

# INTACTE: An Interconnect Area, Delay, and Energy Estimation Tool for Microarchitectural Explorations \*

Rahul Nagpal  
Department of Computer  
Science and Automation  
Indian Institute of Science  
Bangalore, India  
rahul@csa.iisc.ernet.in

Amrutur Bhardwaj  
Department of Electrical  
Communication Engineering  
Indian Institute of Science  
Bangalore, India  
amrutur@ece.iisc.ernet.in

Arvind Madan  
Department of Electrical  
Communication Engineering  
Indian Institute of Science  
Bangalore, India  
marvind@ece.iisc.ernet.in

Y. N. Srikant  
Department of Computer  
Science and Automation  
Indian Institute of Science  
Bangalore, India  
srikant@csa.iisc.ernet.in

## ABSTRACT

Prior work on modeling interconnects has focused on optimizing the wire and repeater design for trading off energy and delay, and is largely based on low level circuit parameters. Hence these models are hard to use directly to make high level microarchitectural trade-offs in the initial exploration phase of a design. In this paper, we propose INTACTE, a tool that can be used by architects to get reasonably accurate interconnect area, delay, and power estimates based on a few architecture level parameters for the interconnect such as length, width (in number of bits), frequency, and latency for a specified technology and voltage.

The tool uses well known models of interconnect delay and energy taking into account the wire pitch, repeater size, and spacing for a range of voltages and technologies. It then solves an optimization problem of finding the lowest energy interconnect design in terms of the low level circuit parameters, which meets the architectural constraints given as inputs. In addition, the tool also provides the area, energy, and delay for a range of supply voltages and degrees of pipelining, which can be used for micro-architectural exploration of a chip. The delay and energy models used by the tool have been validated against low level circuit simulations. We discuss several potential applications of the tool and present an example of optimizing interconnect design in the context of clustered VLIW architectures.

---

\*This research was supported in part by DRDO-CAIR (Center for Artificial Intelligence and Robotics), India

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CASES'07, September 30–October 3, 2007, Salzburg, Austria.  
Copyright 2007 ACM 978-1-59593-826-8/07/0009 ...\$5.00.

## Categories and Subject Descriptors

B.4.3 [Input/Output and data communications]: Interconnections (Subsystems)-Physical structures;Topology

## General Terms

Algorithms, Design, Measurement, Performance

## Keywords

Interconnect, Energy Modeling, Clustered VLIW Processors, Energy-Aware Scheduling

## 1. INTRODUCTION

Emergence of multi-core architectures reinforces the trend that distribution is the only way to scale in the current and future technologies[1][35][13][4]. In Embedded domain, trend towards using fine grained distribution to achieve scalability has been visible for quite some time[17][19][9]. Multicore architectures take the idea of *scalability by distribution* even further. Though, this trend towards distributed architectures has entered the mainstream computing only recently, embedded chips have been using clustering and even multiple cores (especially in DSPs powering mobile phones[7]) for quite some time. All the major embedded chip manufacturers are designing their next generation architectures exclusively based on multi-core philosophy[7][1].

On-chip interconnect for communication among spatially separate resources introduces major performance, area, and energy bottlenecks for both fine-grained and coarse-grained distributed architectures. It has been observed that interconnects can easily consume power equivalent to one core, area equivalent to three cores, and delay that account for over half the L2 access latency[25]. Even for non-multi-core distributed architectures (such as clustered superscalar and clustered VLIW), interconnects consume significantly high energy and area and are known to be a major source of performance bottlenecks[11]. [25] clearly demonstrates that design trade-offs made considering the interconnect as an independent entity can often be quite opposite to the

design trade-offs that are optimal from power and performance point of view. Co-designing interconnects early along with other components when high level architectural design trade-offs are being made is highly desirable for high level synthesis of embedded SoCs.

In order to quantitatively evaluate different interconnect design trade-offs, one needs a reasonably accurate and fast model for the area, delay and power for these choices. Prior research in interconnect modeling and analysis has mostly dealt with specific circuit level issues [29][39][21] and is not directly usable to make high level micro-architectural trade-offs. For example, an architect would be interested in knowing what are the available trade-offs in terms of pipeline latency and power, for a given bandwidth and interconnect distance. This information could be used at a higher level of design to obtain the overall optimum for the system. Similarly, it will be very useful to know the power and performance of the interconnect at different operating voltages and frequencies, in order to evaluate dynamic voltage and frequency scaling schemes. Hence, there is a need for a tool for the interconnect, which can give reasonably accurate design points and their associated area and power costs for various architecture level constraints like bandwidth, latency etc. Similar models are available for caches[43], register files[38] and functional components[16]. Availability of an interconnect model will be very helpful for architects to involve interconnect in early design trade-offs.

This paper proposes an interconnect modeling tool to get fast but reasonably accurate estimates of interconnect delay, area, and power for a given technology, wire length, bit-width, clock frequency and latency. The tool solves an optimization problem of minimizing power by finding the appropriate wire size, repeater size and repeater spacing for varying degrees of pipelining and area. We are currently limiting our work to cover point-to-point interconnects only, as most high performance long distance interconnects will be of this form[14]. The tool outputs a set of interconnect designs for a cross section of area and degrees of pipelining, all of which meet the frequency and latency constraints. In addition, for each design a set of power and performance numbers are also given across a range of supply voltages. These choices enable the user to explore the micro-architecture design space for the system which includes this interconnect.

The area, delay, and power estimation for the interconnect is built upon the corresponding values for the low level component such as wires, repeaters, flops and buffers, which are in turn obtained via accurate HSPICE[3] characterization. However, this one time characterization is done in advance, and hence the tool itself is fast enough to explore many interconnect choices rapidly. Furthermore, the power model is parameterized with respect to the activity factor (probability of switching of any bit) and the coupling factor (probability of relative switching between adjacent bits). An architect can profile the target workload to get these quantities in order to further improve the accuracy with respect to the target workload. We have also validated the estimates of delay and power obtain by the tool with HSPICE simulation and we found that the error is less than 15% in worst and below 12% on average. Our tool based approach to architectural modeling of interconnect parameters is analogous to that of CACTI[43].

The proposed tool can be used by architects in many different ways. Since different on-chip interconnects have

different performance requirements, the tool can be used to customize the interconnect design to meet these goals at minimum power. The impact of different interconnect choices with latency and power trade-offs can be evaluated at the architectural level in concert with compiler optimizations. We present an example in this regard where we evaluate the energy benefits of heterogeneous interconnects in the context of clustered architectures using the proposed interconnect model. Thus, the tool enables the co-design of interconnects along with other components early in the design phase and its impact on the overall system power and performance can be evaluated upfront. As mentioned earlier this has become very important in new process generations[25]. The major contributions of this work are:

1. A tool which provides estimates of area and power for a power efficient interconnect to meet target bandwidth and latency requirements for a range of technologies. The tool optimizes for power by finding the optimal values of the wire widths, repeater sizes and spacings, which can meet the target bandwidth and latency.
2. For each input requirement, the tool provides a range of design choices with respect to area and degrees of pipelining which can be used by the user to explore micro-architectural trade-offs at the system level. Furthermore, the tool provides estimates of power, bandwidth and latency for a range of voltages, lower than the nominal. This allows characterization of the design for dynamic voltage and frequency scaling.
3. A detailed HSPICE validation of the tool varying different parameters such as length, pitch and technology that confirms that tool has known degree of unidirectional error (15% in worse and below 12% on average).
4. We illustrate two applications of the tool. In one we find the optimum degree of pipelining of the wire which minimizes the overall power (Section 4). In the other application, we use the tool to optimize a heterogeneous interconnect design in the context of clustered architectures (Section 5).

The rest of the paper is organized as follows. We describe the tool and its implementation in Section 2 and the associated delay and power models in Section 3. Section 4 gives the experimental and validation results for the tool. In Section 5, we present an example usage of the interconnect energy model to evaluate the benefits of heterogeneous interconnects in the context of clustered architectures. Section 6 presents discussions on other possible uses and extensions. Section 7 puts our work in the context of existing work followed by our conclusions in Section 8.

## 2. INTACTE TOOL DESCRIPTION

The core motivation behind the tool is to fill the gap between an architect's requirements of the interconnect and what the circuit level interconnect models provide. Figure 1 depicts the tool, its inputs and its outputs. The tool is currently implemented in MATLAB[6] and a port to C language is in progress. The user provides the target technology, wire length, number bits, frequency and latency (in number of cycles). There are a number of other parameters which have default values and can be overridden by the user. The supply

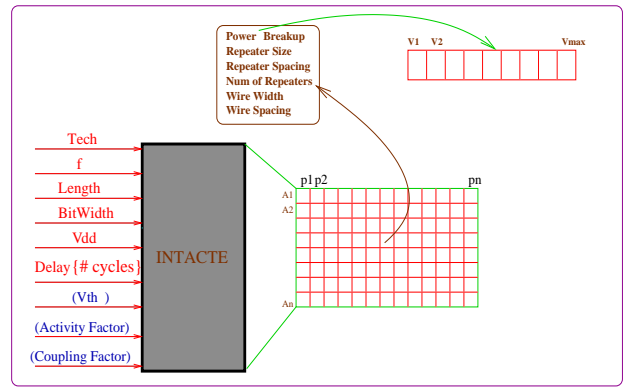
and threshold voltages are automatically derived from the specified technology based on the PTM[8]. Activity factor is the probability of switching of a bit and the coupling factor is probability of relative switching of two adjacent bits. Both can be obtained by profiling the workload to override the default value of 0.5. For long interconnect running at a high frequency, pipelining the interconnect becomes mandatory. Wire length is difficult to obtain accurately during the initial design phase. Estimates can be obtained from a prior design or with some initial rough floor planning and models such as Rents rule[21].

The design variables that the tool considers for the interconnect optimization are as follows:

1. The wire width ( $w$ ) and wire spacing ( $s$ ). Increasing wire width reduces resistance and increasing wire spacing reduces coupling capacitance. Both of these reduce number of repeaters, up to a certain point. Too large a wire width or too small a wire spacing leads to large wire capacitance which is counter productive. Wire width and spacing decides the overall area taken by the interconnect which can be given as an optional constraint by the user or tool work in a loop back manner for set of nominal area values.
2. Repeater Size ( $S$ ) and Spacing ( $l_r$ ). Long wires have to be broken up with periodic repeaters to reduce the impact of the wire resistance. There is an optimal repeater size and spacing for minimum delay. But lower sizes and increased spacing can be used to reduce power while meeting target delay[29].
3. Degree of Pipelining ( $p$ ). Long interconnects will need intermediate flop stages in order to meet the frequency target.
4. Supply ( $V_{dd}$ ) and Threshold Voltage ( $V_{th}$ ). These circuit level parameters can be used to trade off dynamic power and leakage power of the interconnect.

Ideally the tool should find the optimal values for the above variables which will lead to a design with minimum power, while meeting the target performance. Unfortunately, the optimization is very complex to solve as it is a mixed integer nonlinear programming problem. Besides, the analytical formulas relating power and delay to all the design variables are also quite complex. Hence we take a pragmatic approach of a mixed analytical and search technique for finding the optimum values.

The tool explores a limited range of areas and pipeline depths. For any given area, the wire pitch is obtained as the length and number of bits are known. For any pipeline depth, the wire length in any pipeline is obtained by assuming equal pipeline segments. These two calculations result in a smaller optimization problem of finding the optimal wire width, repeater size and spacing for a given wire pitch and unpipelined segment, which meets the target cycle time. This problem is solved by using the well known delay and power models for the repeaters and the wires [29]. We are currently using a built-in optimization function in MATLAB[6] to solve this problem. In addition to considering the activity and coupling factors for dynamic power, we have also considered the leakage power which has been ignored in some of the previous work[21]. We have taken care to include the flop overheads as well as the pre-drivers after



**Figure 1: The INTACTE tool outputs a matrix of values for different areas and degrees of pipelining. Each matrix entry holds the power estimate as well as the design values for the wire width, repeater size and spacing. In addition, for each matrix entry, an additional table is optionally generated which shows the performance and power of that particular design for a range of supply voltages starting from the nominal to a lower value.**

the flop into the timing and power calculations for the unpipelined segments. The delay and power models for the repeaters, buffers, fops and the wires have been calibrated with HSPICE simulations of these components over four different technology nodes using the PTM SPICE models[8]. Once the problem for an unpipelined segment is solved, the total power for the overall interconnect is easily obtained by scaling it by the number of pipe segments. At this level, the flop and clock power are also included. Thus a design which minimizes the power for a given area, length, pipeline depth and target frequency is obtained and this is repeated for a set of areas and pipeline depths. Of course it is also possible to override this iterative behavior to output the results for a specific area and pipeline depth too. Additional information like the breakup of power between different components is also provided which is of interest to a micro-architect. With emerging interest in dynamic voltage and frequency scaling[23][10], it is of interest to see the performance power trade-offs possible in the interconnect. Hence the tool additionally estimates the power and performance (delay of each segment) for a range of supply voltages lower than the nominal value. The other design parameters like width, sizes and spacings are kept the same as that obtained for the nominal value. So in this respect, the power, performance numbers are suboptimal when compared to re-optimizing the design again for specific supply voltages. Nevertheless, these values will be of interest to the architect to evaluate the feasibility and opportunities of dynamic voltage and frequency scaling[10]. One can still obtain optimal design values for any other voltage, by explicitly specifying that voltage, which will then override the default internal voltage value. Thus, the tool allows the architect to choose the best interconnect options that suit their requirements.

The model retains its accuracy because determination of the delay, area, and power are carried out using low level circuit estimation of resistance and capacitance of interconnect components such as wires, repeaters, buffers, and fops using HSPICE[3]. However, these and other technology and

voltage dependent parameters are precomputed for different technology nodes and voltage steps. Thus, the estimation is still fast enough (of the order of seconds) compared to a full blown HSPICE[3] estimation (of the order of hours) attribute to manual work involved in determining low level interconnect parameters. Moreover, pre-estimation of these values for different technology nodes also makes the model capable of providing reasonably accurate estimates for delay, area and power across technologies. We will next briefly go over the detailed models for delay and power used within the tool.

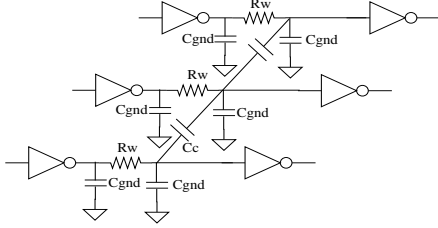


Figure 2: A section of an on-chip bus

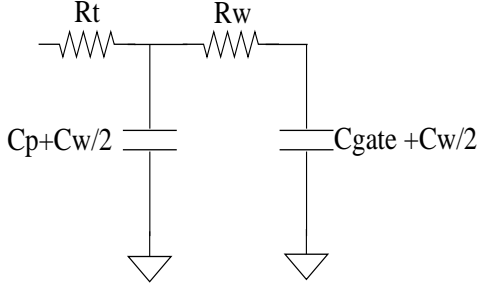


Figure 3:  $\pi$  model of the interconnect

### 3. MODELING THE INTERCONNECT

We consider an interconnect as a set of lines where each line consists of number of pipelined segments. The length of interconnect and the number of lines are given as input by the architect. The architect can also give degree of pipelining as an input or the optimization is performed in an iterative manner for a set of feasible degree of pipelining. The length of a pipelined segment is determined by the length of interconnect and degree of pipelining. Each pipeline segment is made of set of wire segments demarcated by repeaters, a flop, and a set of buffers to drive the first repeater of the pipeline segment. The optimization is essentially performed for a single pipelined segment. The four optimization variables are repeater size, repeater spacing, wire width, and wire spacing. These are varied to obtain a delay that satisfies the latency specified by the architect while minimizing the power. Algorithm 1 gives an outline of the optimization process.

In what follows, we describe how the delay and power of interconnect is characterized in terms of delay and power of a pipeline segment which in turn is determined by delay and power of individual components such as wires, repeaters, flops, and buffers. Fig. 2 shows the schematic of a set of parallel wire segments driven by repeaters at the end. A

repeater is an inverter with equivalent capacitance ( $C_{gate}$ ) at the input, and a series combination of an equivalent resistance ( $R_t$ ) and equivalent capacitance at the output ( $C_p$ ). A wire is modeled as a R-C  $\pi$  section (refer Figure 2). To calculate the power and delay of a wire segment and associated repeaters, all the parasitics such as  $r_t$ ,  $c_p$ ,  $c_{gate}$ ,  $r_w$ , and  $c_w$  are characterized for different technology nodes and voltages as described in Table 1. The power and delay of flops and buffers are calculated by characterizing these values using HSPICE[3] (Refer Table 1 for details).

---

#### Algorithm 1 Outline of Optimization Problem

---

MINIMIZE:

$$P_{total} = BitWidth * p * P_{total}^{seg}$$

WHERE:

$$P_{total}^{seg} = P_{wire}^{seg} + P_{rep}^{seg} + P_{buff}^{seg} + P_{flop}^{seg}$$

$$P_{wire}^{seg} = P_{wire\_dyn}^{seg}$$

$$P_{rep}^{seg} = P_{rep\_dyn}^{seg} + P_{rep\_leak}^{seg} + P_{rep\_short}^{seg}$$

$$P_{buffer}^{seg} = P_{buffer\_dyn}^{seg} + P_{buffer\_leak}^{seg} + P_{buffer\_short}^{seg}$$

$$P_{flop}^{seg} = P_{flop\_dyn}^{seg} + P_{flop\_leak}^{seg} + P_{flop\_short}^{seg}$$

SUBJECT TO:

$$(D_{total} = p * D^{seg}) \leq Delay$$

$$BitWidth * (w + s) \leq W, \quad w \geq 4 * \lambda, \quad s \geq 4 * \lambda$$

WHERE:

$$D^{seg} = D_{wire}^{seg} + D_{rep}^{seg} + D_{buff}^{seg} + D_{flop}^{seg}$$

VARY:

$$rep\_size(S), rep\_space(l_r), wire\_width(w), wire\_space(s)$$


---

### 3.1 Delay Characterization

Delay of a pipeline segment is calculated as sum of the delay of wire segments, repeaters, flops and buffers. A minimum sized flop may not have enough drive strength to drive a repeater at a very high speed. Therefore a series of buffers are introduced such that each stage (including the flop) drives a load of not more than 4 times its size. Thus the number of buffers ( $N_b$ ) is given by  $\lceil (\log(S_r) / \log 4) \rceil$  where  $S_r$  is the ratio of the repeater size to the minimum possible size (i.e.  $4 * \lambda^1$ ) where size of  $i^{th}$  buffer ( $Size_{buff}^i$ ) is  $4^{i-1} * 4 * \lambda$ . The delay equation for an interconnect having p pipelined

<sup>1</sup> $\lambda$  is defined as half the feature size.

**Table 1: Symbols for Various Interconnect components. These Components are characterized for 4 different technology nodes (90,65,45,32) and 32 different voltage steps differing by 15 mV**

$r_t$	Output resistance of 1 $\mu\text{m}$ repeater size <sup>1</sup>
$c_p$	Output capacitance of 1 $\mu\text{m}$ repeater size <sup>1</sup>
$c_{gate}$	Input capacitance of 1 $\mu\text{m}$ repeater size <sup>1</sup>
$r_w$	Resistance of 1 $\mu\text{m}$ wire length <sup>2</sup> .
$c_w$	Capacitance of 1 $\mu\text{m}$ wire length <sup>2</sup> .
$c_{gnd}, c_c, c_f$	Ground, coupling, and fringing capacitance components of $c_w$ . <sup>2</sup>
$D_{flop}$	Delay of min sized (4* $\lambda$ NMOS) flop <sup>1</sup>
$P_{flop\_dy}$	Dynamic power/GHz of min sized flop <sup>1</sup>
$P_{flop\_leak}$	Leakage power of min sized flop <sup>1</sup>
$D_{buff}$	FO4 delay of min sized inverter <sup>1</sup>
$P_{buff\_dy}$	Dynamic power/GHz of min sized inverter <sup>1</sup>
$P_{buff\_leak}$	Leakage power of min sized inverter <sup>1</sup>

segment each of length  $L^{seg}$  and  $n_r$  repeaters (per segment) is determined as follows :

$$D_{total} = p * D^{seg} \quad (1)$$

Equation 2 calculates the delay of a pipelined segment which has four components namely delay of wire, delay of all repeaters in segment, delay of flop at the beginning of pipe segment, and sum of delay of all buffers required to drive the first repeater respectively (refer Table 1. for definitions of symbols) .

$$D^{seg} = (R_t * ((C_p + C_{gate}) * n_r + C_w) + R_w * (C_{gate} * n_r + C_w/2)) + D_{flop} + \sum_{i \in (1..Nb)} D_{buff}^i \quad (2)$$

where

$$R_t = r_t/S, C_p = c_p * S, C_{gate} = c_{gate} * S, C_w = c_w * L_{seg} \text{ and } R_w = r_w * L_{seg}$$

## 3.2 Power Characterization

The total power is determined by multiplying power of a pipeline segment ( $P_{total}^{seg}$ ) with the total number of pipelined segments (p) and total number of wires (BitWidth). Thus the total Interconnect power is given by Equation 3:

$$P_{total} = BitWidth * p * P_{total}^{seg} \quad (3)$$

Whereas the calculation of power for each pipeline segment for a given repeater size, spacing, wire width and wire spacing is done by calculating the dynamic, leakage, and short circuit power for each of the component as follows:

$$P_{total}^{seg} = P_{dy}^{seg} + P_{sc}^{seg} + P_{leak}^{seg} \quad (4)$$

### 3.2.1 Dynamic Power

Dynamic or switching power due to switching of repeaters, pipeline registers and its corresponding buffers, and the wires

<sup>1</sup>Spice characterized

<sup>2</sup>Calculated using PTM[8] models and ITRS parameters[5]

is given by Equation 5 where,  $f$  is the frequency of operation, AF is activity factor<sup>4</sup> and CF is coupling factor<sup>5</sup>

$$P_{dy}^{seg} = (AF * (C_{gate} + C_p) * n_r + C_{wp}) * f * V_{dd}^2 + P_{flop\_dy} * f + \sum_{i \in (1..Nb)} (Size_{buff}^i * P_{buff\_dy}^1 * f) \quad (5)$$

$$C_{wp} = ((c_{gnd} + c_f) * AF + c_c * CF) * L^{seg} \quad (6)$$

### 3.2.2 Static Power

Static power is consumed when the transistors are idle. This is due to the finite OFF state current flowing in transistors in sub-threshold region and is given by Equation 7 where leakage current of a 1  $\mu\text{m}$  repeater ( $I_{leak}$ ) is determined using equations in [2].

$$P_{leak}^{seg} = (I_{leak} * S * n_r + P_{flop\_leak} + \sum_{i \in (1..Nb)} (Size_{buff}^i * P_{buff\_leak}^1)) \quad (7)$$

### 3.2.3 Short-circuit power

The short circuit power of repeaters (the finite duration ( $t_r$ ) in which both PMOS and NMOS are on) is calculated by equation 8 where  $I_{sc}$  is the short circuit current of 1  $\mu\text{m}$  repeater<sup>6</sup>.

$$P_{sc}^{seg} = I_{sc} * S * V_{dd} * t_r * f \quad (8)$$

## 4. EXPERIMENTAL RESULTS

In this section, we present a small subset of results that we obtained using the tool. These results exhibit various trends in the interconnect energy and serve to demonstrate how accurately our tool models the interconnects. The results are presented for interconnects of different lengths modeled at different technology nodes, with varying degree of pipelining, pitch values and operating at different frequencies. We also present and describe validation results for different interconnect configurations obtained using HSPICE[3].

Figure 4 shows the change in the power as degree of pipelining is increased for two different technology nodes (90 nm and 65 nm) and for two different frequency values (2 Ghz and 1 Ghz). Increasing the pipeline stages for a particular frequency and technology first reduced the power and then there is an increase in the power. In the left part of the graph, power reduction due to decrease in repeater size and number (as a result of increase in degree of pipelining) overwhelms the power overheads due to flops and buffers. However, the situation is opposite for higher degree of pipelining (as shown in right part of the graph) where the power overheads due to flops and buffers exceed the benefits because of already small repeaters. Thus, the inflexion point corresponding to the optimal degree of pipelining shifts to the right for higher frequencies and for lower technology nodes.

<sup>4</sup>Activity factor is determined by averaging the transitions on each line for a execution trace of a benchmark

<sup>5</sup>Coupling factor is determined by averaging the coupling between adjacent lines (depending on the direction of switching) for a execution trace of a benchmark

<sup>6</sup>The short circuit power of flops and buffers are included in the dynamic power while characterizing these elements.

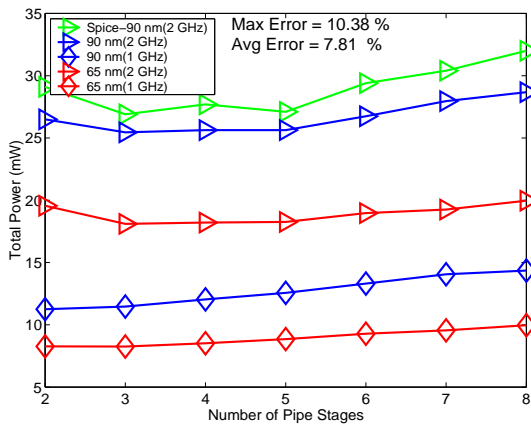


Figure 4: Degree of Pipelining vs Power for 5 mm interconnect with  $12 * \lambda$  pitch

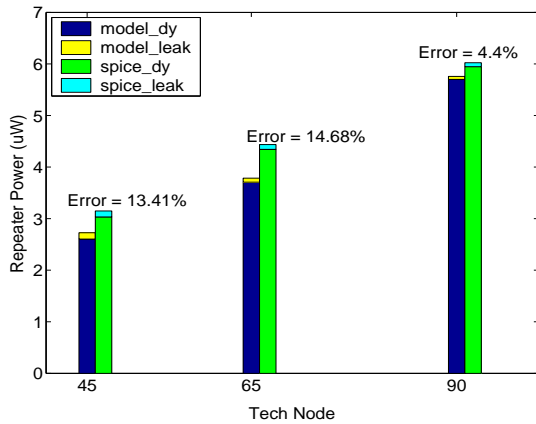


Figure 5: Validation of Dynamic and Leakage power for  $1 \mu\text{m}$  Repeater operated at 1GHz for Different Technology Nodes

This reinforces the need for higher degree of pipelining in interconnects running at high frequencies and/or smaller technologies. The reduction in the power of interconnect for smaller technologies is attributed to reduction of transistor capacitance that leads to lower dynamic and short circuit power of repeaters and flops. Figure 5 brings out this fact more clearly by showing that the dynamic power of repeater reduces significantly whereas leakage power of repeater increases for smaller technology nodes. However, the leakage power is a small fraction of overall power of repeater in Figure 5 or interconnect in Figure 4 because we consider a workload with high activity factor in these configurations. The component wise power breakup and leakage trend in interconnect for different activity factors are presented in Figure 6 and Figure 7 respectively which are discussed later.

Figure 4 also shows the HSPICE simulated power estimation for the 90 nm (2 GHz) interconnect to validate our tool. We observe that the error in estimating power using our model is 10.3% at the worst and 7.8% on an average for this configuration. The error estimates for a  $1 \mu\text{m}$  repeater running at 1 GHz is shown separately in Figure 5 which shows that the error in estimation of repeater power is at most 14.6% across technologies. The important point to note is that our model has a discreet unidirectional error.

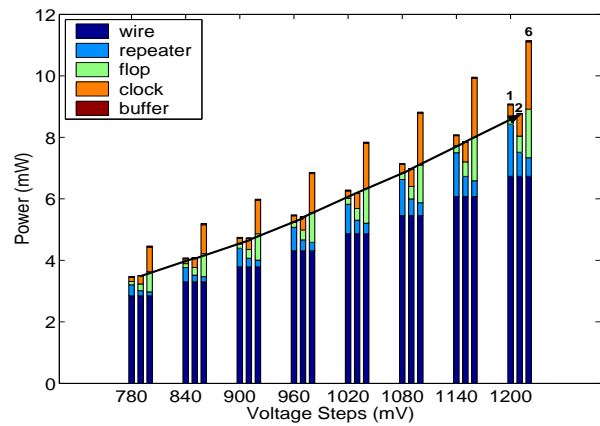


Figure 6: Component wise power breakup for a 5 mm interconnect with  $16 * \lambda$  pitch in 90 nm tech node running at 1 GHz.

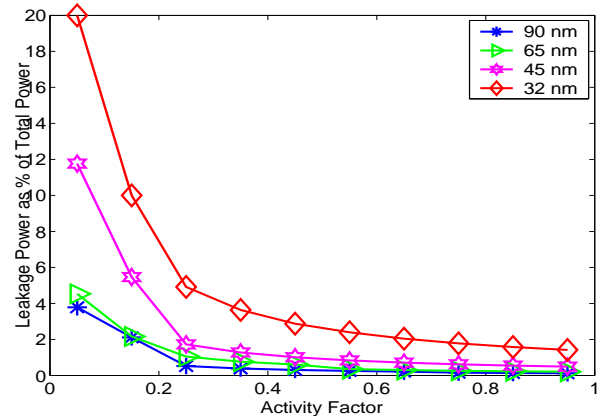
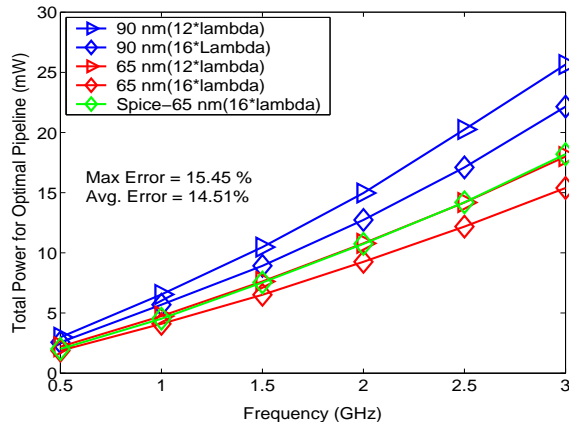


Figure 7: Leakage as % of Total Power for Different Activity Factors for Optimally Pipelined 5 mm Interconnect with  $12 * \lambda$  Pitch Running at 1 GHz

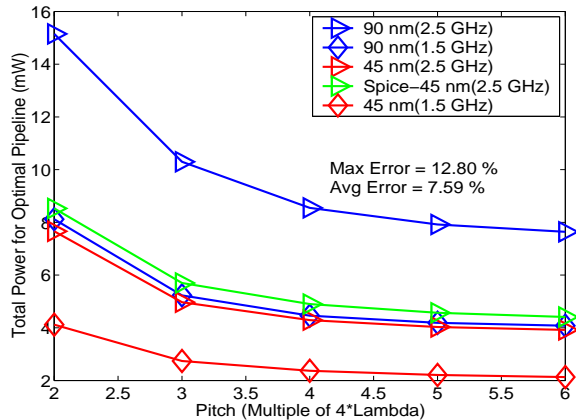
Figure 6 depicts component wise power breakup for 8 different voltage steps decreasing by 60 mV from operating voltage (1.2 V) for three different degrees of pipelining (2 being the optimal degree of pipelining in this configuration). It is clear from the graph that the wire power is the major component in the overall power of interconnect and clock power is the next top contributor. Figure 6 also shows the recurring trend that increasing the degree of pipelining first reduces power till optimal degree of pipelining (middle bar in this case) and then there is an increase because of reason explained earlier. The another trend depicted is the reduction in overall power w.r.t reduction in voltage which is quadratic in nature as shown by plot connecting the high points of the middle bar for different voltage steps.

Figure 7 depicts the leakage power percentage of total power for a 5 mm interconnect which is optimally pipelined and running at 1 GHz for a range of activity factors. The leakage power is high (20%) for smaller technologies such as (32 nm and 45 nm) and for low activity factor as expected. Though the fraction is not as high as in combinational circuits because as Figure 6 depicts that wire (which doesn't have a leakage component) makes a major fraction of interconnect power.

Figure 8 shows the change in power w.r.t frequency for optimal degree of pipelining for two different technology nodes (90nm and 65nm) and for two different wire pitch values ( $12 * \lambda$  and  $16 * \lambda$ ). The graph clearly shows the linear change in power w.r.t to the frequency for both the technology nodes. Increasing wire pitch within a technology decreases coupling capacitance which in turn reduces repeater size and number that leads to reduction in power. Again the power reduces in smaller technologies because of the reason explained above. The HSPICE validated graph for 65 nm and  $16 * \lambda$  shows that the maximum error is 15.45% whereas the average error is 14.51% for varying frequency.



**Figure 8: Frequency vs. Power for optimal degree of pipelining for 4 mm interconnect**



**Figure 9: Pitch Vs Power for optimal degree of pipelining for 2 mm interconnect**

Figure 9 clearly brings out the trend of reduction in power for optimal degree of pipelining with increasing wire pitch for two different frequencies (2.5 GHz and 1.5 GHz) in two different technology nodes (90 nm and 45 nm). The reduction in power is proportional to inverse of the pitch. As mentioned above, increasing pitch actually reduces coupling capacitance which in turn decreases the load on repeaters and makes it possible to reduce repeater size and number of repeaters. Increasing wire pitch also reduces the optimal degree of pipelining as the signal can travel more distance for the same time period. Reduction in the degree of optimal pipeline reduces required number of flops which further reduces power. The trend towards linear reduction in power

with reduced frequency is also visible in this graph so as the trend towards reduction in power in smaller technologies for the same length of interconnect. The HSPICE validation for 45 nm at 2.5 GHz shows that the maximum error is 12.8% whereas the average error is 7.59%.

## 5. EXAMPLE

This section gives an example usage of INTACTE to evaluate an architectural design trade-off and associated compiler optimization in context of clustered VLIW architectures[17]. Clustered VLIW architectures resolve the scalability problem associated with centralized VLIW architectures and are very popular in the embedded domain[36][18][20][34][15]. A clustered VLIW architecture has more than one register file and connects only a subset of functional units to a register file. Groups of small computation clusters can be fully or partially connected using either a *point-to-point* network or a *bus-based* network. The compiler is responsible for spatial and temporal scheduling of instructions in a clustered architecture[24][31][30].

Though clustering helps to combat the scalability problem by making components simpler and thereby improving performance and reducing energy consumption, an interconnection network is required for the communication of data values among different clusters. This communication happens over long wires having high load capacitance, which in effect takes more time and consumes more energy[28][22]. Earlier Studies report that a very high percentage (30% to 50%) of the total processor energy consumption is attributed to interconnects[27][40]. Clearly, clustered architectures are attractive only if their benefits outweigh the performance and energy penalties due to interconnections. Thus efficient means of using interconnects are important for clustered VLIW architectures.

Previous studies have reported that performance degrades by 12% when the latency of communication is doubled for a four clustered architecture, and that increasing the interconnection bandwidth from one to two improves the performance by as much as 10%[24]. It has been observed that though few of the communications are critical and delaying them can have severe impact on performance, the huge majority of communications are known to be non-critical (attributed to data dependencies and resource constraints) and can still happen on a slow path without affecting performance. Figure 10 presents quantitative results to substantiate our arguments. This figure present the percentage of required communication that has a slack of three cycles (two cycles and four cycles) or more for a two cluster and a four cluster machine having two high speed bidirectional cross-paths between clusters. It is clear that all the benchmarks have many communications with high slack values. On an average, we observe that 60.88% (82.51% and 43.16% respectively) and 65.55% (86.21% and 48.34% respectively) of communications can sustain a latency of three cycles (two cycles and four cycles respectively) for a set of media benchmark for a 2-clustered and 4-clustered machine respectively. Thus, even though having a cross-path with inter-cluster communication bandwidth of two is desirable from a performance point of view, having both the wires optimized for low latency is an over kill. Based on these observations, a more suitable design option for interconnect from the point of view of an architect would be to design some paths optimized for latency and others for energy[29]. This

is based on the insight that critical communication can take place over fast but more energy-consuming wires, and the compiler can steer other not-so-critical communication over slower but energy-efficient wires[33][32].

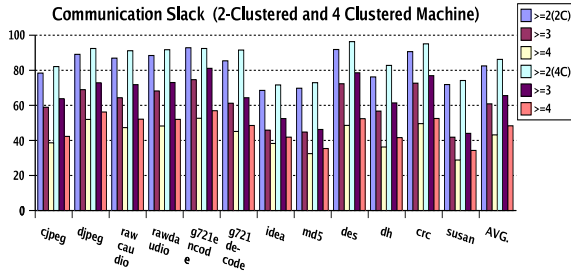


Figure 10: Communication Slack for 2-Clustered and 4-Clustered Machine

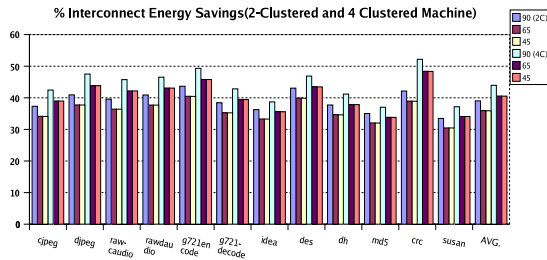


Figure 11: Communication Energy Savings for 2-Clustered and 4-Clustered Machine

Our methodology can be used to easily evaluate the potential of such an architectural trade-off. Architect needs to provide only the length, number of bits, target technology, operating voltage and delay estimates to explore the desired interconnect path under investigation and the proposed model can be used to get a set of possible interconnect design options to choose from. For example, an architect can seek the benefit of using one fast 32-bit path and one slow 32-bit path for inter-cluster communication for a 2 cluster and 4 cluster machine. Based on interconnect length estimates of 1.4 mm, the realizable benefits of the proposed heterogeneous interconnect (with one 32-bit path with single cycle latency and another 32-bit path with 3 cycle latency) over a homogeneous interconnect (with both paths optimized for 1 cycle latency) for 2-clustered and 4-clustered architecture in three different technology nodes (90 nm (1.2V), 65 nm (1V), 45 nm (1V)) are plotted in figure 11. These results are evaluated using a set of media benchmark and an energy efficient scheduling algorithm implemented in the trimaran compiler. The reader is referred to [33][32] for details of the scheduling framework and a detailed analysis of benchmark specific results. The heterogeneous interconnects give 35% to 39% improvement in interconnect energy across different technologies for a 2-clustered machine whereas for 4-cluster machine the benefits are between 38% to 44%. The benefit has slight variation across technologies. For smaller technologies, the effective cluster size and inter-cluster length decreases that reduces the benefit to some extent (as seen in Figure 11 in going from 90 nm to 65 nm) but at the same

time increase in leakage fraction of power causes the increase in energy (as seen in Figure 11 in going from 65 nm to 45 nm). We also observe that the benefit increases further by 5%-8% in all technologies if wire pitch is doubled for slow interconnect. This is one example of how our model makes it very easy for an architect to make high level design trade-offs without requiring a detailed knowledge of circuit level details.

## 6. DISCUSSION

There have been many proposals for reducing interconnect energy at the compiler level or architecture level with an indirect evaluation based on reduction in activity or by guesstimating based on earlier circuit level studies. Such an evaluation is inherently limited because it does not take into account the impact of reducing one component of power on other or the power overheads of the optimization itself. The proposed tool can be used to directly quantify exact benefits of various architectural and compiler optimization for overall interconnect energy saving. For example, [26] reduces the transitions to optimize energy of instruction bus evaluated by aggregate reduction in bit transitions on the consecutive wires. However, this ignores the leakage in various interconnect components such as repeaters, flops, and buffers as well as coupling between the wires which can potentially limit the benefits of such an optimization. Similarly, [11] argues that interconnects composed of wires having different delay and power characteristics improve the overall  $ED^2$  of processor significantly. However, evaluation is based on earlier circuit level studies[12][29] ignoring many important components of power such as power due to pipelining buffers.

Apart from compiler and architectural optimizations, the proposed model is also useful for an early evaluation of a circuit level implementation of desired interconnects. For example, for certain interconnects it might be more beneficial to implement a logical 32 bit interconnect with eighth fast physical lines (meeting delay constraints) and transfer data using serialization. An early evaluation suggests that such an implementation gives up to 25% to 35% energy benefits for interconnects with moderate pitch values. The benefits of serialization are even more for small pitch values. Detailed knowledge of energy breakup in different components of desired interconnect based on workload parameters (such as activity factor and coupling factor) also helps to develop new workload based interconnect optimization techniques such as dynamic voltage scaling, leakage energy savings, and power gating in interconnect components. Finally, co-designing the interconnect with micro-architectural design of rest of the processor modules which is specifically more important for high level synthesis and design of embedded SoCs has been a major motivation behind the development of INTACTE.

## 7. RELATED WORK

Banerjee et al., analyze the effect of changing repeater size and spacing on the power and delay of interconnects[12]. They observe that the delay variation is very shallow near the minimum delay point, which can be utilized to minimize power consumption. However, the wire width and spacing is fixed and its impact on power is not considered in this work. [29] considers the effects of wire dimension on bandwidth (irrespective of power) by considering two cases



of same wire width and spacing and minimal spacing. In contrast, we propose a complete tool for modeling different interconnects across technologies. INTACTE optimizes the power by varying all the four parameters (i.e. repeater size, repeater spacing, wire width and wire spacing) in order to obtain minimal power for the desired interconnect. As our results show, wire width and spacing has significant impact on power and minimal spacing leads to comparatively higher power consumption.

[25] presents strong evidences of interconnects being one of major performance and power bottleneck in multi-core systems and a methodology of co-designing interconnect with other processor components. The study is based on earlier circuit level estimates of interconnect parameters[37][42][22]. [11] observes that different interconnects in processor have different bandwidth and latency requirements and interconnects composed of wires with different characteristics meet the power-performance goals of a system in a much better way. The evaluation has been performed based on guesstimates on circuit level study performed in[29].

The closest to our work is the work by Gupta et al.,[21]. They propose a methodology for first level power estimation of interconnect. They take into account activity factor and coupling factor in a similar fashion. They also propose a wire length estimation model which is complementary to our work. The most important limitation of their method is non-consideration of pipelining in interconnect and its overheads in terms of power and delay which is indispensable for global and semi-global interconnect they target. Many important components of power (such as leakage in repeaters and clock power) are not modeled in their work. It is also not clear how easy it is to obtain power estimates of desired interconnect across technologies by using their model.

[39] considers the impact of coupling between adjacent wires on power using a sophisticated method. The proposed method takes into account the time difference between transitions on adjacent wires using a timescale parameter called charge time which essentially represents the correlation time length between two events. The proposed method relies on layout information to be able to calculate coupling in a better fashion. Since we propose a high level methodology for interconnect energy modeling, in absence of detailed layout information, a simple calculation of coupling as done by profiling workload suffices to give reasonable accuracy in our model.

Orion is a simulator proposed for delay and power modeling specifically targeting off-chip interconnects[41]. The approach used is event driven that takes into account the events during execution to determine the power consumption in various logical interconnect components such as FIFO, arbiter, and crossbar. They lack a link model and rely on standard published data for accounting the power of links. However, link is an essential part of the communication and they also recognize the need for a parameterizable model for link power to be able to perform architectural design trade-offs[41]. Our work complements their work by providing a thoroughly validated model for optimizing the power of link used to connect the logic modules.

## 8. CONCLUSION AND FUTURE DIRECTIONS

In this paper, we proposed a tool that fills the gap between architect's need and circuit level models for design of interconnects. The tool takes architectural parameters such as length, bit-width, latency and target technology and provides a set of interconnect options with varying degree of area, pipelining, and power budget using pre-characterized estimates of circuit parameters for different interconnect components. The major motivation behind development of this tool has been co-designing interconnect with other architectural components that is highly desirable for high level synthesis and design of embedded SoCs. The proposed tool is not only useful to make micro-architectural and architectural trade-offs but also to evaluate various architectural and compiler optimizations. We presented examples of quantitative evaluation of some design choices using the tool such as optimal degree of pipelining and heterogeneous interconnect and discussed other possible usage of the tool. Currently INTACTE is limited to design of point-to-point interconnects which represent most important and major fraction of all on-chip interconnects. In future, we are planning to extend INTACTE for other kind of interconnects such as buses. Porting the tool to C and making it available for the use of general public is another direction in which we plan to venture our efforts.

## 9. REFERENCES

- [1] ARM MPCore. <http://www.arm.com>.
- [2] BSIM4.6.0. <http://www-device.eecs.berkeley.edu/~bsim3/bsim4.html>.
- [3] HSPICE. <http://www.synopsys.com/products/hspice.html>.
- [4] Intel Multi-Core. <http://www.intel.com/multi-core/index.htm>.
- [5] International Technology Roadmap for Semiconductors. <http://www.itrs.net/>.
- [6] MATLAB. <http://www.mathworks.com/products/matlab/>.
- [7] OMAP. [focus.ti.com/omap/docs/omaphomepage.tsp](http://focus.ti.com/omap/docs/omaphomepage.tsp).
- [8] Predictive Technology Model. <http://www.eas.asu.edu/~ptm/>.
- [9] J. H. Ahn, W. J. Dally, B. Khailany, U. J. Kapasi, and A. Das. Evaluating the imagine stream architecture. In *Proc. of intl. symp. on Computer architecture*, page 14, 2004.
- [10] A. Aleta, J. M. Codina, A. Gonzalez, and D. Kaeli. Heterogeneous clustered vliw microarchitectures. In *Proc. of Intl. Symp. on Code Generation and Optimization*, March 2007.
- [11] R. Balasubramonian, N. Muralimanohar, K. Ramani, and V. Venkatachalapathy. Microarchitectural wire management for performance and power in partitioned architectures. In *Proc. of the Intl. Symp. on High-Performance Computer Architecture*, pages 28–39, 2005.
- [12] K. Banerjee and A. Mehrotra. A Power-Optimal Repeater Insertion Methodology for Global Interconnects in Nanometer Designs. In *Proc. of IEEE Transactions on Electron Devices*, pages 2001–2007, November 2002.

- [13] M. Baxter. Amd64 opteron: first look. *Linux J.*, 2003(111):2, 2003.
- [14] W. J. Dally and B. Towles. Route packets, not wires: on-chip interconnection networks. In *Proc. of the conf. on Design automation*, pages 684–689, 2001.
- [15] J. Derby and J. Moreno. A High-performance Embedded DSP Core with Novel SIMD Features. In *Proc. of 2003 Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2003.
- [16] S. Dropsho, V. Kursun, D. H. Albonese, S. Dwarkadas, and E. G. Friedman. Managing static leakage energy in microprocessor functional units. In *Proc. of the intl. symp. on Microarchitecture*, pages 321–332, 2002.
- [17] P. Faraboschi, G. Brown, J. A. Fisher, and G. Desoli. Clustered Instruction-level Parallel Processors. Technical report, Hewlett-Packard, 1998.
- [18] P. Faraboschi, G. Brown, J. A. Fisher, G. Desoli, and F. Homewood. Lx: A Technology Platform for Customizable VLIW Embedded Processing. In *Proc. of 27th annual Intl. Symp. on Computer architecture*, pages 203–213, 2000.
- [19] K. I. Farkas, P. Chow, N. P. Jouppi, and Z. Vranesic. The multicluster architecture: reducing cycle time through partitioning. In *Proc. of the intl. symp. on Microarchitecture*, pages 149–159, 1997.
- [20] J. Fridman and Z. Greefield. The TigerSHARC DSP architecture. *IEEE Micro*, pages 66–76, 2000.
- [21] P. Gupta, L. Zhong, and N. K. Jha. A high-level interconnect power model for design space exploration. In *Proc. of the intl. conf. on Computer-aided design*, page 551, 2003.
- [22] R. Ho, K. Mai, and M. Horowitz. The Future of Wires. *Proc. of IEEE*, 89(4):490–504, 2001.
- [23] C. H. Hsu and U. Kremer. The Design, Implementation, and Evaluation of a Compiler Algorithm for CPU Energy Reduction. In *Proc. of Conf. on Programming language design and implementation*, pages 38–48, 2003.
- [24] K. Kailas, A. Agrawala, and K. Ebcioğlu. CARS: A New Code Generation Framework for Clustered ILP Processors. In *Proc. of intl. Symp. on High-Performance Computer Architecture*, page 133, 2001.
- [25] R. Kumar, V. Zyuban, and D. M. Tullsen. Interconnections in multi-core architectures: Understanding mechanisms, overheads and scaling. In *Proc. of the Intl. Symp. on Computer Architecture*, pages 408–419, 2005.
- [26] C. Lee, J. K. Lee, T. Hwang, and S.-C. Tsai. Compiler optimization on VLIW instruction scheduling for low power. *ACM Trans. Des. Autom. Electron. Syst.*, 8(2):252–268, 2003.
- [27] N. Magen, A. Kolodny, U. Weiser, and N. Shamir. Interconnect-power Dissipation in a Microprocessor. In *Proc. of Intl. workshop on System Level Interconnect Prediction*, pages 7–13, 2004.
- [28] D. Matzke. Will Physical Scalability Sabotage Performance Gains. *IEEE Computer*, September 1997.
- [29] M. L. Mui, K. Banerjee, and A. Mehrotra. A Global Interconnect Optimization Scheme for Nanometer Scale VLSI with Implications for Latency, Bandwidth and Power Dissipation. In *IEEE Transactions on Electron Devices*, pages 195–203, 2004.
- [30] R. Nagpal and Y. N. Srikant. A Graph Matching Based Integrated Scheduling Framework for Clustered VLIW Processors. In *Proc. of ICPP Workshop on Compile and Runtime Techniques Parallel Computing*, pages 530–537, 2004.
- [31] R. Nagpal and Y. N. Srikant. Integrated Temporal and Spatial Scheduling for Extended Operand Clustered VLIW Processors. In *Proc. of Conf. on computing frontiers*, pages 457–470, 2004.
- [32] R. Nagpal and Y. N. Srikant. Exploring Energy-Performance Trade-offs for Heterogeneous Interconnect Clustered VLIW Processors. Technical Report, Dept. of CSA, Indian Institute of Science (<http://www.archive.csa.iisc.ernet.in/TR>), 2005.
- [33] R. Nagpal and Y. N. Srikant. Exploring energy-performance trade-offs for heterogeneous interconnect clustered vliw processors. In *Proc. of Intl. Conf. on High Performance Computing*, pages 497–508, 2006.
- [34] G. G. Pechanek and S. Vassiliadis. The ManArray Embedded Processor Architecture. In *Proc. of Euromicro Conf.*, pages 348–355, 2000.
- [35] B. Sinharoy, R. N. Kalla, J. M. Tandler, R. J. Eickemeyer, and J. B. Joyner. Power5 system microarchitecture. *IBM Journal of Research and Development*, 49(4/5):505–521, 2005.
- [36] Texas Instruments Inc. TMS320C6000 CPU and Instruction Set reference Guide. <http://www.ti.com/sc/docs/products/dsp/c6000/index.htm>, 1998.
- [37] T. N. Theis. The future of interconnection technology. *IBM Journal of Research and Development*, 44(3), 2000.
- [38] J. H. Tseng and K. Asanovic. Energy-efficient register access. In *Proc. of the symp. on Integrated circuits and systems design*, page 377, 2000.
- [39] T. Uchino and J. Cong. An interconnect energy model considering coupling effects. In *Proc. of the conf. on Design automation*, pages 555–558, 2001.
- [40] H. Wang, L.-S. Peh, and S. Malik. Power-driven Design of Router Microarchitectures in On-chip Networks. In *Proc. of Symp. on Microarchitecture*, page 105, 2003.
- [41] H.-S. Wang, X. Zhu, L.-S. Peh, and S. Malik. Orion: a power-performance simulator for interconnection networks. In *Proc. of the intl. symp. on Microarchitecture*, pages 294–305, 2002.
- [42] J. D. Warnock, J. M. Keaty, J. G. C. J. Petrovick, C. J. Kircher, B. L. Krauter, P. J. Restle, B. A. Zoric, and C. J. Anderson. The circuit and physical design of the power4 microprocessor. *IBM Journal of Research and Development*, 46(1), 2002.
- [43] S. Wilton and N. Jouppi. CACTI: An enhanced cache access and cycle time model. In *IEEE Journal of Solid-State Circuits*, volume 31, pages 677–688, May 1996.