

Enhancement and Analysis of a Simple and Efficient VLSI Model

BERNHARD FECHNER; JÖRG KELLER

Department of Computer Science
FernUniversität in Hagen
58084 Hagen
{Bernhard.Fechner|Joerg.Keller}@fernuni-hagen.de

Abstract

To estimate the delay and area of a VLSI circuit before its implementation demands a cost and delay model which should be simple enough to allow a quick and therefore cost-effective paper and pencil analysis. However, common VLSI models are either too complicated resulting in long simulation times or do not consider wire-delays, neglecting their importance in sub-micron and nanotechnologies. Paul and Seidel introduced such a technology-independent model in 1998 [Paul, Wolfgang J.; Seidel, Peter-Michael: On the Complexity of Booth Recoding, 1998]. Based on this model, we will introduce a new VLSI cost and delay model. It consists of two parts – a basic technology-independent part based on reasonable enhancements of the Paul and Seidel model and a technology-specific part, where a specific technology can be implied to get the effective circuit cost and delay. We analyze the accuracy by comparing the cost and performance of synthesized adder circuits with the results derived from the model. The comparison reveals that the model closely matches the cost and delay of the standard-cell and freely placeable element-based implementations in the majority of the tested circuits. Therefore, it could enhance existing conventional VLSI models while retaining their simplicity.

1 Introduction

All information processing systems have one thing in common: The constantly growing complexity and performance, caused by higher speed of operation and integration density. The more dense integration gets, the more circuits can be realized on a chip. Without the help of computer-based simulation, it is impossible to calculate the performance of larger circuits; predictions from simple VLSI models, e.g. the gate delay model are not accurate enough. Simulation time increases with the complexity of such a system. To simulate the behavior of a circuit, an implementation in a hardware description language (HDL) is necessary. This increases the time-to-market and hence the development costs. A way to reduce these costs is a simple and effective VLSI model that allows performance prediction prior to the implementation in a HDL and pays attention to the more and more domi-

nant wire-delays occurring in sub-micron and nanotechnologies. Paul and Seidel [4][5] introduced such a model that even allowed analysis by hand.

This paper makes the following contributions:

- It clarifies some obscurities of the model by Paul and Seidel.
- We introduce a new model, based on reasonable technology-independent enhancements to the Paul-Seidel model.
- It presents a technology-specific part to convert the values from the technology-independent model into units (ps, μm^2) for particular technologies.
- It accomplishes an analysis of the model's accuracy regarding cost and delay.
- The model is validated by a comparison with the results gained from the layout synthesis.

The paper is organized as follows. As technology-independent model, the model of Paul and Seidel and the new technology-independent model based on Paul-Seidel are presented in Section 2. In Section 3, we calculate the minimal delay in maximal circuit area within the model. In Section 4, we introduce the technology-dependent part of the model by calculating conversion factors so that the values from the technology-independent model can be translated into units (ns/ μm). Furthermore, an analysis and comparison of the model with other VLSI models is accomplished and further extensions are proposed. Section 5 summarizes and concludes the paper.

2 Technology-independent Modeling

2.1 The Model of Paul and Seidel

A circuit S in the model of Paul-Seidel consists of $n \in \mathbb{N}$ simple, axis-parallel, rectangular and non-overlapping circuits S_i , $i \in \{1, \dots, n\}$, each containing only a few gates ($k \in \mathbb{N}$) plus N nets connecting the S_i , inputs and outputs.

We will determine k in the next section. Within such a rectangle, wire delays and cost are negligible. Figure 1 shows an example such a structure.

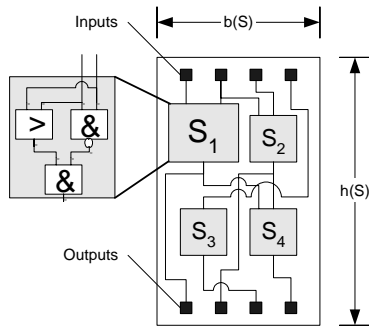


Figure 1: A circuit S

The nets consist of horizontal and vertical wires. They must keep a minimum distance δ to each other and from other circuits. The influence of wires onto circuit area is modeled by δ . The area (cost) of S is the area of the smallest axis-parallel rectangle comprising all nets and all simple circuits S_i . The delay of a wire of unit length on those nets is represented with the help of $v \in [0,1]$. It thus models wire delay relative to an inverter that has unit area and unit delay. The larger v , the larger the influence of wire delays. The delay of a path through S is the sum of the wire and circuit delays on the path.

The cost and delay of gates for a specific technology are given in Table 1 (from [5]). The area of a simple circuit S_i is the sum of its gate costs. The delay of a path through S_i is the sum of the gate delays on the path.

Table 1: Gate costs and delays

Gate	Costs	Delay
XOR/XNOR	4	2
HA	6	2
AND/NAND/OR/NOR	2	2
NOT	1	1

Different delays at the in- and outputs and the placement of the connection pins (at the upper and lower side of the rectangle) are not considered. Inputs are on the upper, outputs on the lower side of the rectangle. For the dimension of S containing all S_i and wires

$$\frac{h(S)}{2} \leq b(S) \leq 2h(S) \quad (*)$$

is demanded, i.e. S may be not too narrow or too wide. Table 2 contains an overview of the symbols used by Paul and Seidel.

Table 2: Summary of the symbols in the model

Symbol	Meaning
δ	Minimal wire-circuit, wire-wire distance
v	Influence of wire delays on the total delay
$\delta(t+1)$	Minimal distance of t wires
$b(S)$	Width of circuit S
$h(S)$	Height of circuit S
$C(S)$	Cost of S. $b(S)h(S)=C(S)$
$D(S)$	Gate delay of S
$ N $	Size of net N (sum of wire lengths of the net)

2.2 Maximal number of In- and Outputs

With $\delta=1$ the number of inputs of a gate is limited to two (distance δ to both sides of the rectangle). The situation is similar for the AND gate. It has two inputs. Here the outcomes are $b(\text{AND})=1$ and $h(\text{AND})=2$. With $\delta=1$ the contradiction $b(\text{AND})=3$ results. There are three ways out of this situation:

1. Arbitrary placement of in- and outputs on all sides of the rectangle.
2. Relaxing the regulation for the minimum height/width of a structure S.
3. Limiting δ in a way sufficient in- and output pins can be placed on the upper and lower side.

An arbitrary placement of the in- and output pins or other mini-/maximum lengths and widths offend against the restrictions of the Paul-Seidel model. Only the third possibility is applicable. Let us examine the layout (done with Alliance¹) of the exemplary operator G_f where (+ represents logic OR),

$$G_f : \{0,1\}^3 \rightarrow \{0,1\}$$

$$G_f(a_0, b_0, c_{in}) := a_0 b_0 + (c_{in}(a_0 + b_0)).$$

We have chosen this operator, since it is used in all examined circuits in this work (s. Table 4). Figure 2 shows the layout of the G_f -Operator in 0.35 μm 5-layer CMOS-technology. From the dimensions of 100 \times 66 μm an aspect ratio of 3:2 results. This corresponds exactly to the relationship, won from the area (=6 [2 OR, 1 AND gate]) of G_f , in accordance to the geometrical restrictions (*) of the model.

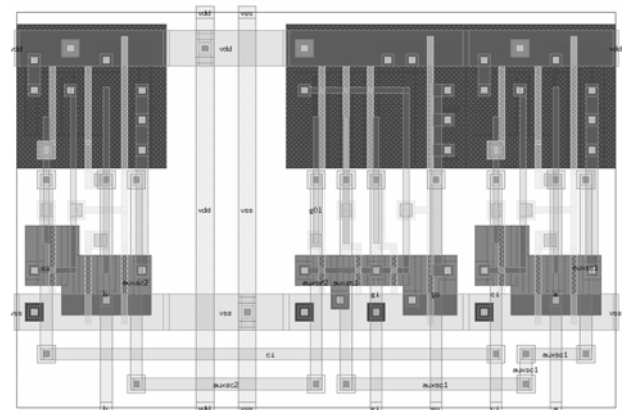


Figure 2: Layout of the G_f -Operator

If we choose $\delta=1$, $b(G_f)=5$ results since G_f has four inputs. In order to reach $b(G_f)=3$, $\delta \leq 3/5=0.6$ must be selected. Analogue to $b(S)=2$, $\delta \leq 2/5=0.4$ results. This is an acceptable value if we consider the following:

- 0.35 μm -technologies and below are current standards.
- With an increasing number of layers, a smaller δ could be selected.
- G_f from Figure 2 has seven pins (including VDD, GND) at the lower side with an aspect ratio of 3:2.

¹ Available at <http://www-asim.lip6.fr/recherche/alliance/>

For these reasons, it shall be permitted to limit δ in such a way that a sufficient number of input and output pins can be placed on the upper and lower side. Due to the structures in this work it is

$$\delta \leq 2/5 = 0.4.$$

If $b(S)$ would only depend on the number of input and output pins t , $b(S) \geq \delta(t+1)$ applies.

From this follows:

$$t \leq \left\lfloor \frac{b(S) - \delta}{\delta} \right\rfloor.$$

For example, with $\delta = 0.07$ and $b(S) = 4$, $t = 56$ results. This number of in- and output pins is not realistic.

Before this thought is continued, we note the following: In [5] there is no given upper limit of how many logical elements k can be combined into a single rectangle. An orientation is given by the largest cells in standard-cell libraries. These are e.g. adders, 8-1 multiplexers and complex flip-flops.

In our opinion, the maximal number of logic gates in a simple circuit S_i shall be limited to five (the number of gates for a full adder). Larger cells cannot be modeled due to limited resources for the internal wiring. Regarding the number of pins in this context again, e.g. a maximum of 5 NOT or 5 XOR gates can be accommodated in a simple circuit. The maximal number of gate in- and outputs in Table 1 is two and one, respectively. A maximum of 10 in- and 5 outputs results, whereby it is allowed to connect one input pin of S_i with several internal gate inputs. To limit the output pins solely by δ , these are connected one-to-one with a gate output.

The maximal number of inputs of a simple circuit S is therefore limited to 10; the number of outputs is limited to 5. To get an upper bound for δ , we solve

$$t \leq \frac{b(S) - \delta}{\delta} \Leftrightarrow \delta \leq \frac{b(S)}{t+1}.$$

$b(S)$ should be minimal and t maximal. With gates having two inputs, $\max(t) = 10$ follows. Since these gates should be as small as possible, only NAND, NOR, AND and OR gates are applicable. From these, five can be integrated into S without hurting condition (*). Hence $b(S) = 3$, $h(S) = 4$ and $\delta \leq 3/11$ follow. This should only be seen as an upper bound if structures have more than 10 inputs. Otherwise, δ can be selected as small as necessary.

2.3 Fan-In and Fan-Out

Paul and Seidel do not limit fan-out. One may argue that it is implicitly modeled via the parameter v . In practice, it is limited between four and six. This restriction demands driver trees, special algorithms and strategies [3]. The fan-in is limited by the area boundaries of a gate. Since all gates in Table 1 have at most two inputs, the largest fan-in allowed by Paul and Seidel is two. Occasionally gates with a higher fan-in are used. We can consult standard-cell libraries again. Here, the largest commonly allowed

fan-in is four. If a higher fan-in should be possible, fan-in trees must be designed. In order not to complicate the analysis and layouts, no driver trees will be designed neither to compensate fan-in nor fan-out. This will not harm the analysis, since the circuits were generated in a way that their speed is affected neither by fan-in nor by fan-out. Maximal fan-in will therefore be limited to four.

2.4 Layout and Wire-Delay

The comparison between wire and gate delay leads to the restriction of v between zero and one. A wire led along a simple circuit, should never have more delay than this gate.

If $(D(S) \geq b(S)) \wedge (D(S) \geq h(S))$ no problems occur (e.g. G, XOR). If $(D(S) < b(S)) \vee (D(S) < h(S))$ and $v = 1$ (see Figure 3), the time a signal needs from the upper to the lower end of the half-adder (HA) is larger than the gate delay.

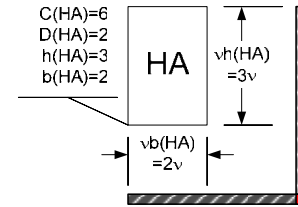


Figure 3: Wire-delay along a simple circuit (HA)

Half-adders are the only elements used in this work, causing such difficulties. Therefore, it should be assured that a situation as in Figure 3 does not occur during the element placement in the layouts. If wires are led along such structures, a constant, marginal deviation develops. Since v represents the metal layer resistance, no problems arise if these wires are not affected by fan-out. Physical factors are modeled over a single parameter v . Therefore, the model may become inaccurate, but allows simple computation and is partially technology-independent. E.g., the 3D-CMOS SOI technology developed at the Institute for Microelectronics Stuttgart could only be modeled by skillful adjustment of δ and v . Today's usual routing in several layers over cells and wires is not considered, but can be compensated by adjustment of δ . Broader wires can become longer without drivers but need more area. This can be balanced by a weighting factor of the concerned wires. For a straight analysis, we did not take any of these factors into account.

3 Minimal Delay in maximal Circuit Area

In order to get an impression of how small the delay of a circuit examined with the model can become, a theoretical computation of these barriers has to take place. With the help of our model we gained the formulae given in this Section. This enables us to compute an upper bound for the area and minimal delay of any rectangular circuit structure. It thus gives an upper bound for standard-cell or FPGA-based designs. Let K be the examined circuit (see Figure 4).

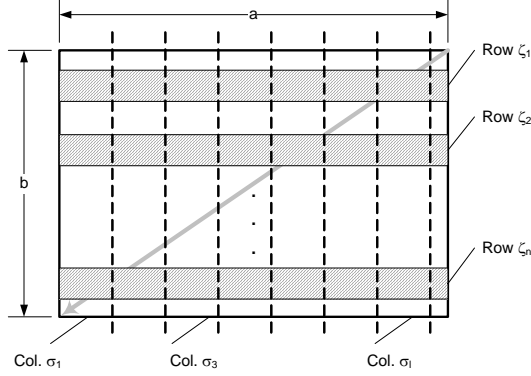


Figure 4: The circuit K

We assume the following:

- K has a rectangular form, whereby $a, b \in \mathbb{R}^+$ are the side lengths of K .
- There are $n \in \mathbb{N}$ rows $\zeta_i; i \in \{1, \dots, n\}$, $l \in \mathbb{N}$ columns $\sigma_j; j \in \{1, \dots, l\}$.
- Between the rows, $n-1$ wire channels exist, but none between individual columns.
- Only two neighboring wires are interlaced. Wires do not cross.
- The connection points of the structures S are equally distributed.

Within the circuit reside ln structures $S_{1,1}, \dots, S_{n,l}$, which are dimensioned according to the Paul-Seidel model (*). First, we try to measure the maximum width of circuit K : the minimum/maximum costs C_{min} and C_{max} a structure $S \in \{S_{1,1}, \dots, S_{n,l}\}$ can have are:

$$C_{min}=1 \text{ (a NOT-gate)}$$

$$C_{max}=20 \text{ (5 XOR-gates)} \Rightarrow C(S) \in \{1, \dots, 20\}.$$

Let $\max(b(S))$ and $\max(h(S))$ be the maximal width and/or height of the structure S under the condition that

$$C_{max}=\max(b(S))\max(h(S)).$$

Since broader structures have more connection points on the upper and lower side, the condition

$$\max(b(S))=5, \max(h(S))=4$$

can be derived. The maximum number of output pins of a structure is five, so

$$a = \max(B(K)) = \max(b(S))l = 5l$$

applies.

If all structures of a row ζ_i are connected with ζ_{i+1} , excluding multiple wiring of inputs and wirings within a

column, the maximal height of a wire channel $h(W_{\zeta_i})$ (with maximal 5 outputs) results to:

$$\max(h(W_{\zeta_i})) = \delta l \left[\frac{b(S) - \delta}{\delta} \right] + \delta = \delta l t + \delta = \delta(5l + 1).$$

Consequently,

$$\begin{aligned} \max(H(K)) &= \max(h(S))n + (n-1)\max(h(W_{\zeta_i})) \\ &= 4n + \delta(5nl + n - 5l - 1) \end{aligned}$$

follows.

Thus

$$\begin{aligned} \max(C(K)) &= \max(H(K))\max(B(K)) \\ &= 20ln + (n-1)(25\delta l^2 + 5\delta l) \end{aligned}$$

gives us the maximal cost of a circuit. We get a lower bound of the runtime with maximal area on the assumption that a signal exists that at least must pass K diagonally. Then there is (using Manhattan-metric) a path of length $L=a+b$. On this path n rows, n structures, $n-1$ wire channels and a horizontal wiring of length a will be passed. The wire delay $T(W)$ sums up to

$$\begin{aligned} T(W) &= v((n-1)\max(h(W_{\zeta_i})) + a) \\ &= v(\delta(5l(n-1) + n - 1) + 5l). \end{aligned}$$

If each structure S has depth 10, the entire delay is:

$$\begin{aligned} T(K) &= T(W) + 10n \\ &= v(\delta(5l(n-1) + n - 1) + 5l) + 10n. \end{aligned}$$

As an example, imagine a quadratic structure $l=n$ and let $\delta=0.1=v$. Then we get: $T(K) = 0.05n^2 + 10.46n - 0.01$ and

$$\max(C(K)) = \underbrace{18n^2}_{\text{gate cost}} + n \underbrace{(2.5n^2 - 0.5)}_{\text{wire area}}.$$

For $n=10$, wire delay is approximately 10.43% of the gate delay. For $n \geq \frac{18}{5} + \frac{1}{5} \cdot \sqrt{329} \approx 7.3$ (52 structures S) wire

area gets larger than gate cost. This is not surprising, as complete bipartite interconnection of n modules of two adjacent rows requires quadratic space.

4 Technology-dependent Modeling

In this Section, we develop a way to adapt the results from the technology-independent model so that they can be transferred into units (ps/ns, μm^2). Table 3 shows the resistances of materials from the technology files (Alliance: `cmos8.rds`, `MicroWind2` (MW2): `cmos035.rul`). The VHDL and Verilog sources of the examined adder circuits shown in Table 4 were generated for the bit widths $n \in \{4, 8, 16, 32, 64\}$ by an application based on [10]. The layouts were done by the program packages Alliance and `MicroWind2`. The data received by simulation was compared with the values from the model. By calculating the relative error, an evaluation of the model results. As pro-

² Available at <http://www.microwind.org>

duction technology, we selected CMOS (5-Layer, 3.5V, 0.35 μm , two metal layers).

Table 3: Material characteristics in chip-production

Material Name	$\Omega/\mu\text{m}^2$	
	Alliance	MW2
Aluminum Layer 1	0.1	0.25
Aluminum Layer 2	0.05	0.05
Polysilicon	50	30

Table 4: The modeled adders








Name	Synthesis	Model
Carry-Incrementer [9]	✓	✓
Carry-Skip [1]	-	✓
Han-Carlson [2]	✓	✓
Multilevel Carry-Skip [8]	-	✓
Ripple-Block Brent-Kung	-	✓
Ripple-Block Carry-Incrementer	✓	✓
Ripple-Block Multilevel Carry-Incrementer	-	✓
Ripple-Block Sklansky [7]	✓	✓
Ripple-Block 8-Bit Sklansky/Carry-Incrementer	✓	✓
Ripple-Carry	✓	✓
Ripple-Carry/ CINC	✓	✓
Variable Carry-Skip	-	✓

The layouts were routed in a way that a vertical bus was available.

4.1 Unit Conversion

A possibility to convert the costs from the area model directly into μm^2 is to keep the gate area variable and adapt the minimum feature size (mfs). Table 5 shows the cost for the standard-cells used.

Table 5: Cost for the Used Standard-Cells

S	C(S)	Cost (C_2)	$C_2/C=\rho^2$	Symbol
XOR	4	45×50=2250	562.5	
HA	6	80×50=4000 70×50=3500	666.6 583.3	
Δ'	4	30×50=1500	375	
G_7	6	100×66=6600	1100	
Δ	6	80×61=4880	813.3	
BUF	2	20×50=1000	500	
AND ₄	4	40×50=2000	500	

The cost C_2 of the implementation is given in μm^2 (length×width). For the basic gates AND, OR, XOR and their negation, optimized standard-cells with equal height were used.

From Table 5 we derive the necessary conversion factor $\rho \in \mathbb{R}$, $b(S)\rho h(S)\rho = C(S)\rho^2$ for the conversion into real-world values by calculating the average value from the real-world circuit costs for each circuit i ($C_2(i)$) divided by the theoretical costs $C(i)$ from Table 5. Thus, we get the average area:

$$\rho = \left[\frac{1}{n} \sum_{i=1}^n \frac{C_2(i)}{C(i)} \right] = \left[\frac{1}{n} 658.2 \right] = 25$$

Since this equation implies the minimum feature size, we adapt δ with the minimum feature size. If the traditional-fan-in model is incorporated, a variable gate area can be implicitly modeled over δ , since the area of a gate grows with the number of the in- and outputs.

With a $mfs=0.35\mu\text{m}$, a maximal fan-in of 4, and a quadratic correction factor for the whole circuit,

$$\delta = 4 \cdot \rho \cdot mfs + (mfs)^2 \approx 0.35 \cdot 100 + 0.35^2 = 35.1225$$

results.

To arrange the results of our model in context, a first analysis of the delays was accomplished with all available MOS-delay models in MicroWind2.

These MOS-delay models are:

- The Model 1: A simple and fast model and
- the SPICE Level 3 Model.

For details on all models, see [6]. To compare the models, all inputs were provided with a clock (pulse width 20ns) and the maximal time of a 0→1, and/or 1→0 transition of $A(0)$ until a change of $S(i)$ at $n \in \{4,8,16\}$ was measured with MicroWind2 (operating temperature of 27°). v was set to the resistance of metal layer 1 in Table 3. Remember that v is the only correction factor for wire-delays. Conservatively this is the maximal resistance of a metal-layer. For the technology used with MicroWind2 this is:

$$v = 0.25(\Omega).$$

The results from the model-specific cost and runtime functions of the analyzed circuits were converted and compared with the cost and running times gained with Alliance and MicroWind. With both programs, the area was measured in μm^2 for $n \in \{4,8,16,32,64\}$. For $n \in \{4,8,16\}$ the delay in ns was calculated with MicroWind. We did not select higher values of n because the time to calculate the delay with both models then was unacceptable. From the obtained values, the relative errors were calculated. The arithmetic means of these errors lead to an evaluation of the model. Since the layouts were generated with dynamic algorithms, deviations in height and width from $\pm 10\%$ can occur. The situation is similar for the runtimes, since these are affected by the layout.

4.2 Analysis of the Area Model

The comparison with MicroWind results in an average relative area error of 0.999261252560948 (Alliance: -0.0306956135439548). Figure 5 clarifies this by showing a diagram of the relative errors in respect to the VLSI implementations. A correlation of 0.90596769380099 exists between both measurement series. Consequently, good predictions can be made with the area model. These get even better at increasing bit widths.

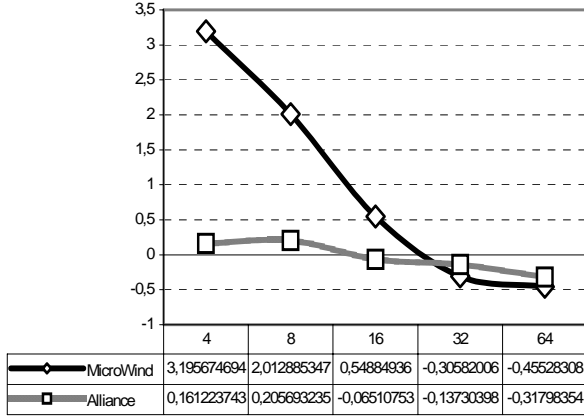


Figure 5: Relative errors for the area model

4.3 Analysis of the Delay Model

The comparison of the delay model with MOS-transistor models SPICE Level 3 and Model 1 from MicroWind for $n \in \{4, 8, 16\}$ at a temperature of 27° resulted in the average relative error of -0.62 (Model 1) and -0.83 (SPICE Level 3), respectively. The values of the delay model were always below those gained by the SPICE Level 3 and Model 1. Therefore, the relative errors are negative. Experiments showed that the actual running time of a circuit is about 4 times higher than the one obtained from Model 1 [6]. The delay model forecasted the smallest running times. Similar to Figure 5, a strong correlation between both measurement series can be recognized in Figure 6.

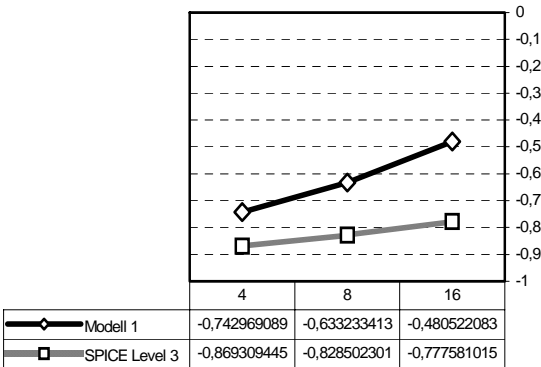


Figure 6: Relative errors for the delay model

It is approximately one (0.999529192533167). Thus, a nearly linear correlation exists between the results. A linear function could adapt the results so that the relative errors will decrease.

4.4 Extensions to the Model

In order to approximate such a function, we could try to integrate technology-conditioned parameters into the model without making it too complicated. In the model from Paul and Seidel and our model, wire delay is modeled over the parameter v . Vertical and horizontal wires in different layers have different resistances and therefore different delays. An extension of the model could contain a representation of these parameters by $v_{\text{horizontal}}$ and v_{vertical} . Transitions from layer i to $i+1$ and back are supposed to be delay-free. Since the fan-out is modeled over the parameter $v \in [0, 1]$, it suggests that the maximal fan-out $\max(\text{fanout}_{\text{circuit}})$ of a circuit could be modeled over v . Let $\max(\text{fanout}_{\text{technology}})$ be the maximal technology-dependent fan-out of a gate (mostly 4). An approximation for the number of drivers that must be inserted is:

$$\#_{\text{driver}} = \left\lceil \frac{\max(\text{fanout}_{\text{circuit}})}{\max(\text{fanout}_{\text{technology}})} \right\rceil.$$

This number could be added to an area estimation, which respects fan-out. v is a global parameter. A change of v involves an overall change of the results. Alternatively v could be modeled locally as $v_L \in [0, 1]$ by e.g. weighting the wire L concerned by fan-out with the maximal fan-out for this technology. v grows with fan-out x . $T(\text{BUF})$ represents the delay of a buffer. Orientating on $\#_{\text{driver}}$, the delay of a driver and thus the (additional) delay can be modeled as the following function:

$$v_L : \mathbb{N} \rightarrow [0, 1]$$

$$v_L(x) := 1 - \left(T(\text{BUF}) \left[\frac{x}{\max(\text{fanout}_{\text{technology}})} \right] + 1 \right)^{-1}$$

$$v_L(0) = 0, \lim_{x \rightarrow \infty} v_L(x) = 1.$$

This choice of v_L only represents a rough estimation, since it strongly depends on the technology used.

5 Summary and Conclusion

With a VLSI model derived from Paul and Seidel [5] we evaluated the area and delay of different synthesized adder circuits for input widths $n \in \{4, 8, 16, 32, 64\}$ and $n \in \{4, 8, 16\}$ respectively. The considerations showed that the area model always gives an upper bound for the costs up to $n=16$. From the small area deviation to the circuits synthesized with Alliance (rel. error: ≈ -0.031) we conclude that the area model gives a very good approximation for the cost of a standard-cell design without adjustment. With freely placeable elements (MicroWind), the relative errors increase for bit widths $n \leq 16$, since the automatic

place-and-route algorithms then work very efficiently. Since the relative errors decrease in a linear fashion up to $n=32$ we conclude that a linear function can adapt the results from the area model up to 32 bits in such a way that good approximations are accomplished. Starting from $n=32$ such a function is no longer necessary. The new delay model was compared with the MOS transistor models SPICE Level 3 and Model 1 from MicroWind at a simulation temperature of 27° . A clock with 20ns pulse width was applied to all inputs of the simulated circuit. The maximal time from a $0 \rightarrow 1 / 1 \rightarrow 0$ transition at the inputs until a change of $S(i)$ was measured. The comparison was done for $n \in \{4, 8, 16\}$ of all synthesized circuits. Our model always predicts the smallest running times. From the trend of the relative errors and the strong correlation to each other we conclude that the values from the model can be adapted by a linear function in such a way that the results deviate marginally from those of the Model 1 or even SPICE Level 3.

References

- [1] Chan P. K. et al: *Delay optimization of carry- skip adders and block carry- lookahead adders*. In Proc. 10th Computer Arithmetic Symp., Grenoble, S. 154–164, June 1991.
- [2] Han, T.; Carlson, D. A.: *Fast area-efficient VLSI adders*. In Proc. 8th Computer Arithmetic Symp., Como, S. 49–56, 1987.
- [3] Kreutzer, M.: *Das Fanout-Problem*, University of Saarbrücken, Department of Computer Science, Master thesis, 1996.
- [4] Paul, W. J.; Seidel, P.-M.: *To Booth or not to Booth*. In Integration, the VLSI Journal, Elsevier, 2002.
- [5] Paul, W. J.; Seidel, P.-M.: *On The Complexity Of Booth Recoding*. In Proc. 3rd Conf. On Real Numbers and Computers, pp. 199-218, 1998.
- [6] Sicard, E.: *MicroWind & Dsch User's Manual Version 2*, Toulouse, INSA, Department of Electrical & Computer Engineering, S. 18-21, May 2002.
- [7] Sklansky, J.: *Conditional sum addition logic*. IRE Trans. on Electronic Computers, EC-9(6): 226–231, June 1960.
- [8] Turrini, S.: *Optimal Group Distribution in Carry-Skip Adders*. Western Research Laboratory (WRL) Research Report 89/2, 1989.
- [9] Tyagi, A.: *A Reduced-Area Scheme for Carry-Select Adders*. IEEE Trans. on Computers, 42(10):1162–1170, 1993.
- [10] Zimmermann, R.; Kunz, Hanspeter: *High-Performance Adder Circuit Generators in Parameterized Structural VHDL*. Zürich, Eidgenössische Technische Hochschule, Institut für Integrierte Systeme. Technical Report No. 96/7, 1996.