Inversed Temperature Dependence Aware Clock Skew Scheduling for Sequential Circuits

Jieyi Long and Seda Ogrenci Memik

Dept. of EECS, Northwestern Univ., Evaston, IL 60208
{jlo198, seda}@eecs.northwestern.edu

***Abstract* — We present an Inversed Temperature Dependence (ITD) aware clock skew scheduling framework. Specifically, we demonstrate how our framework can assist dual-V_{th} assignment in preventing timing violations arising due to ITD effect. We formulate the ITD aware synthesis problem and prove that it is NP-Hard. Then, we propose an algorithm for synergistic temperature aware clock skew scheduling and dual-V_{th} assignment. Experiments on ISCAS89 benchmarks reveal that several circuits synthesized by the traditional high-temperature corner based flow with a commercial tool exhibit timing violations in the low temperature range while all circuits generated using our methodology for the same timing constraints have guaranteed timing.**

I. Introduction

With the shrinking of feature sizes and increasing integration density, leakage power consumption has become a major concern. Dual-V_{th} synthesis is one of the approaches for reducing leakage power. In a typical dual-V_{th} design flow, circuits are synthesized at the high-temperature corner, as it is generally assumed that cell delay depends positively on temperature. At newer technologies this assumption becomes invalid. Under nominal supply voltages, delay of high-V_{th} cells might decrease with rising temperature [1-4]. As a result, at low temperature levels, paths containing high-V_{th} cells might exhibit longer delay and violate the timing constraint. This phenomenon, known as the Inversed Temperature Dependence (ITD), poses threats for the corner based dual-V_{th} synthesis flow.

A plausible way to meet timing constraints in the presence of ITD is to add timing margins. V_{th} assignment can be done at high temperature under an overly constrained clock period. However, determining a robust yet not too pessimistic margin is challenging [3]. In this work, we instead view temperature as a dimension of timing optimization and temperature dependent timing slacks as manageable resources. We propose to perform ITD aware clock skew scheduling during dual-V_{th} assignment to ensure timing correctness. Clock skew scheduling is a well known technique. However, traditional approaches define a static distribution of slacks among combinational logic stages. After dual-V_{th} synthesis, different combinational logic stages will be composed of different types of cells. Thus, they can exhibit different delay-temperature dependencies due to ITD. Hence, the temperature dependent timing slacks can be utilized more effectively if they can flow across the registers according to the actual need of different combinational logic stages as temperature fluctuates.

Circuit designs that adapt their slack dynamically has been studied [5, 6]. In a particular implementation [5] adaptive skew buffers were used, which adjust their delays proportional to temperature and redistribute slacks within sequential circuits. However, it was assumed that the delays of the combinational logic blocks always increase with rising temperature and hence, the ITD effect is not addressed. Despite this limitation, the basic concept of adaptive skew buffers provides a useful starting point towards coupling temperature with clock skew generation in presence of ITD.

Motivated by this observation, we propose to perform synergistic ITD aware clock skew scheduling and dual-V_{th} assignment to dynamically redistribute timing slacks globally within a sequential circuit to guarantee timing correctness. Experimental results on various benchmarks confirm that our technique can ensure timing closure. While the corner based flow using a commercial dual-V_{th} design tool can generate circuits exhibiting timing violations in low temperature range, our flow produces circuits with guaranteed timing. In terms of leakage power, our flow is at least as good as the corner based flow in most of the cases. Moreover, compared to a margin based synthesis flow which can eliminate the timing violations in most cases, our flow is superior in terms of leakage power reduction. The leakage power saving can be as large as 44.28% and 14.60% on average in comparison to the margin based flow.

The remainder of the paper is organized as follows. Section II provides an overview of related work. In Section III, we elaborate on the implications of the ITD on synthesis. Section IV formulates the ITD aware synthesis problem and proves the problem is NP-Hard. Section V describes our proposed synthesis framework. Experimental evaluation of our proposed methodology is presented in Section VI.

II. Related Work

Park et al. conducted one of the earliest studies on the ITD effect in low voltage devices [2]. Dasdan et al. considered ITD during static timing analysis and pointed out that the delay of a path can be a non-monotonic function of temperature [7]. Wu et al. investigated the impact of ITD on timing sign-off [3].

Calimera et al. proposed the first treatment for the problem of combinational logic synthesis considering the ITD effect [1, 4]. A path based V_{th} assignment is proposed to make the design insensitive to temperature fluctuations. Our methodology is different from their approach in many aspects. Firstly, while the timing correctness of the circuits generated by their approach relies largely on an empirically parameter α (the target clock period is scaled by α) [4], our methodology provides timing guarantees for the synthesized designs across the target temperature range. Secondly, besides V_{th} assignment, our framework leverages temperature aware clock skews to further reduce leakage power. Finally, while their technique is limited to combinational circuits, our approach handles sequential logic as well.

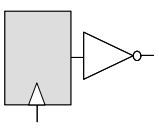
Temperature can be viewed as a generic source of timing variability. In principle, existing timing variation aware V_{th} assignment techniques could be applied to handle ITD [8, 9]. However, most of these works assume a simple analytical delay-V_{th} model. Several important factors, such as the dependence of delay on input signal slope and the relation between gate input capacitance and V_{th} are not captured. Unlike these approaches, we have used an industrial cell delay model that accounts for these factors.

Recently, Ni et al. showed that clock skew can be exploited for leakage power reduction [10]. However, cell delay variation induced by temperature fluctuation is not considered in their approach. Without considering the ITD effect, clock skew scheduling might lead to timing violations at the lower temperature range.

III. Implications of ITD on Dual-V_{TH} Assignment

Let us first illustrate how corner based V_{th} assignment can result in an incorrect design due to ITD. Figure 1 shows a simple circuit and the delay specifications of high-low-V_{th} alternatives for the circuit elements. The target period is 0.5 ns. Note that high-V_{th} cells exhibit an inverse pattern, i.e., their delays decrease as temperature increases. If we synthesize this circuit at 125°C, we might assign low-V_{th} to the register and high-V_{th} to the inverter. While this satisfies the clock period constraint at 125°C, it fails at 25°C. Re-synthesizing the circuit using parameters at 25°C might result in assigning the register

This work was supported by the National Science Foundation under Grant No. CNS-0546305



	Delay (ns)	25°C	125°C
INV	High-V_{th}	0.300	0.280
	Low-V_{th}	0.205	0.215
DFF	High-V_{th}	0.295	0.290
	Low-V_{th}	0.210	0.220

Figure 1. A simple circuit and standard cell delay specifications.

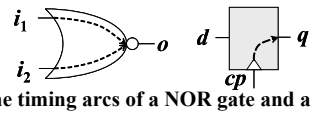


Figure 2. The timing arcs of a NOR gate and a register.

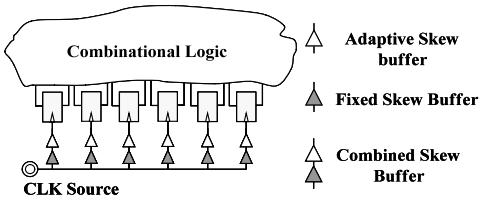


Figure 3. Temperature-dependent clock skew generation.

a high-V_{th} cell and the inverter a low-V_{th} cell. Now, the circuit meets timing at 25°C, but fails at 125°C. This poses a *dilemma for the corner based synthesis methodology*. Designs meeting their timing constraints at the temperature corner for which they were synthesized for may fail at other temperatures. To resolve this dilemma, we need to establish the necessary and sufficient condition, which guarantees timing correctness in the presence of ITD. We will discuss this in Section III.C. Before that, we first present the models of the standard cell delay and temperature dependent clock skew.

A. Modeling Standard Cell Delay

In a typical industrial standard cell library, each input-output pin pair of a cell is associated with a *timing arc* (see Figure 2). The delay and output slew of a timing arc are non-linear functions of multiple parameters. Lookup tables are used to specify delay and output slew of the timing arcs as functions of input slews and load capacitances [11]. Essentially, the lookup table specifies the delay and output slew as *multivariate piecewise linear functions*. We extend this model to include temperature. Delay and output slew of each timing arc is sampled at grid points in the slew-capacitance-temperature space and can be calculated by linear interpolation for other points in the space.

B. Modeling Temperature Dependent Clock Skew

Special adaptive skew buffers can be designed to provide delays linearly dependent on temperature across a wide range (from 25°C to 125°C) [5]. There are also techniques for designing a fixed skew buffer whose delay is independent of temperature. We start with these two concepts and build our temperature dependent skew generation system (shown in Figure 3). Before each register, we place a pair of serially connected fixed buffer and adaptive buffer. The inputs of the fixed buffers are connected to the global zero-skew clock tree. We will refer to the serial connection of these two buffers as a *combined skew buffer*. Its delay is related to temperature by:

$$X(r,\theta):\quad x_r(\theta) = x_r^0 + k_r \cdot (\theta - \theta_{min}) \quad \forall r \in R \qquad (1)$$

where x^0_r is the delay of the combined skew buffer at the minimum target temperature, and k_r is the slope of the linear dependence between delay and temperature. *Note that k_r can be positive or negative.* The adaptive skew buffers are simple structures comprised of chains of repeaters. The x^0_r and k_r parameters can be set to desired values by performing a specific sizing of the repeaters. Note that the temperature dependent skew generation system shown in Figure 3 is illustrative. In fact, the global clock tree already contains repeaters that can be sized and biased to provide functionality equivalent to the adaptive and fixed skew buffers. This allows generating temperature dependent skew with negligible area and power overhead.

C. Timing Correctness in the Presence of ITD

To guarantee the timing correctness of a sequential circuit, the setup/hold constraints must be satisfied. Denoting the set of registers by R, the setup/hold constraints can be written as:

$$S(r,\theta):\quad AT_d(\theta) \le x_r(\theta) + T_{cp} - s_r(\theta) \qquad \forall r \in R \qquad (2)$$

$$H(r,\theta):\quad at_d(\theta) \ge x_r(\theta) + h_r(\theta) \qquad \forall r \in R \qquad (3)$$

where $AT_d(\theta)$ and $at_d(\theta)$ represent the latest and earliest arrival time at the data input node d of register r at temperature θ, respectively. $x_r(\theta)$ is the clock signal arrival time expressed as a linear function of temperature (as shown in Equation (1)). $s_r(\theta)$ and $h_r(\theta)$ denote the setup and hold time. For each node v in the circuit, we have the following constraints on the latest/earliest arrival times:

$$A(u,v,\theta):\quad AT_v(\theta) \ge AT_u(\theta) + \delta_{uv}(\theta) \quad \forall (u,v) \in TArc \qquad (4)$$

$$a(u,v,\theta):\quad at_v(\theta) \le at_u(\theta) + \delta_{uv}(\theta) \quad \forall (u,v) \in TArc \qquad (5)$$

where set $TArc$ is the set of timing arcs in the circuit and $\delta_{uv}(\theta)$ is the delay of the timing arc (u, v). Constraint sets (2-5) in fact contain infinitely many constraints, as the temperature can vary continuously. However, within a sufficiently small temperature range $[\theta_i, \theta_{i+1}]$, the delay of a timing arc can be assumed to be linearly depending on temperature. We thus have the following lemma which reduces the constraints to a finite number (the proof is omitted due to page limit).

Lemma 1. Given a sufficiently small temperature range $[\theta_i, \theta_{i+1}]$, the timing constraints of a given circuit are satisfied across $[\theta_i, \theta_{i+1}]$, if and only if they are satisfied at θ_i and θ_{i+1}.

IV. Formulation and Hardness of the Problem

In the following, "temperature aware clock skew schedule" refers to the configurations (i.e., the x^0_r and k_r values in Equation (1)) of the combined skew buffers in the design.

Problem 1 (ITD Aware Synergistic Clock Skew Scheduling and Dual-V_{th} Assignment). Given the gate-level netlist, target clock period T_{cp}, and a dual-V_{th} standard cell library specifying the delays and output slews as functions of input slew, load capacitance, and temperature, determine whether there exists a temperature aware clock skew schedule and a mapping of gates to the cells, such that the timing is met across the target temperature range $[\theta_{min}, \theta_{max}]$.

Problem 2 (ITD Aware Leakage Optimization). Given the same conditions as in Problem 1, determine the temperature aware clock skew schedule and cell mapping achieving the minimal amount of leakage power consumption.

Theorem 1. Problem 1 and Problem 2 are both NP-Hard for both combinational and sequential logic.

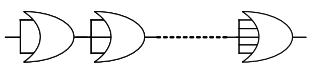
Proof: Problem 1 is a special case of Problem 2. Moreover combinational logic is a special case of sequential logic. Hence, we only need to prove the NP-Hardness of Problem 1 for combinational logic. We will prove the hardness by transforming NP-Complete Problem PARTITION [12] to Problem 1. Given a finite set $A = \{\alpha_i \mid \alpha_i \in \mathbf{Z}^+\}$, Problem PARTITION asks for a subset $A' \subseteq A$, such that

$$\sum_{\alpha_i \in A'} \alpha_i = \sum_{\alpha_j \in A \setminus A'} \alpha_j \qquad (6)$$

Given an instance of PARTITION, we construct a simple circuit depicted in Figure 4 with target clock period T_{cp}

$$T_{cp} = \tfrac{1}{2} \sum_{\alpha_i \in A} \alpha_i \qquad (7)$$

We also construct a dual-V_{th} cell library where the available alternatives for the i^{th} $(i + 1)$-input OR gate in the chain are limited to one high-V_{th} cell and one low-V_{th} cell. The delays of different OR cells at θ_{min} and θ_{max} are given in Figure 4. Within $[\theta_{min}, \theta_{max}]$, the delay of any cell depends linearly on temperature. We need to show that there is a partition satisfying Equation (6) if and only if there is a cell mapping achieving T_{cp}. If a legal mapping exists, in polynomial



	Delay	θ_{min}	θ_{max}
$(i+1)$-input OR Cell	High-V_{th}	α_i	0
	Low-V_{th}	0	α_i

Figure 4. Simple circuit assisting the hardness proof.

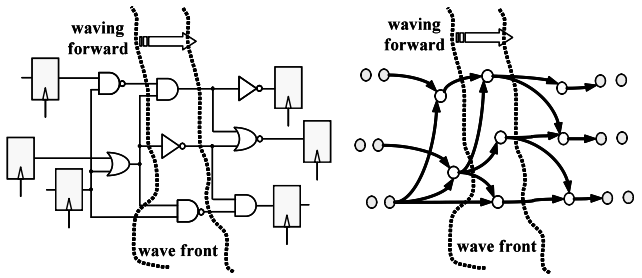


Figure 5. (a) An example sequential circuit (b) The corresponding DAG representation and the wave front.

```
ALGORITHM ITDAwareLeakOpt(NL)
INPUT: NL // netlist of the given sequential circuit
OUTPUT: {μ_v}, {x_v(θ)} // V_th-Assignment and clock skew scheduling
1  BEGIN
2     Construct DAG representing the circuit;
3     Assign low-V_th to each gate; // initial solution
4     REPEAT
5        WF = {vertices with zero in-degree};
6        FOR EACH gate v∈WF BEGIN
7           Calculate the effective delay increments
                of v if it were assigned high-V_th;
8        END
9        Solve relaxed-MILP-LeakOpt;
10       Round fractional μ_v to zero, recalculate {x_v(θ)};
11       Remove WF from DAG;
12    UNTIL DAG = ∅;
13    RETURN {μ_v}, {x_v(θ)};
14 END
```

Figure 6. Pseudo code for ITD aware leakage power optimization.

time we can construct a set $A' = \{\alpha_i \mid$ the i^{th} gate is mapped to a high-V_{th} cell$\}$. Let us denote the set of high-V_{th} and low-V_{th} gates by G_h and G_l, the delay of gate g at temperature θ by $\delta_g(\theta)$. Since the timing constraints must be satisfied at θ_{min} and θ_{max}, we have

$$\text{At } \theta_{min}: \sum_{g\in G_h} \delta_g(\theta_{min}) = \sum_{\alpha_i\in A'} \alpha_i \le T_{cp} = \frac{1}{2}\sum_{\alpha_i\in A} \alpha_i \quad (8)$$

$$\text{At } \theta_{max}: \sum_{g\in G_l} \delta_g(\theta_{max}) = \sum_{\alpha_i\in A\setminus A'} \alpha_i \le T_{cp} = \frac{1}{2}\sum_{\alpha_i\in A} \alpha_i \quad (9)$$

which means that the sum of the integers in set A' must be equal to the sum of those in $A\backslash A'$. On the other hand, if there exists a subset A' satisfying Equation (6), we will map the i^{th} OR gate in the chain to a high-V_{th} cell if and only if $a_i \in A'$. Since Equation (6) is satisfied, according to Inequalities (8) and (9), the setup constraint is met at both θ_{min} and θ_{max}. In addition, as the delay of each cell depends linearly on temperature, according to Lemma 1, the timing constraints are satisfied across $[\theta_{min}, \theta_{max}]$. ■

Note that without the ITD effect, for combinational circuits, determining whether there is a cell mapping achieving a given T_{cp} is polynomial time solvable since we only need to map each gate to the fastest available cell at θ_{max} and examine whether the critical path delay is more than T_{cp}. This observation indicates that the ITD effect increases the complexity of the synthesis problem.

V. ITD AWARE FRAMWORK

As Problem 2 contains Problem 1 as a special case, we will focus on the ITD aware leakage optimization problem. To solve it, we need to determine the clock skew schedule and cell mapping. Cell mapping involves gate sizing and V_{th} assignment. As sizing of the gates is not directly related to the ITD effect, we decide to focus on V_{th} assignment only. In the following, we first present an iterative algorithm for synergistic clock skew scheduling and V_{th} assignment.

Our algorithm processes the given sequential circuit in a "waving forward" manner (see Figure 5). In each iteration, we determine the clock skew schedule and V_{th} assignment of the gates in the current "wave front" by solving a relaxed Mixed-Integer Linear Programming (MILP) formulation. Figure 6 provides the pseudo code of the algorithm. In the beginning, it constructs a graph representation of the circuit (Line 2). Gates/registers are mapped to vertices and interconnects are mapped to directed edges. Sequential circuits contain loops. We break the loops by splitting each vertex representing a register into two (as shown in Figure 5b), thus obtaining a Directed Acyclic Graph (DAG) representation of the circuit. Initially, we assign each gate to low-V_{th} (Line 3). Then, the algorithm enters the main loop which processes the DAG iteratively. In each iteration of the main loop, the algorithm determines the skews and assigns threshold voltages to the gates on the "wave front". The "wave front", denoted as *WF* in the pseudo code, consists of the vertices whose in-degree in the remaining *DAG* is zero (Line 5). Figure 5 depicts an example of a wave front.

In order to determine whether a gate on the wave front can be assigned to high-V_{th}, we need to evaluate the increase in arrival time at other nodes in the circuit if this gate were assigned to high-V_{th}. Raising V_{th} increases the delay of the timing arcs of that gate. Furthermore, the output slew of the gate increases, which impacts its fan-out. To estimate the increase in arrival time at the other nodes, we propagate the output slew in the *downstream tree* rooted at this gate. The maximal increase in arrival time at the nodes is used as the upper bound. On the other hand, the increase in the intrinsic delay of the gate can be used as a lower bound.

The above discussion reveals the rationale behind the idea of "waving forward". Changing the threshold voltage of a gate impacts the slew of its output signal. Hence, if we consider simultaneous V_{th} assignment of a gate and its downstream gates, we have to account for the changes in the slew of the signals and the threshold voltages in order to determine the propagation delay of each gate. This would result in a highly complex set of relationships. A simplified yet effective solution is to consider simultaneous V_{th} assignment for the set of gates lying on the same wave front only. Gates along the same wave front are guaranteed not to affect each other's slew, hence, decisions regarding them can be made simultaneously.

For each gate on the wave front, we introduce a binary decision variable μ_v to represent the V_{th} assignment. μ_v is set to 1 if the gate driving node v is assigned to high-V_{th}, 0 otherwise. The upper and lower bound of the increase (as a result of this assignment) in arrival times at node v are denoted by $\Delta_v^{sup}(\theta)$ and $\Delta_v^{inf}(\theta)$, respectively. The arrival time constraints for timing arcs belonging to the gates in the current wave front then become

$$A'(u,v,\theta): AT_v(\theta) \ge AT_u(\theta) + \delta_{uv}(\theta) + \mu_v \cdot \Delta_v^{sup}(\theta), \forall (u,v) \in WF \quad (10)$$

$$a'(u,v,\theta): at_v(\theta) \le at_u(\theta) + \delta_{uv}(\theta) + \mu_v \cdot \Delta_v^{inf}(\theta), \ \forall (u,v) \in WF \quad (11)$$

The V_{th} assignment of the gates in the wave front and the clock skew schedule, which yields the minimal amount of leakage power, can then be found by solving the following MILP (Line 9) (note that for sake of brevity in the presentation we consolidated the satisfaction of several constraints into a single logical AND " $\wedge$ " expression. This is meant only for ease of presentation.)

$$\min \sum \left(\mu_v \cdot P_{leak,v}^{high}(\theta_{max}) + (1-\mu_v)\cdot P_{leak,v}^{low}(\theta_{max})\right) \quad (12)$$

$$s.t. \quad A'(u,v,\theta) \wedge a'(u,v,\theta) \wedge \left(\mu_v \in \{0,1\}\right), \forall (u,v) \in WF \quad (13)$$

$$A(u,v,\theta) \wedge a(u,v,\theta), \forall (u,v) \notin WF \quad (14)$$

$$S(r,\theta) \wedge H(r,\theta) \wedge X(r,\theta), \forall r \in R \quad (15)$$

$$x_r^0 \in \mathbf{R}^+, k_r \in \mathbf{R}, \forall r \in R \quad (16)$$

$$\theta = \theta_1, \theta_2, ..., \theta_n \quad (17)$$

The cost function of the MILP is the leakage power of the gates in the current wave front at the maximum temperature. The variables of the MILP include the 0-1 decision variables $\{\mu_v\}$ and the real-value configuration parameters of the combined skew buffers ($\{x^0_r\}$ and $\{k_r\}$). The MILP incorporates constraints bounding the increase in

TABLE I. Experimental results of the three different flows.

Bench	*Target* T_{cp} (ns)	Traditional Flow (TF)				Margin Based Flow (MF)				Our Flow				
		Pass/ Fail	T^*_{cp}(ns) 25°C	T^*_{cp}(ns) 125°C	P_{leak}(nW) 125°C	*Pass/ Fail*	T^*_{cp}(ns) 25°C	T^*_{cp}(ns) 125°C	P_{leak}(nW) 125°C	T^*_{cp}(ns) 25°C	T^*_{cp}(ns) 125°C	P_{leak}(nW) 125°C	*Impr. vs.MF*	*Runtime* (s)
s298	**0.4300**	Fail	0.4327	0.4288	1820.5		0.4116	0.4022	1820.8	0.4278	0.4276	1589.7	12.69%	1.83
s349	**0.4200**	Fail	0.4300	0.4179	2767.7		0.3966	0.3928	3230.7	0.4169	0.4167	2634.2	18.46%	3.08
s444	**0.4150**	Fail	0.4190	0.4129	2829.2		0.4000	0.3881	3204.0	0.4119	0.4118	2574.9	19.64%	2.42
s526	**0.4250**		0.4223	0.4229	2771.6		0.4137	0.3975	3180.4	0.4223	0.4222	2689.7	15.43%	5.26
s838	**1.1000**	Fail	1.1292	1.0945	3851.2		1.0571	1.0288	3977.6	1.0970	1.0960	3382.8	14.95%	8.83
s953	**0.5300**	Fail	0.5606	0.5274	4328.8		0.5271	0.4957	4615.7	0.5277	0.5275	4025.3	12.79%	6.68
s1238	**1.1200**	Fail	1.1920	1.1144	1851.9	Fail	1.1309	1.0475	2062.3	1.1180	1.1180	1966.1	4.66%	13.75
s1488	**0.9050**	Fail	0.9479	0.9005	1235.3		0.9030	0.8464	1622.0	0.9034	0.9028	903.8	44.28%	18.84
s9234	**1.0750**	Fail	1.1238	1.0696	23250.5		1.0475	1.0055	23566.2	1.0740	1.0730	20871.0	11.44%	61.47
s15850	**1.3900**	Fail	1.4548	1.3831	55718.9		1.3742	1.3001	56451.8	1.3850	1.3850	55232.0	2.16%	269.03
s35932	**0.5350**	Fail	0.5643	0.5323	182169.0		0.5169	0.5004	188016.0	0.5316	0.5311	180220.0	4.15%	240.77
avg													**14.60%**	

arrival times (Inequalities (10 and 11) applied), the constraints (2-5) presented in Section III.C, and the delay-temperature relationship of the combined skew buffers (Equation (1)). Besides, according to Lemma 1, only the temperatures θ_1, θ_2, … θ_n need to be taken into account to guarantee timing across the target temperature range.

In general, MILP formulations are difficult to solve. In order to reduce the runtime, we relax the binary variables $\{\mu_v\}$ to be continuous within range [0, 1] (Line 9). Then, we round all the fractional μ_v values down to zero (Line 10). Note that this might lead to infringement of the hold-time constraint. To avoid timing violation, we assign the V_{th} level to the gates on the current wave front according to the rounded μ_v values and recalculate the clock skew schedule ($\{x^0_r\}$ and [6]) by solving the same relaxed-MILP (Line 10). At the end of each iteration we push the wave front forward by deleting the vertices in set *WF* from the *DAG* (Line 11). The main loop continues until all the gates are processed.

VI. IMPLEMENTATION AND RESULTS

A. Implementation and Experimental Setup

We implemented our proposed technique using C++. Static timing and power analyzers have been integrated into our experimental flow. To solve the relaxed-MILP formulations, we embedded the ILOG CPLEX 10.1 C++ library into our programs. Note that in previous sections we did not differentiate cell rise delay from fall delay for simplicity. However, to be in line with industrial practices, they were treated separately in our implementation. We conducted experiments on sequential circuits from the ISCAS89 benchmark suite.

In our experiment, we first synthesized the benchmarks onto the TSMC 65nm dual-V_{th} standard cell library using Synopsys Design Compiler. We then fed the generated netlists into our framework to perform synergistic clock skew scheduling and V_{th} assignment, as well as timing/power evaluation. 25°C and 125°C were used as the sampling temperatures. We compared our framework against the traditional flow (*TF*), which performs synthesis at high-temperature corner (i.e., the synthesizer takes the delay of each cell at the maximum allowable temperature). We also compared our framework with the margin based flow (*MF*), which also synthesizes the circuits at high temperature corner, but places a 5% margin on the clock period. We carried out experiments for these two flows using Synopsys Design Compiler with the 65nm dual-V_{th} TSMC standard cell library, enabling the leakage power optimization feature.

B. Experimental Results

TABLE I. presents the experimental results. The column "*Target* T_{cp}" gives the target clock period for each benchmark. For each synthesis flow, the table reports the minimal allowable clock periods T^*_{cp} at 25°C and 125°C, and leakage power of the synthesized designs at 125°C. The column *Pass/Fail* indicates whether the synthesized circuit meets the timing constraints across the target temperature range. We also calculated the improvements of leakage power of our flow over the margin based flow, which is listed in column "*Impr. vs. MF*". The last column provides the runtime of our synergistic V_{th} assignment and skew scheduling algorithm.

We first examine the results for the traditional flow. Except for s526, each of them exhibits the inversed delay pattern. For instance, the minimum allowable clock periods of benchmark s953 are 0.5606ns and 0.5274ns at 25°C and 125°C, respectively. Although these designs meet the timing constraints at 125°C, they fail at 25°C.

Compared to the traditional flow, margin based strategy reduces the ITD induced timing error. However, there is still one benchmark (s1238) failing at 25°C. For the rest of the benchmarks, the difference between the target and the minimal allowable clock period varies. This clearly indicates that there is no optimal margin common to all designs. In fact, the proper margin depends on both the structure of the specific design and the properties of the cells in the library.

In contrast, our synthesis framework preserves the timing correctness of each benchmark at both extreme temperatures. Furthermore, the margin based strategy eliminates the temperature induced timing error at the expense of leakage power. This is because overly constraining the clock period forces Design Compiler to place more low-V_{th} cells in the design than necessary. Compared to the margin based flow, our framework can reduce the leakage power by as much as 44.28%, and by 14.60% on average.

It is noticeable that our synthesis framework generates designs with critical path delays that are more balanced at two temperature extremes. Compared to the other two flows, the difference between the minimal allowable clock periods at 25°C and 125°C are much smaller in general. This confirms our claim that our synthesis framework exploits the temperature dependent slacks and turns them into leakage power savings in a more efficient manner.

The runtime of our algorithm is also quite reasonable. The smaller benchmarks can be processed in seconds. For the largest benchmarks (containing approximately 10K gates), the algorithm terminates in about 4 minutes.

REFERENCES

[1] Calimera, A., et al. *Reducing leakage power by accounting for temperature inversion dependence in dual-Vt synthesized circuits*. in ISLPED 2008.

[2] Park, C., et al. *Reversal of Temperature Dependence of Integrated Circuits Operating at Very Low Voltages*. in IEDM 1995.

[3] Wu, S., et al. *How does inversed temperature dependence affect timing sign-off*. in ICICDTT 2008.

[4] Calimera, A., et al. *Temperature-Insensitive Synthesis Using Multi-Vt Libraries*. in GLSVLSI. 2008.

[5] Long, J., et al. *A Self-Adjusting Clock Tree Architecture to Cope with Temperature Variations*. in ICCAD 2007.

[6] Chakraborty, A., et al. *Dynamic Thermal Clock Skew Compensation Using Tunable Delay Buffers*. IEEE TVLSI, 2008. **16**(6): p. 639-649.

[7] Dasdan, A. et al. *Handling Inverted Temperature Dependence in Static Timing Analysis*. ACM TODAES 2006. **11**(2): p. 306-324.

[8] Khandelwal, V. et al. *Simultaneous Vt Selection and Assignment for Leakage Optimization*. IEEE TVLSI, 2005. **13**(6): p. 762-765.

[9] Srivastava, A. et al. *Statistical Optimization of Leakage Power Considering Process Variations using Dual-Vth and Sizing*. in DAC 2004.

[10] Ni, M. et al. *Leakage Power-Aware Clock Skew Scheduling: Converting Stolen Time into Leakage Power Reduction*. in DAC 2008.

[11] *Library Compiler User Guide: Modeling Timing, Signal Integrity, and Power in Technology Libraries*. Synopsys Online Document. **Z-2007.03**.

[12] Garey, M. et al. *Computers and Intractability - A Guide to the Theory of NP-Completeness*. 1979, New York: W. H. Freeman and Company.