CRISTA: A New Paradigm for Low-Power, Variation-Tolerant, and Adaptive Circuit Synthesis Using Critical Path Isolation

Swaroop Ghosh, Swarup Bhunia, *Member, IEEE*, and Kaushik Roy, *Fellow, IEEE*

***Abstract*—Design considerations for robustness with respect to variations and low-power operations typically impose contradictory design requirements. Low-power design techniques such as voltage scaling, dual-$V_{\rm th}$, etc., can have a large negative impact on parametric yield. In this paper, we propose a novel paradigm for low-power variation-tolerant circuit design called CRitical path ISolation for Timing Adaptiveness (CRISTA), which allows aggressive voltage scaling. The principal idea includes the following: 1) isolate and predict the set of possible paths that may become critical under process variations; 2) ensure that they are activated rarely; and 3) avoid possible delay failures in the critical paths by dynamically switching to two-cycle operation (assuming all standard operations are single cycle), when they are activated. This allows us to operate the circuit at reduced supply voltage while achieving the required yield. Simulation results on a set of benchmark circuits with Berkeley-predictive-technology-model [BPTM 70 nm: Berkeley predictive technology model] 70-nm devices that show an average of 60% improvement in power with small overhead in performance and 18% overhead in die area compared to conventional design. We also present two applications of the proposed methodology that include the following: 1) pipeline design for low power and 2) temperature-adaptive circuit design.**

***Index Terms*—Low power, process variation-tolerant design, supply voltage scaling, temperature-aware design.**

I. Introduction

It is well-known that process parameter variations (both systematic and random) may cause parametric failures in logic circuits leading to yield loss. With technology scaling in nanometer regime, parameter variations play increasingly an important role in circuit marginalities and, hence, pose a major design concern. Conventional wisdom dictates a conservative design approach (e.g., scaling up the supply voltage or upsizing logic gates) to avoid a large number of chip failures. However, such techniques come at the cost of power and/or die area. Process tolerance and low power, therefore, represent contradictory design requirements.

Over the past few years, statistical design approach has been widely investigated as an effective method to ensure yield under process variations. In this approach, the design space is explored to optimize certain design parameter (e.g., power) to meet a timing yield and a target frequency. Several gate-level sizing and/or $V_{\rm th}$ assignment techniques [1] have been proposed recently addressing the minimization of total power while maintaining the timing yield. On the other end of the spectrum, design techniques have been proposed for postsilicon process compensation and process adaptation (e.g., adaptive body biasing [2]) to deal with process-related timing failures. Due to quadratic dependence of dynamic power of a circuit on its operating voltage, supply voltage scaling has been extremely effective in reducing the power dissipation. Researchers have investigated logic design approaches that are robust with respect to process variations and, at the same time, suitable for aggressive voltage scaling. One such technique called Razor [3] uses dynamic detection and correction of circuit timing errors to tune processor supply voltage. This technique eliminates the need of voltage margins and supports dynamic voltage scaling (DVS) for power reduction. Design optimization techniques using gate sizing and dual-$V_{\rm th}$ assignment to improve power/area typically increase the number of critical paths in a circuit, giving rise to the so-called "wall effect" [4]. The uncertainty-aware design technique [4] describes an optimization process to reduce the wall effect. However, it does not address the problem of power dissipation.

In this paper, we present CRISTA, a novel design paradigm, which achieves robustness with respect to timing failure and provides the opportunity for aggressive voltage scaling by critical path isolation. The notion critical path isolation is used throughout this paper to indicate the confinement of critical paths of synthesized design to known logic block (or cofactor, as we will see later). Such isolation leads to a design methodology for low power dissipation by making the critical paths predictable and rare under parametric variations. Any possible delay errors (that may occur under a single-cycle operation) are predicted ahead of time and are avoided by two-cycle operations (assuming all standard operations are single cycle). This lets us scale the supply voltage aggressively for low power dissipation. In particular, CRISTA:

1) isolates the critical paths and makes them predictable (based on few primary inputs) under parametric variation so that with reduced supply voltage, possible delay errors

Manuscript received July 17, 2006; revised January 18, 2007. This work was supported in part by Gigascale System Research Center (GSRC), and in part by Semiconductor Research Corporation (SRC) under Grant 1078.002. This paper was recommended by Associate Editor S. Vrudhula.

S. Ghosh and K. Roy are with the School of Electrical Engineering, Purdue University, West Lafayette, IN 47906 USA (e-mail: ghosh3@purdue.edu; kaushik@purdue.edu).

S. Bhunia is with Case Western Reserve University, Cleveland, OH 44106 USA (e-mail: skb21@case.edu).

Digital Object Identifier 10.1109/TCAD.2007.896305

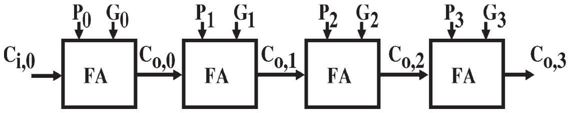

Fig. 1. Ripple carry adder [5].

under single-cycle operation are deterministic and can be avoided by a two-cycle operation.

2) restricts the occurrences of the previous two-cycle operations by reducing the activation probability of critical paths.
3) increases the delay margin between critical and noncritical paths by both logic synthesis and proper gate sizing for improved yield, reliability of operations, and aggressive voltage scaling.

We present two applications of CRISTA pipeline-based design for low power and temperature-adaptive circuit design. Both are based on the concept of making the possible delay errors (under single-cycle operations) predictable and rare under parametric variations and avoiding them by two-cycle operations. In pipeline design, the circuit is designed to operate at fixed low-supply voltage with occasional two-cycle operations. The two-cycle operations are implemented by stalling the pipeline, as discussed in Section IV. On the other hand, in temperature-adaptive design, the circuit is designed such that it operates in a single cycle at normal temperature (and nominal supply) while at high temperature, it operates at lower supply voltage with few two-cycle operations. This reduces the temperature with small performance overhead.

This paper is organized as follows. In Section II, we explain the design flow to synthesize an input netlist for critical path isolation. Implementation details of CRISTA along with experimental results are presented in Section III. In Section IV, we present the application of CRISTA in pipeline design. In Section V, we propose its application in temperature-adaptive circuit design. Practical challenges associated with the proposed technique are addressed in Section VI. Conclusions are drawn in Section VII.

II. Preliminary Analysis

In this section, first we present an example of an adder to illustrate the proposed approach for low-power robust circuit design. Next, we present the design flow which allows us to apply a similar approach to any random logic circuit.

A. Voltage Scaling and Two-Cycle Operations in an Adder

For the sake of simplicity, we choose a 4-b ripple carry adder, as shown in Fig. 1. Signals $P_0 - P_3 (G_0 - G_3)$ are the propagate (generate) signals, whereas $C_{\mathrm{i},0} (C_{\mathrm{o},1} - C_{\mathrm{o},3})$ are carry-in (carry-out) signals [5]. As evident, the path from carry-in to carry-out is critical and determines the frequency of operation of the adder. However, note that the critical path is activated only when $C_{\mathrm{i},0} = 1$, and at the same time, $P_0 P_1 P_2 P_3 = 1$. Since the probability of such occurrences is very low (as $p(P_0 P_1 P_2 P_3 C_{\mathrm{i},0} = 1) = p(P_0) p(P_1) p(P_2) p(P_3) p(C_{\mathrm{i},0})$ is very low), one can reduce the supply voltage such that all operations with $P_0 P_1 P_2 P_3 = 0$ and/or $C_{\mathrm{i},0} = 0$ can still be performed in one cycle. However, when the critical path is activated, the correct results are obtained by evaluating the adder in two clock cycles (called two-cycle operation). The activation of critical path can be predicted by precomputation of $P_0 P_1 P_2 P_3$. In a nutshell, by making the critical path predictable and utilizing the available slack between critical and noncritical path, it is possible to operate the circuit at reduced supply voltage. Note that this approach incurs a penalty of an extra clock cycle when the critical path is activated. However, by ensuring low activation probability of critical paths, it may be possible to reduce the active and leakage power by rarely paying penalty of an extra clock cycle.

To evaluate the feasibility of this idea, we simulated a 4-b ripple carry adder with 1-V supply in Hspice. We used BPTM [6] 70-nm devices for simulation. The critical path delay was found to be 260 ps, and average power consumption was 13.03 μW. Assuming the clock period to be 260 ps, we reduced the supply to 0.8 V. Now, the noncritical paths were within the single-cycle delay bound; however, the critical path delay increased to 330 ps and was evaluated with two cycles. The new power consumption was 7.32 μW, leading to 44% saving in total power (active and leakage).

Note that the aforementioned technique could also be used for supply voltage reduction based on temperature. Instead of permanently operating the adder at low voltage, we may operate it at nominal supply during normal temperature and lower supply only during increased temperature. This approach results in performance penalty only during increased temperature.

B. Generalization to Random Logic

In the previous section, we presented the idea of supply voltage scaling for an adder where the critical path was unique (assuming no process variation). However, a random logic can have many critical paths and corresponding input conditions to predetermine their activation. Furthermore, the critical paths may vary from chip to chip due to parametric variations. In such situations, the overhead associated with predecoding logic can overshadow the power savings. To exercise a similar supply scaling technique on random logic circuits, we need to make sure that the following conditions are met: 1) the critical paths are confined to a predictable logic section and 2) the noncritical paths remain noncritical under process variation by providing a safe timing slack. The timing slack between critical and noncritical paths will be the enabling factor for supply voltage scaling. An example of a possible path delay distribution (cartoon) is shown in Fig. 2. It illustrates that critical paths are isolated from noncritical paths due to presence of timing slack. Furthermore, the critical paths may fail the delay target as determined by the clock frequency (as shown by solid bar in Fig. 2) under process variation. However, if the critical path activations are rare and predictable, then possible delay failures may be avoided by providing an extra clock cycle occasionally.

To obtain the delay distribution shown in Fig. 2, the design needs to be partitioned and synthesized in such a way that the paths are divided into several logic blocks. The partitioning

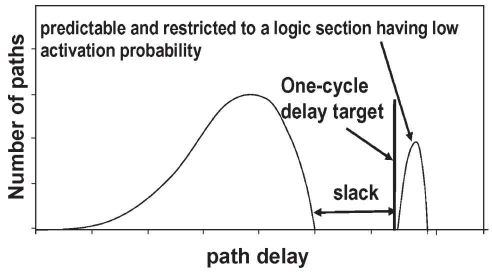

Fig. 2. Path delay distribution required for CRISTA.

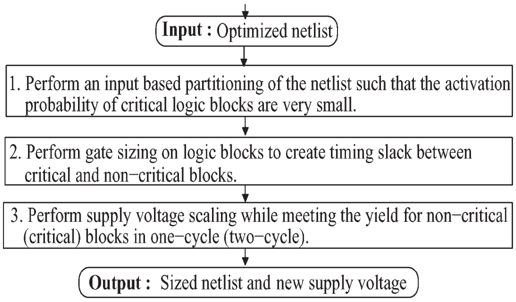
Input : Optimized netlist
1. Perform an input based partitioning of the netlist such that the activation probability of critical logic blocks are very small.
2. Perform gate sizing on logic blocks to create timing slack between critical and non−critical blocks.
3. Perform supply voltage scaling while meeting the yield for non−critical (critical) blocks in one−cycle (two−cycle).
Output : Sized netlist and new supply voltage

Fig. 3. CRISTA design methodology.

procedure should consider the following: 1) these logic blocks can be active or remain idle based on the state of primary inputs and 2) the probabilities of activation of the logic blocks containing critical paths (called critical block) are very low. Therefore, it will be possible to predict the activation of a logic block (and the corresponding paths) just by decoding the states of inputs. Next, gate sizing can be performed on the partitioned logic blocks to maximize the slack between critical and noncritical blocks leading to further isolation of critical paths. By performing the partitioning and sizing, a path delay distribution similar to the one shown in Fig. 2 can be achieved. Finally, supply voltage scaling can be done such that noncritical blocks meet the desired timing yield with respect to one-cycle delay target, whereas critical block meet the yield with respect to two-cycle delay target. Since the probability of activation of the critical block is low, the new design operating at a scaled voltage will have minimum impact on performance. The overall design flow is shown in Fig. 3.

It can be concluded that the power saving in CRISTA mainly comes from quadratic dependence of power on voltage. Power reduces quadratically while the delay and switching capacitance (due to decoding logic) increases only linearly, letting us reduce the EDP.

III. CRISTA Design Methodology

Based on the analysis and the guidelines derived previously, we describe the details of each step of the design flow (Fig. 3). This is followed by simulation results on a set of benchmark circuits.

A. Partitioning and Synthesis for Critical Path Isolation

To perform an input-based partitioning of the circuit for isolating critical paths and reducing their activation probabilities, we have used Shannon-expansion-based partitioning [4]–[9] scheme. This method is similar to [4] and will be described here briefly for the sake of clarity. Shannon expansion partitions a Boolean expression (f) as follows:

$$\begin{aligned} f(x_1,\ldots,x_i,\ldots,x_n) &= x_i \cdot f(x_1,\ldots,x_i=1,\ldots,x_n) \\ &\quad + \bar{x}_i \cdot f(x_1,\ldots,x_i=0,\ldots,x_n) \\ &= x_i \cdot \mathrm{CF}_1 + \bar{x}_i.\mathrm{CF}_2 \end{aligned} \tag{1}$$

where x_i is called the control variable, and CF_1 and CF_2 are called cofactors. The outputs of cofactors are merged using a *mux* controlled by x_i. If f contains subexpressions independent of control variable x_i, then we may also have a shared cofactor. Note that, in (1), the activation probability of each cofactor is 50%. However, by performing multilevel expansion, it is possible to reduce the activation probability of the resulting cofactors further. For example, a second level expansion of f (2) reduces activation probability of resulting cofactors to 25%

$$\begin{aligned} f(x_1,\ldots,x_i,\ldots,x_n) &= x_i x_j \cdot \mathrm{CF}_1 + x_i\bar{x}_j \cdot \mathrm{CF}_2 \\ &\quad + \bar{x}_i x_j \cdot \mathrm{CF}_3 + \bar{x}_i\bar{x}_j \cdot \mathrm{CF}_4. \end{aligned} \tag{2}$$

Control variable selection plays a very important role in achieving the desired goals in Shannon's based partitioning. If $a_i(b_i)$ is the literal count of variable x_i in true (complement) form in f, then the following metric can be used for critical path isolation of the circuit after expansion [4]:

$$M_i = \frac{|a_i - b_i|}{\max(a_i, b_i)}. \tag{3}$$

To achieve the dual objectives of isolating the critical paths to a cofactor and reducing its activation probability during partitioning and synthesis, first, the circuit is partitioned, and the cofactors containing critical paths are determined. Then, the cofactors are marked for further expansion (to reduce the activation probability of critical paths). The process is repeated under a given area and delay constraint. Note that Synopsys Design Compiler [10] has been used for synthesizing the new cofactors. The overall synthesis flow is shown in Fig. 4. A complete example of hierarchical partitioning and synthesis is also illustrated in Fig. 5 where the original circuit is partitioned into four cofactors, CF_{20}, CF_{32}, CF_{53}, and CF_{63}. The critical paths have been isolated to CF_{53} (which is activated by three inputs, i.e., $x_1x_2'x_3$). Note that, in this example, we do not have the shared cofactor which is important in avoiding the logic duplication during partitioning. However, they are independent of control variable. Therefore, our synthesis flow (Fig. 4) automatically chooses it for further expansion (if critical paths are isolated to it).

B. Gate Sizing for Further Isolation

In the previous section, we presented a circuit partitioning method to isolate the critical paths to a cofactor with small activation probability. The next step is to size the resulting cofactors individually: 1) to further isolate the critical paths and 2) to create timing slack between critical and noncritical

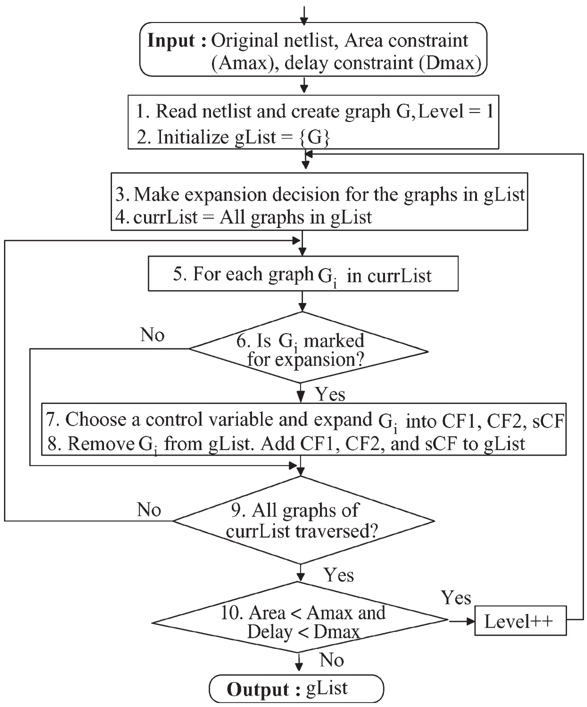
Input : Original netlist, Area constraint (Amax), delay constraint (Dmax)
1. Read netlist and create graph G, Level = 1
2. Initialize gList = {G}
3. Make expansion decision for the graphs in gList
4. currList = All graphs in gList
5. For each graph G_i in currList
No
6. Is G_i marked for expansion?
Yes
7. Choose a control variable and expand G_i into CF1, CF2, sCF
8. Remove G_i from gList. Add CF1, CF2, and sCF to gList
No
9. All graphs of currList traversed?
Yes
Yes
10. Area < Amax and Delay < Dmax
Level++
No
Output : gList

Fig. 4. Automated synthesis flow.

cofactors to allow lowering of supply voltage. To achieve this goal, all gates of the critical cofactor are downsized to make the corresponding paths further critical. The gates belonging to the remaining cofactors are selectively upsized to make them more noncritical and increase the slack. An example of the proposed sizing approach after partitioning is shown in Fig. 5. As shown, cofactor CF_{53} is downsized to make it further critical while other cofactors are upsized to make them more noncritical. Note that the proposed sizing approach is counter intuitive because in this case, the critical paths are made slower while noncritical paths are made faster.

We follow the aforementioned sizing strategy in a Lagrangian-relaxation-based gate sizing [12], as shown in Table I. Given a delay target (T_c), it tries to meet the yield requirement with minimum area. The procedure takes *gList* (i.e., list of cofactors) and determines the cofactor at highest level of hierarchy, *maxLevel* for downsizing it (Step 1). To compute the mean delays of the paths, Statistical Static Timing Analysis (SSTA) [11] is performed in Step 2 on each of the cofactors G_i from *gList*. In Step 3, a cofactor at the maximum hierarchical level (i.e., *maxLevel*) is selected as a critical logic block candidate. Next, a cofactor with hierarchy equal or one less than *maxLevel* is selected as *noncritCF*. Once the critical (noncritical) logic block candidates are decided, it is easy to perform downsizing (upsizing) based on their slack constraints. The multiplexer delays are computed and subtracted (Steps 5–7) from the specified delay target to derive the cofactor delay targets (with $\alpha = 1.2$, determined empirically). The delay targets of noncritical cofactors are obtained by subtracting T_c and multiplexer delays from overall critical path delay (Steps 8–12). The noncritical cofactor candidates are now upsized while meeting the yield target (Step 13). Finally, a complete graph G is created (Step 15) and returned in Step 16.

C. Determination of Supply Voltage

After partitioning and sizing, we obtain the path delay distribution similar to Fig. 2. Now, we may assign a lower supply voltage to reduce the power dissipation while meeting robustness. To achieve this, we start from nominal supply and iteratively reduce it with two stopping criterions: 1) delay violation of any of the noncritical cofactors (one-cycle delay target) for the given yield constraint and 2) delay violation of the critical cofactor (two-cycle target) for the target yield. Finally, another stopping criterion is the $3V_\mathrm{th}$ limit for reliable superthreshold operations [5]. The new supply voltages for a set of Microelectrons Center of North Carolina (MCNC) benchmarks are shown in Section III-D.

D. Simulation Results

In the previous sections, we presented CRISTA methodology to make the possible delay errors (that may occur under single-cycle operation) predictable and rare (using partitioning, synthesis and sizing). We also discussed the determination of new supply voltage. In this section, we present the simulation result on a set of MCNC benchmarks to demonstrate the feasibility of this methodology. In particular, we show the following: 1) isolation of critical paths to a cofactor (having low activation probability) and 2) reduction of supply voltage while maintaining robustness. In the following paragraphs, we present simulation setup followed by the results and discussions.

For logic optimization in our synthesis flow, we have used Synopsys Design Compiler [10]. For a basis of comparison, the original benchmarks are also optimized for area in Synopsys. The mapping is done to a standard cell library. The circuit delays are computed by using SSTA for BPTM 70-nm technology device parameters. The parametric variations (L, W, T_ox, doping, etc.) have been lumped into threshold voltage variation. The change in V_th due to interdie $(\Delta\mathrm{Vt}_\mathrm{inter})$ and intradie $(\Delta\mathrm{Vt}_\mathrm{intra})$ process variations are modeled as Gaussian variables with zero mean and standard deviations of 80 and 40 mV, respectively. The total change in transistor V_th is given by the summation of $\Delta\mathrm{Vt}_\mathrm{inter}$ and $\Delta\mathrm{Vt}_\mathrm{intra}$. The delay target (T_c) for sizing procedure is chosen by plotting the area-delay curve of the circuit and selecting the delay at which the slope of the curve is -1. The area and delay constraints for Shannon-based partitioning are kept at 40% and 20% more than the original circuit area and delay, respectively. The yield targets of original circuit and the cofactors for gate sizing are set to 95%. The yield target of cofactors operating on one cycle (two cycle) after application of reduced supply is fixed to 95% (100%). For power estimation, the circuits are simulated in Hspice by applying a set of 200 random input patterns having input switching probabilities of 0.2 as well as 0.5. The runtime of the entire methodology is found to be small (6.03 s for the largest benchmark *cht* on SUN blade 1000 workstation).

To illustrate the isolation of critical paths to the critical cofactor, we have plotted the path delay distribution of an example MCNC benchmark circuit (i.e., sct) after partitioning and sizing [Fig. 6(a)]. We have illustrated the path delay distribution of the partitioned and sized circuit (indicated by new) and also the

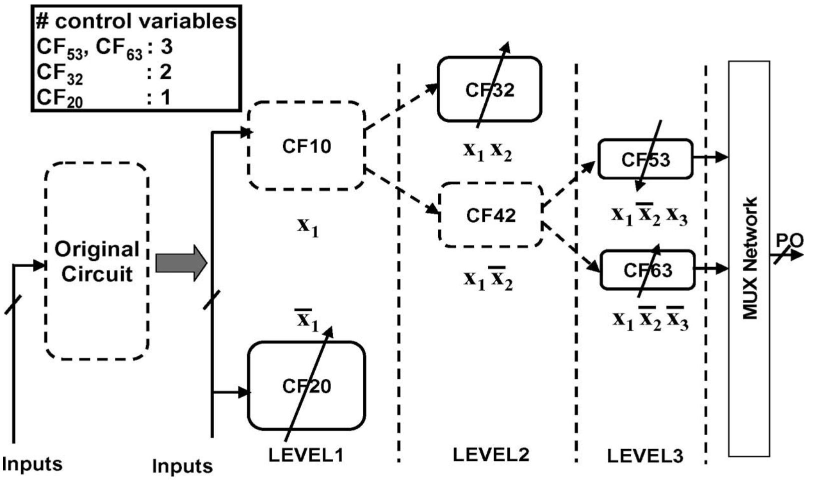

Fig. 5. Hierarchical expansion and sizing of cofactors.

TABLE I
SIZING PROCEDURE

```
Procedure performSizing()
Input : target delay ($T_c$), yield (Y), list of cofactors (gList);
Output : sized netlist;
maxLevel = maximum hierarchy of the cofactors in gList ;
run SSTA on $G_i \in gList$;
critCF=cofactor with critical paths at maxLevel hierarchy;
for each cofactors $G_i \in gList$
    calculate $G_i \rightarrow muxdelay$;
end for
dTarget = $\alpha T_c$ – critCF→muxDelay;
downSize(critCF, dTarget, Y);
critDelay=critCF→maxDelay+critCF→muxDelay;
for each cofactors $G_i \in gList$
    if $G_i \neq critCF$
        dTarget = critDelay - $T_c$ - $G_i \rightarrow muxDelay$ ;
        upSize($G_i$, dTarget, Y);
end for
Add mux's in $G_i \in gList$ to create a complete graph G;
return G;
```

path delay distribution of the critical cofactor (i.e., CF_1). This figure clearly indicates that all critical paths of the synthesized design are limited to the critical cofactor. We also present its cofactor-wise critical path delay distribution under process variation (V_{th} variation) in Fig. 6(b). From this figure, note the following: 1) CF_1 remains critical even under parametric variation while the other cofactors remain noncritical and 2) there is a delay slack present between CF_1 and other cofactors. Also, note that the critical cofactor CF_1 is activated based on the states of four control variables. The delay distribution at reduced supply is shown in Fig. 6(c). It shows that CF_1 operates in two cycles while the rest of the cofactors operates in a single cycle.

In Fig. 7, we show the area, power, and new supply voltages of a set of MCNC benchmark circuits. It can be observed from Fig. 7(a) that supply voltages required for the designed circuits are significantly less than nominal supply (1 V). This results in 60% average saving in total power, as shown in Fig. 7(b). The partitioning is performed such that the critical cofactors of all the benchmarks are at the fourth level of hierarchy. Therefore, the activation probability of critical paths (thereby, the number of two-cycle operations) is very low. The performance penalty due to two-cycle operations is elaborated in the next section for a pipeline-based system. Fig. 7(c) shows the area overhead (18% on average) associated with the proposed design methodology. This comes from two sources: 1) logic

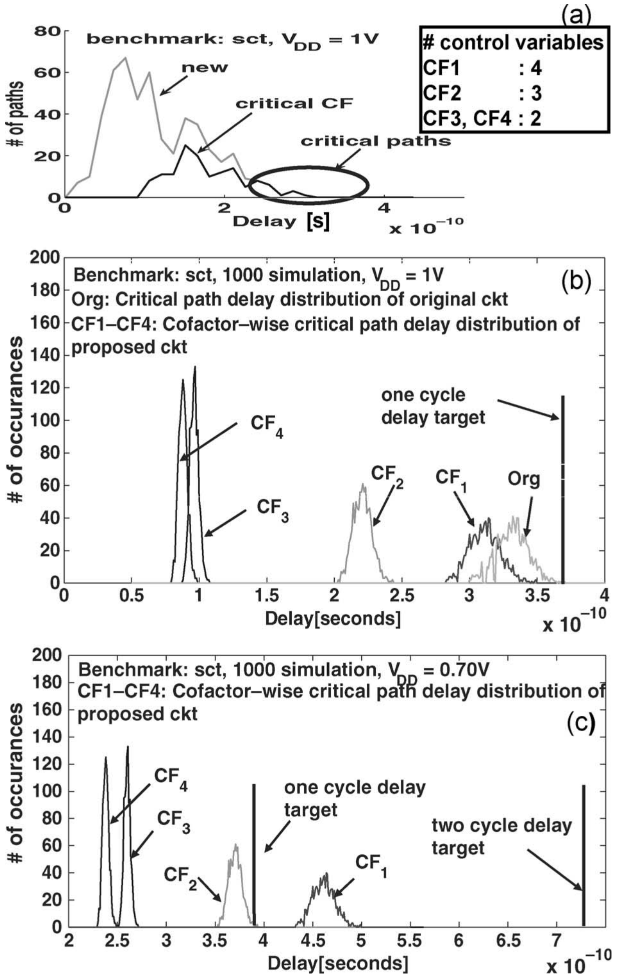

Fig. 6. Results for benchmark *sct*. (a) Path delay distribution after partitioning and sizing. (b) Cofactor-wise critical path delay distribution under V_{th} variation ($V_{\mathrm{DD}} = 1$ V). (c) $V_{\mathrm{DD}} = 0.7$ V.

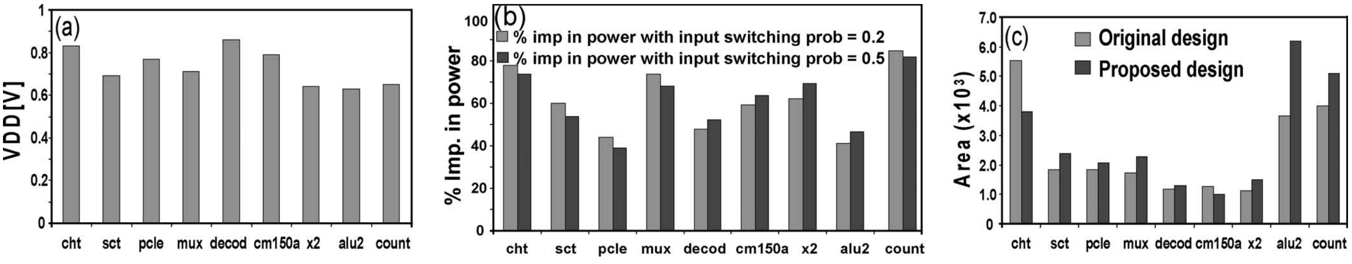

Fig. 7. (a) Supply voltage of proposed design. (b) Percent improvement in power. (c) Area overhead.

TABLE II
PROCEDURE FOR PIPILINE DESIGN

```
Procedure pipelineDesign()
Input: yield (Y), list of circuits(dList), $V_{DDL}$; //$V_{DDL}$< 1V
Output : list of re-designed circuits (dList);
1.  target delay ($T_c$) = max(stage delays);
2.  for each design $D_i \in$ dList
3.      gList = performPartitioning($D_i$, $V_{DDL}$);//Fig. 5
4       $D_i$ = performSizing(gList, $T_c$, Y, $V_{DDL}$); //Table 1
5.  end for
6.  return dList;
```

duplications during Shannon-based partitioning and 2) upsizing of certain cofactors. However, for two benchmarks, namely, *cht* and *cm150a*, we observed an area improvement. This is mainly due to better optimization after control variable isolation and multilevel Shannon partitioning [8].

IV. Application in Pipeline-Based Design

In this section, we present an application of CRISTA methodology in pipeline-based design. Here, each stage is designed using the technique explained in Section III. We also discuss the performance overhead due to pipeline stalls. This is followed by simulation results for a three-stage pipeline.

A. Pipeline Design Methodology and Performance Analysis

Our pipeline design methodology is based on a given reduced supply voltage constraint. The procedure of the aforementioned pipeline design method is shown in Table II. It takes target yield, the list of pipeline stage designs, and target supply voltage as input, applies CRISTA methodology and produces the redesigned and synthesized list of designs as output. The stage delays are computed by sizing the original design, as explained in Section III-D. The maximum stage delay is chosen as target delay (T_c) for all the stages (Step 1). Next, one design is picked at a time, and circuit partitioning is performed, as explained in Section III (Step 3). Note that the delays are computed by using SSTA at specified supply voltage $(V_{\rm DDL})$. The output of Step 3 is a list of cofactors which is sized to meet the required delay target at supplied voltage (Step 4). Steps 2–5 are repeated for each of the pipeline stages, and the list of redesigned stages is returned as output.

Next, let us evaluate the performance of the new pipeline design. Consider a three-stage linear pipeline after partitioning and synthesis [Fig. 8] where decoders D_1, D_2, and D_3 predict the activation of critical cofactors of the individual stages. A two-cycle operation is needed whenever the critical cofactor of any of the pipeline stages is activated. Under such circumstances, the pipeline is stalled by gating the clock (using signal freeze in Fig. 8). Let p_i be the activation probability of critical cofactor of ith stage, and $p_{\rm total}$ is the probability of two-cycle operation in the pipeline. Further, we assume that critical cofactors of each of the stages have the same number of control variables. So

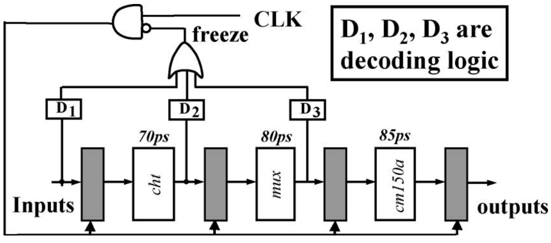

Fig. 8. Example of a pipeline design using the proposed method.

$$p_1 = p_2 = \cdots p_N = p = (\text{input switching activity})^k \quad (4)$$

where k is the hierarchy (or, number of control variables) of critical cofactor. Then, $p_{\rm total}$ is given by

$$p_{\rm total} = 1 - (1-p)^N. \quad (5)$$

If the ideal clock cycle-per instruction (CPI) of the pipeline is given by $\text{CPI}_{\rm ideal}$, then the new CPI is given by

$$\text{CPI}_{\rm new} = \text{CPI}_{\rm ideal} + p_{\rm total} \cdot (\text{stall penalty}). \quad (6)$$

The performance penalty due to occasional two-cycle operation is given by

$$\begin{aligned}\text{Perf. penalty} &= \frac{\text{CPI}_{\rm new} - \text{CPI}_{\rm ideal}}{\text{CPI}_{\rm new}} \\ &= \frac{p_{\rm total} \cdot (\text{stall penalty})}{\text{CPI}_{\rm ideal} + p_{\rm total} \cdot (\text{stall penalty})} \\ &= \frac{p_{\rm total}}{1 + p_{\rm total}}. \quad (7)\end{aligned}$$

The performance penalty for different N and input switching activities is shown in Fig. 9. In this plot, we assume that the critical cofactor of each stage is activated by four inputs (i.e., $k = 4$). It can be observed from this plot that if the control variables have low switching activities (approximately 0.1–0.3), then the penalty can be restricted within 10%. It can be noted that penalty can be large for deep pipeline designs (i.e., large N). We suggest the following techniques for reducing the performance penalty: 1) tune the control variable

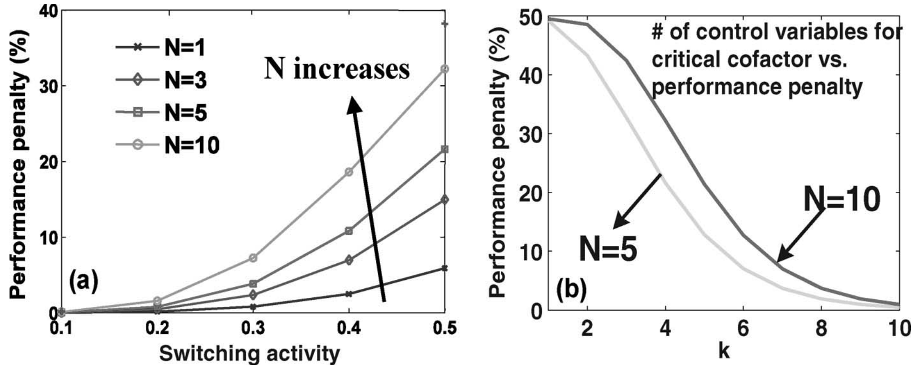

Fig. 9. Performance penalty for (a) critical cofactor at $k = 4$ and (b) different values of k.

TABLE III
SIMULATION RESULTS FOR THREE-STAGE PIPELINE

V_{DDL} (V)	% imp in power			Overall imp (%)	% area overhead			overall area penalty (%)	# cycles reqd. [*cht, mux, cm150a*]	perf. penalty(%)
	cht	*mux*	*cm150a*		*cht*	*mux*	*cm150a*			
0.75	40.06	59.70	-5.24	41.99	62.79	28.96	159.7	75.75	[1,2,2]	10.9
0.80	64.64	58.20	56.69	62.16	-10.69	15.17	11.87	0.50	[1,2,2]	10.9
0.85	**61.27**	**56.71**	**54.59**	**59.43**	**-17.51**	**5.86**	**0.76**	**-7.85**	**[1,1,2]**	**5.89**
0.90	49.49	50.74	39.63	49.05	-12.55	1.37	6.51	-5.01	[1,1,2]	5.89

selection metric to pick low switching inputs as control variables and 2) reduce the activation probability of critical blocks further (i.e., by increasing k).

B. Simulation Results

We performed the experiment on a three-stage pipeline where each stage is an MCNC benchmark circuit. The pipeline with the stage delays (for 95% yield with BPTM 70-nm devices) is shown in Fig. 8. After performing Step 1 of the *pipelineDesign()*, delay target is chosen to be 85 ps. We redesign the pipeline stages for $V_{\rm DDL}$s, ranging from 0.75 to 0.90 V. The constraints for partitioning and sizing are the same as discussed in Section III-D. After design, the entire pipeline is simulated in Hspice using 200 random test patterns with uniform switching activity of 0.5. The results are shown in Table III. Columns 2–4 in Table III show the improvement in power for each of the individual stages. The overall pipeline power saving is shown in column 5. Columns 6–8 indicate the area overheads of the individual stages, whereas column 9 shows the overall area overhead. The maximum number of clock cycles required by each stage is shown in column 10 while the performance penalty is shown in column 11.

It is interesting to note from Table III that the overall power saving increases initially but declines at lower supply voltages. This is due to increased switching capacitance to meet the delay target at low-supply voltage. The negative value of area overhead for *cht* is due to better optimization (Section III-D). It should also be noted that the critical cofactor of *cht* does not need two-cycle operations for the given range of $V_{\rm DDL}$. This is due to the increased delay target (i.e., 85 ps). Circuit *mux* may need two-cycle operations only when $V_{\rm DDL} = 0.8$ and 0.75 V (column 10). However, circuit *cm150a* may need two-cycle operations for the entire voltage range. Therefore, the pipeline performance penalty varies between 6%–11%. Table III clearly shows that it is beneficial to design the pipeline for $V_{\rm DDL} = 0.85$ V where the power saving is significant (approximately 60%) with low-performance penalty (approximately 6%). Similar technique could also be extended for any N-stage pipeline.

V. APPLICATION IN TEMPERATURE-ADAPTIVE DESIGN

In the previous section, we discussed an application of CRISTA design paradigm in pipeline-based design which allows it to consume low power while being robust to parametric variations (by adaptively operating in one cycle/two cycle). Along with power, another important problem of present day high-performance circuits is "die overheating." Techniques like logic shutdown, clock gating, simultaneous voltage–frequency (V, f) throttling, etc. [13], [14] have been proposed to address this problem. However, these techniques may be complex and may lead to pipeline stalls during the V/f tuning process. In this section, we propose an application of CRISTA which makes the circuit intrinsically suitable for DVS for thermal management with minimum performance overhead.

The main idea of temperature-adaptive circuit design is again based on CRISTA, i.e.; making the possible delay errors (under single-cycle operations) predictable and rare under parametric variations. We follow similar partitioning technique as discussed in Section III-A. However, the sizing routine is modified to attain the path delay distribution, as shown by a cartoon in Fig. 10. It is easy to observe that this kind of delay distribution can allow the circuit to operate at two different lower voltages with small performance overhead. On obtaining this delay distribution, a temperature-adaptive DVS can be performed with the following conditions: 1) at normal temperatures, the circuit operates in single cycle with nominal supply; 2) if the temperature violates threshold $T_{\rm REF1}$, a lower supply $V_{\rm DDL1}$ is applied to push the critical cofactor to two-cycle operations; and 3) if temperature crosses threshold $T_{\rm REF2} (> T_{\rm REF1})$, supply

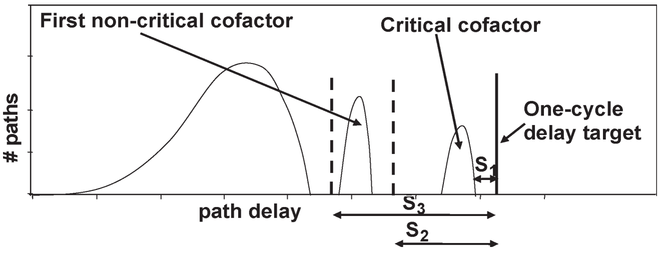

Fig. 10. Path delay distribution required for the temperature-aware design methodology with three voltage domains.

$V_{\text{DDL2}}(< V_{\text{DDL1}})$ is applied so that the first noncritical cofactor (Fig. 10) also operates in two cycles. A thermal sensor can automatically make decision about the supplies (i.e., V_{DDH}, V_{DDL1}, or V_{DDL2}) that can be applied to the circuit. The activation of critical and first noncritical cofactors is predicted by decoders.

For the temperature-adaptive design, the critical cofactor is downsized to meet slack S_1 (Fig. 10) with respect to the clock period T_{c}. Furthermore, one of the noncritical cofactors (i.e., first critical cofactor) is sized to meet a slack of S_2. All other cofactors are upsized to meet the slack S_3. The remaining steps are similar to Table I so the details are omitted.

The proposed technique has the following advantages over conventional thermal management methods (e.g., [13] and [14]): 1) it allows us to dynamically schedule the supply voltage (during overheating) without tuning the clock frequency; 2) the pipeline need not be stalled for more than two cycles at any instant (not even during the voltage assignment process); 3) the control overhead is negligible compared to (V, f) control techniques; and 4) a tradeoff can be made between temperature and performance penalty. In the following sections, first, we discuss the architecture of the temperature-adaptive pipeline design. Then, we present the simulation results.

A. Architecture of Temperature-Adaptive Pipeline Design

The architecture of CRISTA based temperature-adaptive pipeline (with three voltage domains) is shown in Fig. 11. Decoders D_{11}, D_{21}, and D_{31} determine whether the critical cofactor of one of the stages is activated. Similarly, decoders D_{12}, D_{22}, and D_{32} determine the activation of first noncritical cofactor of the stages. These two sets of decoders are ORed to predict the activation of critical cofactor and the first noncritical cofactor. The predictions are gated with the outputs of thermal sensor. The freeze signal is deactivated during the normal die temperatures. However, it is activated when the following conditions are met: 1) temperature of the chip crosses the threshold value, and 2) a two-cycle operation is predicted by the decoders. When the T_{REF1} is crossed, the supply voltage is automatically reduced by the multiplexer to V_{DDL1} where only the critical cofactor starts operating in two cycles. However, if T_{REF2} is violated, then supply voltage is reduced to V_{DDL2}, and both critical and first noncritical cofactors are operated in two cycles. Since the performance of pipeline at V_{DDL2} is lower than that of V_{DDL1}, a tradeoff can be made between temperature and performance.

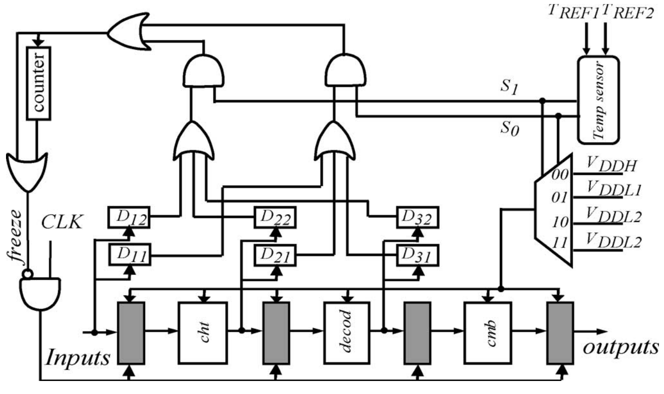

Fig. 11. Temperature-aware pipeline design (three stages with three voltage domains).

An interesting feature of our method is that the pipeline need not be stalled during DVS process. This is true because the possible delay failures are predicted ahead in time, and corrective action (i.e., two-cycle operation) is taken. However, the delay failures can occur when the supply is brought back from V_{DDL2} to V_{DDL1} or V_{DDL1} to V_{DDH}. To avoid such situations, we place a few bits counter as delay element which extends the freeze signal for a few more cycles (as shown in Fig. 11). The proposed DVS technique can also be extended for N-stage pipeline (at the cost of extra AND and OR gates).

B. Thermal Model for Electro-Thermal Simulation

To demonstrate the impact of DVS on the transient behavior of temperature, we follow the lumped RC thermal model [15]. The thermal resistance (R_{chip}) and capacitance (C_{chip}) of the circuit is given by

$$R_{\text{chip}} = \frac{\rho \cdot l}{A}; C_{\text{chip}} = c \cdot l \cdot A \tag{8}$$

where $\rho(= 10^{-2}\ \text{mK/W})$ is the thermal resistivity, $c(= 10^6\ \text{J/m}^3\text{K})$ is the thermal capacitance, $l(= 0.1\ \text{mm})$ is the thickness of the wafer, and A is the die area. The transient temperature at ith instant is determined as follows:

$$\begin{aligned} T_i &= T_{i-1} + [T_{\text{MAX}} - T_{i-1}][1 - \exp(-t/\tau_{\text{chip}}) \\ &\quad \text{if} \quad (T_{\text{ambient}} + R_{\text{chip}}P_i) > T_{i-1} \\ &= T_{i-1}\exp(-t/\tau_{\text{chip}}), \text{ otherwise} \end{aligned} \tag{9}$$

where t is the time step. $\tau_{\text{chip}} = R_{\text{chip}}C_{\text{chip}}$, and $T_{\text{MAX}} =$ maximum allowable temperature. Equation (9) is based on simple RC circuit charge/discharge expression.

C. Experimental Setup and Simulation Results

The yield targets of original circuit and the cofactors for gate sizing are set to 95%. After partitioning and sizing, the supply voltages V_{DDL1} and V_{DDL2} (for DVS) are determined such that the yields of cofactors operating on one cycle (two cycle) are 95% (100%). Once we get the new pipeline design with nominal and reduced supply voltages, we perform

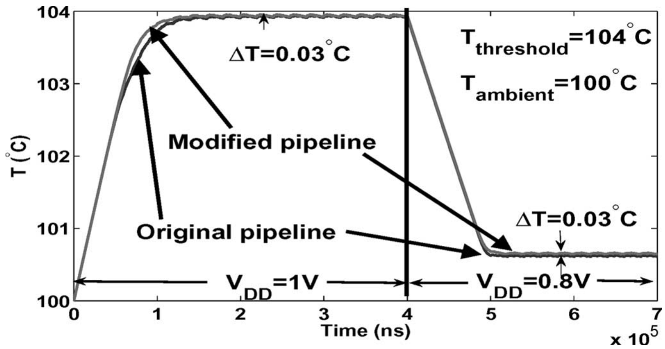

Fig. 12. Transient temperature profile for the three-stage pipeline.

electro-thermal simulation to solve the self-consistent loop of power and temperature at each time step. $R_{\rm chip}$ and $C_{\rm chip}$ of the circuit are computed by using (8). For power estimation, the circuit is simulated in Hspice at ambient temperature (100 °C). After determination of power, the transient temperature is computed using (9). The computed temperature is used for determination of power in the next time step. The $T_{\rm REF1}$ and $T_{\rm REF2}$ are kept at 104 °C and 110 °C, respectively.

The target delay of the pipeline was 100 ps, and $\{V_{\rm DDH}, V_{\rm DDL1}, V_{\rm DDL2}\}$ were found to be $\{1.0\ {\rm V}, 0.8\ {\rm V}, 0.75\ {\rm V}\}$. The thermal simulation result for our example pipeline circuit Fig. 11 is shown in Fig. 12. It can be observed that the temperature rises from the ambient value and saturates after some time. At $t = 4 \times 10^5$ ns, the temperature of the circuit crosses $T_{\rm REF1}$, and the supply voltage is reduced to $V_{\rm DDL1}$ (i.e., 0.8 V) based on the sensor's output. As expected, the temperature reduces to approximately 100.7 °C after some delay determined by the thermal RC constant of the die. At this lower supply, only the critical cofactors of the pipeline stages operate in two cycles. This simulation demonstrates the self-adaptive nature of the system with respect to temperature. For the sake of comparison, we also plotted the temperature profile of the pipeline with conventionally synthesized circuits. It can be noted from the figure that the final temperature of original pipeline is lower than the redesigned pipeline by approximately 0.03 °C. This difference is accounted by extra area overhead of redesigned circuit (approximately 22%) compared to the original circuit. This raises $R_{\rm chip}$ and thereby the steady-state temperature by a small fraction (as $T_{\rm final} = T_{\rm ambient} + {\rm PR}_{\rm chip}$). However, the main difference lies in terms of performance. In the event of high temperature, if the supply of original pipeline is also reduced to $V_{\rm DDL1}$, then it would require two-cycle operations for each computation (unless the operating frequency is also simultaneously reduced) leading to large [as much as 50% as $p_{\rm total} = 1$ in (7)] performance penalty. On the other hand, the proposed design needs only occasional two-cycle operations at $V_{\rm DDL1}$. The performance penalty at $V_{\rm DDL1}$ was found to be 11% (for input switching activities of 0.5).

VI. Practical Issues and Challenges

In this section, we address some of the practical challenges and other issues associated with the CRISTA design methodology.

A. Application in Multiple VDD Islands

Multiple VDD islands with critical paths can be handled assuming that a single frequency of operation for the different islands is used. The proposed technique can be used for such situations by considering voltage domains in timing analysis and gate sizing during the synthesis process.

B. Complexity of Decode Logic

The decode logic (for predicting the activation of critical paths) is nothing but a set of few NAND and NOR gates (since only four primary inputs are being decoded for each stage in our simulations). Furthermore, only the critical cofactor (and secondary critical cofactor, in case of temperature-aware design) requires the predecoders. Therefore, the decoding overhead is very minimal.

C. Considerations for Signal Probability of Input Literals

If the signal probabilities are available, then control variable selection metric should take it into account during partitioning (i.e., by picking the variable with smallest signal probability). If critical paths are not isolated properly due to this modification in control variable selection, then gate sizing can be used as an additional tool for further isolation.

D. Application in Full Chip Synthesis

We demonstrated the application of the proposed technique for random logic circuits. However, only specific circuits consume more power in a chip (e.g., pipelines). Therefore, we believe that the proposed approach can be more suitable for those power hungry portions rather than on a full chip scale.

E. Test Implications

Since we isolate the critical paths and make sure that they meet 100% yield target under two-cycle delay constraint by design, the need to test critical paths is minimal. The OFF-critical paths, on the other hand, are designed to meet a user-defined yield target under one-cycle delay constraints. Therefore, delay testing should be exercised only on the OFF-critical paths at the lower supply voltage. This is contradictory to the conventional delay testing strategy where the testing is performed on timing critical paths. In general, the proposed approach improves testability of internal nodes [16].

VII. Conclusion

We have proposed CRISTA, a new design paradigm based on critical path isolation, which achieves low-power operation while being robust with respect to parametric delay failures. CRISTA makes the possible delay errors (that may occur under single-cycle operations due to critical paths) predictable and rare under parametric variations. The critical paths have been isolated to a known logic block by Shannon partitioning and gate sizing. This leads to a robust circuit design which allows us to reduce the supply voltage aggressively while using the predictability to prevent occurrence of any delay violations. Simulation of circuits designed using this methodology show 60% improvement in power. We have demonstrated that this technique can be effectively applied to low-power pipeline design. We have also shown that the proposed design methodology can be used to design temperature-adaptive circuits, which can maintain a target temperature with small performance penalty.

References

[1] A. Srivastava, D. Sylvester, and D. Blaauw, "Statistical optimization of leakage power considering process variations using dual-V_{th} and sizing," in *Proc. Des. Autom. Conf.*, 2004, pp. 773–778.

[2] S. Borkar, T. Karnik, and V. De, "Design and reliability challenges in nanometer technologies," in *Proc. Des. Autom. Conf.*, 2004, p. 75.

[3] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Proc. IEEE MICRO*, 2003, pp. 7–18.

[4] X. Bai, C. Visweswariah, P. N. Strenski, and D. J. Hathaway, "Uncertainty-aware circuit optimization," in *Proc. Des. Autom. Conf.*, 2002, pp. 58–63.

[5] J. M. Rabaey, *Digital Integrated Circuits*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[6] *BPTM 70 nm: Berkeley Predictive Technology Model*. [Online]. Available: www.eas.asu.edu/~ptm

[7] L. Lavagno, P. C. McGeer, A. Saldanha, and A. L. Sangiovanni-Vincentelli, "Timed Shannon circuits: A power-efficient design style and synthesis tool," in *Proc. Des. Autom. Conf.*, 1995, pp. 254–260.

[8] S. Bhunia, N. Banerjee, Q. Chen, H. Mahmoodi, and K. Roy, "A novel synthesis approach for active leakage power reduction using dynamic supply gating," in *Proc. Des. Autom. Conf.*, 2005, pp. 479–484.

[9] S. Kundu, S. M. Reddy, and N. K. Jha, "Design of robustly testable combinational logic circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 10, no. 8, pp. 1036–1048, Aug. 1991.

[10] 2006*Synopsys Design Compiler*. (2006). [Online]. Available: www.synopsys.com

[11] K. Kang, B. C. Paul, and K. Roy, "Statistical timing analysis using levelized covariance propagation," in *Proc. Des. Autom. Test Eur.*, 2005, pp. 764–769.

[12] S. H. Choi, B. C. Paul, and K. Roy, "Novel sizing algorithm for yield improvement under process variation in nanometer," in *Proc. Des. Autom. Conf.*, 2004, pp. 454–459.

[13] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture: Extended discussion and results," Univ. Virginia, Charlottesville. Tech. Rep. 2003.

[14] C. Poirier, R. McGowen, C. Bostak, and S. Naffziger, "Power and temperature control on a 90 nm Itanium-family processor," *J. Solid-State Circuits*, vol. 41, no. 1, pp. 229–237, Jan. 2006.

[15] K. Skadron, T. Abdelzaher, and M. R. Stan, "Control-theoretic techniques and thermal-RC modeling for accurate and localized dynamic thermal management," in *Proc. Int. Symp. High-Performance Comput. Archit.*, 2002, pp. 17–28.

[16] S. Ghosh, S. Bhunia, and K. Roy, "Shannon expansion based supply-gated logic for improved power and testability," in *Proc. Asian Test Symp.*, 2005, pp. 404–409.

Swaroop Ghosh received the B.E. degree (Hons.) in electrical engineering from Indian Institute of Technology, Roorkee, India, in 2000 and the M.S. degree from the University of Cincinnati, Cincinnati, OH, in 2004. He is currently working toward the Ph.D. degree at the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN.

From 2000 to 2002, he was with Mindtree Technologies Pvt. Ltd., Bangalore, India, as a Very-Large-Scale-Integration Design Engineer. He spent the summer of 2006 in Intel's Test Technology Group. His research interests include low-power process-tolerant circuit and system design, fault-tolerant design, and digital testing for nanometer technologies.

Swarup Bhunia (S'00–M'05) received the B.E. degree (Hons.) from Jadavpur University, Kolkata, India, in 1995, and the M.Tech. degree from the Indian Institute of Technology, Kharagpur, India, in 1997. He received the Ph.D. from Purdue University, West Lafayette, IN, in 2005.

Currently, he is an Assistant Professor of electrical engineering and computer science at Case Western Reserve University, Cleveland, OH. He has worked in the semiconductor industry on synthesis, verification, and low-power design for about three years.

Dr. Bhunia received the 2005 SRC Technical Excellence Award as a team member, Best Paper Award in International Conference on Computer Design (ICCD 2004), Best Paper Award in Latin American Test Workshop (LATW 2003), and Best Paper Nomination in Asia and South Pacific Design Automation Conference (ASP-DAC 2006). He has served in the Technical Program Committee, Design Automation and Test Conference in Europe (DATE 2006–2007), International Symposium on Low Power Electronics and Design (ISLPED 2007), Test Technology Educational Program (TTEP 2006–2007), and in the Program Committee of International Online Test Symposium (IOLTS 2005).

Kaushik Roy (S'83–M'95–SM'95–F'02) received the B.Tech. degree in electronics and electrical communications engineering from the Indian Institute of Technology, Kharagpur, India, and the Ph.D. degree from the Electrical and Computer Engineering Department, University of Illinois, Urbana–Champaign, in 1990.

He was with the Semiconductor Process and Design Center, Texas Instruments, Dallas, where he worked on field-programmable gate array architecture development and low-power circuit design. He was with the Electrical and Computer Engineering Faculty, Purdue University, West Lafayette, IN, in 1993, where he is currently a Professor and holds the Roscoe H. George Chair of Electrical and Computer Engineering. His research interests include very-large-scale-integration (VLSI) design/computer-aided design for nanoscale silicon and nonsilicon technologies, low-power electronics for portable computing and wireless communications, VLSI testing and verification, and reconfigurable computing. He has published more than 400 papers in refereed journals and conferences, is the holder of eight patents, and is coauthor of two books on low-power CMOS VLSI design.

Dr. Roy received the National Science Foundation Career Development Award in 1995, IBM Faculty Partnership Award, ATT/Lucent Foundation Award, 2005 SRC Technical Excellence Award, SRC Inventors Award, and Best Paper Awards at the 1997 International Test Conference, IEEE 2000 International Symposium on Quality of IC Design, 2003 IEEE Latin American Test Workshop, 2003 IEEE Nano, 2004 IEEE International Conference on Computer Design, 2006 IEEE/ACM International Symposium on Low Power Electronics and Design, and 2005 IEEE Circuits and System Society Outstanding Young Author Award (Chris Kim), 2006 IEEE Transactions on VLSI Systems Best Paper Award. He is a Purdue University Faculty Scholar, the Chief Technical Advisor of Zenasis Inc., and Research Visionary Board Member of Motorola Laboratories (2002). He has been in the editorial board of IEEE DESIGN AND TEST, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, and IEEE TRANSACTIONS ON VLSI SYSTEMS. He was Guest Editor for a special issue on low-power VLSI in the IEEE DESIGN AND TEST (1994) and IEEE TRANSACTIONS ON VLSI SYSTEMS (June 2000), and *IEE Proceedings—Computers and Digital Techniques* (July 2002).