Low-Overhead Circuit Synthesis for Temperature Adaptation Using Dynamic Voltage Scheduling

Swaroop Ghosh, Swarup Bhunia*, and, Kaushik Roy
School of Electrical and Computer Engineering, Purdue University, IN, USA
*Electrical Engineering and Computer Science, Case Western Reserve University, OH, USA

***Abstract*—Increasing power density causes die overheating due to limited cooling capacity of the package. Conventional thermal management techniques e.g. logic shutdown, clock gating, frequency scaling, simultaneous voltage-frequency tuning etc. increase the design complexity and/or degrade the performance significantly. In this paper, we propose a novel design technique, which makes a circuit amenable to temperature adaptation using dynamic voltage scheduling (DVS). It is accomplished by a synthesis technique that (a) *isolates and predicts* the set of paths that may become critical under variations, (b) ensures they are *activated rarely*, and (c) *tolerates* possible delay failures (at reduced voltage) in these paths by *adaptive clock stretching*. This allows us to schedule a lower supply voltage during increased temperature without requiring frequency tuning. Simulation results on an example pipeline show that proposed design yields similar temperature reduction as conventional design with only 11% performance penalty and 14% area overhead. The conventional pipeline design, on contrary, leads to 50% performance degradation due to reduced operating frequency.**

I. INTRODUCTION

Aggressive technology scaling has been the driving force of semiconductor industry for decades to achieve nearly exponential growth in computational power and device integration. However, a major design challenge that has emerged with technology scaling is the increasing power density of a chip (due to integration of large number of devices in unit area as well as increased dynamic and leakage power [1]). The increased power density translates to excessive heat generation while the cooling capability of the package remains limited; giving rise to elevation of die temperature as well as localized heating of a chip (called "hot-spots") [2].

Traditionally, a circuit is designed to operate at worst-case temperature which is inefficient. Dynamic thermal management techniques like logic shutdown, clock gating, frequency scaling [1, 2] etc. have been proposed in the past. Although these techniques are capable of bringing down power density and temperature, they also degrade the performance considerably. Dynamic voltage scaling can be an effective way to reduce temperature in the event of overheating because it reduces both the switching as well as leakage power. However, if the clock frequency is not scaled simultaneously, then increased delay (due to lower voltage) may result into wrong computation. Combined *V-f* control technique [3] reduces leads to cubic benefit in terms of power saving at the cost of complex control circuitry. In [3], a dedicated micro-controller is used for *V-f* control.

In this paper, we present a novel design technique which improves the opportunity for aggressive DVS during elevated temperature with small area and performance overhead. We achieve this by employing *critical path isolation* technique [4]. The notion of critical path isolation indicates the confinement of critical paths in a *synthesized design* to known logic blocks. The proposed *design methodology isolates the critical paths and makes them predictable and rare even under variations. Possible delay errors are predicted ahead of time (by decoding few input bits) and avoided by stretching the clock to two clock cycles (assuming all standard operations are single cycle). This lets us scale the supply voltage aggressively for thermal management without large performance penalty or complex logic for frequency tuning.* We propose an automatic synthesis process based on the above design concept. The synthesis process isolates the critical paths and makes them predictable (based on few primary inputs).The isolated critical paths operate with *single cycle* at nominal supply voltage under nominal temperature. However, at increased temperature in a thermally constraint environment, the supply voltage is reduced and since possible delay errors in critical paths are deterministic, they are avoided by *stretching the clock to two-cycles* on critical path activation. The synthesis process restricts occurrences of the above *two-cycle* operations by reducing critical path activation probability. It also increases the delay margin between critical and non-critical paths by both logic synthesis and proper gate sizing to allow aggressive DVS at elevated temperature while maintaining good performance.

The design technique presented in [4] also relies on the idea of critical path isolation. However, the objective of [4] is to re-design the circuit for operation at *fixed low supply voltage* with occasional *clock stretchings*. In this work, the circuit is synthesized in a way that it operates in *single-cycle* at normal temperature (and *nominal supply*) while at high temperature, it operates at lower supply voltage and maintaining frequency with occasional *clock stretchings*. Consequently, reduction in temperature can be achieved with small performance overhead. Possible delay errors due to critical paths are deterministic and can be avoided by *stretching the clock to two cycles*.

The proposed technique has following major advantages over existing thermal management methods [1, 2, 3]: (a) it allows us to dynamically schedule the supply voltage during overheating without tuning the clock frequency; (b) the pipeline need not be stalled for more than *two-cycles* (not even during the DVS process); and (c) the control overhead is negligible compared to existing (*V, f*) control techniques.

II. PRELIMINARY ANALYSIS

In this section, first we present a pipeline design to illustrate the proposed DVS approach for thermal management. Next, we present a design flow that extends the approach to random logic circuits.

A. Voltage scaling and clock stretching in a pipeline design

For the sake of simplicity, we choose two 4-bit ripple carry adders (RCA) as pipeline stages (Fig. 1) to illustrate the concepts. If P_0-P_3 are the propagate signals ($P_i = A_i + B_i$) and $C_{i,0}$ is carry-in signal then the critical path is activated only when $P_0P_1P_2P_3C_{i,0} = 1$. Since the probability of such occurrence is very low, one can scale down the supply voltage when the circuit is overheated (to reduce the temperature) such that; (a) all operations with $P_0P_1P_2P_3C_{i,0} = 0$ can still be performed in one-cycle; (b) during critical path activation, the possible delay error due to scaled supply can be avoided by performing addition in two clock cycles. The activation of critical path can be predicted by pre-computation of $P_0P_1P_2P_3C_{i,0}$. Note that this approach incurs penalty of an extra clock cycle when the set of critical paths are activated at low supply voltage. However, by ensuring small activation probability of critical paths, it may be possible to maintain temperature by rarely paying penalty of an extra clock cycle.

To evaluate the feasibility of this idea, we simulated the pipeline (Fig. 1) with V_{DD} = 1V in Hspice using BPTM 70nm devices [5]. The critical path delay of the RCAs were found to be 260ps. Based on the critical path delays, we selected clock period to be 270ps. The average power dissipation of the entire design was 26.1uW. Assuming clock period to be 270ps, we reduced the supply voltage to 0.80V. The actual critical path delays of RCAs were found to be 330ps with scaled supply. The new power dissipation was 14.7uW (44% saving). The performance penalty due to *two-cycle* operations (by clock stretching) of pipeline stages at 0.8V is observed to be small (about 5.8% assuming 50% switching activity of P_0-P_3 and $C_{i,0}$) and may occur only during overheating of the chip.

978-3-9810801-2-4/DATE07 © 2007 EDAA

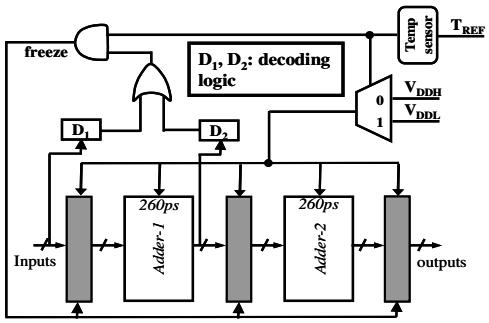

Fig. 1: An example of temperature-aware pipeline design

The overall DVS scheme (Fig. 1) contains two decoders D_1 and D_2 (essentially, few AND gates) to predict the activation of critical paths of *adder-1* and *adder-2* stages. A temperature sensor [6] indicates if the current die temperature violates the thermal constraint. Under nominal temperatures, the pipeline operates in *one-cycle*. However, during elevated die temperature (> reference temperature T_{REF}), the output of the sensor becomes '1' and the supply voltage is reduced from V_{DDH} to V_{DDL}. Under this condition, the pipeline is stalled (controlled by signal *freeze*) whenever the critical path activation is predicted by D_1 or D_2. Once the temperature drops below T_{REF}, the supply is raised back to nominal value and the pipeline operates in nominal condition.

B. Design flow

In subsection IIA, we presented the idea of temperature-aware DVS for a pipeline, where the critical path in a stage was unique. To exercise similar DVS technique in random logic circuits (usually with skewed path distribution as shown in Fig. 2(a)), we need to ensure that, (a) the critical paths are confined to a logic section whose activation can be predicted; and, (b) the non-critical paths remain non-critical under process, temperature and supply voltage variations by maintaining a safe timing margin. Fig. 2(b) illustrates such a path delay distribution.

To obtain the delay distribution shown in Fig. 2(b), the design needs to be partitioned and synthesized in such a way that the paths are divided into several logic blocks so that; (a) these logic blocks are active or remain idle, based on the state of primary inputs; and, (b) 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 just by decoding the states of certain inputs. Next, gate sizing is performed on the partitioned logic to maximize the slack between critical and non-critical blocks. Finally, DVS can be performed during overheating such that *non-critical* blocks meet the desired timing yield with respect to *one-cycle* delay target whereas *critical* block meet the yield with respect to *stretched clock* period. As illustrated in Fig. 2(b), slack S_2 can be utilized to assign low supply voltage at elevated temperature. Under low voltage, the *critical block* will operate in *stretched cycle* while the *non-critical* ones will operate in *single-cycle* with little performance degradation (since activation probability of the *critical block* is very low).

Note that the delay distribution as shown in Fig. 2(b) allows us to assign only a single lower voltage during elevated temperature. However, there can be a situation where the die may require further lower voltages (at the cost of more performance degradation). To encounter such circumstances, one may modify the sizing strategy after partitioning to obtain delay distribution as depicted in Fig. 2(c). This allows assignment of two lower voltages ($V_{DDL2} < V_{DDL1}$). At V_{DDH} (i.e., nominal voltage), there is no performance penalty. However, at V_{DDL1} the design has a small performance penalty (p') due to occasional *clock stretching* operations of *critical block*. At V_{DDL2}, the design has performance penalty p'' ($>p'$).

The overall design flow is shown in Fig. 3. Next, we present a mathematical analysis to obtain the conditions under which the proposed temperature-aware design approach can be advantageous compared to conventional design running at half clock frequency during low voltage in the event of thermal emergency.

C. Analysis of the proposed design methodology

Let us consider two designs for a circuit namely, *design-A* and *design-B* with timings as shown in Fig. 2(a) and (c). *Design-A* is conventional design whereas *design-B* is the proposed design. Critical paths of both *design-A* and *design-B* have a slack of S_1 (at voltage V_{DDH}) with respect to the clock period T_c (Fig. 2(a) and (c)). However, the critical paths of *design-B* are confined to a logic block whose activation can be predicted. A set of non-critical paths (referred as "*secondary critical paths*") in *design-B* maintain a slack of S_2 (Fig. 2(c)). Note that the activation of these paths can also be predicted based on the states of few inputs. The other paths of *design-B* have a slack of S_3. Two extra decoders are needed in *design-B* for pre-determining the occurrences of *critical* and *secondary critical* path activation. Let us now compare the power dissipation (and subsequently temperature) of *design-A* and *design-B*. In this analysis, we assume that *design-A* operates with half clock frequency at reduced voltage. Since voltage is proportional to $(delay)^{-1}$, the scaled voltage (V_{DDL}^{A}) for *design-A* is given by,

$$V_{DDH} \propto \frac{1}{T_c - S_1} \text{ and, } V_{DDL}^{A} \propto \frac{1}{2T_c} \quad \Rightarrow V_{DDL}^{A} = \frac{V_{DDH}}{2}\left(1 - \frac{S_1}{T_c}\right) \tag{1}$$

$$\text{Similarly, } V_{DDL1}^{B} = V_{DDH}\left(1 - \frac{S_2}{T_c}\right) \text{ and, } V_{DDL2}^{B} = V_{DDH}\left(1 - \frac{S_3}{T_c}\right) \tag{2}$$

If the performance penalty due to clock stretching in *design-B* at V_{DDL1}^{B} and V_{DDL1}^{B} is p' and p'' respectively, then change in steady state temperature from ambient value ($\Delta Temp = P.R_{chip}$ where P is power and R_{chip} ($=\rho.l/A$) is thermal resistance as mentioned in equation (14)) of both designs are given by

$$\begin{aligned} \Delta Temp^{A} &= C^{A}(V_{DDL}^{A})^2\left(\frac{1}{2T_c}\right)R^{A} \propto \frac{(V_{DDL}^{A})^2}{2T_c} \text{ as } C \propto A \text{ and } R \propto \frac{1}{A} \\ \Delta Temp_1^{B} &= (C^{B} + C^{d})(V_{DDL1}^{B})^2\left(\frac{1}{T_c + p'T_c}\right)R^{B} \propto \frac{(V_{DDL1}^{B})^2}{T_c(1+p')} \\ \Delta Temp_2^{B} &= (C^{B} + C^{d})(V_{DDL2}^{B})^2\left(\frac{1}{T_c + p''T_c}\right)R^{B} \propto \frac{(V_{DDL2}^{B})^2}{T_c(1+p'')} \end{aligned} \tag{3}$$

where C^A (C^B) is average switched capacitance, R^A (R^B) is thermal resistance of *design-A* (*design-B*) and C^d is the average switched

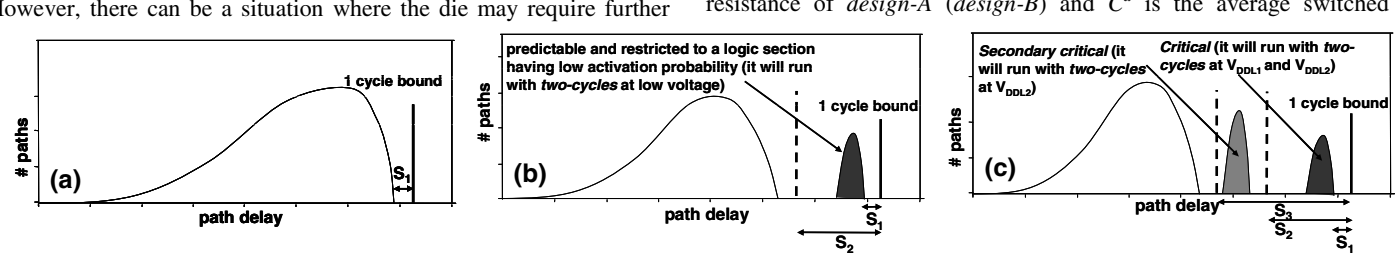

Fig. 2: Path delay distribution: (a) random logic circuit; (b) proposed design methodology with two voltages; and, (c) proposed design methodology with three voltages

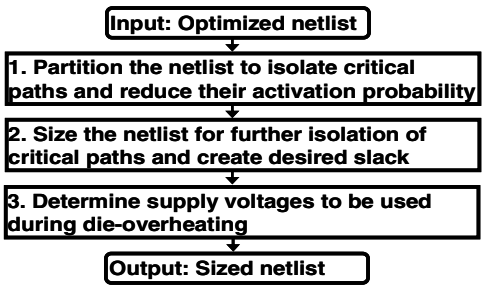

Fig. 3: Design methodology

capacitance of decoding logic (to determine critical path activation).

The $\Delta Temp$ ratio is given by

$$\frac{\Delta Temp^A}{\Delta Temp_1^B} = \frac{(V_{DDL}^A)^2(1+p')T_c}{2(V_{DDL1}^B)^2 T_c} = \left(\frac{V_{DDL}^A}{V_{DDL1}^B}\right)^2 \left(\frac{1+p'}{2}\right)$$

Putting the values of V_{DDL}^A and V_{DDL1}^B, and rearranging we get,

$$\frac{\Delta Temp^A}{\Delta Temp_1^B} = \left(\frac{1-\frac{S_1}{T_c}}{1-\frac{S_2}{T_c}}\right)^2 \left(\frac{1+p'}{8}\right) = \left(\frac{1-S_{1norm}}{1-S_{2norm}}\right)^2 \left(\frac{1+p'}{8}\right) \quad (4)$$

Similarly,

$$\frac{\Delta Temp^A}{\Delta Temp_2^B} = \left(\frac{1-S_{1norm}}{1-S_{3norm}}\right)^2 \left(\frac{1+p''}{8}\right) \quad (5)$$

where S_{1norm} (S_{2norm}, S_{3norm}) is the normalized slack of *design-A* (*design-B*).

From the above discussion, it is obvious that *design-B* can be effective for reducing temperature than *design-A* if $Temp^A / Temp_1^B \geq 1$ and $Temp^A / Temp_2^B \geq 1$. Since $(1+p') < 2$ and $(1+p'') < 2$, a necessary condition for *design-B* to be better or equal to *design-A* is $S_{3norm} \gg S_{2norm} \gg S_{1norm}$ *i.e.*, $S_3 \gg S_2 \gg S_1$. Therefore, a larger value of S_2 and S_3 is better for lowering the temperature. However, the upper bound of S_2 and S_3 is determined by S_2, $S_3 < T_c$ and the area overhead to achieve it. From first order analysis, conventional design may appear to be better than proposed design in terms of temperature reduction. However, the issues involved in *design-A* are as follows: (a) it degrades the performance more than the proposed design ($2T_c$ vs. $T_c(1+p')$ or $T_c(1+p'')$); (b) it may not be possible to scale the supply beyond $3V_{th}$ limit due to increased delay spread at low voltages, and (c) it may be possible to sufficiently reduce temperature in a thermally constrained environment by small supply voltage reduction.

III. DESIGN METHODOLOGY

Based on the analysis presented above, here we describe in details each step of the design flow (Fig. 3) and present simulation results.

A. Circuit 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, 7, 8] 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:

$$\begin{aligned} f(x_1,...,x_i,..., x_n) &= x_i.f(x_1,...,x_i=1,..., x_n) + \bar{x}_i.f(x_1,...,x_i=0,..., x_n) \\ &= x_i.CF_1 + \bar{x}_i.CF_2 \end{aligned} \quad (6)$$

$$CF_1 = f(x_1,...,x_i=1,..., x_n); \quad CF_2 = f(x_1,...,x_i=0,..., x_n)$$

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 sub-expressions independent of control variable x_i, then we may also have a *Shared Cofactor*. Note that, in equation (6), the activation probability of each cofactor is 50%. However, *by performing multi-level expansion, it is possible to reduce the activation probability of the resulting cofactors further*. For example, a 2nd level expansion of f (equation (7)) reduces activation probability of resulting cofactors to 25%.

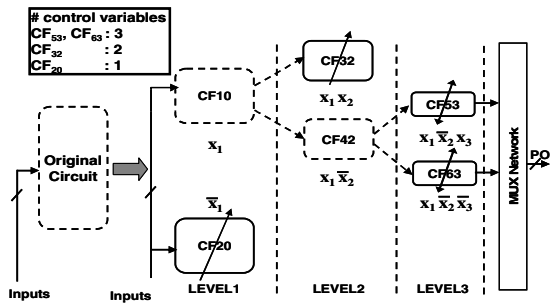

Fig. 4: Hierarchical expansion and sizing of cofactors

$$f(x_1,...,x_i,..., x_n) = x_i x_j.CF_1 + x_i\bar{x}_j.CF_2 + \bar{x}_i x_j.CF_3 + \bar{x}_i\bar{x}_j.CF_4 \quad (7)$$

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 the *critical function* (*or output in f*), then following metric can be used for critical path isolation of the circuit after expansion [4]:

$$M_i = \frac{|a_i - b_i|}{\max(a_i, b_i)} \quad (8)$$

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 marked for further expansion (to reduce the activation probability of critical paths). The process is repeated under a given area and delay constraint.

B. Gate sizing for further isolation and creation of slack

Once the critical paths are isolated to a cofactor with small activation probability, the next step is to size the resulting cofactors individually to (a) further isolate the critical paths and, (b) create sufficient timing slack between *critical* and *non-critical* cofactors to achieve the desired delay distribution (Fig. 2(c)). To achieve it, the *critical* cofactor is downsized to meet a slack of S_1 with respect to the clock period T_c. Further, *secondary critical cofactor* is sized to meet a slack of S_2. All other cofactors are upsized to meet the slack S_3 (where $S_3 > S_2$). An example of the proposed sizing approach after partitioning is shown in Fig. 4. The original circuit is partitioned into four cofactors, CF_{20}, CF_{32}, CF_{53} and CF_{63}. Cofactors CF_{53} and CF_{63} are at the highest level of hierarchy (or # of control variables). CF_{53} is downsized selectively to make it further *critical* while CF_{63} is upsized selectively to make it *secondary-critical*. The other cofactors are upsized aggressively under area constraint to make them more *non-critical*. Note that, in this work *hierarchy* and *# of control variables* are used interchangeably.

A Lagrangian Relaxation (LR) based gate sizing [9] is used in the proposed sizing procedure (Table 1). 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 highest level of hierarchy, *maxLevel* (Step 1). To compute the path delays, statistical static timing analysis (SSTA) [10] is performed (Step 2) on each of the cofactors from *gList*. In Step 3, a cofactor at *maxLevel* with critical paths is selected as *critical block* candidate. Next a cofactor with hierarchy equal or one less than *maxLevel* is selected as *secondary critical* cofactor (*scritCF*) (Step 4). Now, the *critical* (*secondary critical*) logic block candidates are *downsized* (*upsized*) selectively based on their slack requirements (Step 5 to 8). Note that the *mux* delays are subtracted from the delay target to derive the cofactor delay targets. The *non-critical* cofactors are selectively upsized while

TABLE-1

	Procedure performSizing()
	Input : target delay (T_c), target yield (Y), list of cofactors (*gList*), slacks S_1, S_2 and S_3;
	Output : sized netlist;
1	*maxLevel* = maximum hierarchy of the cofactors in *gList* ;
2	run SSTA on $G_i \in gList$;
3	*critCF* = cofactor at *maxLevel* hierarchy;
4	*scritCF*=cofactor at *maxLevel* or (*maxLevel-1*) hierarchy;
5	$dTarget = T_c - critCF \rightarrow muxDelay - S_1$;
6	***downSize***(*critCF, dTarget, Y*);
7	$dTarget = T_c - scritCF \rightarrow muxDelay - S_2$;
8	***upSize***(*scritCF, dTarget, Y*);
9	**for each** cofactors $G_i \in gList$
10	if G_i *critCF or noncritCF*
11	$dTarget = T_c - G_i \rightarrow muxDelay - S_3$;
12	***upSize***(G_i, *dTarget, Y*);
13	**end for**
14	Add mux's in $G_i \in gList$ to create a complete graph *G;*
15	**return** *G*;

meeting the yield target (Steps 9 to 13). Finally, a complete graph *G* is created and returned (Step 14 and 15).

C. Determination of supply voltages

After circuit partitioning and sizing, we obtain a netlist with defined *critical*, *secondary critical* and *non-critical* logic blocks. The last objective is to determine supply voltages V_{DDL1} and V_{DDL2} for DVS. V_{DDL1} is determined by scaling down the supply voltage and performing SSTA till the point where delay of *secondary critical* cofactor meets the yield constraint for *one-cycle* delay target (T_c). Similarly, V_{DDL2} is determined by scaling the supply voltage until the delay of *non-critical* cofactors meets the yield constraint for target T_c.

D. Simulation results and discussion

The experiments are performed on a set of MCNC benchmark circuits. We used Synopsys Design Compiler [11] for logic optimization in our synthesis flow. 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 70nm technology with 1V nominal supply voltage. The delay target (T_c) for sizing procedure is chosen by plotting the area-delay curve of the circuit and selecting the knee point delay. Slack targets S_1, S_2 and S_3 for sizing are chosen to be $0.01T_c$, $0.3T_c$ and $0.4T_c$ respectively. The area and delay constraints for Shannon partitioning are kept 40% and 5% more than actual area and delay of the circuit respectively, while 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* or *stretched cycle* at lower supply voltages during DVS is also fixed to 95%. Reduced voltage of original circuit is determined by iteratively reducing the supply voltage to meet 95% timing yield corresponding to half frequency. Supply voltages V_{DDL1} and V_{DDL2} for proposed circuit are determined as described in previous subsection.

The path delay distribution of two benchmarks (*cht* and *cmb*) after partitioning and sizing is shown (Fig. 5). *Org* and *New* represent path delay distribution of the original and the proposed design, respectively. The path distribution of *critical* cofactor is also plotted indicating that critical paths are isolated to *critical* cofactor. Note that the *critical* cofactor is activated by 4 control variables whereas *secondary critical* cofactor is activated by 3 control variables. The delays of *secondary critical* cofactor and *non-critical* cofactors normalized with respect to *critical* cofactor delay of the benchmarks are shown in Fig 6(a). The corresponding low supply voltages for conventional design (V_{DDL}) and

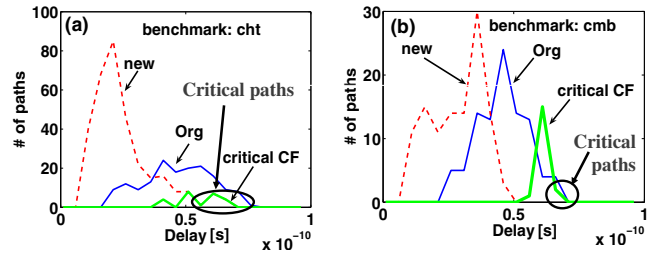

Fig. 5: Path delay distribution for (a) cht; (b) cmb

proposed design (V_{DDL1}, V_{DDL2}) are also shown (Fig 6(b)) for 1V nominal supply voltage.

In Fig. 6(c), we compare the steady-state temperature of original circuit running at nominal voltage (with *rated* frequency) and reduced voltage (V_{DDL} with half frequency) with proposed circuit running at V_{DDL1} and V_{DDL2}. The temperature is given by $T=T_{ambient}+PR_{chip}$ where P is power, R_{chip} is thermal resistance (given by $\rho.l/A$) and $T_{ambient}$ is ambient temperature. In our simulation, we assumed $\rho = 10^{-2}$ *mK/W*, $l = 0.1mm$ [12] and $T_{ambient} = 100^oC$. Power is computed in Hspice by application of 200 random patterns. Fig. 6(c) indicates that proposed design (i.e., *New*) running at V_{DDL1} and V_{DDL2} can reduce the temperature similar to *Org* running at V_{DDL} except for benchmarks *cmb* and *alu2* (due to small available slacks for DVS, Fig. 6(a)). Note that, the temperature changes are small in this experimentation due to small benchmark sizes and corresponding values of power.

The average temperature reduction for *Org*(V_{DDL}), *New*(V_{DDL1}) and *New*(V_{DDL2}) is similar (i.e., 0.78°C, 0.70°C and 0.80°C, respectively). However, *Org*(V_{DDL}) incurs 50% performance degradation whereas *New*(V_{DDL1}) and *New*(V_{DDL2}) incurs only 5.8% and 15.2% degradation respectively, assuming 50% switching activity of control variables. The penalty reduces to 0.16% and 1.47% if switching activity is 20%. The average area overhead of the proposed methodology is about 14% (Fig. 6(d)).

IV. TEMERATURE-ADAPTIVE PIPELINE DESIGN

In Section III, we presented a design methodology for random logic circuits to achieve temperature adaptation. In this section, first we present its application to pipeline-based design where each stage is designed using this methodology and discuss the performance overhead due to pipeline stalls for stretching the clock to two-cycles during die overheating. Next, we propose an architecture suitable for performing temperature-adaptive DVS. Finally, we present simulation results for a 3-stage pipeline design.

A. Performance analysis

Consider an N-stage linear pipeline where each stage is designed using the proposed technique. The pipeline is stalled for an extra cycle whenever a *clock stretching* is performed by one or more stages in the current cycle. Probability of pipeline stall (p_{total}) is given by

$$p_{total} = 1-(1-p_1)(1-p_2)...(1-p_N) \tag{9}$$

where p_i is the activation probability of *critical* block of ith stage. If *critical* block of each stage has same number of control variables

$$p_1 = p_2 = ...p_N = p' = (\text{input switching activity})^k \tag{10}$$

where *k* is the hierarchy or # control variables of *critical* block. Hence,

$$p_{total} = 1-(1-p')^N \tag{11}$$

If the circuit is designed to operate at the two lower voltages (Fig. 2(c)), then the probability of *clock stretching* becomes

$$p_{total} = 1-[(1-p')(1-p'')]^N \tag{12}$$

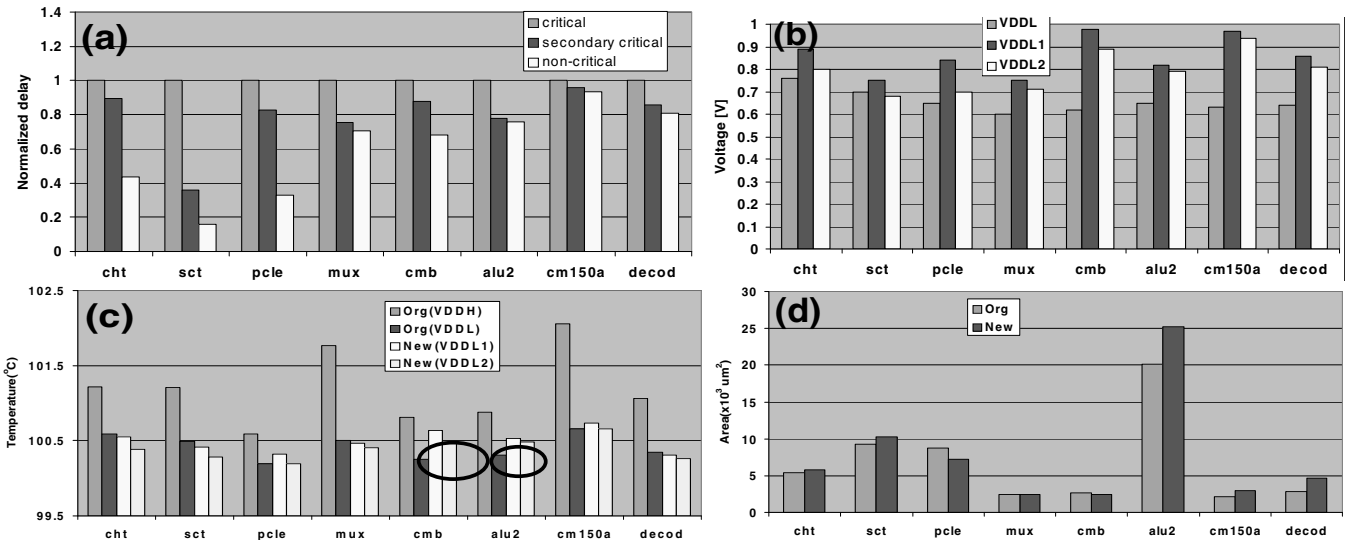

Fig. 6: (a) Normalized delays of *critical*, *secondary critical* and *non-critical* cofactors; (b) temperature of conventional and proposed design at nominal and reduced voltages; and, (c) area of conventional and proposed design

TABLE-2

Procedure pipelineDesign()	
Input	: yield (Y), list of circuits(dList), V_{DDL1}, V_{DDL2};
Output	: list of re-designed circuits (dList);
1	target delay (T_c) = *max*(stage delays);
2	**for each** design $D_i \in dList$
3	$gList = performPartitioning(D_i, V_{DDH})$;
4	S_1 = Slack of *critCF* in *gList* at V_{DDH} to meet yield Y;
5	S_2 = Slack of *scritCF* in *gList* at V_{DDL1} to meet yield Y;
6	S_3 = Slack of other cofactors in *gList* at V_{DDL2} to meet Y;
7	$D_i = performSizing(gList, T_c, Y, S_1, S_2, S_3)$;
8	**end for**
9	**return** *dList*;

where p'' is the activation probability of *secondary critical* block of every stage.

If the ideal clock cycle-per instruction (CPI) of the pipeline is given by CPI_{ideal}, then new CPI is $CPI_{new} = CPI_{ideal} + p_{total}.(\text{stall penalty})$ and the performance penalty due to *clock cycle stretching* is given by

$$\text{Perf. penalty} = \frac{CPI_{new} - CPI_{ideal}}{CPI_{new}} = \frac{p_{total}.(\text{stall penalty})}{CPI_{ideal} + p_{total}.(\text{stall penalty})} = \frac{p_{total}}{1 + p_{total}} \quad (13)$$

The performance penalty for different N and k (for input switching activity of 20%) is shown in Fig. 7(a). The solid (dashed) lines denote penalty at V_{DDL1} (V_{DDL2}). In these plots, we assume that the *secondary critical* cofactor of each stage has one less control variable than *critical* cofactor. The penalties for switching activity of 50% are shown in Fig 7(b). It can be observed from this plot that if the control variables have low switching activity (~20%), then the penalty can be restricted within 10% (25%) for $k=4$ at V_{DDL1} (V_{DDL2}). The penalty can be large for deep pipeline designs (i.e. large N). We suggest the following techniques for reducing the performance penalty, (a) tune the control variable selection metric to pick low switching inputs as control variables; and, (b) reduce the activation probability of *critical* blocks further (i.e., by increasing k).

B. Design methodology

Our pipeline design methodology is based on a *given* set of reduced supply voltage constraints (i.e., V_{DDL1} and V_{DDL2}). The procedure is shown in Table 2. It takes a target yield, the list of pipeline stages and the target voltages as inputs and produce re-designed, synthesized circuits as outputs. The stage delays are

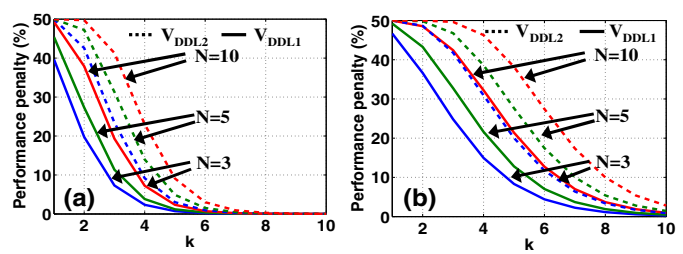

Fig. 7: Performance penalty for different pipeline depths at uniform input switching activity of (a) 20%; (b) 50%

computed by sizing the design as explained in Section IIIB. The maximum stage delay is chosen as the 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 slacks S_1, S_2 and S_3 are computed in Steps 4-6 by using SSTA at specified supply voltages (V_{DDH}, V_{DDL1} and V_{DDL2}). The output of step-3 is a list of cofactors which is sized to meet the required slacks (Step-7). Steps 2-5 are performed on each of the pipeline stages and the list of pipeline stages is returned as output.

The architecture of a temperature-adaptive pipeline (with two lower voltages) is shown in Fig. 8. Decoders D_{11}, D_{21} and D_{31} determine *critical* cofactor activation of one of the stages. Similarly, decoders D_{12}, D_{22} and D_{32} determine the activation of *secondary critical* cofactor of the stages. These two set of decoders are OR-ed to predict the activation of *critical* and *secondary critical* cofactor of pipeline stages. The predictions are gated with thermal sensor outputs. A *freeze* signal is generated only when the temperature of the chip crosses the threshold value and, a *clock stretching* is predicted by the decoders. When the temperature goes above T_{REF1}, the supply voltage is automatically reduced by the multiplexer to V_{DDL1}. However, if T_{REF2} is violated, then supply voltage is reduced to V_{DDL2} and both *critical* and *secondary critical* cofactors are operated in *stretched clock*. Since the performance of pipeline at V_{DDL2} is lower than that of V_{DDL1}, a trade-off can be made between temperature and performance.

Note that traditional supply reduction techniques may require pipeline stalling for few cycles to tune the clock frequency. However, the proposed pipeline need not be stalled during DVS process. This is true because activation of critical paths is predicted ahead in time and corrective action is taken. To avoid delay failures when the supply is brought back from low to nominal voltage, we place a small counter to

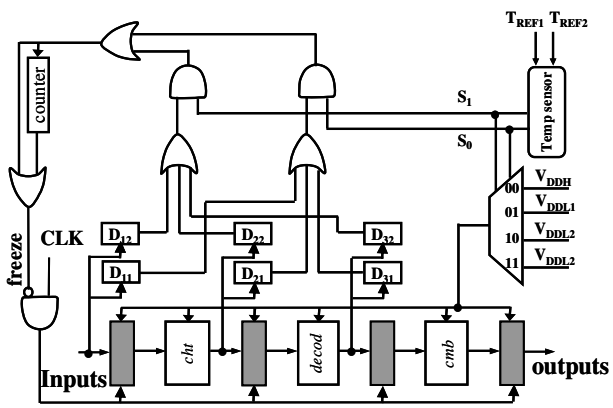

Fig. 8: Temperature-adaptive pipeline design

extend the *freeze* signal (Fig. 8). Although the proposed adaptive DVS technique has been demonstrated for 3-stage pipeline, it can be extended for *N-stages*.

C. Thermal model for electro-thermal simulation

To demonstrate the impact of DVS on the transient behavior of temperature, we follow lumped RC thermal model [12]. The thermal resistance (R_{chip}) and capacitance (C_{chip}) of the circuit is given by

$$R_{chip} = \frac{\rho \,.\, l}{A}; \;\; C_{chip} = c \,.\, l \,.\, A \tag{14}$$

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

$$\begin{aligned} T_i &= T_{i-1} + [T_{MAX} - T_{i-1}][1 - \exp(-t/\tau_{chip}); \;\; \text{if } (T_{ambient} + R_{chip}P_i) > T_{i-1} \\ &= T_{i-1}\exp(-t/\tau_{chip}); \;\; \text{otherwise} \end{aligned} \tag{15}$$

where t is the time step. $\tau_{chip}=R_{chip}C_{chip}$ and T_{MAX} = maximum allowable temperature. Equation (15) is based on simple RC circuit charge/discharge expression.

D. Experimental results and discussion

The timing 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} are determined as explained in Section IIID. Once we get the new pipeline design with nominal and reduced supply voltages, we perform electro-thermal simulation to solve the self-consistent loop of power and temperature at each time step. R_{chip} and C_{chip} of the circuit are computed by using equation (14). For power estimation, the circuit is simulated in Hspice at ambient temperature ($100^{\circ}C$). After determination of power, the transient temperature is computed using equation (15). The computed temperature is used for determination of power in the next time step. The T_{REF1} and T_{REF2} are kept at $104^{\circ}C$ and $110^{\circ}C$ respectively.

The path delay distribution of two of the pipeline stages (i.e. *cht* and *cmb* of Fig. 8) is shown in Fig. 5. The target delay was 100ps and (V_{DDL1}, V_{DDL2}) were found to be (0.8V, 0.75V) with V_{DDH}=1V. The thermal simulation result for our example pipeline (Fig. 8) is shown in Fig. 9. It can be observed that the temperature rises from the ambient value and saturates after some time. At t=4x10^{5}ns, the temperature of the circuit goes above T_{REF1} and the supply voltage is reduced to V_{DDL1} (i.e. 0.8V) based on sensor's output. As expected, the temperature reduces to ~$100.7^{\circ}C$. At this lower supply, only the *critical* cofactors of the pipeline stages operate in *stretched clock*. This simulation demonstrates the self adaptive nature of the system with respect to temperature. For the sake of comparison, we also plotted the

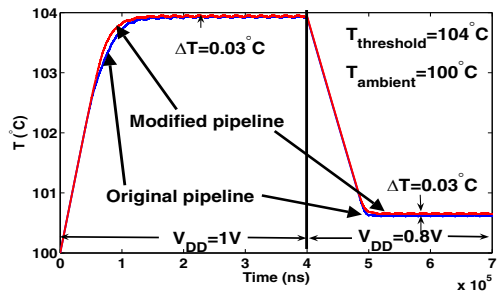

Fig. 9: Transient temperature profile for the 3-stage pipeline

temperature profile of the pipeline with conventionally synthesized circuits. It can be noted from the figure that final temperature of original pipeline is lower than re-designed pipeline by approximately $0.03^{\circ}C$. This is true because area of the re-designed circuit is larger than the original circuit (by approximately 22%). This raises R_{chip} and thereby the steady-state temperature by a small fraction (as $T_{final}=T_{ambient}+PR_{chip}$). However, the difference lies in performance. In the event of high temperature, the supply of original pipeline is also reduced and operated at *half frequency* leading to 50% (as p_{total} = 1 in equation (13)) performance penalty. On the other hand, the proposed design needs only occasional *clock stretching* at V_{DDL1}. The performance penalty at V_{DDL1} was found to be 11% (for input signal probabilities of 50%).

V. CONCLUSION

We proposed a novel temperature-adaptive circuit design technique for dynamic temperature control based on critical path isolation. The proposed technique makes delay errors due to supply voltage scaling predictable and rare such that they are avoided by adaptively stretching the clock period. It helps maintain reasonably high performance even under supply voltage scaling at higher chip temperature. The synthesis technique isolates critical paths to few logic blocks by Shannon decomposition and gate sizing. Simulation results on a pipeline-based design show that the proposed design can maintain a target temperature with small area and performance penalty.

Acknowledgement: This research was partly supported by Gigascale System Research Center and Semiconductor Research Corp (#1078.002). We would like to thank Dr. Saibal Mukhopadhyay, Purdue University for valuable discussions.

REFERENCES

[1] S. Gunther et al., Managing the impact of increasing microprocessor power consumption, *Intel Tech. J.*, 2001.
[2] K. Skadron et al., Temperature aware micro-architecture, ISCA, 2003.
[3] R. McGowen et al., Power and Temperature control on a 90nm Itanium-Family Processor, JSSC, 2006.
[4] S. Ghosh et al., A new paradigm for low-power, variation-tolerant circuit synthesis using critical path isolation, to appear, ICCAD, 2006.
[5] BPTM 70nm: Berkeley predictive technology model.
[6] M. Meterellioz et al., A Leakage Control System for Thermal Stability during Burn-In Test, ITC, 2005.
[7] L. Lavagno et al. Timed Shannon Circuits: A Power-Efficient Design Style and Synthesis Tool, DAC, 1995.
[8] S. Bhunia et al., A novel synthesis approach for active leakage power reduction using dynamic supply gating, DAC, 2005.
[9] B. C. Paul et al., Novel sizing algorithm for yield improvement under process variation in nanometer, DAC, 2004.
[10] K. Kang et al., Statistical timing analysis using levelized covariance propagation, DATE, 2005.
[11] Synopsys Design Compiler, *www.synopsys.com*.
[12] K. Skadron et al., Control-theoretic techniques and thermal-RC modeling for accurate and localized dynamic thermal management, HPCA, 2002.