A 3-D FPGA Wire Resource Prediction Model Validated using a 3-D Placement and Routing Tool

Young-Su Kwon, Payam Lajevardi, Anantha P. Chandrakasan, Frank Honoré, and Donald E. Troxel
Microsystems Technology Laboratory
Massachusetts Institute of Technology
{yskwon,payam,anantha,honore,troxel}@mtl.mit.edu

ABSTRACT

The interconnection architecture of FPGAs such as switches dominates performance of FPGAs. Three-dimensional integration of FPGAs overcomes interconnect limitations by allowing instances to be located and signals to be routed in 3-D space. Wire resource prediction is important for fast and accurate interconnection planning in 3-D FPGA. In this paper, we extend the existing analytic model shown in [13] with a new parameter for our 3-D FPGA which has cluster-based logic blocks. The proposed wire resource prediction model is applied to our 3-D FPGA using a Xilinx Virtex2 slice [18] and our 3-D routing architecture. We validate the effectiveness of the extended model by comparing the required number of channel wires predicted by the extended analytic equation with that of the placed and routed results using 3-D placement and routing algorithm designed for our 3-D FPGA for a number of benchmark circuits. The extended 3-D wire resource prediction model predicts the required channel capacity with an average of 11.1% error for 17 large circuits from LGSynth93 and ISPD2001 Verilog benchmarks.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics—*performance measures*

General Terms

Experimentation, Measurement

Keywords

3-D FPGA, Wire resource prediction

1. INTRODUCTION

As feature size of deep-submicron technologies decreases, interconnects are becoming the critical limiting factor in determining overall delay and power consumption. Three-dimensional (3-D) integration is a promising technology for overcoming interconnect limitation in deep-submicron designs while maintaining logic and memory density [5][6][14]. Three-dimensional integrated circuits are formed by monolithic vertical integration of multiple strata (device layer with multiple metal layers) using wafer bonding. As shown in Figure 1, Cu-Cu bonding between an SOI wafer with thin etched backside and Si wafer forms an integrated circuit with two wafers [10]. This wafer bonding process is repeated to integrate multiple strata with vertical vias interconnecting metal wires in different strata. An analytic wirelength prediction model for 3-D IC based on Rent's Rule proposed in [15] and experimental results for 3-D IC in [6] predicts significant wirelength reduction in four-layer 3-D ICs.

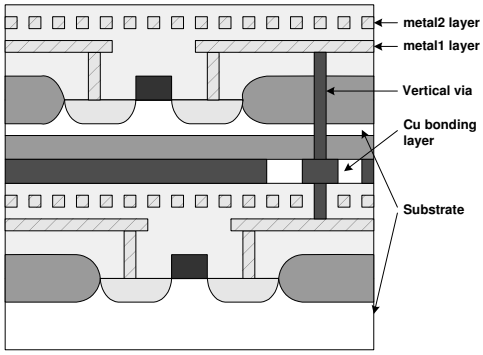

Figure 1: Cross section of three-dimensional integration with Cu-Cu wafer bonding.

FPGAs are currently extending their application area from prototyping to custom applications. The essential limitation in terms of speed and power of FPGAs comes from the well-known fact that interconnect resources in FPGAs account for 40%-80% of the delay, 90% of chip area, and 60-70% of overall power consumption [17]. Therefore, FPGA is an ideal application which can benefit significantly from 3-D integration where reduced wirelength of 3-D integration improves the speed of implemented circuit while reducing power consumption.

The benefit of 3-D integration comes from the 3-D stacking of transistors to overcome the fundamental limits of the next-generation design caused by the interconnect limitation [8]. It is reported that 3-D integration can achieve 50% total wirelength reduction and 50% delay reduction in the

SLIP'05, April 2–3, 2005, San Francisco, California, USA.

critical path. 3-D integration can achieve 80%-90% power reduction with the same operating frequency if we assume that 70% of power in FPGAs goes to the interconnect.

Wire resource prediction of FPGAs plays an important role in determining routability of FPGAs, planning the architecture of FPGAs, determining costs used in FPGA CAD tools, and characterization and comparison of FPGAs. The routability of a design implemented in an FPGA depends not only on the configuration of CLBs (Configurable Logic Block) such as the number of LUTs and I/O pins and routing architecture but also design of placement and routing algorithm. The analytic wire resource prediction model provides the designer with preliminary feedback on tradeoffs for various routing architectures and CLB configurations. The designer gets the earlier insight of FPGA architecture without running time-consuming placement and routing for large circuits iteratively.

There have been several works for wire resource prediction in 2-D FPGAs. The analysis shown in [9] presents that channel capacity W on $N \times N$ array FPGA converges to a Poisson distribution. The channel capacity or the number of routing wires per channel W is given by

$$\overline{W} = \frac{\gamma \overline{L}}{2} \tag{1}$$

where $\overline{L}$ is the average point-to-point wirelength and γ is the average number of nets connected to a logic block. γ is known prior to placement and routing while extraction of $\overline{L}$ depends on the stochastic distribution of wirelengths. In [16], the stochastic model with random variable representing the wirelength on an infinite 2-D array is proposed where $\overline{L}$ depends on the gamma function and Rent's parameter p. This work exploits Rent's Rule for deriving stochastic wirelength distribution. In [4], it is shown that the wire resource is best predicted by estimating the total wirelength in the circuit, not the mean wirelength times pins per cell. The random circuit generation methodology is applied to show that the total wirelength is the best predictor of wire resource. In [12], a 2-D and 3-D wire resource prediction models based on total wirelength predict the required channel width using experimental results with VPR [1].

In this paper, we extend and validate the wire resource prediction model presented in [13] for our 3-D FPGA and compare 3-D placed and routed results with the extended analytic model. To our best knowledge, this work is the first research on the validation of 3-D FPGA wire resource prediction model with experimental results obtained from the 3-D FPGA placer and router although the analytic equation for wirelength distribution function for three-dimensional integrated circuit (not FPGA) is also derived in [11], [15], and [19]. The proposed model is also an extended wire resource prediction model based on the model shown in [13] for our 3-D FPGA with clustered CLBs. We developed a custom 3-D placement and routing for our 3-D FPGA with Xilinx Virtex2-style CLB and our 3-D routing architecture. We obtain the required number of routing wires per channel by applying 3-D placement and routing to successfully place and route several large benchmark circuits.

This paper is organized as follows. In section 2, our 3-D FPGA architecture and CAD flow are described. In section 3, the previous 3-D FPGA wire resource prediction model is explained and our extended 3-D FPGA wire resource prediction model for 3-D FPGA with clustered CLBs is proposed. Experimental results are presented in section 4 with concluding remarks in section 5.

2. 3-D FPGA ARCHITECTURE AND CAD

In this section, we briefly introduce our 3-D FPGA architecture and 3-D placement and routing. The block diagram of our 3-D FPGA is shown in Figure 2. The basic unit of a 3-D FPGA is a tile which is composed of a 3-D switch and a CLB. A 3-D mesh array of tiles constitutes the whole FPGA where wires of each 3-D switch are connected to that of nearest six switches. (For top and bottom strata, each switch is connected to five switches.)

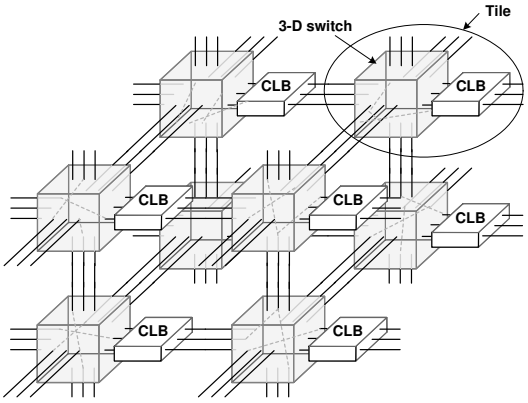

Figure 2: Block diagram of 3-D FPGA with CLBs. Each tile is composed of a 3-D switch and a CLB.

The CLB architecture of our 3-D FPGA is the same as that of Xilinx Virtex2 [18]. Each CLB is composed of four slices where each slice is composed of two 4-input LUTs, two configurable latches, and several multiplexers. The CLB in our 3-D FPGA is a *cluster-based* logic block. We have used a cluster-based logic block to improve circuit area and speed for commonly used circuits (shapes) such as arithmetic circuits, shift registers (up to 128 bits), large multiplexers or sum-of-product circuits. The interconnection wires for shapes do not go through 3-D switches but local interconnection wires. Stochastic wirelength distribution models using Rent's Rule predict the number of nets emanating from a CLB based on the number of gates in a CLB. However, the number of slices in a single CLB in our 3-D FPGA is not deterministic before placement and is usually larger than that of FPGAs with non-clustered CLBs. In our wire resource prediction model, the existing 3-D FPGA wire resource prediction model is extended to estimate the number of terminals for a single CLB based on the experimental data because of the non-deterministic nature in the number of implemented slices in a CLB.

We use a Xilinx FPGA synthesizer and technology mapper to obtain a technology-mapped circuit from HDL designs [18]. The technology-mapped circuit is converted to a XDL (Xilinx Design Language) netlist which describes the technology-mapped slices and the nets interconnecting slices.

3-D placement determines slice locations based on 3-D extension of the simulated annealing. The simulated annealing used in our 3-D placement is composed of two steps: inter-strata force-directed movement and intra-stratum random

movement. In inter-strata optimization phase, simulated annealing moves each slice to the equilibrium position by computing net weights connected to the slice and locations of neighboring slices. As each slice moves to the equilibrium position, slices are located in different strata reducing overall cost. This is the application of well-known force-directed methodology to simulated annealing with combination of the following random movement phase. After several force-directed slice movements, the acceptance ratio which is the portion of cost-saving movement decreases and the second phase starts. In the second phase, the algorithm moves slices only within the same strata where each slice is located for local optimization.

3-D global and detailed routing allocates wire resources to nets in the XDL netlist. In the global routing, the router first determines the priority of each net based on the timing criticality of each net based on the timing slack of the net. The global router routes nets in the order of decreasing priority trying to minimize the wire resource conflict, i.e., it considers routability during global routing. After the global routing, the detailed routing assigns wire resources to each net considering interconnection topology of 3-D switches. We have used an implicit enumeration algorithm which is usually used for a graph coloring algorithm in the detailed routing.

3. EXTENSION OF 3-D WIRE RESOURCE PREDICTION MODEL

Most stochastic models adapt Rent's Rule to predict the wirelength distribution. Rent's Rule is a well-known empirical model observed in subcircuits. It reflects a scaling of the number of external terminals of a given subcircuit with the number of gates in the subcircuit given by

$$T = kN^p \tag{2}$$

where T is the average number of external terminals in a subcircuit or partition, k is Rent's constant and N is the number of gates in a subcircuit. The proposed 3-D FPGA wire resource prediction model is the extended model for our 3-D FPGA based on the prediction model presented in [13].

3.1 3-D FPGA Wire Resource Prediction Model [13]

In [13], the wire resources of 3-D FPGAs are predicted by estimating the required channel capacity, i.e., the required number of wires per channel. The 3-D disjoint switch where each wire segment is connected to wire segments on the other five sides of a cubic box is assumed. The 3-D switch requires more pass transistors and SRAM cells per routing switch than that of 2-D switch. However, the number of routing wires per each channel of 3-D FPGA is expected to be reduced as shown in [13]. The required number of routing wires per channel is estimated by

$$W = \frac{\sum_{l=1}^{2\sqrt{N/N_z}-2+(N_z-1)t_z} l f_{3D}(l)\chi_{fpga}}{2N + \frac{(N_z-1)N}{N_z} \quad e_t} \tag{3}$$

where l is wirelength, N is the number of CLBs, and N_z is the number of strata. $f_{3D}(l)$ is 3-D wirelength distribution which is the function of l. χ_{fpga} is a point-to-point net length conversion factor, t_z is the separation between neighboring strata, and e_t is the utilization factor of wiring tracks. The derivation of Eq. 3 is based on the assumption that the total number of utilized wiring tracks is equal to the total wirelength. χ_{fpga} is the compensation factor for the wirelength of multi-terminal nets. $f_{3D}(l)$ gives the number of nets with the wirelength of l where a point-to-point net (two-terminal nets) is assumed. The wirelength for a multi-terminal net tends to be shorter than the sum of the point-to-point wirelength. χ_{fpga} is to compensate the wirelength for multi-terminal nets. Note that χ_{fpga} is smaller than 1.0. e_t is the compensation factor for the channel usage in the global and detailed routing. In global routing, the placement result causes some wires to take a detour because of limited number of wires in the channel. In detailed routing, the FPGA router normally can not maintain the result of global routing because of the routing resource conflict called "routing anomaly" resulting from the interconnection methodology in the 3-D switch. A simple example

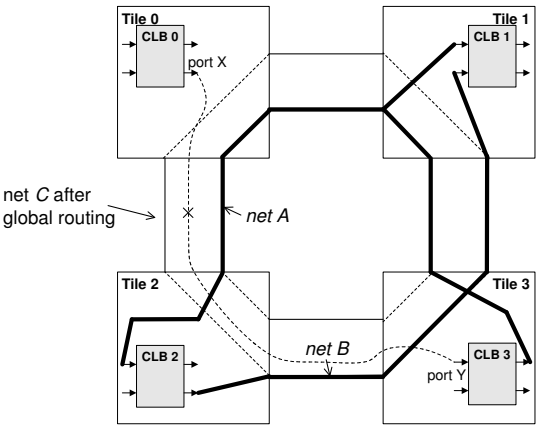

Figure 3: The example showing the necessity of e_t. Net C is not routable although the number of available wiring tracks is larger than the number of required wiring tracks.

in Figure 3 shows the routing anomaly problem. The FPGA shown in Figure 3 has four tiles with four CLBs, two wires per routing channel, and the "disjoint" switch. Each wire is connected to only one of the wires in each side in a disjoint switch shown in Figure 3. After global and detailed routing, the router assigns wire resources to the net A and net B as shown in the figure. The global router assigns the routing path of net C which is connected to port X and port Y as the dotted line but the detailed router fails to route net C because there is not a set of unoccupied wires from CLB 0 to CLB 3. If the switch is a "full crossbar" switch where each wire is connected to all the other wires in the switch, net C is routable but full crossbar switch is not used in FPGAs because of the complexity. In this example, the number of available wire resources is larger than the required number of wires but it needs more wires per channel. The wire utilization factor, e_t is 5/8. In Eq. 3, $(2N + \frac{(N_z-1)N}{N_z})$ is the number of available wire resources and e_t scales the number of available wire resources considering route detour and routing anomaly. Note that W in Eq. 3 is the required number of routing wires per channel, i.e., the maximum number

of routing wires per channel not the average number of routing wires per channel. The experimental results in this work exploits a disjoint 3-D switch.

The 3-D wirelength distribution presented in [15] leverages the 2-D model shown in [7] and is given by

$$f_{3D}(l) = \Gamma M_{3D}(l) I_{3D}(l) \quad (4)$$

where $M_{3D}(l)$ is the number of gate pairs separated by length l and $I_{3D}(l)$ is the number of interconnects of length l gate pitches from a given logic gate. Γ is the normalization factor which is the inverse of the sum of $M_{3D}(l)I_{3D}(l)$ over all valid wirelength l. $M_{3D}(l)$ is computed by counting the number of gate pairs in the 3-D plane given by

$$M_{3D}(l) = \gamma M_{2D}(l) + \sum_{i=0}^{N_z-1} \beta_i M_{2D}(l - it_z) u(l - it_z) \quad (5)$$

where γ and β_i are constants that depend on the number of device layers and $u(l)$ is the unit step function. The 2-D wirelength distribution, M_{2D} is given by

$$M_{2D}(l) = \begin{cases} \frac{l^3}{3} - 2l^2\sqrt{N} + 2Nl, \ 1 \le l < \sqrt{N} \\ \frac{1}{3}(2\sqrt{N} - l)^3, \ \sqrt{N} \le l < 2\sqrt{N} \end{cases} \quad (6)$$

where the maximum length of l is $2\sqrt{N}$ [7]. The number of nets emanating from a CLB is found by applying Rent's Rule and terminal conservation theory. $I_{3D}(l)$ is the number of nets with length l given by

$$\begin{aligned} I_{3D}(l) &= \frac{\#\text{ nets with length } l}{\#\text{ gates on the periphery}} \\ &= \frac{\alpha k}{N_c}[(N_a + N_b)^p - N_b^p + (N_b + N_c)^p \\ &\quad -(N_a + N_b + N_c)^p] \end{aligned} \quad (7)$$

where $\alpha = f_o/(1+f_o)$ (f_o is the average number of fan-outs), $N_a(=1)$ is the number of logic gates under investigation, N_c is the number of gates at manhattan distance l, and N_b is the in-between number of gates [7]. N_a, N_b, and N_c for small N_z are shown in [15].

3.2 Extension of 3-D FPGA Wire Resource Prediction Model

The architecture of the 3-D FPGA in [13] and the CLB architecture of the proposed 3-D FPGA are shown in Figure 4. Figure 4(a) shows a block diagram of a stratum in 3-D FPGA which is composed of switches and CLBs presented in [13]. Each CLB with a single LUT is connected to four switches and each switch is shared by four CLBs. In an infinite 2-D plane, each CLB is approximately connected to a single switch. The number of channel wires, W is the number of wires connected to each side of a switch. The circuit parameters, N, k, and f_o are defined based on the assumption that the basic unit is a CLB which is composed of a single LUT and a single latch. N is the number of CLBs and k is the average number of terminals for the subcircuit implemented in each CLB. f_o is extracted by investigating nets interconnecting CLBs.

The basic unit of our 3-D FPGA shown in Figure 4(b) is a tile which is composed of a switch and four slices. The circuit parameters, N, k and f_o are extracted by analyzing the XDL netlist where the XDL netlist describes the interconnection among slices. In our 3-D FPGA, N is the number of slices, k is the average number of terminals for the

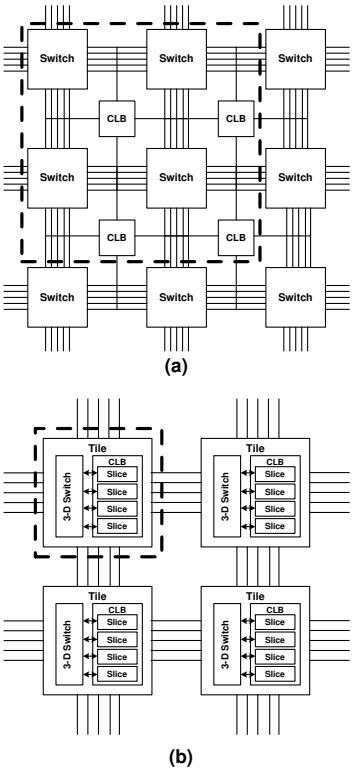

Figure 4: Comparison of CLB and routing architecture of each stratum for (a) 3-D FPGA in [13] and (b) the proposed 3-D FPGA.

subcircuit implemented in each slice, and f_o is the average number of fan-outs for nets interconnecting slices. However, the channel width in Figure 4(b) is the number of wires connected to a 3-D switch which is connected to four slices, not to a single slice. In other words, four switches in Figure 4(a) are equivalent to a single switch in Figure 4(b). The number of "active" slices (a slice where a subcircuit is implemented), moreover, is not pre-defined but is determined after 3-D placement.

The wire resource prediction model shown in Eq. 3 is extended such that the basic unit is a tile, not a slice although parameters N, k, and f_o obtained from the XDL netlist describing interconnections among slices are used without modification. We have modified Eq. 7 by adding a new parameter, n_s which is the average number of "active" slices in a CLB. The modified equation is given by,

$$\begin{aligned} I_{3D}^C(l) &= \frac{\alpha(n_s k)}{N_c} {n_s}^p[(N_a + N_b)^p - N_b^p + (N_b + N_c)^p \\ &\quad -(N_a + N_b + N_c)^p] \end{aligned} \quad (8)$$

where $I_{3D}^C(l)$ is the extended model of $I_{3D}(l)$ for our 3-D FPGA. In Eq. 7, $[(N_a + N_b)^p - N_b^p + (N_b + N_c)^p - (N_a + N_b + N_c)^p]$ is the number of point-to-point interconnections between blocks in A and C. In our 3-D FPGA, a tile is a "gate" and each of N_a, N_b and N_c is multiplied by n_s. In other words, the number of gates, N in Rent's Rule is multiplied by n_s because there are n_s slices contributing to the number of terminals of a CLB in Eq. 8. As shown in Eq. 7, the denominator in I_{3D} is defined as the number of

gates on the periphery of a partial manhattan circle [7]. The number of gates on the periphery in our 3-D FPGA is the same as that of Eq. 7 because N_c is the number of tiles on the periphery. k, the average number of fan-outs of the basic unit, is multiplied by n_s. $I^C_{3D}(l)$ is also represented as

$$I^C_{3D}(l) = {n_s}^{p+1} I_{3D}(l) \tag{9}$$

where n_s^{p+1} is a scaling factor. As shown in 9, the number of interconnections in our 3-D FPGA, increases by a factor of n_s^{p+1} compared to that of 3-D FPGA in [13].

Eq. 5 is the number of gate pairs which are separated by length l. The number of gate pairs is computed by counting the number of logic gates separated by length l from the given logic gate. In our 3-D FPGA, the number of gates is N/n_s which is the number of tiles. The extended model for the number of gate pairs is shown as follows.

$$M^C_{3D}(l, N) = M_{3D}(l, N/n_s) \tag{10}$$

The extended wire resource prediction model is given as

$$W^C = \frac{n_s^{p+1} \sum_{l=1}^{2\sqrt{(N/n_s)/N_z}-2+(N_z-1)t_z} l f_{3D}(l, N/n_s)\chi^C_{fpga}}{2(N/n_s) + \frac{(N_z-1)(N/n_s)}{N_z} \quad e^C_t} \tag{11}$$

where W^C is the extended model for prediction of routing wires per channel, χ^C_{fpga} is net-length conversion factor, and e^C_t is a wiring track utilization factor in our 3-D FPGA. The 3-D wirelength distribution f_{3D} is represented as a function of N and l where N is scaled to N/n_s.

The placer of our 3-D FPGA does the clustering of slices in the circuit into each CLB and n_s is a variable determined by the placement algorithm but we expect that it is a constant as shown in experimental results. Although n_s is determined by the placement algorithm as shown in the experimental results, the constancy of n_s enables the proposed prediction model to predict the number of required wires per channel without running placement and routing repeatedly. If the placement algorithm is changed, n_s should be recomputed based on the placement algorithm. The introduction of parameters obtained from the real experimental results on *warming-up* benchmarks in the prediction model relieves the prediction model of "the dependency on the algorithm optimization". We determine the values of variables in Eq. 11 from the XDL netlist to expect the required number of wiring resources per channel before 3-D placement and routing.

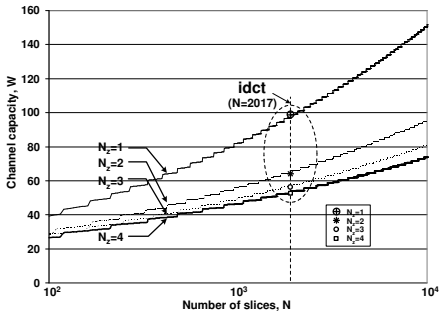

Figure 5: The number of required routing wires determined analytically, W^C as a function of N for the number of strata, $N_z = 1, 2, 3,$ and 4 which is computed by Eq. 11. The measured (placed and routed) results for *idct* are overlapped on the figure. The circuit parameters are $k = 6.68$, $f_o = 2.33$, and $p = 0.72$.

4. EXPERIMENTS

Table 1: Circuit parameters for predicting the number of wires per channel. k and p are Rent's parameters, f_o is the average number of fan-outs of nets, α is $\alpha = f_o/(1 + f_o)$, N is the number of slices in the XDL netlist, and e^C_t is the wiring resource utilization factor.

circuits	k	N	f_o	α	p	e^C_t
addrgen	5.65	304	2.45	0.71	0.57	0.45
bigkey	8.01	2027	3.10	0.76	0.46	0.45
clma	8.60	4132	3.57	0.78	0.65	0.45
diffeq	8.49	538	3.28	0.77	0.49	0.45
dsip	7.36	687	2.98	0.75	0.43	0.45
elliptic	8.48	1518	3.33	0.77	0.77	0.45
frisc	7.72	2584	3.02	0.75	0.74	0.45
idct	6.68	2017	2.33	0.70	0.72	0.45
mac1	7.88	1509	3.35	0.77	0.67	0.45
mac2	7.89	5349	3.42	0.77	0.79	0.45
matrix	6.93	311	2.56	0.72	0.59	0.45
rgb_interp	6.88	255	3.08	0.76	0.48	0.45
s38417	8.25	1870	3.16	0.76	0.63	0.45
s38584	8.65	2282	3.50	0.78	0.57	0.45
sdram	7.68	763	2.94	0.75	0.66	0.45
tseng	7.53	515	3.28	0.77	0.46	0.45
vp2	6.87	1253	2.99	0.75	0.60	0.45

The 3-D FPGA CAD flow is implemented by a HDL synthesizer, technology mapping tool from Xilinx for generating XDL netlist from EDIF, and our tools for placement and routing. 3-D placement and routing tool is implemented in C. Benchmark circuits are from LGSynth93 [2] and ISPD2001 Verilog benchmarks in [3]. The experimental architecture has 40×40 tiles in each layer with intra-stratum wire segments of "DIRECT" and inter-strata wire segments of "VDIRECT". The length of a wire with type of DIRECT and VDIRECT spans 1 tile. All wires are connected to two 3-D switches at their end-points.

Figure 5 shows the number of routing wires per channel as a function of N computed by the analytic model in Eq. 11. The number of required routing wires increases as N increases while it decreases as the number of strata, N_z, increases. The circuit parameters used in Figure 5 are extracted from one of the benchmark circuits, *idct*. The measured numbers of routing channels after placement and routing for *idct* are overlapped on the figure. The analytic results predict the measured number of routing wires per channel in a reasonable way.

We have compared the predicted number of routing wires per channel computed by Eq. 11 with the measured numbers for benchmark circuits. The circuit parameters are shown in Table 1. k is Rent's parameter which is the average number of active terminals of slices. The active terminal of a slice is a terminal which is connected to a net in the XDL netlist. N is the number of slices in the XDL netlist and f_o is the average number of fan-outs of nets interconnecting

Table 2: Comparison of the number of routing wires per channel for analytic results, W^C (CPM) and the actual number (3DPR) obtained from placed and routed results.

	L=1			L=2			L=3			L=4		
circuit	CPM	3DPR	Error	CPM	3DPR	Error	CPM	3DPR	Error	CPM	3DPR	Error
addrgen	41.0	47	-14.6%	30.0	30	0.0%	27.8	24	13.7%	26.5	26	1.9%
bigkey	67.2	86	-28.0%	48.0	30	37.5%	43.4	26	40.1%	41.4	25	39.6%
clma	145.0	156	-7.6%	96.2	91	5.4%	83.2	81	2.6%	77.3	85	-10.0%
diffeq	63.7	72	-13.0%	47.5	49	-3.2%	43.2	38	12.0%	41.5	35	15.7%
dsip	49.2	50	-1.6%	36.4	38	-4.4%	33.3	32	3.9%	32.1	29	9.7%
elliptic	137.6	138	-0.3%	91.7	87	5.1%	79.7	78	2.1%	73.9	81	-9.6%
frisc	226.7	207	8.7%	148.6	137	7.8%	130.4	138	-5.8%	119.2	128	-7.4%
idct	99.1	100	-0.9%	66.0	63	4.5%	57.9	55	5.0%	54.1	53	2.0%
mac1	136.4	108	20.8%	93.0	90	3.2%	81.9	85	-3.8%	76.6	74	3.4%
mac2	324.8	212	34.7%	204.5	178	13.0%	174.0	182	-4.6%	159.7	184	-15.2%
matrix	43.9	51	-16.2%	32.0	28	12.5%	29.6	29	2.0%	28.2	22	22.0%
rgb_interp	36.2	38	-5.0%	27.7	27	2.5%	25.8	20	22.5%	24.7	25	-1.2%
s38417	107.5	129	-20.0%	73.1	71	2.9%	64.7	39	39.7%	60.8	54	11.2%
s38584	103.7	127	-22.5%	71.1	57	19.8%	63.5	46	27.6%	60.1	57	5.2%
sdram	86.0	105	-22.1%	59.1	59	0.2%	52.9	53	-0.2%	50.6	40	20.9%
tseng	52.8	59	-11.7%	38.8	39	-0.5%	36.1	36	0.3%	34.7	29	16.4%
vp2	76.1	68	10.6%	52.6	49	6.8%	47.5	47	1.1%	44.6	48	-7.6%
avg.(abs)			14.0%			7.6%			11.0%			11.7%

terminals of slices. α is computed as $\alpha = f_o/(1 + f_o)$. p is Rent's parameter which is extracted by iteratively partitioning XDL netlist. e_t^C is the wiring resource utilization factor which depends on the efficiency of the routing algorithm. In a general routing algorithm, the global routing assigns a set of routing channels to a net and the detailed routing assigns one of the wiring tracks to the net. Our routability-driven 3-D routing algorithm is a combined global-detail routing algorithm. The net ordering stage which determines the routing priority of nets affects e_t^C. In our experiments, we have assumed that $e_t^C = 0.45$.

n_s is the average number of active slices in a CLB and it is closely related to the placement algorithm. In our experiments we assume that n_s is 3.32 for the proposed 3-D placement algorithm. We believe that n_s depends only on the placement algorithm and a fixed n_s value is used whenever a particular placement algorithm is used. In other words, the proposed prediction model uses the n_s value extracted from several benchmark circuits for other designs without running the placement algorithm. The actual n_s values for benchmark circuits are shown later in this section.

The number of routing wires per channel for analytic results and routed results are compared in Table 2. "L" is the number of strata, "CPM" represents the proposed extended prediction model, and "$3DPR$" represents measured results of 3-D placement and routing. In experiments, χ_{fpga} and e_t^C are set as 0.95 and 0.45, respectively. The values of "CPM" may not be an integer because they are an estimated value for the number of routing wires per channel. "Error" shows the percentage error of the measured number of routing wires. The average of absolute values of errors are shown in the row of "avg.(abs)". The average value of absolute errors is 11.1%. For some benchmark circuits, the number of routing wires per channel for $L = 4$ is not better than that for $L = 3$. It means that there is an optimal number of strata for a given circuit depending on the interconnection type of the circuit. We expect that the number of routing wires per channel does not always decrease monotonically and there is an optimal number of strata for a given circuit. For those circuits, we expect that analytic wire resource prediction model will not be so effective in estimating the required number of wires per channel. The number of required wires for some benchmarks is large because we have used only DIRECT and VDIRECT segment which is a tile-to-tile connection. In Xilinx Virtex2, there are four types of wire segments, DIRECT, HEX which spans six tile-to-tile connections, DOUBLE which spans two tile-to-tile connections, and LONG which spans an entire row or column. The number of wires per channel for each wire segment is 16, 120, 40, and 24, respectively.

Table 3: The average number of active slices in a CLB, n_s of placed results.

circuit	L=1	L=2	L=3	L=4
addrgen	2.69	2.69	2.49	2.47
bigkey	3.70	3.50	3.28	3.29
clma	3.61	3.72	3.74	3.67
diffeq	3.59	3.69	3.54	3.41
dsip	2.99	3.07	2.84	2.84
elliptic	3.78	3.68	3.68	3.68
frisc	3.68	3.71	3.71	3.69
idct	3.21	3.07	3.04	3.05
mac1	3.68	3.70	3.62	3.51
mac2	3.85	3.64	3.66	3.71
matrix	3.08	3.05	2.85	2.70
rgb_interp	2.93	3.00	2.34	2.83
s38417	3.25	3.63	3.43	3.48
s38584	3.02	3.57	3.51	3.41
sdram	3.47	3.28	3.22	3.10
tseng	3.55	3.34	3.20	3.22
vp2	3.28	3.07	3.23	3.41

The actual n_s values for the benchmark circuits are shown in Table 3. The average value of n_s is 3.32 and the standard deviation is 0.36. We have also measured n_s for the

other placement algorithm, simulated annealing with only random slice movement. n_s for this placement algorithm is 2.2 and standard deviation is 0.21. We expect that n_s depends only on the placement algorithm and is independent of the number of strata. The n_s value obtained from some benchmarks is used for other designs without running placement and routing repeatedly. In our experiments, we set n_s to 3.32 which is the average value of the numbers in Table 3.

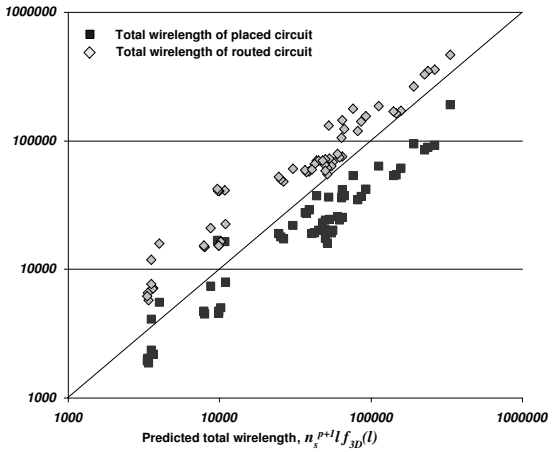

Figure 6: The scatter plot of the wirelengths of placed and routed results versus the predicted wirelength on the X-axis. The unit of total wirelength is a length of DIRECT wire segment. The axes are in log-log scale.

There are several error sources causing the predicted values to deviate from the actual values such as total wirelength, χ_{fpga}, and e_t. Figure 6 shows the scatter plot of the total wirelengths of placed and routed circuits with the predicted total wirelength of the proposed analytic model on the X-axis. As shown in Table 6, the total wirelengths of the routed circuit is usually larger than that of the prediction. The deviation of the total wirelength of the routed circuits from the predicted total wirelength is explained as follows. First, the equation for the total wirelength, $n_s^{p+1} l f_{3D}(l, N/n_s)$ considers only point-to-point interconnection without consideration of router behavior in the global routing stage. In the global routing stage, the router searches for the routing path of each net sequentially. If the number of occupied wires in a channel is not sufficient, the router searches for a detour path of the net. However, the function $f_{3D}(l)$ does not consider this effect and it causes the difference between the routed total wirelength and the predicted one. Second, the results of detailed routing depend on the 3-D switch interconnection topology. The limited number of programmable interconnection points controlled by SRAM configuration bits in the 3-D switch may cause the circuit to be unroutable in the detailed routing stage. If the router fails in the detailed routing stage, it adjusts the capacity of each channel and goes back to the global routing stage for rerouting. The routability issues in FPGAs cause the total wirelength of the routed circuits to deviate from that of the prediction model.

Figure 7 shows the average number of wires over the required (maximum) number of wires. It shows how much the required number of wires per channel deviates from the average number of wires per channel. As shown in Figure 7, the wire utilization is 0.15∼0.68. The deviation comes from two factors: the wiring congestion cost in the placer and 3-D switch interconnection. First, the wiring congestion cost in the placement algorithm tries to maintain the uniformity in the number of allocated wires per channel. The placer uses a wiring congestion estimation to place instances such that there is not a heavily congested routing channel. However, the efficiency of the congestion estimation methodology decreases due to the circuit topology and the router behavior. The inefficiency of the congestion estimation methodology in the placer increases the deviation between the required number of wires per channel and the average number of wires per channel. Second, 3-D switch interconnection topology also increases the deviation. If 3-D switch is a full crossbar where each pin is connected to all the other pins in the switch, the deviation is small because detailed router never fails. However, the detailed router can not fully utilize the result of global routing because of the 3-D switch interconnection topology and it decreases the utilization of the channels as explained in Figure 3. The values shown in Figure 7 are also interpreted as the actual e_t value, the wire utilization of the routed circuits. As shown in the figure, the range of wire utilization is from 0.15 to 0.68. We have used the fixed value of e_t in the equation but we expect that the quality of the prediction model would be improved by extracting the equation of e_t depending on the circuit parameters.

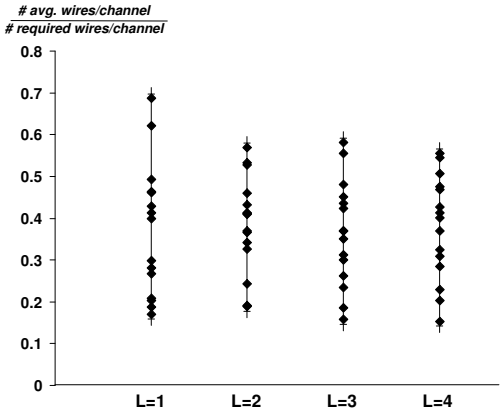

Figure 7: The ratio of the average number of used wires per channel to the required (maximum) number of wires per channel.

In the proposed analytic equation, n_s plays an important role in accounting for the clustered FPGA. The RMS (Root-Mean-Square) prediction errors for various n_s values are shown in Figure 8. The sensitivity of RMS error to the value of n_s around the minimum point (∼3.5) is not so large but it increases as n_s is far from the minimum peak point. In Table 3, most of the n_s values are in between 2.8 and 3.8. If we use the n_s value of 2.2 which is for simulated annealing with only random placement, the RMS error for the results obtained by simulated annealing with two phases is 122.2%. However, the minimum value of n_s, 2.47 for the *addrgen* benchmark still causes a large RMS error. We expect that n_s values represented as an equation based on circuit parameters will be more efficient.

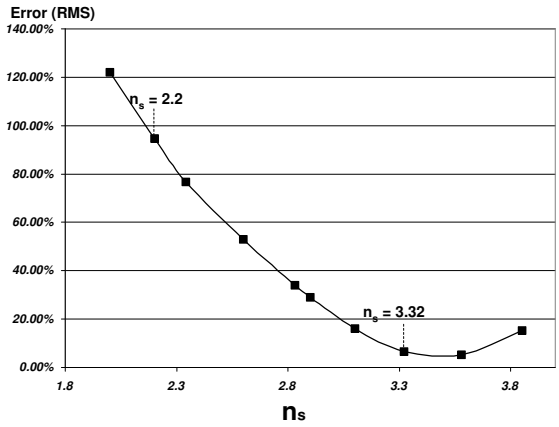

Figure 8: The RMS (Root-Mean-Square) prediction errors versus n_s values. As the distance between the minimum point and n_s increases, the RMS error increases.

5. CONCLUSIONS

We have proposed an extended 3-D wire resource prediction model for the proposed 3-D FPGA. We have shown that the 3-D FPGA wire resource prediction model is valid by comparing the number of routing wires per channel computed by the proposed wire resource prediction model with the required number of routing wires per channel of the placed and routed results.

6. ACKNOWLEDGMENTS

This material is based upon work supported by IBM and DARPA under SSC SAN DIEGO Contract No. N66001-04-C-8032. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of IBM, DARPA or SSC SAN DIEGO. The authors would like to thank Dr. Arifur Rahman for providing the MATLAB code of his wire resource prediction model.

7. REFERENCES

[1] V. Betz and J. Rose. VPR:A new packing placement and routing tool for FPGA research. In *Field-Programmable Logic and Its Applications*, pages 213–222, 1997.

[2] CBL, http://www.cbl.ncsu.edu/CBL_Docs/lgs93.html. *LGSynth93 benchmark [online]*.

[3] Y.-C. Chou and Y.-L. Lin. A Performance-driven Standard Cell Placer based on a Modified Force-directed Algorithm. In *International Symposium on Physical Design*, pages 24–29, 2001.

[4] J. Darnauer and W. W.-M. Dai. A method for generating random circuits and its application to routability measurement. In *ACM International Symposium on Field Programmable Gate Arrays*, pages 66–72, 1996.

[5] S. Das, A. P. Chandrakasan, and R. Reif. Three-dimensional integrated circuits:Performance, design methodology, and CAD tools. In *International Computer Society Annual Symposium on VLSI*, pages 13–18, 2003.

[6] S. Das, A. P. Chandrakasan, and R. Reif. Calibration of Rent's rule models for three-dimensional integrated circuits. *IEEE Transactions on VLSI Systems*, 12(4):359–366, Apr. 2004.

[7] J. A. Davis, V. K. De, and J. D. Meindl. A stochastic wire-length distribution for gigascale integration(GSI)-PartI:Derivation and Validation. *IEEE Transactions on Electron Devices*, 45(3):580–589, Mar. 1998.

[8] J. A. Davis, R. Venkatesan, A. Kaloyeros, M. Beylansky, S. J. Souri, K. Banerjee, K. C. saraswat, A. Rahman, R. Reif, and J. D. Meindl. Interconnect limits on gigascale integration (GSI) in the 21st century. *Proceedings of IEEE*, 89(3):305–324, Mar. 2001.

[9] A. El-Gamal. Two dimensional model for interconnections in master slice integrated circuits. *IEEE Transactions on Circuits and Systems*, 28:127–138, Feb. 1981.

[10] A. Fan and R. Reif. Three-dimensional integration with copper wafer bonding. In *Interconnect Technology Conference*, pages 18–20, 2000.

[11] J. W. Joyner, P. Zarkesh-Ha, J. A. Davis, and J. D. Meindl. A three-dimensional stochastic wire-length distribution for variable separation of strata. In *Interconnect Technology Conference*, pages 126–128, 2000.

[12] A. Rahman, S. Das, A. P. Chandrakasan, and R. Reif. Wiring requirement and three-dimensional integration of field-programmable gate arrays. In *International Workshop on System-level Interconnect Prediction*, pages 107–113, 2001.

[13] A. Rahman, S. Das, A. P. Chandrakasan, and R. Reif. Wiring requirement and three-dimensional integration technology for field programmable gate arrays. *IEEE Transactions on VLSI Systems*, 11(1):44–54, Feb. 2003.

[14] A. Rahman, A. Fan, and R. Reif. Comparison of key performance metrics in two- and three-dimensional integrated circuits. In *Interconnect Technology Conference*, pages 18–20, 2000.

[15] A. Rahman and R. Reif. System-level performance evaluation of three-dimensional integrated circuits. *IEEE Transactions on VLSI Systems*, 8(6):671–678, Dec. 2000.

[16] S. Sastry and A. C. Parker. Stochastic models for wireability analysis of gate arrays. In *IEEE Transactions on CAD*, pages 52–65, 1986.

[17] T. Tuan and B. Lai. Leakage power analysis of a 90nm FPGA. In *IEEE Custom Integrated Circuits Conference*, pages 57–60, 2003.

[18] Xilinx Inc. http://www.xilinx.com, San Jose, CA. *Xilinx Virtex-II Platform FPGA [online]*, 2002.

[19] R. Zhang, K. Roy, C.-K. Koh, and D. B. Janes. Stochastic wire-length and delay distributions of 3-dimensional circuits. In *IEEE/ACM International Conference on CAD*, pages 208–213, 2000.