described in [6] (Fig. 1).

The following sections describe analytic derviations of the segment lengths and distributions under one- and two-segment routing models. We have assumed generic routing algorithms that choose the shortest segments at the first available track for completing connections. This model is consistent with the way most segmented channel routers operate.

A. Channel Architecture Model

We will use the following notation. L denotes the number of columns in the channel and N denotes the total number of connections in the channel that must be routed. The origination point of a connection is wherever a horizontal segment is to be connected to a vertical segment. The distribution of origination points along the length of the channel varies depending on the specific application. We assume this distribution is either given by a histogram or by a closed-form equation. In this way, we can regard the origination point as a random variable with a probability density function $f(x)$; the domain of $f(x)$ is $[0, L]$, where L is the total channel length. Similarly, we assume that the length of a connection is a random variable with a probability density function $h(y)$; the domain of $h(y)$ is $[0, L]$. Using the information embedded in f and h, we can predict the required number of tracks for successful routing or evaluate different routing architectures in terms of their routing completion rate.

Our formulation is general and holds for arbitrary f and h. The distributions of origination point f and connection length h greatly affect the estimations and, hence, realistic choices of f and h are needed. If f and h are not analytically known, we can either fit curves on their corresponding histograms and treat them as analytical functions or use numerical integration techniques. It should be noted that particular forms of f and h are highly dependent on the type of the data. That is, their distributions will be different for data path modules as compared to controller modules. This is one strength of our approach as it enables us to design the best segmentation architecture for different input data types as described below.

We first describe a procedure similar to [5] for constructing a segmented channel. We determine a set Λ_k, $0 \leq k \leq K$ such that $0 = \Lambda_0 < \Lambda_1 < \cdots < \Lambda_K = L$. The set of Λ_k's divides the possible net lengths y into a number of ranges

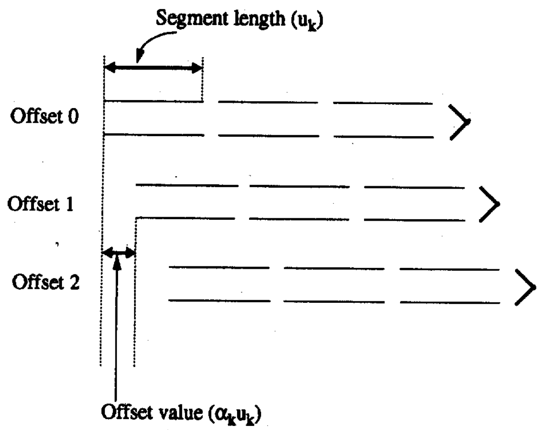


Fig. 2. The staggered non-uniform segmentation model (shown for tracks of type k).

$\Lambda_{k-1} < y \le \Lambda_k, k = 1, \cdots K$. A number of tracks is set aside for nets with lengths in each range (Fig. 2). A group of $\tau^{k,0}$ identically segmented tracks is placed first, where each track is divided into segments of length Λ_k. Then a second group of $\tau^{k,1}$ segmented tracks is placed by shifting the previous group of tracks to the right by $\alpha_k \Lambda_k$ columns for some constant $0 < \alpha_k \le 1$. The procedure repeats $1/\alpha_k$ times where $1/\alpha_k$ is a natural number. These groups of staggered segmented tracks are placed for every length range to construct a segmented channel.

We adopt this staggered non-uniform segmented channel model with the following assumptions. Let $u_1 = \Lambda_1 \ge 2$ denote the segment length within the set of tracks allocated for net lengths $\Lambda_0 < y \le \Lambda_1$. The segment length within the set of tracks allocated for net lengths $\Lambda_{k-1} < y \le \Lambda_k$ will be u_k. A track that contains segments of length u_k will be referred to as a track of type k. Note that K denotes the largest available track type. The length of the largest segment equals the total length. Short segments are created only at the track boundaries and we will ignore their effects in our analysis. This model makes the channel analysis manageable and reasonably fits the actual architectures.

Formally, we can distinguish between two types of routing, i.e. one-segment routing versus M-segment routing. In one-segment routing, no horizontal fuse is allowed between two segments. In M-segment routing, at most M adjacent segments can be connected together and this corresponds to $M-1$ horizontal fuses. In the remainder of this paper, we will focus on one- and two-segment routing.

In the following analysis, we will determine the number $\tau^{k,i}$ of tracks of type k and offset $i\alpha_k u_k$ for $1 \le k \le K$ and $0 \le i \le 1/\alpha_k - 1$ needed for successful routing using one-segment or two-segment routing. The total number of tracks T is the sum of $\tau^{k,i}$'s over the k and i indices.

B. Routing with One Segment

One-segment routing is the case when no horizontal fuse is used. This has the advantage of eliminating the fuse delay. The disadvantage is that when a connection point originates near the end of a segment and spans to a neighboring segment, it cannot be routed by those two segments and must be deferred to a track with larger segment lengths.

The following "one-segment channel routing" scheme is assumed.

Algorithm 2.1:

S1: Given the connection length y, select the candidate track of segments u_k, such that $\Lambda_{k-1} < y \le \Lambda_k$;
S2: If routable with one segment then go to S4;
S3: Switch to track of type $k+1$;
S4: Find a segment whose starting point is closest to the origination point of the connection. Route the connection using that segment;

This is a generic one-segment routing algorithm that captures the common features of many industrial routers and is similar to the left-edge algorithm for segmented channel routing (e.g., [6]).

In general, Algorithm 2.1 may defer a connection more than once to tracks with higher segment lengths until the last track and finish the routing there. However, we prove that a proper choice of offset ratio avoids the multiple deferral problem.

If a connection is deferred, in the worst case, it starts at distance $\alpha_{k+1}u_{k+1}$ from the beginning of a segment in track $k+1$ and its connection length is at most u_k. In order to avoid a second deferral, the following condition must be satisfied: $\alpha_{k+1}u_{k+1} + u_k \le u_{k+1}$.

$$\alpha_{k+1} \le \frac{u_{k+1} - u_k}{u_{k+1}}.$$

This formula ensures that no matter what kind of a distribution for the segment lengths is assumed, a proper value of α_k will result in completion of routing in at most two steps.

Lemma 2.1: Algorithm 2.1 completes the routing of a connection with at most one deferral if $\frac{u_{k+1}}{u_k} > 1$

Proof: In order to avoid more than one deferral, the relation

$$\alpha_{k+1} \le \frac{u_{k+1} - u_k}{u_{k+1}}$$

should hold. However $\alpha_k > 0$, $\forall k$ implying that $\frac{u_{k+1}-u_k}{u_{k+1}} > 0 \Rightarrow \frac{u_{k+1}}{u_k} > 1$. □

We conclude that the segment lengths must increase at least **geometrically** with a ratio that is strictly larger than 1.

A connection with origination point x and length y is routed either in a segment of length u_k with span $\langle (j-1)u_k, ju_k \rangle$ such that $(j-1)u_k \le x < (j-1)u_k + \alpha_k u_k$ or in one segment of length u_{k+1} with span $\langle (j-1)u_{k+1}, ju_{k+1} \rangle$ such that $(j-1)u_{k+1} \le x < (j-1)u_{k+1} + \alpha_k u_{k+1}$. Therefore, the expected number of tracks of type $k+1$ is the sum of the expected number of tracks used by connections whose length is in the range u_k to u_{k+1} and can be routed in one segment, and those whose length is in the range u_{k-1} to u_k and could not be routed using one segment of tracks of type k.

Consider the cases in which $S2$ in Algorithm 2.1 succeeds (Fig. 3).

Clearly $u_{k-1} < y \le u_k$. Consider a channel segment of length u_k whose leftmost point is at $(j-1)u_k$. The staggered segments of type u_k have their leftmost points

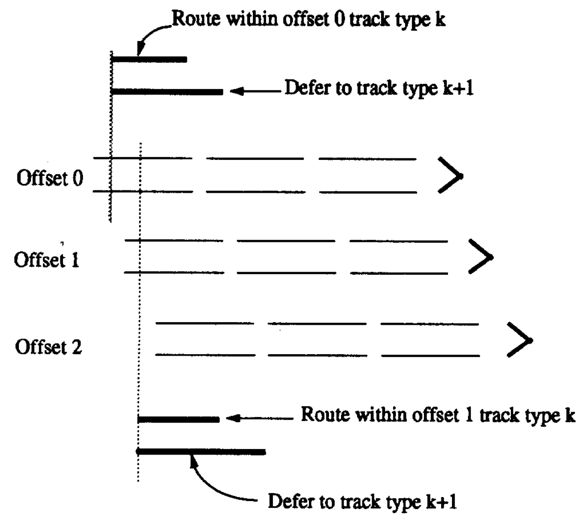


Fig. 3. One-segment routing scheme.

$(j-1)u_k + i\alpha_k u_k$ for $0 \le i \le (1/\alpha_k - 1)$. In the following cases, the connections with origination point x and length y will be routed using exactly one of these staggered segments.

$$\begin{array}{ll}
(j-1)u_k < x \le (j-1)u_k + \alpha_k u_k & u_{k-1} < y \le ju_k - x \\
(j-1)u_k + \alpha_k u_k < x \le (j-1)u_k + 2\alpha_k u_k & u_{k-1} < y \le ju_k + \alpha_k u_k - x \\
\vdots & \vdots \\
(j-1)u_{k_i} + i\alpha_k u_k < x \le (j-1)u_k + (i+1)\alpha_k u_k & u_{k-1} < y \le ju_k + i\alpha_k u_k - x \\
\vdots & \vdots \\
ju_k - \alpha_k u_k < x \le ju_k & u_{k-1} < y \le (j+1)u_k - \alpha_k u_k - x
\end{array}$$

Let $n_1^{k,j,i}$ denote the number of segments of length u_k whose starting points are at $(j-1)u_k + i\alpha_k u_k$ and which are used to complete the routing for non-deferred connections with lengths between u_{k-1} and u_k. Then

$$n_1^{k,j,i} = N \int_{(j-1)u_k + i\alpha_k u_k}^{(j-1)u_k+(i+1)\alpha_k u_k} \times f(x) \int_{u_{k-1}}^{ju_k + i\alpha_k u_k - x} h(y)dydx.$$

Next, consider cases in which $S2$ in Algorithm 2.1 fails (Fig. 3). Note that these connections will be "deferred" to next larger track type and will be successfully routed there (without any further deferrals).

$$\begin{array}{ll}
(j-1)u_k < x \le (j-1)u_k + \alpha_k u_k & ju_k - x < y \le u_k \\
(j-1)u_k + \alpha_k u_k < x \le (j-1)u_k + 2\alpha_k u_k & ju_k + \alpha_k u_k - x < y \le u_k \\
\vdots & \vdots \\
(j-1)u_k + i\alpha_k u_k < x \le (j-1)u_k + (i+1)\alpha_k u_k & ju_k + i\alpha_k u_k - x < y \le u_k \\
\vdots & \vdots \\
ju_k - \alpha_k u_k < x \le ju_k & (j+1)u_k - \alpha_k u_k - x < y \le u_k
\end{array}$$

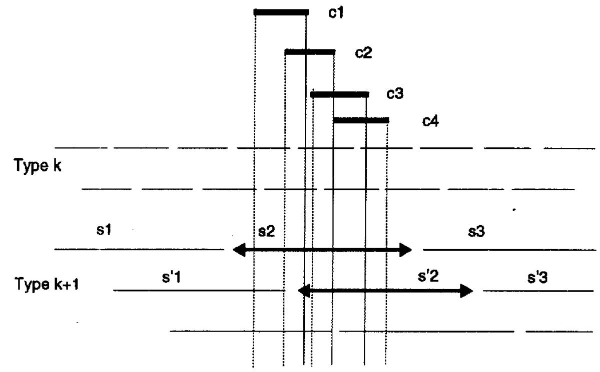


Fig. 4. Deferred connection in one-segment routing.

Let $m_1^{k,j,i}$ denote the number of segments of length u_{k+1} whose starting points are at $(j-1)u_{k+1} + i\alpha_{k+1}u_{k+1}$ and that are used to complete the routing for deferred connections with lengths between u_{k-1} and u_k. Then

$$m_1^{k,j,i} = N \sum_{l=0}^{R_k-1} \int_{(j-1)u_{k+1}+lr_k+i\alpha_{k+1}u_{k+1}}^{(j-1)u_{k+1}+(l+1)r_k+i\alpha_{k+1}u_{k+1}} \times f(x) \int_{(j-1)u_{k+1}+u_k+lr_k+i\alpha_{k+1}u_{k+1}-x}^{u_k} h(y)dydx$$

where R_k and r_k are defined in the sequel. If $\alpha_{k+1}u_{k+1} > \alpha_k u_k$ then $R_k = \frac{\alpha_{k+1}u_{k+1}}{\alpha_k u_k}$ and $r_k = \alpha_{k+1}u_{k+1}$ else $R_k = \frac{1}{\alpha_k}$ and $r_k = \alpha_k u_k$ if $\alpha_k u_k > \alpha_{k+1}u_{k+1}$.

For example, consider Fig. 4. Connections $c1$-$c4$ in this figure are initially assigned to tracks of type k, but are all deferred since they cannot be routed using one segment. Connections $c1$ and $c2$ will be routed using $s2$ in track of type $k+1$ because their origination points are closer to the left edge of $s2$ (they will contribute to $m_1^{k+1,2,0}$). In contrast, connections $c3$ and $c4$ will be routed using $s'2$ since their origination points are closer to the left edge of $s'2$ (they will contribute to $m_1^{k+1,2,1}$).

$m_1^{k,j,i}$ is the number of all deferred connections whose lengths are between u_{k-1} and u_k and are mapped into the segments of length u_{k+1} that begin at $(j-1)u_{k+1} + i\alpha_{k+1}u_{k+1}$.

Finally, the number of tracks of type k with offset $i\alpha_k u_k$ to complete the routing of connections that are assigned to these tracks by the one-segment channel router is given by:

$$\tau_1^{k,i} = \mathrm{MAX}_{j=1}^{L/u_k}\left(n_1^{k,j,i} + m_1^{k-1,j,i}\right)$$

where $n_1^{k,j,i}$ is the contribution of connections of length $\langle u_{k-1}, u_k\rangle$ and $m_1^{k-1,j,i}$ is the contribution of connections of length $\langle u_{k-2}, u_{k-1}\rangle$. The MAX operator is needed to ensure the worst case requirement.

C. Routing with Two Segments

Two-segment routing is the case when at most one fuse is used for any connection. The added flexibility in the use of segments makes two-segment routing more similar to the traditional channel routing.

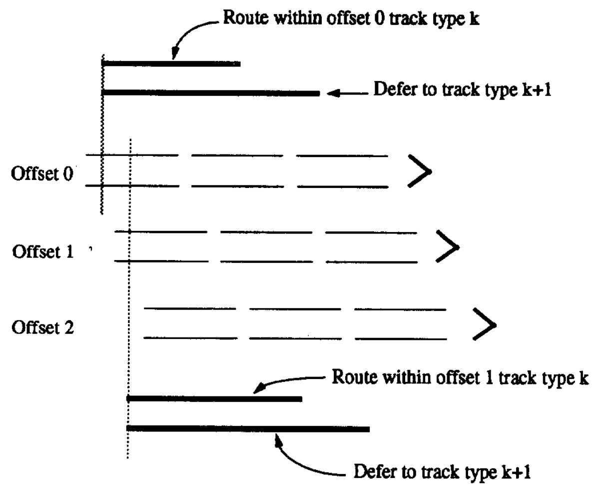


Fig. 5. Two-segment routing scheme.

The following "two segment channel routing" scheme is assumed.

Algorithm 2.2:

S1: Given the connection length y, select the candidate track of type k such that $\Lambda_k \leq y < \Lambda_{k+1}$;

S2: If routable with two segments then go to S4;

S3: Switch to track of type $k+1$;

S4: Find a segment whose starting point is closest to the origination point of the connection. Route the connection using that segment and, if necessary, the one that follows it.

Lemma 2.2: Algorithm 2.2. does not defer more than once.

Proof: If a connection is deferred, in the worst case it starts at distance $\alpha_{k+1}u_{k+1}$ from the beginning of a segment in track k, and its connection length is at most u_{k+1}. However, $\alpha_{k+1}u_{k+1} + u_{k+1} < 2u_{k+1}, \forall \alpha_{k+1} < 1$. □

Clearly $u_k < y \leq u_{k+1}$. Cases in which $S2$ in Algorithm 2.2 succeeds are the following (Fig. 5):

$$\begin{array}{ll}
(j-1)u_k < x \leq (j-1)u_k + \alpha_k u_k & u_k < y \leq (j+1)u_k - x \\
(j-1)u_k + \alpha_k u_k < x \leq (j-1)u_k + 2\alpha_k u_k & u_k < y \leq (j+1)u_k + \alpha_k u_k - x \\
\vdots & \vdots \\
(j-1)u_k + i\alpha_k u_k < x \leq (j-1)u_k + (i+1)\alpha_k u_k & u_k < y \leq (j+1)u_k + i\alpha_k u_k - x \\
\vdots & \vdots \\
ju_k - \alpha_k u_k < x \leq ju_k & u_k < y \leq (j+2)u_k - \alpha_k u_k - x
\end{array}$$

The non-deferred connections can be classified into two groups: $\nu_2^{k,j,i}$, which corresponds to those originating in $\langle (j-1)u_k, (j-1)u_k + i\alpha_k u_k \rangle$ and $\nu_2'^{k,j,i}$, which corresponds to those originating in $\langle (j-2)u_k, (j-2)u_k + i\alpha_k u_k \rangle$ but terminating in $\langle (j-1)u_k, (j-1)u_k + i\alpha_k u_k \rangle$. We can write:

$$\nu_2^{k,j,i} = N \int_{(j-1)u_k + i\alpha_k u_k}^{(j-1)u_k + (i+1)\alpha_k u_k} \times f(x) \int_{u_k}^{(j-1)u_k + 2u_k + i\alpha_k u_k - x} h(y)\,dy\,dx,$$
$$\nu_2'^{k,j,i} = \nu_2^{k,j-1,i}.$$

Thus,

$$n_2^{k,j,i} = \nu_2^{k,j,i} + \nu_2'^{k,j,i}.$$

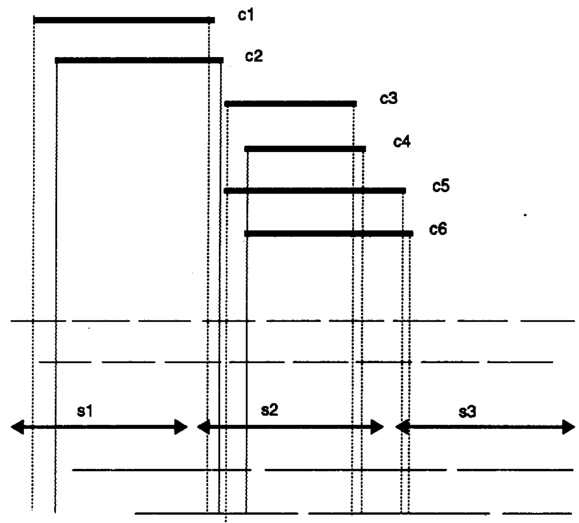


Fig. 6. The two types of deferred connections.

Cases in which $S2$ in Algorithm 2.2 fails are the following (Fig. 5). Note that these connections will be deferred to next larger track type and will be successfully routed there (without any further deferrals).

$$\begin{array}{ll}
(j-1)u_k < x \leq (j-1)u_k + \alpha_k u_k & (j+1)u_k - x < y \leq u_{k+1} \\
(j-1)u_k + \alpha_k u_k < x \leq (j-1)u_k + 2\alpha_k u_k & (j+1)u_k + \alpha_k u_k - x < y \leq u_{k+1} \\
\vdots & \vdots \\
(j-1)u_k + i\alpha_k uk < x \leq (j-1)u_k + (i+1)\alpha_k u_k & (j+1)u_k + i\alpha_k u_k - x < y \leq u_{k+1} \\
\vdots & \vdots \\
ju_k - \alpha_k u_k < x \leq ju_k & (j+2)u_k - \alpha_k u_k - x < y \leq u_{k+1}
\end{array}$$

Consider Fig. 6. Here, we distinguish between the deferred connections that originate in the jth segment of track of type $k+1$ (connections $c3$–$c6$) and those that terminate there (connections $c1$ and $c2$). We denote the number of these connections by $m_2^{k,j,i}$ and $p_2^{k,j,i}$, respectively. Note that all these connections will use $s2$ for routing ($c1$ and $c2$ use $s1$ and $s2$, $c3$ and $c4$ use only $s2$, $c5$ and $c6$ use $s2$ and $s3$).

We can write:

$$m_2^{k,j,i} = N \sum_{l=0}^{R_{k-1}} \int_{(j-1)u_{k+1}+lr_k+i\alpha_{k+1}u_{k+1}}^{(j-1)u_{k+1}+(l+1)r_k+i\alpha_{k+1}u_{k+1}} \times f(x) \int_{(j-1)u_{k+1}+2u_k+lr_k+i\alpha_{k+1}u_{k+1}-x}^{uk+1} h(y)dydx,$$

$$p_2^{k,j,i} = N \sum_{l=0}^{R_k} \int_{(j-2)u_{k+1}+lr_k+i\alpha_{k+1}u_{k+1}}^{(j-2)u_{k+1}+(l+1)r_k+i\alpha_{k+1}u_{k+1}} \times f(x) \int_{(j-1)u_{k+1}+i\alpha_{k+1}u_{k+1}-x}^{u_{k+1}} h(y)dydx.$$

Finally, the number of tracks of type k with offset $i\alpha_k u_k$ needed to complete the routing of connections that are assigned to these tracks by the two-segment channel router is given by:

$$\tau_2^{k,i} = \mathrm{MAX}_{j=1}^{L/u_k} \left(n_2^{k,j,i} + m_2^{k-1,j,i} + p_2^{k-1,j,i} \right)$$

where $n_2^{k,j,i}$ is the contribution of connections of length $\langle u_k, u_{k+1} \rangle$ and $m_2^{k-1,j,i}$ and $p_2^{k-1,j,i}$ are the contribution of connections of length $\langle u_{k-1}, u_k \rangle$.

D. Remarks

We have shown how to obtain the track type distributions $\tau_1^{k,i}$ and $\tau_2^{k,i}$ under the one- and two-segment routing models. This information is used to calculate the required number of tracks of type k subject to given number of tracks in the routing channel. Our derivations are general and independent of the particular forms of distributions of f (origination point) and h (connection length). Once these distributions are known, the optimal staggered non-uniform channel architecture can be generated.

III. Study of Architectures for Specific Distributions

In this section, we give closed-form expressions for three interesting and commonly used special cases. We have used continuous density functions. In the case of discrete density functions that are represented by histograms, a curve-fitting technique that can be used to produce the continuous functions, and then closed form expressions can be derived. Alternatively, numerical integration techniques may be applied.

The staggered non-uniform segmented-channel model already fixes the architecture to some extent. The free parameters of the architecture are u_k, α_k and $\tau^{k,i}$. We show how to apply the derivations in Section II to some prevalent distributions. We have picked three example distributions. Similar calculations could be done for other types of distributions. In all three cases we assume that the origination point has *uniform* distribution. This choice of distribution for the origination point is very close to reality as confirmed by empirical studies by the authors as well as others [9]. We prove that *uniform* distribution of origination point results in omission of the MAX opeator and this simplifies the problem to a large extent.

Lemma 3.1: When the origination point has a uniform distribution, the MAX operator in equations for $\tau_1^{k,i}$ and $\tau_2^{k,i}$ can be omitted.

Proof: We prove that $n_1^{k,j,i}$ is independent of j. A similar proof can be given for $m_1^{k,j,i}$, $n_2^{k,j,i}$, $m_2^{k,j,i}$ and $p_2^{k,j,i}$.

The first (inner) integral in the equation for $n_1^{k,j,i}$ evaluates to

$$H(ju_k + i\alpha_k u_k - x) - H(u_{k-1}).$$

The $H(u_{k-1})$ term is a constant and independent of the x. After the second (outer) integration, this term is simply multiplied by $(j-1)u_k + (i+1)\alpha_k\, u_k - (j-1)u_k + i\alpha_k u_k = \alpha_k u_k$, which is independent of j. So, MAX operator does not affect the second term. The $H(ju_k + i\alpha_k u_k - x)$ term is also independent of j because

$$\int_{(j-1)u_k+i\alpha_k u_k}^{(j-1)u_k+(i+1)\alpha_k u_k} H(ju_k + i\alpha_k u_k - x)dx = -\int_{u_k}^{(1-\alpha_k)u_k} H(z)dz$$

where $z = ju_k + i\alpha_k u_k - x$. Both the lower and upper bounds of the integral on the right-hand side are independent of j, and so is the total result. Therefore, $n_1^{k,j,i}$ is not affected by the MAX operator, whose running index is j. □

A. Uniform Connection Length

If we assume that both the connection length and origination point have uniform distributions, $n_1^{k,j,i}$ and $m_1^{k,j,i}$ can be expressed in closed forms. The choice of uniform distribution for the connection length corresponds to a random placement of modules on the FPGA device. We are presenting this case only for the sake of illustration. Uniform distribution implies that $f(x) = 1/L$ and $h(x) = 1/L$. N denotes the total number of connections. The relevant closed form expressions are as follows.[2]

For one-segment routing:

$$n_1^{k,j,i} = \frac{N}{L^2}\alpha_k\left(u_k - \frac{\alpha_k u_k}{2} - u_{k-1}\right)u_k$$
$$m_1^{k,j,i} = \frac{N}{2L^2}\alpha_k \alpha_{k+1} u_k u_{k+1}$$

For two segment-routing:

$$n_2^{k,j,i} = \frac{2N}{L^2}\alpha_k\left(1 - \frac{\alpha_k}{2}\right)u_k^2$$
$$m_2^{k,j,i} = \frac{N}{L^2}\alpha_{k+1}\left(u_{k+1} - 2u_k + \frac{\alpha_k u_k}{2}\right)u_{k+1}$$
$$p_2^{k,j,i} = \frac{N}{2L^2}\alpha_{k+1}^2 u_{k+1}^2$$

These formulas indicate an exponential increase in the number of required tracks as k increases. This makes sense if we recall that the uniform connection length suggests that the number of short and long connections are equal, and that the segment lengths on FPGA tracks with short and long segments differ in an exponential fashion. Thus, exponentially more tracks with long segments are required to accommodate the same number of connections.

[2] For the sake of brevity, we omit the derivation of these expressions that are lengthy, but rather straightforward.

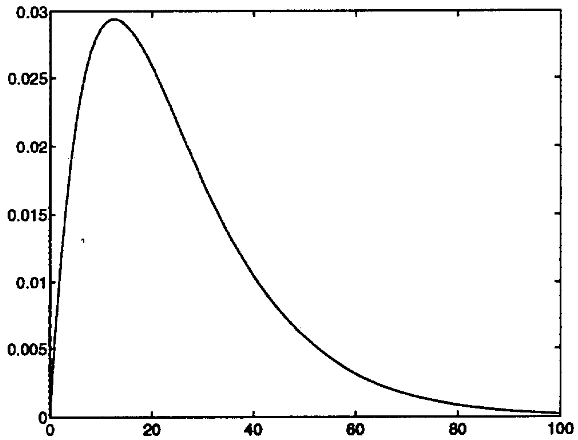


Fig. 7. Gamma distribution ($b = 1$).

B. Exponential Connection Length

Another interesting case is when the origination point is uniform, but the connection length has an exponential distribution, $f(x) = 1/L$ and $h(x) = \lambda e^{-\lambda x}$. λ is the inverse of the expected value of connection length. This corresponds to the wire length distribution after placement optimization. Keeping in mind that exponential distribution is the continuous counter part of geometric distribution, it seems to fit better to empirical data. The relevant closed form expressions are as follows:

For one-segment routing:

$$n_1^{k,j,i} = \frac{N\alpha_k}{L}\left(u_k e^{-\lambda u_{k-1}} - \frac{1}{\alpha_k \lambda} \times \left(e^{-(1-\alpha_k)\lambda u_k} - e^{-\lambda u_k}\right)\right)$$

$$m_1^{k,j,i} = \frac{N\alpha_{k+1}}{L\alpha_k}\left(\left(\frac{-1}{\lambda} - \alpha_k u_k\right)e^{-\lambda u_k} + \frac{1}{\lambda}e^{-\lambda(1-\alpha_k)u_k}\right) \times \frac{u_{k+1}}{u_k}$$

For two-segment routing:

$$n_2^{k,j,i} = \frac{2N}{L}\left(\alpha_k u_k e^{-\lambda u_k} - \frac{1}{\lambda}e^{-\lambda u_k(2-\alpha_k)} + \frac{1}{\lambda}e^{-2\lambda u_k}\right)$$

$$m_2^{k,j,i} = \frac{N_{\alpha_{k+1}}}{L\alpha_k}\left(\frac{1}{\lambda}e^{-2\lambda u_k}\left(e^{\lambda\alpha_k u_k} - 1\right) - \alpha_k u_k e^{-\lambda u_{k+1}}\right) \times \frac{u_{k+1}}{u_k}$$

$$p_2^{k,j,i} = \frac{N}{L}e^{-\lambda u_{k+1}}\left(\frac{1}{\lambda}\left(e^{\lambda\alpha_{k+1}u_{k+1}} - 1\right) - \alpha_{k+1}u_{k+1}\right)$$

C. Gamma Connection Length

Another important case is when the origination point is uniform, but the connection length has a *gamma* distribution, $f(x) = 1/L$ and $h(x) = Ax^b e^{-cx}$. A is equal to $\frac{c^{b+1}}{\Gamma(b+1)}$ where if $b = n$ is an integer, then $\Gamma(n+1) = n!$. The expected value of the *gamma* distributions is equal to $E(x) = \frac{b+1}{c}$. To simplify the problem we consider $b = 1$, which results in $A = c^2$ where $2/c$ is equal to the expected value of the connection length (Fig. 7).

The relevant closed-form expressions are as follows: For one-segment routing:

$$n_1^{k,j,i} = \frac{N}{L}\left(e^{-cu_k}\left(\frac{2}{c} + u_k\right) + e^{-(1-\alpha_k)cu_k} \times \left(-\frac{2}{c} - (1-\alpha_k)u_k\right) + e^{-cu_{k-1}}\alpha_k u_k(cu_{k-1} + 1)\right)$$

$$m_1^{k,j,i} = \frac{N_{\alpha_{k+1}}}{L\alpha_k}\left(\left(-\frac{2}{c} - (1 + \alpha_k + \alpha_k c u_k)u_k\right)e^{-cu_k} + \left(\frac{2}{c} + (1-\alpha_k)u_k\right)e^{-c(1-\alpha_k)u_k}\right)\frac{u_{k+1}}{u_k}$$

For two-segment routing:

$$n_2^{k,j,i} = \frac{2N}{L}\left(\left(\frac{2}{c} + 2u_k\right)e^{-2cu_k} + \left(\alpha_k u_k - \frac{2}{c} - 2u_k\right) \times e^{-cu_k(2-\alpha_k)} + \alpha_k u_k(cu_k + 1)e^{-cu_k}\right)$$

$$m_2^{k,j,i} = \frac{N\alpha_{k+1}}{L\alpha_k}\left(e^{-2cu_k}\left(-2u_k - \frac{2}{c}\right) + e^{-cu_k(2-\alpha_k)} \times \left(2u_k + \frac{2}{c} - \alpha_k u_k\right) - \alpha_k u_k e^{-cu_{k+1}}(cu_{k+1} + 1)\right)\frac{u_{k+1}}{u_k}$$

$$p_2^{k,j,i} = \frac{N}{L}\left(\left(\frac{1}{e^{c\alpha_k u_k - 1}}\right)\right)\left(\left(\frac{2}{c} + (1-\alpha_{k+1})u_{k+1}\right) \times e^{-c((1-\alpha_{k+1})u_{k+1} - \alpha_k u_k)} - \left(u_{k+1} + \frac{2}{c}\right)e^{-c(u_{k+1} - \alpha_k u_k)} - \left(u_{k+1} + \frac{2}{c} - \alpha_{k+1}u_{k+1}\right)e^{-c((1-\alpha_{k+1})u_{k+1}}\right) + \left(u_{k+1} + \frac{2}{c}\right)e^{-cu_{k+1}}\right) - \alpha_{k+1}u_{k+1}(cu_{k+1} + 1)e^{-cu_{k+1}}\right)$$

IV. Empirical Results and Discussions

A C program was developed to compute the segmented channel parameters and generate the segmented channel architecture. We assumed that the lengths of segments follow a geometric progression model, i.e., $u_k = u^k$ where $u > 2$ denotes the basic segment length (the length of the segments in track of type 1). We used the same offset ratio for all of the tracks $\alpha_k = \alpha$. Short segments were generated only at the boundaries of the segmented track. The algorithms (and the code) for one-segment and two-segment routing are from Zhu *et al.* [12].

The fixed parameters of the routing channel are given in Table I.

In the tables that follows, λ and c are the parameters of the exponential and gamma distributions, respectively. NT is the total number of tracks generated in a routing channel. We generated 100 random routing problems. The success rate (SR) denotes the percentage of routing problems that were successfully routed on the specified architecture using NT

TABLE I
PARAMETERS FOR THE RANDOM CHANNELS

Maximum number of nets per channel	121
Minimum number of nets per channel	31
Maximum net length	99
Minimum net length	1
Maximum channel density	36
Number of columns	100
Number of channels generated	100
Number of connections	68

TABLE II
ONE-SEGMENT ROUTING WITH GAMMA DISTRIBUTION

Architecture				Results	
u	α	c	NT	SR	d_T
3	0.33	0.08	39	73%	27
3	0.33	0.044	40	85%	31
3	0.33	0.044	44	96%	34

TABLE III
TWO-SEGMENT ROUTING WITH EXPONENTIAL DISTRIBUTION

Architecture				Results	
u	α	λ	NT	SR	d_T
4	0.25	0.033	41	92%	34
4	0.25	0.04	42	93%	33
4	0.25	0.03	41	93%	32
3	0.33	0.04	42	94%	34
3	0.33	0.03	42	96%	34
3	0.33	0.033	42	99%	34

tracks. Threshold Density d_T is defined as the smallest channel density d such that more than 10% of the channels with density d in the distribution fail to be routed. Clearly, a higher d_T is more desirable.

Based on the maximum channel density restriction, we scaled and rounded up our estimations of τ^k. The maximum number of tracks put in a channel was never more than 44 (22% above the channel density). The length of the longest segment was taken to be the channel length and no offset was considered for the last track. The percentage of successful routing was highly dependent on the type of the distributions and their parameters. The best result obtained for one-segment routing was obtained with gamma distribution as seen in Table II.

Tables III and IV contain the two-segment routing statistics for a number of architectures generated with exponential and gamma distributions, respectively. The success rate of 100% was obtained with the gamma distribution.

The empirical results can be further improved by using a non-geometric progression for the u_k's and variable α_k's. A better modeling of the segmented channel router used for routing the example problems would have improved our results. However, we decided to keep the architectural model simple and make the routing procedures (Algorithms 2.1 and 2.2) as general as possible. Despite the above-mentioned facts, the channel architectures designed based on our equations are nearly as efficient as the mask-programmable channel architectures.

TABLE IV
TWO-SEGMENT ROUTING WITH GAMMA DISTRIBUTION

Architecture				Results	
u	α	c	NT	SR	d_T
4	0.25	0.044	39	90%	32
4	0.25	0.041	39	91%	33
4	0.25	0.051	41	95%	33
3	0.33	0.051	42	96%	36
3	0.33	0.043	43	98%	38
3	0.33	0.044	43	100%	-

Now we have a tool at our disposal to evaluate the effectiveness of candidate segmented-channel architectures. We show that when an architecture is already given, there is a measure that can be used to predict its success rate, i.e. the percentage of connections that can be successfully routed. The formula presented hereafter compares the results of estimations for a required number of tracks with the actual architecture.

Let a_k denote the number of tracks of type k in a given segmented-channel architecture and τ^k denote the expected number of tracks of type k calculated in Section II for one- or two-segment routing schemes, whatever the case may be).[3] In general, f and h are different for control-dominated versus data-path dominated designs, thus τ^k's calculated for these classes of designs will be different.

Consider choosing between two-channel segmentation architectures for a new class of circuits with its specific f and h. The following analysis will determine which of the two architectures is more suitable. We exemplify the problem with arbitrary data, though it can be extracted from real architectures.

We define a function called *deficit*(k) as follows:

$$\begin{aligned} deficit(0) &= 0;\\ deficit(k) &= (\tau^k - a_k + deficit(k-1))\\ &\quad \times S(\tau^k - a_k + deficit(k-1)); \end{aligned}$$

where $S(x)$ is the unit step function.

Since K denotes the largest available track type, *deficit*(K) gives the number of tracks that cannot be routed. We conclude that the percentage of unsuccessful routing is:

$$\frac{deficit(K)}{\sum_{k=1}^{K} \tau^k}$$

Architecture 1:

a_k	τ_k	$deficit(k)$
7	10	$10-7+0=3$
8	10	$10-8+3=5$
18	10	0
9	10	$10-9+0=1$
8	10	$10-8+1=3$

[3] Recall that $\tau^k = \sum_{i=0}^{1/\alpha_k - 1} \tau^{k,i}$.

this is, $3/50 = 6\%$ of the connections cannot be successfully routed.

Architecture 2:

a_k	τ_k	$deficit(k)$
12	10	0
9	10	$10 - 9 + 0 = 1$
8	10	$10 - 8 + 1 = 3$
11	10	$10 - 11 + 3 = 2$
10	10	$10 + 2 - 10 = 2$

that is, $2/50 = 4\%$ of the connections cannot be successfully routed. We therefore conclude that the second segmented architecture is superior for the given design data.

V. Concluding Remarks

We addressed the channel segmentation design and the routability analysis for row-based FPGA devices. Our formulation is quite general and can handle arbitrary density functions for the origination point and length of connections. At the same time, it captures the essential features of the segmented routing scheme that is to be used. We relied on the staggered non-uniform segmentation model for our analysis and derived expressions for the desired number of tracks of each type. Although we derived expressions for one- and two-segment routing only, generalization to M-segment routing is straightforward.

The main benchmark of our calculations is that given the corresponding data histograms, for any application we can predict the number of necessary tracks. Our calculations facilitate the evaluation of the different architectures and find the one that has the highest probability of routing completion.

References

[1] Stephen D. Brown, Robert J. Francis, Jonathan Rose, and Zvonko G. Vranesic, *Field-Programmeable Gate Arrays*. Boston, MA: Kulwer Academic Publishers, 1992.

[2] S. Burman, C. Kamalanathan, and N. Sherwani, "New channel segmentation and associated routing algorithm for high performance fpgas," in *Proc. IEEE Int. Conf. Computer Aided-Design*, 1992, pp. 22–25.

[3] X. Chen and M. L. Bushnell, "A module area estimator for VLSI layout," in *Proc. 25th Design Automation Conf.*, June 1988, pp. 54–59.

[4] A. El Gamal, J. Greene, J. Reyneri, E. Rogoyski, K. El-Ayat, and A. Mohsen, "An architecture for electrically configurable gate arrya," *IEEE J. Solid State Circuits*, vol. 24, no. 2, pp. 394–398, Apr. 1989.

[5] A. El Gamal, J. Greene, and V. Roychowdhury, "Segmented channel routing is nearly as efficient as channel routing," *Advanced Research in VLSI*, pp. 193–211, 1991.

[6] J. Greene, V. Roychowdhuri, S. Kaptanoglu, and A. El Gamal, "Segmented channel routing," in *Proc. 27th Design Automation Conf.*, 1990, pp. 567–572.

[7] F. J. Kurdahi and A. C. Parker, "Techniques for area estimation of VLSI layouts," *IEEE Trans. Computer-Aided Design*, vol. CAD-8, no. 1, pp. 81–92, Jan. 1989.

[8] M. Pedram and B. T. Preas, "Interconnection length estimation for optimized standard cell layouts," in *Proc. Int. Conf. Computer Design*, Oct. 1989, pp. 390–393.

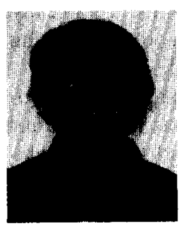
[9] B. T. Preas and M. Lorenzetti, *Physical Design Automation of VLSI Systems*. Menlo Park, CA: The Benjamin/Cummings Publishing Company, Inc., 1988.

[10] S. Sastry and A. C. Parker, "Stochastic models for wirability analysis of gate arrays," *IEEE Trans. Computer-Aided Design*, vol. CAD-5, no. 1, pp. 52–65, Jan. 1986.

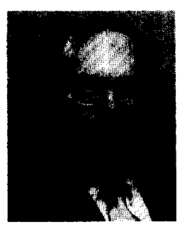
[11] C. Sechen, "Average interconnection length estimation for random and optimized placements," in *Proc. IEEE Int. Conf.*, Nov. 1987, pp. 190–193.

[12] K. Zhu and D. F. Wong, "On channel segmentation design for row-based FPGAs," in *Proc. IEEE Int. Conf. Computer Aided-Design*, 1992, pp. 26–29.

Massoud Pedram (S'88–M'90) is an assistant professor of Electrical Engineering - Systems at the University of Southern California. He received his B.S. degree in electrical engineering from the California Institute of Technology in 1986 and his M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley in 1989 and 1991, respectively. Dr. Pedram has received the best paper award in the CAD track at the IEEE International Conf. on Computer Design (1989), the National Science Foundation's Research Initiation (1992) and Young Investigator (1994) Awards. He has served on the program committees of several conferences and workshops (including the Design Automation Conference) and is the Co-founder and General Chair of the International Workshop on Low Power Design. His research interests and contributions have been in logic synthesis, physical design and CAD for low power.



Bahman S. Nobandegani received the B.Sc. degree in electrical engineering in 1989 and M.Sc. degree in communication engineering in 1991 from the Technical University of Budapest, Hungary. He is currently working toward the Ph.D. degree in Computer Engineering at the University of Southern California, Los Angeles. His research interests are physical design algorithms and logic synthesis.



Bryan T. Preas (M'82) obtained the B.S. degree from Texas A&M University in 1968, the M.S. degree from Carnegie Mellon University in 1969, and the Ph.D. degree from Stanford University in 1979. He was with Bell Telephone Laboratories, Whippany, NJ, from 1968 to 1973, where he worked on hardware and software design for large, multiprocessor computers. From 1973 to 1981 he worked at Sandia National Laboratories, Albuquerque, NM, with the CAD development group. He was Vice resident of Research and Development at VR Information Systems in Austin, TX, from 1981 to 1983 where he was responsible for development of CAD software. In 1983, he joined the Xerox Palo Alto Research Center as Area Manger of Design and Architecture and Principal Scientist. Dr. Preas was awarded the Humboldt Senior Scientist prize and spent 1990 at the University of Paderborn, Germany. Dr. Preas has published numerous technical papers and is active in the ACM and IEEE. He serves on the program committees of several conferences and is an associate editor of the *IEEE Transactions on Computer-Aided Design*. Recently, He has been General Chair of the Design Automation Conference.