

# Buffering Global Interconnects in Structured ASIC Design

Tianpei Zhang and Sachin S. Sapatnekar

*Department of Electrical and Computer Engineering, University of Minnesota,  
Minneapolis, Minnesota 55455, USA  
zhangt, sachin@ece.umn.edu*

---

## Abstract

Structured ASICs present an attractive alternative to reducing design costs and turnaround times in nanometer designs. As with conventional ASICs, such designs require global wires to be buffered. However, via-programmable designs must prefabricate and preplace buffers in the layout. This paper proposes a novel and accurate statistical estimation technique for distributing prefabricated buffers through a layout. It employs Rent's rule to estimate the buffer distribution required for the layout, so that an appropriate structured ASIC may be selected for the design. Experimental results show that the buffer distribution estimation is accurate and economic, and that a uniform buffer distribution can maintain a high degree of regularity in design and shows a good timing performance, comparable with nonuniform buffer distribution.

*Key words:* Structured ASIC, Rent's rule, buffer insertion, interconnect, physical design

---

## 1 Introduction

In the nanometer regime, cell-based ASIC designs are increasingly hard-pressed to produce affordable design solutions, due to the challenges associated with skyrocketing mask costs and manufacturability issues for complex designs [1]. As an alternative, field programmable gate arrays (FPGAs) may be used since they provide a regular prefabricated structure that can avoid many of the manufacturing problems associated with ASICs. However, for many designs, the performance gaps, in terms of speed, power, and area, between FPGAs and cell-based ASIC designs are too large for an FPGA to be a realistic alternative. In this context, structured ASIC design has emerged as a promising new design style to fill the gap.

Structured ASICs are composed of regular arrays of prefabricated standard building blocks, with fixed mask structures. Design with structured ASICs involves many

fewer masks than for cell-based ASICs. One paradigm that is used involves *via-configurability*, where the building blocks and interconnect skeletons are prefabricated and are then connected with appropriate via connections by programming only a small number of masks [2–6]. A standard building block is composed of combinational logic and memory elements (such as flip-flops) and has enough flexibility that it can be programmed to various functions through via configurations. The wires that electrically connect the building blocks are also prefabricated regular fabrics, and the routing of nets can be configured with a via definition. This strategy provides a low NRE (non-recurring engineering) cost, and yet with relatively high-performance solutions. Additionally, since structured ASICs use well-characterized logic blocks and regular, fixed interconnect structures, they are well suited to combat problems associated with manufacturability, yield, noise, or crosstalk.

However, many of the other problems of nanometer design continue to plague structured ASICs. Most importantly, as in cell-based ASICs, the dominant role of interconnect in determining system performance remains a major hurdle, and a key issue in overcoming this is through the use of buffer insertion along global wires. Buffer insertion can not only improve timing performance, but can also effectively reduce functional noise by recovering noise margins [7]. The number of buffers necessary to achieve timing closure and meet noise requirements continues to rise with decreasing feature sizes, and it is projected that at the 32nm technology node, a very large proportion of cells will be buffers [8]. Although interconnects are generally buffered in the later phases of physical design, buffer resources must be planned earlier in the design process, so that enough resources are available for the later insertion phase. Several buffer resource planning strategies for cell-based ASIC design have been proposed in [9, 10].

This problem is more acute for via-programmable structured ASICs, where, in addition to the basic standard building blocks, buffers must be prefabricated and distributed in the layout. Although it is possible to reconfigure the basic building blocks to work as buffers, this is not an economical approach since (a) the number of buffers can be very large, (b) the sizes of the buffers are typically larger than the sizes of regular gates, (c) configuring a large and general standard building blocks as buffers is an inefficient use of resources, and (d) these blocks do not have the driving ability of dedicated buffers. Therefore, it is essential to distribute dedicated buffers in structured ASICs, and to plan for them well, prior to the fabrication of the chip.

The buffer insertion problem for structured ASIC design has not been fully addressed in publications so far. The only work we know of that considers this issue is [11], where it is assumed that a uniform distribution of dedicated buffers is placed throughout the layout, and there is a ratio of 2:1 between number of logic cells and buffers everywhere in the circuit. However, this does not recognize that the demand for buffers depends on the interconnect complexity of circuits, and assuming a single 2:1 ratio for all kind of circuits may result in a large waste of buffer resources.

Clearly, the choice of this ratio should depend on the topology and structure of the circuit that is being mapped to the chip and the number of interconnects. On the other hand, given that this ratio must be predetermined in a structured ASIC, it is clearly not possible or realistic to tune each chip individually to a design, and a “good” set of buffer-to-logic-cell ratios must be chosen.

This paper uses Rent’s rule to develop a family of “good” ratios, and proposes a distributed buffer insertion methodology for the use of dedicated buffers in structured ASIC design. For each range of  $(p, k)$  values, where  $p$  is the Rent’s exponent and  $k$  the Rent’s coefficient, our algorithm (described in section 3.3) finds a statistical estimate of the buffer distribution for circuits falling in that range. Thus for each range, we can prefabricate an off-the-shelf structured ASIC chip with buffers preplaced according to the estimated buffer distribution of that  $(p, k)$  range.

In the implementation phase, a designer may choose the appropriate prefabricated chip for implementing custom design according to values of  $(p, k)$  for the design. Experimental results show that the buffer resource estimation is accurate and adequate for interconnect buffering purpose of circuits in each range, and with an average uniform buffer distribution based on the estimation, we can maintain a good timing performance as well as a highly regular structured ASIC.

The organization of the paper is as follows. In section 2, we describe the buffer insertion methodology. section 3 provides a statistical buffer distribution estimation based on Rent’s rule, and a classification method of circuits based on their Rent’s exponent and coefficient. In section 4, we present the experimental results, which is followed by conclusion in section 5.

## 2 Buffer Insertion Scheme for Structured ASIC Design

Various buffer insertion models have been proposed for cell-based ASIC designs, prominent among which are the buffer block approach [9], in which blocks of buffers are placed *between* the building blocks of the circuit, and the distributed buffer approach [10], in which buffers are interspersed *within* the building blocks, with their exact location being undetermined until later in the design process. The distributed buffer insertion model has several advantages over the buffer block model: it spreads the routing wires around and avoids excessive routing congestion, while satisfying the requirement of buffer insertion.

For structured ASIC design, in the same spirit of interspersing buffers with logic units across the circuit, we adopt a buffer insertion model in which the prefabricated buffers are scattered through the structured ASIC, and the distribution of buffers should be adequate enough for buffering global wires. We also refer to this as a distributed buffer insertion model to capture the distributed nature of this scheme,

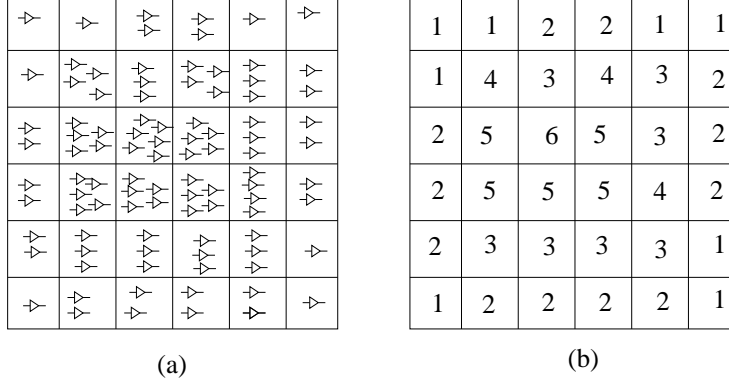


Fig. 1. (a) A schematic showing a chip that is tessellated into tiles, with buffers dispersed within the tiles. For simplicity, the VCCs are not shown here. (b) The corresponding tessellation with the buffer capacity listed for each tile.

although unlike [10], the buffers are actually prefabricated in structured ASICs. We define the structured ASIC to be a two dimensional array composed of via-configurable standard building blocks, denoted as *via-configurable cells* (VCC), which are connected by via-configurable interconnect wires. To facilitate the distributed buffering model, we divide the circuit into an array of *tiles*, and each tile is a square area containing  $m \times m$  VCCs as well as a predetermined number of dedicated buffers for interconnect buffering usage. For a tile positioned at  $(i, j)$ , we refer to this number as the *buffer capacity*, denoted as  $B_{i,j}$ . If  $B_{i,j}$  is uniform for all  $(i, j)$ , we refer to this as a *uniform* buffer distribution; otherwise the buffer distribution is said to be *nonuniform*. A routing solution may use some or all of these buffers: the actual number of utilized buffers in tile  $(i, j)$  is referred to as the *buffer usage*, denoted as  $b_{i,j}$ . If  $b_{i,j} > B_{i,j}$ , the routing solution is invalid, and we refer to this situation as *buffer overflow*. Figure 1(a) shows a  $6 \times 6$  tile graph for a structured ASIC design, with buffers distributed within tiles. The corresponding buffer capacity of each tile in the circuit with a nonuniform buffer distribution is shown in Figure 1(b).

In this paradigm, buffers are prefabricated into the layout but are connected to the global lines that require buffering only in the later phases of physical design. To enable this, we utilize a via-defined buffer insertion (VDBI) scheme, which allows a buffer in a tile to be inserted along any interconnect wire traversing the tile, by means of a via configuration. Figure 2(a) shows a representative buffer in a tile: its input and output are connected to horizontal and vertical wires that go across the tile. Figure 2(b) shows that any wire that crosses the tile can choose to be via-configurably connected to either this buffer through “insertion vias,” or through “jumping vias” to a metal strip that can skip the buffer entirely.

Our model uses a simple yet effective distance-based criterion for buffer insertion. As noted in [9], the delay of a wire is relatively insensitive to the precise location of a buffer, and a buffer can be inserted within a feasible region instead of at a specific location without greatly affecting the timing performance. This is supported by

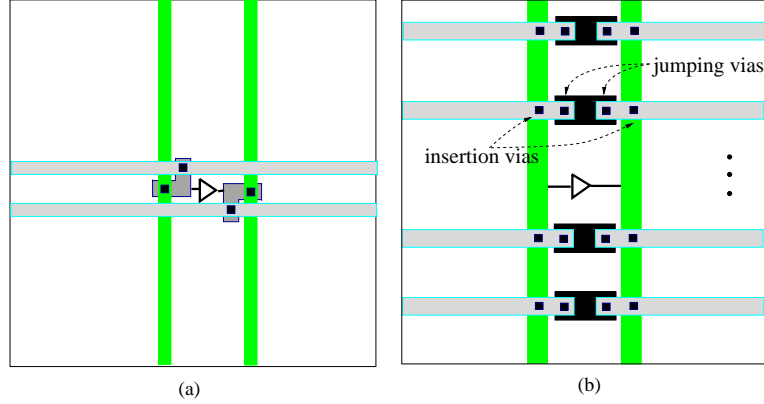


Fig. 2. (a) The input and output of a buffer are each connected to two wires that stretch across the tile, one in the horizontal direction and one in the vertical direction. (b) Buffer insertion on a wire crossing the tile is carried out by means of via definition. An “insertion via” connects a buffer, while a “jumping via” is an electrical short circuit.

[12, 13], where the authors observed that there is a maximum distance between two consecutive buffers so that the timing performance and sharp slew rate can be ensured. Our distance-based model is similar to that used in [10, 12, 13]: the maximum length of interconnect can be driven by a gate (buffer) is  $L$ , and this is referred to as the *critical length*. For a two-pin net, this implies that the separation distance between two buffers is at most  $L$ , and for a multi-pin net, this distance may be shorter between some pair(s) of buffers. This simple buffer insertion constraint is effective and flexible enough and can also work with various routing algorithms.

### 3 Statistical Buffer Distribution Estimation

Structured ASICs provide a reliable and economic platform to implement various designs, but the trade-off is the reduction in flexibility. The basic VCCs, buffers and interconnects must be prefabricated, and this necessitates early and accurate estimation of physical design properties without knowing the details of circuit. To ensure that adequate numbers of buffers are available for routing the circuit, we must determine the buffer distribution, i.e., the buffer capacity of each tile, prior to fabrication. This estimate of the buffer distribution should have the following properties:

- (1) It is desirable for this estimate to be accurate, and should guarantee that enough buffers will be allocated so that it can meet the timing-optimal buffer usage in the final layout. On the other hand, the estimate can not be too pessimistic: if too many buffers are prefabricated, it will result in wasted silicon area.
- (2) The estimation should be based on basic circuit properties instead of any specific circuit implementation details, so that a set of prefabricated chips can be developed on the basis of these properties. Since these chips will be used to

implement a variety of designs, and may work with various physical design tools, the solution should not be specific to any particular tools, and should only be based on basic circuit properties.

With the above considerations, our buffer distribution estimation is based on Rent’s rule. We use this to determine a statistical estimate of the interconnect wire distribution, and the buffer distribution. In the remainder of this section, we will describe the background and details of our estimation technique.

### 3.1 Rent’s Rule

Rent’s rule is an empirical relationship that correlates the number of signal input and output (I/O) terminals  $T$ , to the number of gates,  $N$ , in a random logic network [14]. It is given by a simple power law expression,

$$T = kN^p \tag{1}$$

where  $k$  and  $p$  are called the *Rent’s coefficient* and the *Rent’s exponent*, respectively. These parameters reflect the complexity of a circuit, and can be derived in the process of partitioning the circuit netlist. Our work assumes that these parameters have been computed for the circuit to be mapped to the structured ASIC.

When  $N$  exceeds some critical size  $N_{crit}$ , the relationship between  $T$  and  $N$  deviates from the exponential curve and enters region II of Rent’s rule [14];  $T$  will keep constant or drop while  $N$  increases.

### 3.2 Estimating the Statistics of Buffer Distribution

The number of buffers required in a tile is highly correlated to the total length of external interconnect wires crossing this tile. If a larger number of wires traverse a given tile, it is likely that a larger number of buffers will have to be inserted in the tile. From the Rent’s exponent  $p$  and Rent’s coefficient  $k$  for a circuit, we can apply Rent’s rule to statistically estimate the length of interconnect wires crossing a specific tile  $D$ , and further, to estimate the number of buffers required in the tile.

A schematic of a circuit layout is shown in Figure 3(a). The layout is a square consisting of  $n \times n$  tiles<sup>1</sup>. Each tile is a square consisting of  $m \times m$  VCCs, and the geometrical size of a tile is  $t \times t$  units. While considering the estimation for a tile  $D$ , we divide the circuit, for convenience, into 9 blocks, labeled  $A$  through  $I$ , as shown in the figure, with block  $D$  in the center. Block  $A$  consists of all tiles northwest of

---

<sup>1</sup> For a general circuit of rectangular form, the analysis is very similar to the square case.

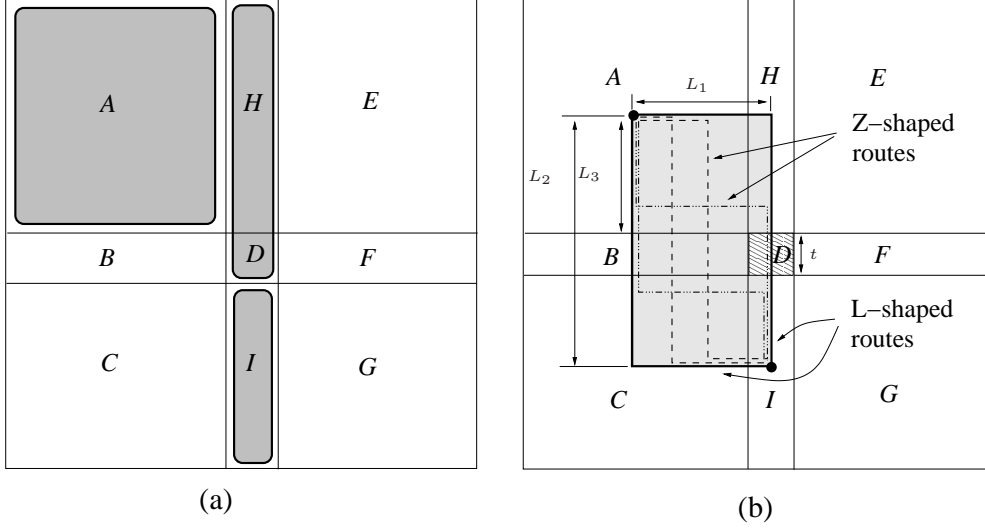


Fig. 3. (a) We divide the circuit into 9 blocks,  $A$  through  $I$ , and for estimation purpose, we merge two blocks  $H$  and  $D$  into a larger block  $HD$ . (b) Estimation of the length of interconnect wires passing tile  $D$ .

$D$ ; block  $B$  is composed of all tiles to the east of  $E$ ; block  $C$  comprises the set of tiles to the southwest of  $D$ , and so on. We will use  $(i, j)$  to refer to the coordinates of tile  $D$  in the  $n \times n$  tiling.

For estimation purposes, it is reasonable to assume that the routing will remain within the bounding box set by the pins of a net. Under this assumption, the interconnect wires that cross block  $D$  consist of the contributions of the wires connecting block pairs in set  $S$  defined as

$$S = \{(A, F), (A, G), (A, I), (B, E), (B, F), (B, G), (B, H), (B, I), (C, E), (C, F), (B, H), (B, I), (C, E), (C, F), (C, H), (E, I), (F, H), (F, I), (G, H), (H, I)\}$$

The total wire length passing through tile  $D$ ,  $W_D$ , is given by

$$W_D = \sum_{\text{all block pairs } (x,y) \in S} W_{x,y}, \quad (2)$$

where  $W_{x,y}$  is the total wirelength of all interconnects crossing tile  $D$ , connecting VCCs in blocks  $x$  and  $y$ , where  $(x, y) \in S$ . Since the tile dimension<sup>2</sup> is generally much smaller than the critical length  $L$ , the interconnects originating from tile  $D$  will not be likely to consume buffer resources at  $D$ , and we only consider the contribution from those wires *passing*  $D$ . Knowing  $W_D$ , if the maximum interconnect length driven by any buffer is length  $L$  units from the insertion model described

<sup>2</sup> It is easy to extend this analysis to the case where  $t$  is larger than  $L$ .



in section 2, then a *unit length* interconnect segment in a wire crossing tile  $D$  will have a probability of  $1/L$  of requiring a buffer to be inserted in tile  $D$ . This implies that an external wire passing a tile of dimension  $t$  horizontally will probabilistically insert  $t/L$  buffers from this tile. We can use this idea to estimate the buffer capacity,  $B_D$ , required for tile  $D$  as the probabilistic usage of buffers in the tile. In other words,

$$B_D = \frac{W_D}{L} \quad (3)$$

The  $W_{x,y}$  components of  $W_D$  can be estimated by using Rent's rule. As an example, we now illustrate how the value of  $W_{A,I}$  may be estimated; other  $W_{x,y}$  components may be estimated in a similar way.

As in [15], we merge two neighboring blocks  $H$  and  $D$  into a larger block  $HD$  as shown in Figure 3(a), and apply the I/O terminal conservation rules to the three blocks  $A$ ,  $HD$  and  $I$ , which are shown as shaded regions in Figure 3(a). **Similar to [15]**, we have the number of I/Os connecting blocks  $A$  and  $I$ , denoted as  $T_{A\text{to}I}$ , to be

$$T_{A\text{to}I} = T_{AHD} + T_{HDI} - T_{HD} - T_{AHD I} \quad (4)$$

in which the  $T_{block}$ ,  $block \in \{AHD, HDI, HD, AHD I\}$  is the number of I/Os of the combinational blocks, and they can be estimated using Rent's rule as

$$\begin{aligned} T_{AHD} &= k(N_A + N_H + N_D)^p \\ T_{HDI} &= k(N_H + N_D + N_I)^p \\ T_{HD} &= k(N_H + N_D)^p \\ T_{AHD I} &= k(N_A + N_H + N_D + N_I)^p \end{aligned} \quad (5)$$

The parameters  $N_A$ ,  $N_H$ ,  $N_D$  and  $N_I$  are the number of VCCs in blocks  $A$ ,  $H$ ,  $D$  and  $I$ , and they are simple expressions in the variables  $i$ ,  $j$ ,  $n$  and  $m$ . However, in applying this formula, we may find that the estimate moves into region II of Rent's rule as described in section 3.1. We use a simplified approach to handle this deviation: when the number of VCCs in the combinational block from equation (5) exceeds  $N_{crit}$ , we substitute this number with  $N_{crit}$  into equation (5). Experimental curves show that  $N_{crit}$  is between 150 to 200 for various circuits, and we simplify it by taking  $N_{crit} = 175$  for all experimental circuits. Experimental results also show that small variation in the choice of  $N_{crit}$  does not affect estimation results much.

To calculate the number of interconnects between blocks  $A$  and  $I$ , we define a variable  $\alpha$  that is the fraction of terminals that are sinks. Thus we can obtain the number of point-to-point interconnects between block  $A$  and block  $I$ ,  $I_{A\text{to}I}$  as:



$$I_{A.to.I} = \alpha T_{A.to.I} \quad (6)$$

and  $\alpha$  can be expressed as **in terms of** the average fanout of the system [15], as

$$\alpha = \frac{fanout}{fanout + 1} \quad (7)$$

Using equation (6) to obtain the number of interconnects between blocks  $A$  and  $I$ , we can further combine it with a simplified L-Z shaped routing model to estimate the wire length crossing tile  $D$  due to interconnects between  $A$  and  $I$ ,  $W_{A,I}$ . Figure 3(b) shows a set of possible L-shaped and Z-shaped connections between the blocks  $A$  and  $I$ . Probabilistically, we can assume that the average position of the terminals of the interconnects are at the center of blocks  $A$  and  $I$ . Thus the routing of interconnects will follow the bounding-box path, and falls in the dotted box in Figure 3 (b). We denote the distance between the centers of  $A$  and  $H$  by  $L_1$ ; the distance between the centers of  $A$  and  $C$  by  $L_2$ ; and the distance between the center of  $A$  and the northern edge of  $D$  as  $L_3$ . The parameters  $L_1$ ,  $L_2$  and  $L_3$  are pictorially illustrated in Figure 3, and these can be expressed in terms of  $i$ ,  $j$ ,  $n$  and  $t$ .

In practice, it is observed that a router will route the bulk of the nets with simple L-shaped and Z-shaped patterns [16]. Hence, we can assume that the routing of an interconnect will utilize one of these two patterns, and the probability of using an L-shaped and Z-shaped route are  $P_L$  and  $P_Z = 1 - P_L$ , respectively. As in [16], we assume  $P_L = 0.7$  in the estimation. Under this routing model, we can estimate the wirelength crossing tile  $D$  due to L-shaped routes as:

$$W_{L,(A,I)} = \frac{1}{2} \cdot t \cdot I_{A.to.I} \cdot P_L \quad (8)$$

The factor “1/2” in the above equation is due to the fact that there are two possible L-shaped routes, and only the upper-L route will pass the  $D$  tile.

Similarly, there are two kinds of Z-shaped routes, type I: with two horizontal segments, and type II: with two vertical segments. If every Z-shaped route has an equal probability of being taken, the type I Z-shaped routes will have a probability of  $L_1/(L_1 + L_2)$  of being taken, while the type II Z-shaped routes have a probability of  $L_2/(L_1 + L_2)$ . Both types of routes can pass tile  $D$ , and we can probabilistically estimate the wirelength of Z-shaped routes crossing tile  $D$  as:

$$W_{Z,(A,I)} = \left( \frac{L_1}{L_1 + L_2} \cdot \frac{t/2}{L_1} \cdot t + \frac{L_2}{L_1 + L_2} \cdot \frac{(L_3 + t)}{L_2} \cdot t \right) \cdot I_{A.to.I} \cdot (1 - P_L) \quad (9)$$

Finally, we can compute the total interconnect wire length from  $A$  to  $I$  that traverses  $D$  as

$$W_{A,I} = W_{L,(A,I)} + W_{Z,(A,I)} \quad (10)$$

The wirelength contribution from other block pairs in set  $S$  can be computed in a similar way as  $W_{A,I}$ , and their values can be substituted into equation (2) and (3) to yield the estimated buffer capacity  $B_D$  for tile  $D$  at position  $(i, j)$ .

### 3.3 Application to Structured ASICs

#### 3.3.1 Using $(p, k)$ Range to Prefabricate Structured ASIC Templates

Structured ASICs consist of predetermined regular architectures, and the buffer resource planning must be addressed at the prefabrication phase. However, due to the unavailability of specific circuit information at this phase, we must preallocate buffers so that it can satisfy the buffer insertion requirements of a range of circuits. As stated earlier, our approach determines the buffer distribution that is necessary for a circuit, based on its basic characteristics, namely, the Rent's exponent  $p$  and Rent's coefficient  $k$ . The practical way of employing structured ASICs in design is that they are prefabricated with other hard Intellectual Property (IP) blocks, which are embedded processors, I/O controllers, etc. The structured ASIC part can be used to implement users' specific logic, and we can assume that there is only one Rent's exponent  $p$  and one Rent's coefficient  $k$  associated with this logic.

We now describe how the buffer distribution determined in section 3.2 is used to design off-the-shelf structured ASIC parts that can be used for a specific circuit. We can divide the spectrum of Rent's exponent values,  $p$ , and Rent's coefficient values,  $k$ , into a set of ranges:  $R_p = \{[p_1, p_2], [p_2, p_3], [p_3, p_4], \dots\}$ ,  $R_k = \{[k_1, k_2], [k_2, k_3], [k_3, k_4], \dots\}$ . For the circuits of tile array size  $n \times n$  in a specific range pair  $[p_i, p_{i+1}]$  and  $[k_j, k_{j+1}]$ , we can predetermine the maximum number of buffers required in each tile for these circuits with the algorithm shown in Figure 4. Using this estimation technique, a set of structured ASIC chips can be prefabricated. When a given circuit is to be mapped onto this fabric, its Rent's parameters are first computed. Based on their values, the appropriate prefabricated structured ASIC chip is chosen, and the circuit is mapped onto that chip. If the Rent's parameter is exactly at the boundary between two ranges, we choose the structured ASIC representing the lower range to avoid waste of buffer resource.

To find a buffer distribution fitting the requirements of all circuits in the range  $[k_j, k_{j+1}]$  and  $[p_i, p_{i+1}]$ , we must find the maximum estimated buffers in each tile. The estimation procedure in Figure 4 is based on the following theorem and observation:

**Theorem 1.** *The estimated number of buffers in a tile  $D$  is a monotonically increasing function of the Rent's coefficient,  $k$ .*

*Proof.* According to the statistical estimation in Section 3.2, the estimated number of buffers  $B_D$  in a tile  $D$  has a general expression as follows:

$$B_D = \sum_{\text{all block pairs } i \in S_D} \eta_i \cdot k [(N_{\alpha,i} + N_{\beta,i})^p + (N_{\beta,i} + N_{\gamma,i})^p - (N_{\alpha,i} + N_{\beta,i} + N_{\gamma,i})^p - N_{\beta,i}^p] \quad (11)$$

where  $S_D$  is the set of block pairs, interconnects between which can pass tile  $D$ ;  $\eta_i$  is a constant determined for block pair  $i$ ;  $N_{\alpha,i}$ ,  $N_{\beta,i}$  and  $N_{\gamma,i}$  are the number of VCCs in three consecutive neighboring blocks  $\alpha_i$ ,  $\beta_i$  and  $\gamma_i$ . From this expression, we can conclude that the estimated number of buffers in a tile  $D$  is linearly dependent on  $k$ , hence it is a monotonically increasing function of  $k$ .  $\square$

**Observation 1.** *The estimated number of buffers in a tile  $D$  in general does not vary monotonically with Rent's exponent  $p$ .*

This observation can be illustrated as follows. From equation (11) above, The derivative of the estimated number buffers  $B_D$  in tile  $D$ , with respect to Rent's exponent,  $p$ , can be found as:

$$\begin{aligned} dB_D/dp = & \sum_{\text{all block pairs } i \in S_D} \eta_i \cdot k [\log(N_{\alpha,i} + N_{\beta,i}) \cdot (N_{\alpha,i} + N_{\beta,i})^p \\ & + \log(N_{\beta,i} + N_{\gamma,i}) \cdot (N_{\beta,i} + N_{\gamma,i})^p - \log(N_{\alpha,i} + N_{\beta,i} + N_{\gamma,i}) \\ & \cdot (N_{\alpha,i} + N_{\beta,i} + N_{\gamma,i})^p - \log N_{\beta,i} \cdot N_{\beta,i}^p] \end{aligned}$$

With the values of  $N_{\alpha,i}$ ,  $N_{\beta,i}$ ,  $N_{\gamma,i}$ ,  $\eta_i$  and  $p$  varying, this derivative can be of positive or negative value. Therefore, the estimated number of buffers in a tile  $D$  generally does not vary monotonically with Rent's exponent  $p$ .

Based on the above theorem and observation, we design the algorithm in Figure 4 to estimate the maximum number of buffers required in each tile. According to Theorem 1, line 1 sets  $k$  to be at the upper limit of the range, i.e.,  $k_{j+1}$  so as to find the maximum buffers in each tile. However, with Observation 1, the dependence between the estimated number of buffers and  $p$  is not monotonic, and therefore, line 3 through 9 performs an enumeration of  $p$  with a step size of  $\delta$  to find the maximum number of estimated buffers in a tile  $D$  for that range. In our experiments, we find  $\delta = 0.01$  is an appropriate choice of the step size. Finally, this enumeration is embedded in an outer loop (line 2 through 10) to perform such estimation for all tiles.

With this estimated buffer capacity for circuits that lie within a range of Rent's parameter values, we can predetermine a single buffer distribution to satisfy the interconnect buffering requirement of all of these circuits, thus creating only one structured ASIC chip that can meet the requirements of all of these circuits. The finer the granularity of these ranges, the more accurate the buffer distribution will

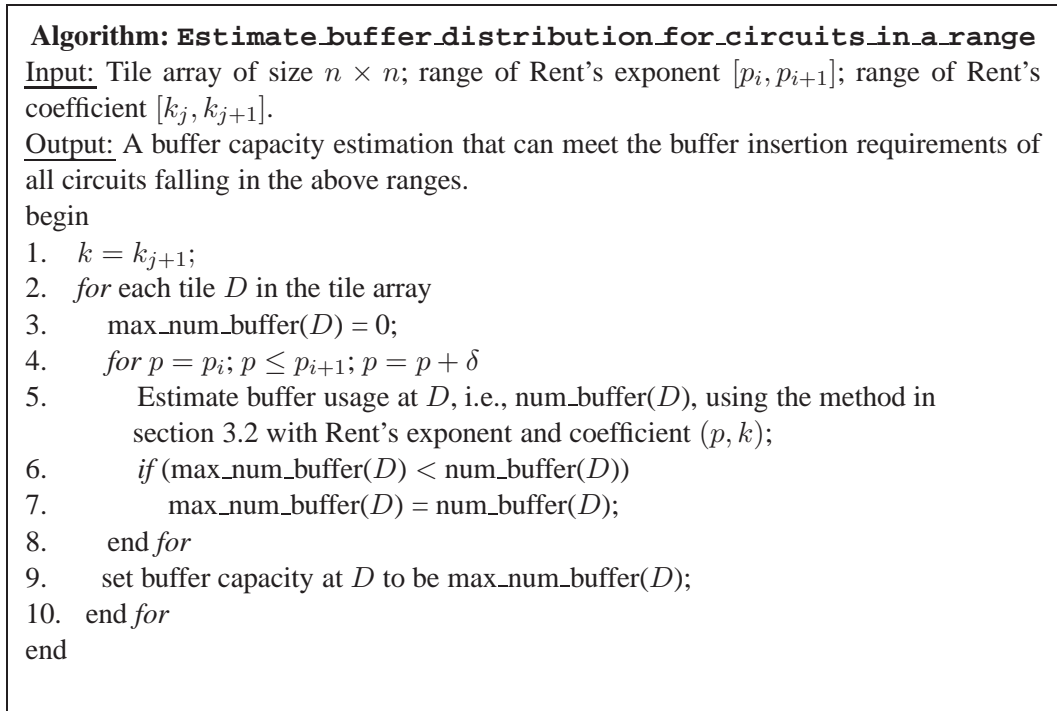


Fig. 4. Estimation of buffer distribution for a range of circuits.

be, but the trade-off is that a larger number of base chips will have to be prefabricated. The size of the structured ASICs is determined at the prefabrication phase and it may be larger than the circuit to be implemented. However, for the same  $(p, k)$  range, it can be shown that the estimated buffer capacity for a larger prefabricated structured ASIC will be greater than the estimated buffer usage for a smaller circuit, if we place the smaller circuit at the center of the structured ASIC. Thus if we estimate and distribute buffers aiming at a larger base structured ASIC, it will always satisfy the buffering requirement of smaller implemented circuits.

### 3.3.2 Uniform Buffer Distribution

The buffer distribution estimation obtained above is not a uniform one, i.e., the number of buffers at different tile locations are different. An important feature of structured ASICs is the regularity in design, which helps much in improving the manufacturability and reducing the design complexity. If a single buffer-to-logic-cell ratio is chosen, a single tile can be designed and optimized, and then repeated throughout the layout. In other words, an alternative to the above *nonuniform* estimated buffer distribution is a *uniform* buffer distribution. We set the number of buffers each tile in the uniform distribution as the *average* buffer number over all the tiles in the nonuniform distribution, which is obtained from the algorithm in Figure 4. This uniform buffer distribution allocates the same amount of total buffers as the nonuniform distribution in the prefabrication phase, and more importantly, it maintains the regularity of the structured ASICs. Since its buffer level is based on the average buffer number from our estimation, it could still satisfy the require-

ment of interconnect buffering under our flexible buffer insertion model, and this is verified in experimental part.

## 4 Experimental Results

Our experiments are performed on 18 of the largest MCNC benchmarks, ranging from 1047 to 8383 logic blocks [17]. These circuits have been technology-mapped to 4-LUTs and flip flops using Flowmap [18], and then the 4-LUTs and flip flops are combined into basic logic blocks with VPACK tools [19]. The benchmarks are then placed and routed with Versatile Place and Route (VPR) tools [19] under a  $0.09\mu\text{m}$  technology. In placement and routing, the VCC block used is the same as the VPGA based 4-LUT CLB designed in [2], and the routing architecture is the switch block architecture also from [2]. The technology parameters of  $0.09\mu\text{m}$  technology used in physical design are listed in Table 1, and they are derived from the scaling of parameters in [2] as well as from [20] and [21].

Parameter	Value
VCC intrinsic delay (ps)	90
VCC area ( $\mu\text{m}^2$ )	52
VCC input capacitance ( $fF$ )	6.0
VCC output resistance ( $\Omega$ )	916
Unit length wire resistance ( $\Omega/\mu\text{m}$ )	0.143
Unit length wire capacitance ( $\text{fF}/\mu\text{m}$ )	0.25
Buffer input capacitance (fF)	13.65
Buffer output resistance ( $\Omega$ )	231
Buffer intrinsic delay (ps)	28

Table 1  
Technology parameters used in placement and routing.

We take each tile to include  $8 \times 8 = 64$  VCCs, and compute the tile array size for each circuit, which is listed in Table 2. For the buffer-related parameters, the critical length  $L$  for buffer insertion is estimated using the similar approach from [13], which results in  $L \approx 500\mu\text{m}$ . We divide the Rent’s exponent and coefficient spectrum into range sets  $R_p = \{[0.3, 0.4], [0.4, 0.5], [0.5, 0.6], [0.6, 0.7]\}$ , and  $R_k = \{[3.0, 4.0], [4.0, 5.0]\}$ , corresponding to a total of  $|R_p| \times |R_n| = 8$  different types of structured ASICs to be fabricated. For each benchmark circuit, we can derive the Rent’s exponent  $p$  and coefficient  $k$  in a recursive partitioning process using hMetis [22] and they are listed in Table 2. Once this is calculated, ~~we then select~~ **we can determine the  $p$  and  $k$  range that each circuit falls into, and select the cor-**

Circuit	$p$	$k$	Tile array	Circuit	$p$	$k$	Tile array
<i>alu4</i>	0.628	4.38	$5 \times 5$	<i>frisc</i>	0.695	4.39	$7 \times 7$
<i>apex2</i>	0.640	4.28	$5 \times 5$	<i>misex3</i>	0.628	4.38	$5 \times 5$
<i>apex4</i>	0.657	4.23	$4 \times 4$	<i>pdv</i>	0.674	3.93	$8 \times 8$
<i>clma</i>	0.578	4.37	$12 \times 12$	<i>s298</i>	0.528	4.28	$6 \times 6$
<i>des</i>	0.389	4.34	$8 \times 8$	<i>s38417</i>	0.437	4.15	$10 \times 10$
<i>diffeq</i>	0.460	4.07	$5 \times 5$	<i>s38584.1</i>	0.413	4.18	$10 \times 10$
<i>elliptic</i>	0.641	3.87	$8 \times 8$	<i>seq</i>	0.616	3.98	$5 \times 5$
<i>ex1010</i>	0.573	4.49	$8 \times 8$	<i>spla</i>	0.676	3.99	$8 \times 8$
<i>ex5p</i>	0.675	4.39	$4 \times 4$	<i>tseng</i>	0.496	3.94	$4 \times 4$

Table 2

Basic information about circuits: Rent’s exponent, Rent’s Coefficient and tile array size.

**responding base structured ASIC for implementation.** Assuming the prefabricated base structured ASIC has similar tile array size as the actual circuit implementation, we can apply the estimation algorithm in Figure 4 to find the *estimated* number of buffers in each tile. We use this as the estimated *nonuniform* buffer capacity for the prefabricated structured ASIC characterized by the corresponding ranges in  $R_p$  and  $R_k$ , and we denote this as nonuniform (*NU*) distribution. Based on this, we can further calculate the average number of buffers per tile as the level of the estimated *uniform* buffer capacity, and we denote this as uniform (*UNI*) buffer distribution. We also experiment on a buffer distribution **as in [11]** that the ratio between logic cells and buffers is 2:1 everywhere in the structured ASIC, and we denote this as a uniform 2:1 (*U21*) distribution. In our experimental setup, there are 32 buffers in each tile for the *U21* distribution as 64 VCCs are contained in each tile. **Furthermore, we experiment on another uniform buffer distribution with a fixed 4:1 ratio between logic cells and buffers across the circuit, which represents a more conservative fixed-number buffering approach than the *U21* model. We denote this model as a uniform 4:1 (*U41*) distribution, and similar to the calculation above, there are 16 buffers in each tile. We then compare the accuracy and the timing performance of the four buffer distribution models.**

To verify the choices that have been made, we apply a buffer insertion algorithm from [10] to actually insert buffers into the above placed and routed circuits, under the *NU*, *UNI*, *U21* and *U41* buffer capacity models. This will produce the *actual* number of buffers used in each tile. Comparing this actual buffer number distribution with that from the estimation models, we can examine the accuracy of our buffer distribution estimation, and the performance of four buffer distribution models. The results of experiments are listed in Table 3.

The first column of Table 3 lists the circuit names. **The second column shows the**

Circuit	$EBC$	Ave. # buffers per tile				# buffer overflow per tile				Ave. buffer usage rate			
		$NU$	$UNI$	$U21$	$U41$	$NU$	$UNI$	$U21$	$U41$	$NU$	$UNI$	$U21$	$U41$
<i>alu4</i>	11	9.48	9.52	9.24	9.12	0	0.2	0	0	85%	86%	29%	57%
<i>apex2</i>	13	9.60	9.44	9.20	9.48	0	0	0	0	74%	73%	29%	59%
<i>apex4</i>	10	8.06	8.19	7.75	8.13	0	0.13	0	0	85%	86%	24%	51%
<i>clma</i>	14	10.62	11.15	10.54	10.48	0	0.6	0	0	76%	79%	33%	65%
<i>des</i>	4	3.02	3.33	2.90	2.91	0	0.72	0	0	79%	88%	9%	18%
<i>diffeq</i>	4	3.64	3.60	3.48	3.56	0.08	0.32	0	0	88%	87%	11%	22%
<i>elliptic</i>	10	8.19	8.53	8.30	7.73	0	0.44	0	0	81%	85%	26%	48%
<i>ex1010</i>	11	7.81	8.06	7.58	7.83	0.02	0.08	0	0	69%	71%	24%	49%
<i>ex5p</i>	8	6.25	6.81	6.69	6.19	0	0.13	0	0	83%	90%	21%	39%
<i>frisc</i>	16	11.92	13.71	12.22	13.29	0.02	0.73	0	0	76%	87%	38%	83%
<i>misex3</i>	7	5.40	4.92	5.28	4.40	0	0	0	0	78%	71%	17%	28%
<i>pdca</i>	14	11.38	11.28	11.26	11.22	0	0.09	0	0	83%	82%	35%	70%
<i>s298</i>	8	5.50	5.58	5.50	5.61	0	0	0	0	69%	70%	17%	35%
<i>s38417</i>	7	5.81	5.54	5.26	5.30	0.01	0.27	0	0	82%	83%	16%	33%
<i>s38584.1</i>	7	4.06	4.62	3.99	4.12	0.01	0.05	0	0	62%	69%	13%	26%
<i>seq</i>	9	8.36	8.08	8.20	8.36	0	0.32	0	0	90%	87%	26%	52%
<i>spla</i>	12	7.69	7.83	7.81	8.27	0	0.03	0	0	64%	62%	24%	52%
<i>tseng</i>	3	2.44	2.19	2.31	1.94	0	0	0	0	81%	73%	7%	12%

Table 3

Comparison of buffer resource usage from buffer insertion under four buffer distribution models: nonuniform ( $NU$ ) distribution, uniform ( $UNI$ ) distribution, uniform 2:1 ( $U21$ ) distribution, and uniform 4:1 ( $U41$ ) distribution.  $EBC$  is the estimated buffer capacity from our estimation algorithm.

average number of buffers per tile which is denoted as estimated buffer capacity ( $EBC$ ). The value of  $EBC$  has been rounded to the nearest integer, and it is exactly the uniform buffer capacity used in the uniform ( $UNI$ ) distribution model. The uniform buffer capacity for the  $U21$  and  $U41$  models are 32 and 16, respectively, for all structured ASICs, and are not explicitly listed in the table. The next four columns list the actual buffer usages from buffer insertion under the four different buffer distribution models. As we can see, the average number of buffers inserted in the cases of  $NU$  and  $UNI$  models are close to those computed from our statistical estimation; in fact, the actual number of buffers required is always a little smaller than the estimated value. This is due to the fact that our buffer estimation method determines the maximum number of buffers required for a set of circuits falling in a range of Rent's exponent and coefficient, and therefore, for a specific circuit with a particular set of Rent's parameters, this estimation can be larger than its actual usage. On the other hand, this pessimism also increase the robustness of the estimated capacity, so that the estimated buffer capacity will satisfy the buffering requirement of circuits even in the presence of fluctuations in the buffer usages of practical circuits. For the average buffer usage under  $U21$  model, the average number of buffers used is much less than the capacity, 32, and this suggests an inefficient use of buffer resources in all of the benchmark circuits. For the  $U41$  model,



although more conservative than the  $U21$  model, it still shows a high level of buffer resource waste as the buffer usage for most of the circuits is still much less than the capacity, 16. The large buffer resource waste from both  $U21$  and  $U41$  models suggests that buffer distribution for structured ASICs should be based on the statistical estimation for a spectrum of circuits, instead of on a simple fixed number.

The columns labeled “# buffer overflow per tile” report the average buffer usage overflow per tile for each of the four buffer distribution models.  $U21$  model, due to its extreme pessimism, results in no overflow at all, but at the price of large waste of resources. Similar observations also applies to the  $U41$  model, although the waste is less due to its more conservative nature compared with  $U21$  model. For the  $NU$  model, it is observed that most of the circuits are free from buffer overflow, and the occurrence of instances that do have overflows is very low, less than 0.08 per tile. The  $UNI$  buffer distribution model results in more overflows than the  $NU$  model, because it is derived from the  $UNI$  model, but the uniformity of the distribution incurs overflow, and it can be considered as a trade-off for the regularity in buffer distribution. However, it is observed these overflows are of low values, and even the worst case has less than 0.73 overflow per tile. These results show that our buffer estimation method can produce an adequate solution, so that the solution can successfully satisfy the buffer insertion requirements of practical circuits. Moreover, even at the trade-off of regularity, our uniform distribution estimation still shows good estimation of buffer levels. In practice, to further reduce the buffer usage overflow under the  $UNI$  buffer distribution model, we can introduce a “fudge factor” to inflate the  $EBC$  value used in the  $UNI$  model. We set the inflated  $EBC$  value  $EBC_{inf} = (1 + \epsilon)EBC$ , thus increase the uniform buffer number to compensate for the trade-off resulted from the regularity. In experiments, we find that by choosing  $\epsilon = 0.2$ , most of the benchmark circuits will be overflow-free, and only three of them still have trivial overflow of less than 0.1 per tile.

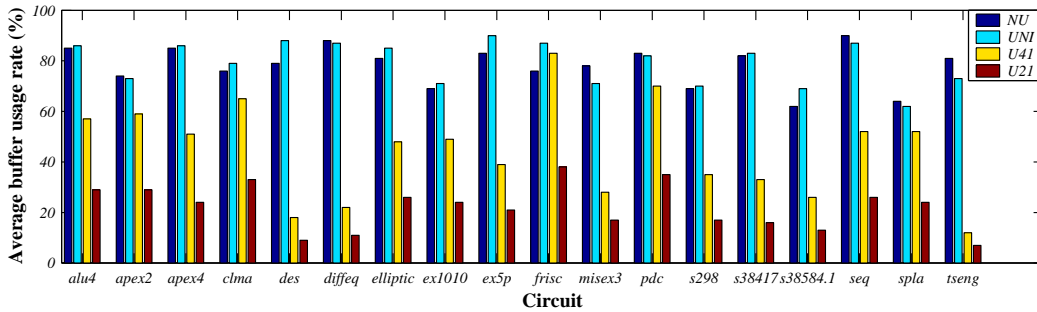


Fig. 5. Comparison of average buffer usage rate among  $NU$ ,  $UNI$ ,  $U41$  and  $U21$  buffer distribution models.

The last four columns in Table 3 list the average buffer usage rate of all circuits under different buffer distribution models. For better visual effects, the corresponding bar chart is shown in Figure 5. It shows that for most of the circuits, the average buffer usage rate is between 70%-90% for both  $NU$  and  $UNI$  models, which means that the pessimism of our estimation algorithm is low, and that it actually

produces reliable and economical *a priori* buffer distribution solutions. However, due to the extreme high buffer capacity, the *U21* buffer distribution shows a very poor usage rate, and thus this solution is not economical. **The *U41* model distributes less buffers in structure ASICs compared with *U21* model; however, with a fixed buffer number for all circuits and not adaptive, the buffer usage rate can be as low as 12% (circuit *tseng*), which can result in a great buffer resource waste.**

Circuit	Critical path delay (ns)					Circuit	Critical path delay (ns)				
	<i>NU</i>	<i>UNI</i>	<i>U21</i>	<i>U41</i>	<i>NB</i>		<i>NU</i>	<i>UNI</i>	<i>U21</i>	<i>U41</i>	<i>NB</i>
<i>alu4</i>	2.02	1.91	2.00	1.93	3.90	<i>frisc</i>	4.19	4.59	4.27	4.11	6.68
<i>apex2</i>	1.80	1.90	1.83	1.90	3.51	<i>misex3</i>	1.45	1.68	1.47	1.70	3.35
<i>apex4</i>	2.04	1.95	2.05	1.99	4.07	<i>pdc</i>	2.46	2.55	2.48	2.44	6.16
<i>clma</i>	3.35	3.33	3.30	3.59	6.27	<i>s298</i>	4.15	4.43	4.16	3.98	7.27
<i>des</i>	1.54	1.61	1.51	1.46	3.22	<i>s38417</i>	3.03	3.71	3.06	2.89	6.42
<i>diffeq</i>	2.12	2.43	2.11	2.12	5.72	<i>s38584.1</i>	2.70	2.99	2.58	2.43	12.59
<i>elliptic</i>	3.55	3.95	3.62	3.75	5.57	<i>seq</i>	1.87	1.97	1.76	1.93	4.74
<i>ex1010</i>	2.31	2.50	2.29	2.40	6.02	<i>spla</i>	2.03	2.30	2.05	2.32	5.35
<i>ex5p</i>	2.02	1.88	1.84	1.75	4.40	<i>tseng</i>	2.08	2.08	2.09	2.09	2.69

Table 4

Comparison of timing performance results from buffer insertion under four buffer distribution models: nonuniform (*NU*) distribution, uniform (*UNI*) distribution, uniform 2:1 (*U21*) distribution, and uniform 4:1 (*U41*) distribution; together with the timing results from the case that no buffer (*NB*) is inserted.

To examine the timing performance of structured ASICs under buffer insertion, we perform timing simulation to obtain the critical path delay after buffers are inserted by the distance-based insertion algorithm from [10]. Table 4 shows the critical path delays for benchmark circuits under four buffer distribution models. For the overflowed buffers, we remove them from buffer insertion solution, and then calculate the critical path delay. For comparison purpose, we also list the critical path delays for the case that no buffers are inserted, and we denote this case as *NB*. Moreover, the critical path delays are also presented as bar chart in Figure 6. From Table 4 and Figure 6, we can find there are not much difference in the timing performance of the four buffer capacity models. Although there are some buffer overflow for the *UNI* model, rip-up of those buffers does not affect the timing performance much. This is because (a) the number of overflowed buffers is small, (b) they may not lie on the critical path. These facts legitimize the use of the *uniform* buffer capacity model based on our estimation in structured ASICs; we can thus acquire an adequate and economic buffer solution with great regularity to improve the manufacturability and reduce cost, while not affecting the timing performance significantly. Furthermore, it can be seen that the critical path delay in the case with no buffers (*NB*) inserted is significantly longer than all four buffer insertion cases. Averagely, the critical path delay is about 120% more without buffering, and there can be as high as 373% more delay for the worst case (circuit *s38584.1*). This performance degradation shows the necessity and importance of buffer insertion for structured ASICs, and therefore our buffer planning method is crucial to improve

the performance of this new design style.

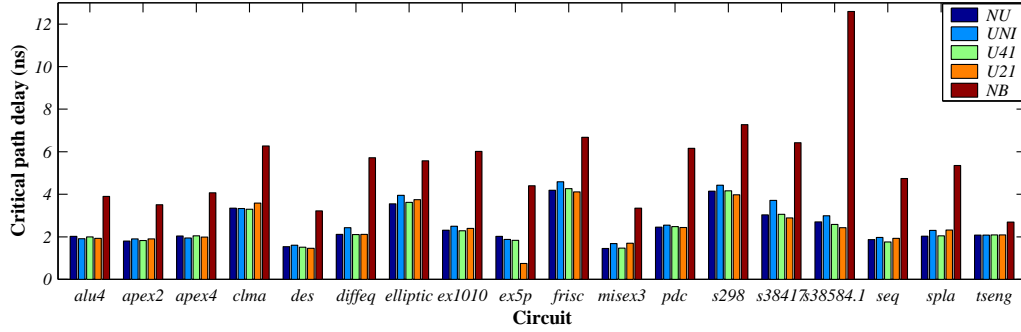


Fig. 6. Comparison of critical path delay among *NU*, *UNI*, *U41*, *U21* and *NB* buffer distribution models.

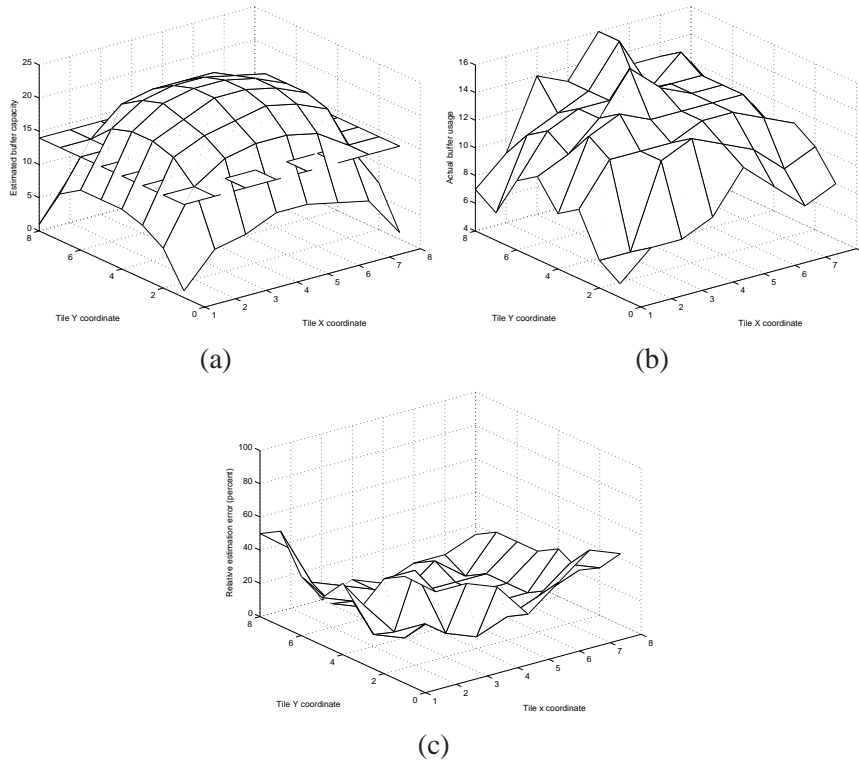


Fig. 7. Accuracy of the statistical buffer distribution estimation for a representative circuit *pdc*. (a) Estimated buffer capacity distributions for the nonuniform and uniform distributions. (b) Actual buffer usage distribution under the uniform buffer capacity. (c) Relative error of the buffer capacity estimation compared with actual buffer usage under the uniform buffer capacity.

Figure 7 further shows the two dimensional buffer distribution for a specific representative circuit, *pdc*. The three graphs show, respectively, the estimated buffer capacity distributions for nonuniform and uniform models, the actual buffer usage distribution for uniform model and the relative errors. We can see from Figures 7(a) that the distribution curve for the nonuniform distribution is of a “bell curve” form, but with a flat region in the center and a sharp dropoff near the boundary. Thus it

does not deviate much from an average uniform distribution as shown in the figure. This property makes it reasonable to use a uniform buffer distribution based on the average of the nonuniform distribution, but still get good buffer usage and small buffer overflow. Figure 7(b) shows that the distribution of the actual buffer usage under the uniform buffer distribution model. For this and all other benchmarks, the general trend is that the usage of buffers fits the uniform buffer capacity well in most part of circuits, but less buffers are used at the periphery. This is due to the fact there is generally less interconnect wires at periphery. In practice, we may pre-allocate some buffer areas on periphery to be decoupling capacitors to save some resources.

The relative error curve in Figure 7(c) shows that the estimated buffer capacity is generally a little higher than the actual buffer number, because our estimation is aimed at the maximum buffer capacity for a range of circuits, and naturally builds in some pessimism. Also this error is seen to a greater degree near the periphery, due to the smaller number of interconnect wires around there. However, for the most part, this error is less than 20%, which shows that our *a priori* buffer distribution estimation provides an economic solution.

## 5 Conclusion

In this paper, we have presented a practical distributed length-based buffer insertion model for structured ASIC design. Based on this model, we have proposed an early statistical buffer distribution estimation using Rent's rule and a simplified L/Z-shaped routing model, and also proposed a uniform buffer distribution model of great regularity. Experimental results show that the buffer distribution estimation and models, although not based on physical design details, can accurately and economically plan buffer resources on structured ASIC chip, and it is shown the buffer capacity prediction matches actual buffer usage well.

## References

- [1] P. Gupta and A. B. Kahng, "Manufacturing-Aware Physical Design", *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 2003, pp. 681-687.
- [2] C. Patel, A. Cozzie, H. Schmit and L. Pileggi, "An Architectural Exploration of Via Patterned Gate Arrays," *Proceedings of the ACM International Symposium on Physical Design*, 2003, pp. 184-189.
- [3] L. Pileggi, H. Schmit, A. J. Strojwas, P. Gopalakrishnan, V. Kheterpal, A. Koorapaty, C. Patel, V. Rovner and K. Y. Tong, "Exploring Regular Fabrics to Optimize the

- Performance Cost Trade-off,” *Proceedings of the ACM/IEEE Design Automation Conference*, 2003, pp. 782-787.
- [4] Y. Ran and M. Marek-Sadowska, “On Designing Via-Configurable Cell Blocks for Regular Fabrics,” *Proceedings of the ACM/IEEE Design Automation Conference*, 2004, pp. 198-203.
  - [5] B. Hu, H. Jiang, Q. Liu and M. Marek-Sadowska, “Synthesis and Placement Flow for Gain-based Programmable Regular Fabrics,” *Proceedings of the ACM International Symposium on Physical Design*, 2003, pp. 197-203.
  - [6] V. Kheterpal and L. Pileggi, “Routing Architecture Exploration for Regular Fabrics,” *Proceedings of the ACM/IEEE Design Automation Conference*, 2004, pp. 204-207.
  - [7] C. Alpert, A. Devgan and S. T. Quay, “Buffer Insertion for Noise and Delay Optimization,” *IEEE Transactions on Computer-Aided Design*, 18(11), pp. 1633-1645, November 1999.
  - [8] P. Saxena, N. Menezes, P. Cocchini, D. A. Kirkpatrick, “The Scaling Challenge: Can Correct-by-Construction Design Help,” *Proceedings of the ACM International Symposium on Physical Design*, 2003, pp. 51-58.
  - [9] J. Cong, T. Kong and D. Z. Pan, “Buffer Block Planning for Interconnect-Driven Floorplanning,” *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 1999, pp. 358-363.
  - [10] C. Alpert, J. Hu, S. S. Sapatnekar and P. Villarrubia, “A Practical Methodology for Early Buffer and Wire Resource Allocation,” *Proceedings of the ACM/IEEE Design Automation Conference*, 2001, pp. 189-194.
  - [11] K. Wu and Y. Tsai, “Structured ASIC, Evolution or Revolution,” *Proceedings of the ACM International Symposium on Physical Design*, 2004, pp. 103-106.
  - [12] F. F. Dragan, A. B. Kahng, I. Mandoiu and S. Muddu, “Provably Good Global Buffering Using an Available Buffer Block Plan,” *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 2000, pp. 358-363.
  - [13] C. W. Sham and E. F. Y. Young, “Routability Driven Floorplanner with Buffer Block Planning,” *IEEE Transactions on Computer-Aided Design*, 22(4), pp. 470-480, April, 2003.
  - [14] B. S. Landman and R. L. Russo, “On a Pin Versus Block Relationship for Partitions of Logic Graphs,” *IEEE Transactions on Computer*, vol. C-20, pp. 1469-1479, December 1971.
  - [15] J. A. Davis, V. K. De and J. D. Meindl, “A Stochastic Wire-Length Distribution for Gigascale Integration (GSI) - Part I: Derivation and Validation,” *IEEE Transactions on Electron Devices*, 45(3), pp. 580-589, March 1998.
  - [16] J. Westra, C. Bartels and P. Groeneveld, “Probabilistic Congestion Prediction,” *Proceedings of the ACM International Symposium on Physical Design*, 2004, pp. 204-209.

- [17] V. Betz, "VPR and T-VPack: Versatile Packing, Placement and Routing for FPGAs," Version 4.30, March 2000. Available at <http://www.eecg.toronto.edu/~vaughn/vpr/download.html>
- [18] J. Cong, Y. Ding, "Flowmap: An Optimal Technology Mapping Algorithm for Delay Optimization in Lookup-Table Based FPGA Designs," *IEEE Transactions on Computer-Aided Design*, 13(1), pp. 1-12, January 1994.
- [19] V. Betz and J. Rose, "VPR: A New Packing, Placement and Routing Tool for FPGA Research," *Proceedings of the ACM International Workshop on Field Programmable Logic and Applications*, 1997, pp. 213-222.
- [20] J. Cong, "Challenges and Opportunities for Design Innovations in Nanometer Technologies," (Available at <http://ballade.cs.ucla.edu/~cong/papers/final1.pdf>), *Proceedings of the International Symposium on VLSI Technology, Systems, and Applications*, pp. 54-57, 1999.
- [21] Semiconductor Industry Association, *International Technology Roadmap for Semiconductors*, 2003.
- [22] G. Karypis, R. Aggarwal, V. Kumar and S. Shekhar, "hMETIS-Hypergraph & Circuit Partitioning," Version 1.5.3, November 1998. Available at <http://www-users.cs.umn.edu/~karypis/metis/hmetis/>