

Interconnect Modeling for Improved System-Level Design Optimization

Luca Carloni[‡], Andrew B. Kahng[†], Swamy Muddu[†], Alessandro Pinto⁺, Kambiz Samadi[†], Puneet Sharma[†]

[‡] CS Department, Columbia University, New York, NY

[†] ECE Department, University of California San Diego, La Jolla, CA

⁺ EECS Department, University of California Berkeley, Berkeley, CA

Email: luca@cs.columbia.edu, {abk,smuddu,ksamadi,sharma}@ucsd.edu, apinto@eecs.berkeley.edu

Abstract

Accurate modeling of delay, power, and area of interconnections early in the design phase is crucial for effective system-level optimization. Models presently used in system-level optimizations, such as network-on-chip (NoC) synthesis, are inaccurate in the presence of deep-submicron effects. In this paper, we propose new, highly accurate models for delay and power in buffered interconnects; these models are usable by system-level designers for existing and future technologies. We present a general and transferable methodology to construct our models from a wide variety of reliable sources (Liberty, LEF/ITF, ITRS, PTM, etc.). The modeling infrastructure, and a number of characterized technologies, are available as open source. Our models comprehend key interconnect circuit and layout design styles, and a power-efficient buffering technique that overcomes unrealities of previous delay-driven buffering techniques. We show that our models are significantly more accurate than previous models for global and intermediate buffered interconnects in 90nm and 65nm foundry processes - essentially matching signoff analyses. We also integrate our models in the COSI-OCC synthesis tool and show that the more accurate modeling significantly affects optimal/achievable architectures that are synthesized by the tool. The increased accuracy provided by our models enables system-level designers to obtain better assessments of the achievable performance/power/area tradeoffs for (communication-centric aspects of) system design, with negligible setup and overhead burdens.

1 Introduction

Due to increasing complexity of Systems-on-Chip (SoCs) and poor scaling of interconnects with technology, on-chip communication is becoming a performance bottleneck and a significant consumer of power and area budgets [11, 28]. Decisions made in the early stages of the design process have the maximum potential to optimize the system for objectives such as power [21]. Therefore, in order to drive meaningful optimizations and to reduce guardbanding, it is crucial to account for interconnects during system-level design by modeling their performance, power, and area.

During system design, organizational and technological choices are performed. At this stage, we are concerned with implementing the hardware architecture sketched in the conceptualization and modeling steps. Design is supported by hardware synthesis tools and software compilers. Energy efficiency can be obtained by leveraging the degrees of freedom of the underlying hardware technology. Even within the framework of a standard implementation technology, there are ample possibilities for reducing power consumption. System-level decisions affect primarily the global interconnects by setting their lengths, bit widths, and speed requirements. Local interconnects are typically less affected as they are either already routed in IP blocks or are routed by automatic back-end routing tools.

This paper focuses on interconnect delay, power, and area models that are usable by the system-level designer at an early phase of the design process. We first study the requirements of a system-level designer to model global interconnects and discuss the shortcomings of the models that are presently used. We then describe our mod-

els and present a reproducible methodology to obtain them. Since the accuracy of such models relies on the accuracy of the underlying technology parameters, we also highlight reliable sources that are easily available to the system-level designer for present and future technologies. We compare predictions from our models with existing models and show the impact of improved accuracy on system-level design choices by contrasting the NoC topologies generated by COSI-OCC [19] with the existing models and with our models.

The remainder of the paper is organized as follows. In Section 2 we describe a CAD tool for the automatic synthesis of networks-on-chip. Section 3 investigates the model requirements for such automatic tool. In Section 4 we develop accurate physical models for wires and repeaters, which are building block components of NoCs. In Section 5 we validate the accuracy of our buffered interconnect delay model against industry's golden tool (i.e., PrimeTime SI) and also show the impact of the new models on the optimal NoC configurations that can be achieved with the CAD tool. Finally, Section 6 concludes and gives directions for future work.

2 Communication Synthesis

Packet-switched networks-on-chip (NoC) have been proposed as the solution to the problem of connecting an increasing number of processing cores on the same die [4, 8, 10]. Key steps in the optimization of the NoC design include topology selection and assignment of routes for packets as they travel from a source core to a destination core. Some network design ideas can be borrowed from the computer science community that addressed the same problems for local area networks and supercomputer networks. However, the challenge is leveraging the intrinsic characteristics of on-chip communication to achieve both energy efficiency and high performance [15].

Each target silicon technology offers a variety of possibilities to the NoC designers who, for instance, can decide the number and positions of network access points and routers as well as which metal layer to use for implementing each given channel. Because the design space of the possible topologies is so large, choosing the best one is a difficult problem that cannot be solved only by experience. In fact, the problem is even harder given the heterogeneity of cores, and the traffic patterns among them.

Therefore, the development of automatic tools to design NoC is a key enabler for the success of the NoC design paradigm. COSI-OCC is an open-source software infrastructure for the automatic synthesis of On-Chip Communication (OCC) [19]. Figure 1 shows the design flow implemented in COSI-OCC. The input is a *project* file that contains pointers to the *communication constraint* file and to the *library* file. The constraint file contains the description of the IP cores and the communication constraints among them. An IP core can be manually placed on the chip, thereby having fixed position and dimensions, or it can be characterized by its area only. If there are unplaced IP cores, PARQUET [1] is used to floorplan the chip. An end-to-end communication constraint is defined by a source core, a destination core, a minimum bandwidth and a maximum number of hops.

The library file contains the description of the library elements.

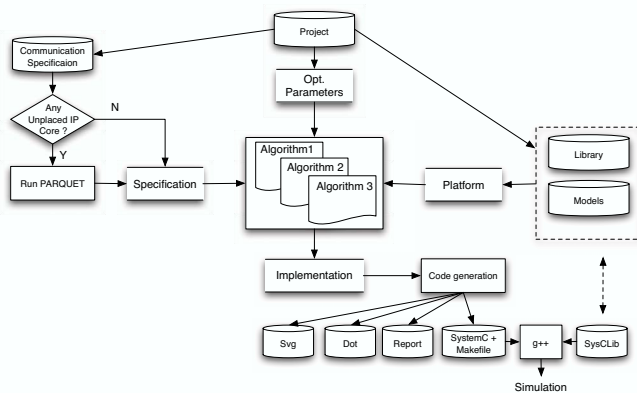


Figure 1: COSI-OCC design flow.

Each element is characterized by a set of architectural parameters (such as flit width, maximum number of input and output ports of a router, etc.) and a model that defines its performance and cost (in terms of area and power). The user can select the appropriate synthesis algorithm to derive an implementation depending on the optimization goal (minimum power, minimum area or minimum delay). The development of new synthesis algorithms is simplified by the simple standard interface with the library. This interface defines an API to retrieve the performance and cost of a component (e.g. a point-to-point link) given its configuration (e.g. clock speed, total bandwidth). Such simple API, available for each component, is of extreme importance to system level designers that are not concerned with low-level technology details.

COSI-OCC provides a set of code generators (SYSTEMC, SVG, DOT) including a textual report of the properties of the communication implementation like power consumption, area, number of hops, total wire-length, number of routers etc.

3 Model Requirements

System-level designers require *accurate, yet simple* models of implementation fabrics (i.e., communicating entities and interconnections between them) in order to bridge planning and implementation, and enable meaningful system design optimization choices. Today, performance and power modeling for system-level optimization suffers from:

- poor definition of inputs required to make design choices;
- *ad hoc* selection of models as well as sources of model inputs;
- lack of model extensibility across multiple/future technologies; and,
- inability to explore different implementation choices and design styles.

In this section, we discuss the accuracy and extensibility of previous models as well as key modeling deficiencies that our work addresses.

3.1 Accuracy

Communication mechanisms between subsystems (such as IP blocks and routers) are realized using high-speed bus structures or point-to-point interconnects. The delay and bandwidth envelope of such interconnects is defined by optimally buffered structures, and must be accurately modeled to enable synthesis of optimal (i.e., minimum-latency or minimum-power) communication topologies. Just as with technology mapping in logic synthesis, on-chip communication synthesis is driven by models of latency and power consumption. The accuracy of such models should be comparable to that available during physical synthesis due to the high sensitivity of design outcomes. For example, poor models of interconnect latency can increase hop

count and introduce unnecessary routers between communicating blocks; this in turn can increase chip area and power consumption.

Existing methods for on-chip communication synthesis [18] and analysis [10] primarily use “classic” delay and power models of Bakoglu [3], or more recently of Pamunuwa *et al.* [16]. Popularity of these models with the NoC research community is likely due to the following reasons:

- *Simplicity and ease of use.* Bakoglu’s delay model for buffered interconnect lines [3] is based on lumped approximation of the distributed parasitics of the interconnect. Driver and buffers are modeled as simple voltage sources with series resistances connected to the interconnect load. These approximations make the buffered line delay model amenable to analytical, closed-form representation, and hence adoptable in NoC synthesis flows.
- *First-order accuracy.* Bakoglu and Pamunuwa *et al.* use a simple step voltage source with series resistance to represent a driver/buffer. Interconnect load is lumped at the output of the cell to compute cell delay. Interconnect delay is computed as Elmore delay, i.e., the first moment of the impulse response of the distributed RC line. These models of buffers and wires are accurate to first order and capture significant contributors to delay.
- *Inertia.* There have not been any compelling reasons to use alternative, more accurate models. To this point, our present work shows that accurate models can still be simple, and that different optimization results and trends follow from use of improved models.

The remainder of this subsection lists key factors that are not addressed by existing delay models. In 180nm and below process nodes, these factors lead to inaccuracy in delay (latency) computation.

Transition time (slew) dependence. The simple cell delay model (step voltage source with constant series resistance) no longer captures delay impact. A finite input slew rate changes the drive resistance and, consequently, cell delay as well as the output voltage waveform that drives other cells. To the best of our knowledge, none of the delay models used in NoC literature consider the impact of slew on delay.

Interconnect resistivity. Resistance affects interconnect delay directly and it shields the load capacitance experienced by driving buffers. As the interconnect dimensions continue to scale, electron scattering has started to affect the resistivity. Copper interconnect manufacturing requires use of a barrier layer that reduces the effective width of the metal. Existing delay models ignore these two effects and sacrifice considerable accuracy.

Coupling capacitance. Crosstalk from capacitive coupling affects signal transition times and delay along interconnects. Classic models such as Bakoglu’s do not consider coupling between neighboring interconnects, and hence are oblivious to resulting delay and transition time changes. Pamunuwa *et al.* consider the impact of switching activity on the ‘Miller’ coupling between neighboring lines and hence delay, but fails to model the impact on transition time. This leads to inaccurate delay computation for cells driven by the affected signal.

The aforementioned deficiencies in gate and wire delay models are addressed to some extent in the large body of work on gate delay [2, 9] and interconnect delay [17, 22] modeling. However, such models (e.g., AWE-based approaches [22]) need detailed interconnect parasitic information which is unavailable at the system-level design phase. For gate delay, works such as that of Arunachalam *et al.* [2] model input voltage as a piecewise-linear function and choose the value of series resistance more elaborately. The main drawbacks of such approaches is that they model drive resistance independent of input transition time (slew). In reality, drive resistance (R_d) varies with input slew. This also affects output slew. Shao *et al.* [25] recently proposed a gate delay model that relies on a second-order RC model of the gate. They propose analytical formulas for computing

the output voltage waveform for a given ramp input waveform. However, they do not address gate loading during model construction. For a gate delay model to be accurate, drive resistance dependence on input slew, and output slew dependence on load capacitance and input slew, must both be considered.

3.2 Design Styles and Buffering Schemes

System-level designers usually ignore design-level degrees of freedom such as wire shielding, wire width and spacing perturbation, etc. when modeling interconnect latency and power. Yet, optimizations of design styles or buffering schemes can have huge impact on the envelope of achievable system performance. For example, shielding an interconnect with quiescent lines on both sides reduces worst-case capacitive coupling and improves delay. Wire width sizing and spacing also improve delay. In addition to design style choices, the buffering objective can also be significant. Interconnect delay models of Bakoglu and Pamunuwa *et al.* incorporate buffering schemes that minimize end-to-end delay (min-delay buffering), and are used extensively in the NoC literature. However, min-delay buffering can result in unrealistically large buffer sizes, and high dynamic and leakage power. It is necessary for system-level design optimization to comprehend power-aware buffering schemes, and more generally the key circuit-level choices that maximize achievable performance.

3.3 Model Inputs and Technology Capture

Perhaps the most critical gap in existing system-level and NoC optimizations has been the lack of well-defined pathways to capture necessary technology and device parameters from the wide range of available sources. Since exploration of the system-level performance and power envelope is typically done for current and future technologies, the models driving system-level design must be derivable from standard technology files (e.g., Liberty format [14], LEF [13]), as well as extrapolatable models of process (e.g., PTM [20], ITRS [12]). Earlier works on NoC design space exploration and synthesis [18, 27] collect inputs from *ad hoc* sources to drive internal models of performance, power and area. However, the use of non-standard interfaces and data sources can often lead to misleading conclusions that can have significant impact on the final outcome precisely because exploration is being performed at a very-high level. Instead, inputs that accompany system-level models must come from standard sources and be conveyed through standardized interfaces and formats.

4 Buffered Interconnect Model

In this section we describe our models and present a methodology to construct them from reliable and easily accessible sources for existing and future technologies. We account for previously-ignored crucial effects such as slew-dependent delay and scattering-dependent wire resistivity change. Our models are by construction calibrated against SPICE and contain well-defined parameters.

4.1 Repeater Delay Model

We now present our repeater delay model and describe its derivation.¹ For brevity, the following study is presented only for rise transitions in inverters implemented in 65nm technology. The derived functional forms are identical for fall transitions, for buffers, and for 90nm technology; only the function coefficients change.

Repeater delay can be decomposed into load independent and load dependent components as follows:

$$d_r = i + r_d \cdot c_l \quad (1)$$

where d_r is the repeater delay and i is the load-independent or intrinsic delay of the gate. $r_d \cdot c_l$ is the load dependent delay term, where r_d is the drive resistance and c_l is the load capacitance.

¹We use the term ‘repeater’ to denote both an inverter and a buffer.

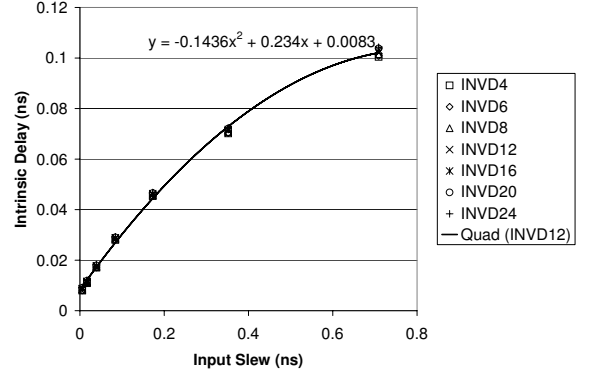


Figure 2: Dependence of repeater intrinsic delay on input slew and inverter size. Intrinsic delay is essentially independent of repeater size, and depends quadratically on input slew.

The **intrinsic delay**, i , can potentially depend on the input slew of the gate and the gate size. However, as seen in Figure 2, i is practically independent of the gate size but depends nearly quadratically on the input slew.²

The quadratic dependence of intrinsic delay on input slew is captured by the equation

$$i(s_i) = \alpha_0 + \alpha_1 \cdot s_i + \alpha_2 \cdot s_i^2 \quad (2)$$

where s_i denotes the input slew and α_0 , α_1 , and α_2 are the coefficients determined by quadratic regression. The dependence of **drive resistance** on input slew has often been ignored [3, 16, 7], but this can contribute to substantial error in delay prediction. Figure 3 shows the dependence of r_d on input slew and repeater size. We observe that r_d is nearly linear with input slew especially for larger input slew values. We also note that both the intercept and slope vary with repeater size; hence, r_d can be written as

$$r_d = r_{d0} + r_{d1} \cdot s_i \quad (3)$$

where r_{d0} and r_{d1} are the coefficients, both of which can depend on the repeater size.

Both r_{d0} and r_{d1} can be readily calculated using linear regression for a few repeater sizes and their dependence on repeater size studied. Previous works (e.g., [3]) have assumed r_d to be inversely proportional to the repeater size. We have confirmed this relationship to be sufficiently accurate for sub-90nm technology modeling. To be precise, we use the PMOS (NMOS) device width as the repeater size for rise (fall) transitions. As shown in Figure 4, both r_{d0} and r_{d1} are linearly proportional to the inverse of the repeater size (i.e., are inversely proportional to the repeater size), and the exact coefficients can be calculated using linear regression with zero intercept³. I.e.,

$$r_{d0}(w_r) = \beta_0 / w_r \quad (4)$$

$$r_{d1}(w_r) = \beta_1 / w_r \quad (5)$$

²The independence of intrinsic delay from gate size can be understood as follows. For inverters, larger sizes are attained by connecting in parallel multiple identical devices (fingers), which switch simultaneously and have negligible impact on each other. As the inverter size increases, the number of parallel-connected devices increases but the intrinsic delay remains unaffected due to the independent switching of the devices. For buffers, the intrinsic delay additionally comprises of the delay of the inverter in the first stage which drives the inverter in the second stage. As the buffer size increases, the size of the second stage inverter increases but the size of the first stage inverter is also increased to maintain small intrinsic delay. Consequently, the total intrinsic delay of buffers is nearly independent of the buffer size.

³All graphs are generated using simple SPICE simulations for a set of input slew values, output capacitance values, and repeater sizes.

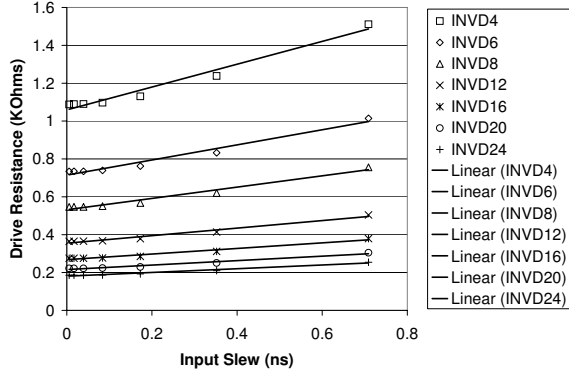


Figure 3: Dependence of drive resistance on input slew and repeater size. Drive resistance depends linearly on the input slew. Both the intercept and the slew are affected by the repeater size.

where w_r is the repeater size and is equal to the PMOS (NMOS) width for rise (fall) transitions, and β_0 and β_1 are the fitted coefficients.

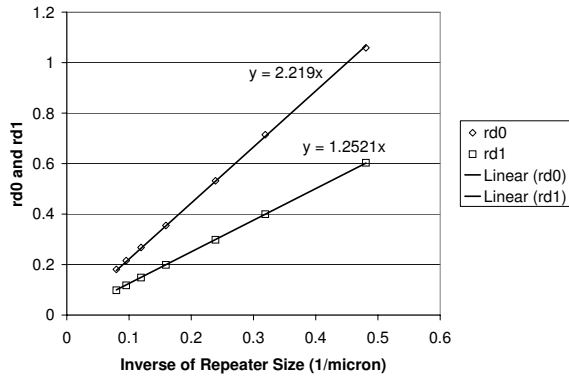


Figure 4: Coefficients r_{d0} and r_{d1} vary linearly with the inverse of the repeater size with zero intercept.

Repeater Output Slew Model

Since our gate delay model depends on input slew, we must also model output slew of the previous stage of the buffered interconnect. Slew is not a crucial metric at the system level, and its only use arises in delay calculation. Furthermore, while repeater delay depends on slew, inaccuracies arising in slew estimation tend to be masked in delay calculation. As a result, accuracy requirements for the slew model are less stringent than those for the delay model.

As with gate delay, slew depends on repeater size, input slew, and load capacitance. Figure 5 shows the dependence of output slew on load capacitance and input slew. Slew depends strongly on the load capacitance, and we have found a linear relationship to be a good tradeoff between simplicity and accuracy. We note that the slope is nearly independent from the input slew, while the intercept is dependent on it. Assuming that the intercept depends linearly on the input slew, the output slew for a given repeater can be written as

$$s_o(c_l, s_i) = s_{o0} + s_{o1} \cdot s_i + s_{o2} \cdot c_l \quad (6)$$

where s_o is the output slew, and s_{o0} , s_{o1} , and s_{o2} are the fitting coefficients readily derived from multiple linear regressions.

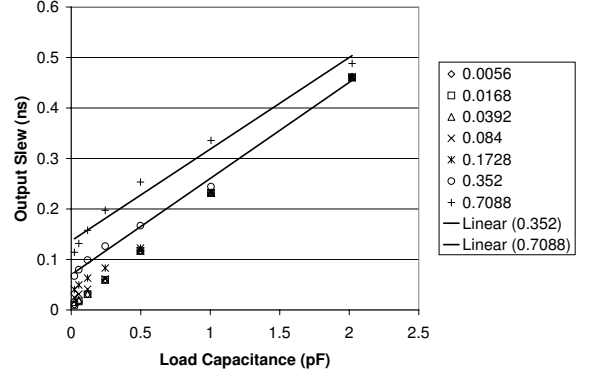


Figure 5: Dependence of output slew on load capacitance and input slew. Output slew depends nearly linearly on load capacitance. The slope of the linear fit is nearly independent of the input slew, but the intercept depends on it.

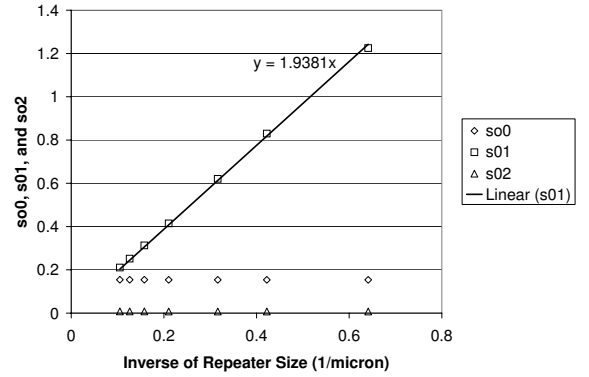


Figure 6: Dependence of coefficients s_{o0} , s_{o1} and s_{o2} on inverse of repeater size. s_{o0} and s_{o2} are independent of repeater size, while s_{o1} varies inversely with repeater size.

The impact of repeater size on the coefficients s_{o0} , s_{o1} , and s_{o2} is shown in Figure 6. We consistently observe that s_{o0} and s_{o2} are independent of the repeater size, but s_{o1} varies inversely with repeater size. Hence, output slew can be calculated as

$$s_o(c_l, s_i, w_r) = \gamma_0 + \frac{\gamma_1 \cdot c_l}{w_r} + \gamma_2 \cdot s_i \quad (7)$$

where γ_0 , γ_1 , and γ_2 are constants.

Repeater Input Capacitance Model

The input capacitance of a repeater is required to calculate the load capacitance of the previous stage. As expected, the input capacitance is proportional to the repeater size. Typically, the P/N ratio is kept constant for repeaters of all sizes and the previous models (e.g., [3]) are sufficient. However, even if the P/N ratio changes with repeater size, input capacitance can be modeled as

$$c_i = \eta \times (w_p + w_n) \quad (8)$$

where c_i is the input capacitance, w_p and w_n are PMOS and NMOS widths respectively, and η is a coefficient derived using linear regression with zero intercept.

4.2 Wire Delay Model

For wire delay we use the model proposed by Pamunuwa *et al.* [16] which accounts for crosstalk-induced delay:

$$d_w = r_w (0.4c_g + \frac{\lambda_i}{2} c_c + 0.7c_i) \quad (9)$$

where d_w , r_w , c_g , c_c and c_i respectively denote wire delay, wire resistance, ground capacitance, coupling capacitance, and input capacitance of the next-stage repeater. λ_r is a coefficient due to switching patterns of the neighboring wires and is equal to 1.51 for worst-case switching. We enhance the quality of the wire delay model by considering two important factors that affect wire resistance:

- *Scattering-aware resistivity.* The rapid rise of wire resistivity due to electron scattering effects (grain boundaries and interfaces) at small cross-sections poses a critical challenge for on-chip interconnect delay. For 65nm and beyond, scattering can degrade delay by up to 70% [26] and must be accounted in delay modeling. We adopt the following closed-form width-dependent resistivity equation [26]:

$$\rho(w) = \rho_B + \frac{K_\rho}{w_w} \quad (10)$$

where w_w is the wire width, $\rho_B = 2.202 \mu\Omega \cdot cm$, and $K_\rho = 1.030 \times 10^{-15} \Omega m^2$. The above model has been verified against measurement data from [23] and is used in the 2004 ITRS [12].

- *Interconnect Barrier.* To prevent copper from diffusing into surrounding oxide, a thin barrier layer is added to three sides of a wire. This barrier affects the wire resistance calculation as [11]:

$$r_w = \frac{\rho \cdot l_w}{(t_m - t_b)(w_w - 2t_b)} \quad (11)$$

where t_m and t_b respectively denote the metal and barrier thicknesses, l_w is length of the wire, and ρ is computed using Equation (10).

4.3 Power Models

Power is a first-class design objective and must be modeled early in the design flow [21]. In current technologies, leakage and dynamic power are the primary forms of power dissipation, and we develop models for them. In repeaters, leakage occurs in both output states. NMOS devices leak when the output is high, while PMOS devices leak when the output is low. This is applicable for buffers also because the second stage devices are the primary contributors due to their large sizes. Leakage power has two main components: (1) sub-threshold leakage, and (2) gate-tunneling current. Both components depend linearly on device size. Thus, leakage can be calculated using:

$$p_s = \frac{p_s^n + p_s^p}{2} \quad (12)$$

$$p_s^n = \kappa_0^n + \kappa_1^n \cdot w_n \quad (13)$$

$$p_s^p = \kappa_0^p + \kappa_1^p \cdot w_p \quad (14)$$

where p_s^n and p_s^p are the leakage for NMOS and PMOS devices, respectively, and κ_0^n , κ_1^n , κ_0^p and κ_1^p are coefficients determined using linear regression. State-dependent leakage modeling can also be performed using Equations (13) and (14) separately.

In present and future technologies, the dynamic power of devices is primarily due to charging and discharging of capacitive loads (wire and input capacitance of next-stage repeater). Internal power dissipation, arising from charging and discharging of internal capacitances and short-circuit power, is noticeable for repeaters only when the input slews are extremely large. Dynamic power is given by the well-known equations:

$$p_d = a \cdot c_l \cdot v_{dd}^2 \cdot f \quad (15)$$

$$c_l = c_i + c_g + c_c \quad (16)$$

where p_d , a , c_l , v_{dd} and f respectively denote the dynamic power, activity factor, load capacitance, supply voltage, and frequency. The load capacitance is the sum of the input capacitance of the next repeater, c_i , and the ground (c_g) and coupling (c_c) capacitances of the wire driven.

4.4 Area Models

Since repeaters are composed of several fingered devices connected in parallel, repeater area grows linearly with the repeater size. For existing technologies, the area can be calculated as

$$a_r = \tau_0 + \tau_1 \cdot w_n \quad (17)$$

where a_r denotes repeater area, and τ_0 and τ_1 are coefficients found using linear regression. For future technologies, area values may not be available for performing linear regression. Hence, we propose the use of feature size, contacted pitch, and row height - all of which become available early in process and library development and are also predictable - to estimate area as:

$$NF = (w_p + w_n + 2 \cdot F) / RH \quad (18)$$

$$RW = NF \times (F + CP) + CP \quad (19)$$

$$a_r = RH \times RW \quad (20)$$

where NF is the calculated number of fingers, F is the feature size, RH is the row height, RW is the calculated row width, and CP is the contacted pitch.

The area of global wiring can be calculated as

$$a_w = n \times (w_w + s_w) + s_w \quad (21)$$

where a_w denotes the wire area, n is the bit width of the bus, and w_w and s_w are the wire width and spacing computed from the width and spacing of the layer (global or intermediate) on which the wire is routed, and from the design style.

4.5 Overall Modeling Methodology

Our delay, power, and area models can be mathematically derived from the following inputs.

- For *repeater delay calculation*, delay and slew values for a set of input slew and load capacitance values, along with input capacitance values, are required for a few repeaters. Since the coefficients are derived using regression, a larger data set improves accuracy. The required data set is available from Liberty/TLF library files or can be generated using SPICE simulations for existing technologies. Since libraries are not available for future technologies, SPICE simulations must be used along with SPICE netlists for repeaters and predictive device models such as PTM [20]. To construct the repeater netlists, a PMOS/NMOS ratio is assumed (from previous technology experience or from expected PMOS/NMOS drive strengths, and is kept constant for all repeaters), and a variety of repeaters are constructed for different device sizes.
- For *wire delay calculation*, we require the wire dimensions and inter-wire spacings for global and intermediate layers. These values are available in LEF (lateral dimensions) and ITF (vertical dimensions) files for existing technologies, and in the ITRS for future and existing technologies.
- For *power calculations*, input capacitance (computed in repeater delay calculation) and wire parasitics (computed in wire delay calculation) are used. Additionally, device leakage is required and can be computed from the Liberty/TLF library files or SPICE simulations.

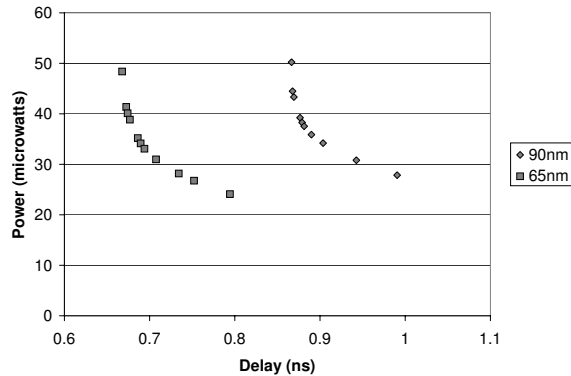


Figure 7: Pareto-optimal frontier of the delay-power tradeoff for a 5mm buffered interconnect in 90nm and 65nm technologies.

- For *area calculations*, wire dimensions used in wire delay calculation are used for wire area. Repeater area is readily available for existing technologies in Liberty or LEF files or from layouts. For future technologies, ITRS A-factors can be used or Equations 18-20 can be used along with the feature size, row height, and contacted pitch, all of which values are available early in process and library development.

Finally, the total delay of a buffered interconnect is the sum of the delays of all repeaters and wire segments in it. We assume that there is negligible slew degradation and resistive shielding (of capacitive load) due to the wires. Table 3 lists the coefficients derived for TSMC 90nm and 65nm high-speed technologies.

4.6 Interconnect Optimization

Delay-optimal buffering optimizes the size and number of repeaters, and has been addressed under simple delay models in, e.g., [3, 16, 7]. However, delay-optimal buffering results in extremely large repeaters having sizes that are never used in practice due to area and power consumption considerations. Cao *et al.* [5] showed that use of smaller buffers improves the energy-delay product significantly while only marginally worsening delay.

While previously-proposed closed-form optimal buffering solutions are efficient to compute, they are difficult to adapt to more complex and accurate delay models such as ours. Furthermore, hybrid objective functions that optimize delay, power, and area are even more difficult to handle. With this in mind, we have developed an iterative optimization technique that evaluates a given objective function for a given number and size of repeaters, while searching for the optimal (number, size) values. We have found that realistic objective functions are convex, making binary search for the optimal repeater size feasible.

Our iterative optimization is easily extensible to other interconnect optimizations such as wire sizing and wire spacing, but the runtime grows exponentially with the number of optimization knobs. In general, wire sizing and wire spacing are weaker optimization knobs and their effect at the system-level can be ignored. We optimize only the number and size of repeaters during interconnect optimization. However, we support the use of double-width and double-spacing design styles which the system designer can invoke to optimize interconnect area, delay, noise, and power.

Figure 7 shows the Pareto-optimal delay-power tradeoff for a 5mm global buffered wire in 90nm and 65nm technologies. We note that for both technologies, power can be reduced by 20% at the cost of under 2% degradation in delay.

4.7 Publicly-Available Framework

Finally, we have developed a framework [29] capable of modeling and optimizing buffered interconnects for various technologies and under different design styles. The framework is accessible through XML files or through a C++ API. We have packaged the framework with XML files corresponding to a number of future and existing technologies corresponding to commercial foundry processes (90nm and 65nm), ITRS, and PTM.

5 Validation and Significance Assessment

We now assess the accuracy of our model and compare it with that of previously-proposed models ([3] and [16]). We perform the accuracy comparison on a 5mm long buffered interconnect for two technology choices (90nm and 65nm), two routing layer regimes (global and intermediate), two design styles (single-width-single-spacing and single-width-double-spacing). Since delay is linear with length for buffered interconnects, a length of 5mm is representative of other lengths that require buffering.

To create the layout of a 5mm long buffered interconnect, we first create the layout to define the chip area using *Cadence SoC Encounter (version 6.1)*. Repeaters are then placed at equal distances along the length to buffer the interconnect uniformly. Connections between inputs, outputs and the buffers are created by *Cadence NanoRoute*. The values of minimum wire spacing and wire width are chosen from the input LEF file. Parasitic extraction on the buffered lines is performed using *SoC Encounter's* built-in extractor. To perform timing analysis, we read in the parasitics output from *SoC Encounter* in SPEF format and the timing library (Liberty format) into *PrimeTime (version 2006.12)* for signoff delay calculation. We use TSMC 90nm and 65nm Liberty and technology files in our experimental setup. Results of our accuracy studies are presented in Table 1. Column 4 shows the delay of the buffered line evaluated in PrimeTime (input transition time = 300ps). Columns 5, 6, and 7 show the error in delay prediction from Bakoglu, Pamunuwa and our model respectively. From the table we can observe that the delay from our proposed method matches that from PrimeTime within 15%. In comparison, previous models have error in the range of -65% to 160%.

To assess the impact of improved accuracy on system-level design-space exploration, we integrate our models in COSI-OCC. We use two representative SoC designs as test cases. The first design (VPROC) is a video processor with 42 cores and 128-bit datawidths. The second design is based on a dual video object plane decoder (dVOPD), where two video streams are decoded in parallel by utilizing 26 cores and 128-bit datawidths. Table 2 compares the interconnect power, delay, and area when the original [19] and proposed models are used. The clock frequencies used are 1.5 GHz and 2.25 GHz for 90nm, 65nm technology nodes, respectively. Hop count, which captures the communication latency, is also shown. The main differences between the NoC obtained using the original and the proposed models are in the area and hop-count. The critical sequential length, i.e. the maximum distance that a signal can travel in an optimally-sized and optimally-buffered interconnect within a single clock period [24], that was computed with the original model turns out to be very optimistic, allowing the use of excessively long wires. This is the case of a non-conservative abstraction that leads to design solutions that are actually not implementable. We note that the difference in area estimates between the original and proposed models is very large because of the simplistic area modeling in the original models.

6 Conclusions

Accurate estimation of delay, power, and area of interconnections early in the design phase can drive effective system-level exploration. Existing models of buffered interconnections are inaccurate for current and future technologies (due to deep-submicron effects) and can

Table 1: Comparison of accuracy of Bakoglu [3], Pamunuwa [16], and the proposed models with respect to PrimeTime.

Tech. Node	Layer	Design Style	PrimeTime (ns)	Bakoglu (%)	Pamunuwa (%)	Proposed (%)
90nm	Global	SW-SS	0.670	97.0	66.4	-8.2
		SW-DS	0.515	120.4	49.5	-9.7
	Intermediate	SW-SS	0.620	9.7	160.5	0.8
		SW-DS	0.650	-6.2	60.0	6.9
65nm	Global	SW-SS	0.505	-6.9	33.7	-5.0
		SW-DS	0.395	6.3	31.6	1.3
	Intermediate	SW-SS	0.735	-65.3	42.2	10.9
		SW-DS	0.505	-52.5	42.6	14.9

Table 2: Comparison of dynamic power (P_{dyn}), static power (P_{leak}), device area (A_d) and total area (A_{tot}) metrics relative to wires, and number of hops, between the original and proposed models.

SoC		P_{dyn} (mW)		P_{leak} (mW)		A_d (mm ²)		A_{tot} (mm ²)		Ave. # of hops		Max. # of hops	
		Orig.	Prop.	Orig.	Prop.	Orig.	Prop.	Orig.	Prop.	Orig.	Prop.	Orig.	Prop.
VPROC	90nm	117.3	364.8	38.1	99.6	0.070	0.009	0.370	0.346	3.09	3.01	4	5
	65nm	51.1	179.9	69.9	86.7	0.036	0.007	0.217	0.223	3.10	3.42	4	6
dVOPD	90nm	63.4	88.0	14.2	32.5	0.026	0.003	0.141	0.162	1.76	1.76	3	3
	65nm	27.3	73.2	25.7	33.2	0.013	0.003	0.082	0.085	1.76	1.91	3	4

Table 3: Coefficients for our model derived from TSMC 90nm and 65nm technologies. α , β , and γ are for the rise transition.

	α_0	α_1	α_2	β_0	β_1
90nm	0.013	0.217	-0.088	3.008	1.494
65nm	0.008	0.234	-0.144	2.219	1.252
	γ_0	γ_1	γ_2	η	κ_0^r
90nm	0.015	5.553	0.128	0.0015	-6.128
65nm	0.012	4.162	0.142	0.0011	-6.034
	κ_1^r	κ_0^d	κ_1^d	τ_0	τ_1
90nm	29.313	1.261	13.274	1.312	1.099
65nm	26.561	1.238	27.082	0.657	0.866

lead to misleading design targets. We have proposed accurate models for buffered interconnects that are easily usable by system-level designers. We have presented a reproducible methodology for extracting inputs to our models from reliable sources. Existing delay-driven buffering techniques minimize interconnect delay without any consideration for power and area impact. This can often result in buffered interconnects that are infeasible during implementation. We have proposed a power-efficient buffering technique that minimizes total power with minimal delay impact.

To demonstrate the accuracy of our model, we evaluated its delay prediction for buffered interconnects in global and intermediate wiring layers across 90nm and 65nm technologies. Our results showed that delay from our proposed model matches that from a commercial sign-off tool within 15%. We integrated our model in an NoC topology synthesis tool (COSI-OCC) and found that accurate models substantially affect the explored topology solution.

Our future work is in two directions: (1) development of models for higher-level communication architectures such as NoC's and AMBA, and (2) extension of modeling to other upcoming metrics such as variability and reliability.

7 Acknowledgment

This research is partially supported by the GSRC Focus Center, one of the five research centers funded under the Focus Center Research Program, a Semiconductor Research Corporation program.

References

- [1] S. N. Adya and I. L. Markov, "Fixed-Outline Floorplanning: Enabling Hierarchical Design", *IEEE Trans. VLSI Systems*, 11(2), 2003, pp. 1120-1135.
- [2] R. Arunachalam, F. Dartu and L. Pileggi, "CMOS Gate Delay Models for General RLC Loading", *Proc. ICCD*, 1997, pp. 224-229.
- [3] H. Bakoglu, *Circuits, Interconnections and Packaging for VLSI*, Addison-Wesley, 1990.

- [4] L. Benini and G. D. Micheli, "A New SoC Paradigm", *IEEE Computer*, 35(1), 2002, pp. 70-78.
- [5] Y. Cao, C. M. Hu, X. J. Huang, A. B. Kahng, S. Muddu, D. Stroobandt and D. Sylvester, "Effects of Global Interconnect Optimizations on Performance Estimation of Deep Submicron Design", *Proc. IEEE ICCAD*, 2000, pp. 56-61.
- [6] L. P. Carloni, R. Passerone, A. Pinto and A. L. Sangiovanni-Vincentelli, "Languages and Tools for Hybrid Systems Design", *Foundations and Trends in Electronic Design Automation*, 1, 2006, pp. 1-194.
- [7] J. Cong and D. Z. Pan, "Interconnect Delay Estimation Models for Synthesis and Design Planning", *Proc. IEEE ASPDAC*, 1999, pp. 507-510.
- [8] W. J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks", *Proc. ACM/IEEE DAC*, 2001, pp. 684-689.
- [9] F. Dartu, N. Menezes and L. Pileggi, "Performance Computation for Precharacterized CMOS Gate with RC Load", *IEEE Trans. on CAD*, 1996, pp. 544-553.
- [10] S. Heo and K. Asanovic, "Replacing Global Wires With an On-Chip Network: A Power Analysis", *Proc. ISLPED*, 2005, pp. 369-374.
- [11] K. W. Mai, R. Ho and M. A. Horowitz, "The Future of Wires", *Proc. IEEE*, 2001, pp. 490-504.
- [12] *International Technology Roadmap for Semiconductors*, <http://www.itrs.net>.
- [13] *LEF/DEF Exchange Format*, <http://openeda.si2.org/projects/lefdef>.
- [14] *Liberty File Format*, <http://www.synopsys.com/products/libertyccs/libertyccs.html>.
- [15] G. D. Micheli and L. Benini, *Networks on Chip*, Morgan Kaufmann, 2006.
- [16] D. Pamunuwa, L.-R. Zheng and H. Tenhunen, "Maximizing Throughput over Parallel Wire Structures in the Deep Submicrometer Regime", *IEEE Trans. on VLSI Systems* 11, 2003, pp. 224-243.
- [17] L. Pillage and R. Rohrer, "Asymptotic Waveform Evaluation for Timing Analysis", *IEEE Trans. on CAD* 9, 1990, pp. 352-366.
- [18] A. Pinto, A. Bonivento, A. Sangiovanni-Vincentelli, R. Passerone and M. Sgroi, "System Level Design Paradigms: Platform-Based Design and Communication Synthesis", *ACM TODAES*, 11(3), 2006, pp. 537-563.
- [19] A. Pinto, L. P. Carloni and A. L. Sangiovanni-Vincentelli, "A Methodology and an Open Software Infrastructure for Constraint-Driven Synthesis of On-Chip Communications", *Technical Report UCB/EECS-2007-130*, Nov. 2007.
- [20] *Predictive Technology Model*, <http://www.eas.asu.edu/~ptm/>
- [21] A. Raghunathan, N. K. Niraj and S. Dey, *High-Level Power Analysis and Optimization*, Kluwer, 1998.
- [22] C. Ratzlaff and L. Pillage, "RICE: Rapid Interconnect Circuit Evaluation using AWE", *IEEE Trans. on CAD* 13, 1994, pp. 763-776.
- [23] S. M. Rosnsagel and T. S. Kuan, "Alteration of Cu Conductivity in The Size Effect Regime", *J. Vacuum Science Tech. B* 22, 2004, pp. 240-247.
- [24] P. Saxena, N. Menezes, P. Cocchini and D. A. Kirkpatrick, "Repeater Scaling and its Impact on CAD", *IEEE Trans. on CAD* 23, 2004, pp. 451-462.
- [25] M. Shao, M. Wong, H. Cao, Y. Gao, L.-P. Yuan, L.-D. Huang and S. Lee, "Explicit Gate Delay Model for Timing Evaluation", *Proc. ACM/IEEE ISPD*, 2003, pp. 32-38.
- [26] S. X. Shi and D. Z. Pan, "Wire Sizing and Shaping with Scattering Effect for Nanoscale Interconnection", *Proc. IEEE ASPDAC*, 2006, pp. 503-508.
- [27] V. Soteriou, N. Easley, H. Wang and L.-S. Peh, "Polaris: A System-Level Roadmap for On-Chip Interconnection Networks", *Proc. ICCD*, 2006, pp. 134-142.
- [28] D. Sylvester and K. Keutzer, "A Global Wiring Paradigm for Deep Submicron Design", *IEEE Trans. on CAD* 19, 2000, pp. 242-252.
- [29] <http://vlsicad.ucsd.edu/GSRC/>.