Minimizing Leakage Power of Sequential Circuits through Mixed-V_t Flip-Flops and Multi-V_t Combinational Gates

JAEHYUN KIM AND CHUNGKI OH
Samsung Electronics
and
YOUNGSOO SHIN
KAIST

4

The current use of multi-V_t to control leakage power targets combinational gates, even though sequential elements such as flip-flops and latches also contribute appreciable leakage. We can, nevertheless, apply multi-V_t to flip-flops, but few can take advantage of high-V_t, which causes abrupt changes in timing. We combine low- and high-V_t at the transistor level to design mixed-V_t flip-flops with reduced leakage, an unchanged footprint, and a small increase in either setup time or clock-to-Q delay, but not both. An allocation algorithm for two V_ts determines the V_t (mixed, high, or low) of each flip-flop and the V_t of each combinational gate (high or low) in a sequential circuit. Experiments with 65-nm technology show an average leakage saving of 42% compared to conventional multi-V_t approaches; the leakage of flip-flops alone is cut by 78%. This saving is largely unaffected by die-to-die or within-die process variations, which we show through simulations. Standard deviation of leakage caused by process variation is also reduced due to less use of low-V_t devices. We also extend our approach to three V_ts, and obtain a further 14% reduction in leakage.

Categories and Subject Descriptors: B.6.1 [**Logic Design**]: Design Styles—*Sequential circuits*; B.7.1 [**Integrated Circuits**]: Types and Design Styles—*VLSI (very large scale integration)*

General Terms: Algorithms, Design

Additional Key Words and Phrases: Flip-flop, leakage current, low power, mixed-V_t, sequential circuit

ACM Reference Format:

Kim, J., Oh, C., and Shin, Y. 2009. Minimizing leakage power of sequential circuits through mixed-V_t flip-flops and multi-V_t combinational gates. ACM Trans. Des. Autom. Electron. Syst., 15, 1, Article 4 (December 2009), 22 pages.
DOI = 10.1145/1640457.1640461 http://doi.acm.org/10.1145/1640457.1640461

Authors' addresses: J. Kim and C. Oh, Samsung Electronics, Yongin, Gyeonggi-Do 449-711, Korea; Y. Shin, Department of Electrical Engineering, KAIST, Daejeon 305-701, Korea; email: youngsoo@ee.kaist.ac.kr.

DOI 10.1145/1640457.1640461 http://doi.acm.org/10.1145/1640457.1640461

1. INTRODUCTION

Leakage current has been growing continuously and is now comparable to switching power. In recent nanometer CMOS technologies, such as 90 and 65 nm, it is common for leakage current to be responsible for almost half of total power consumption [Friedrich et al. 2007], so it is important to reduce it as far as possible. Leakage current consists of many components [Roy et al. 2003], but subthreshold leakage is the largest proportion in most technologies.

There are many circuit techniques to suppress subthreshold leakage, of which the most popular are power gating [Mutoh et al. 1995; Inukai et al. 2000; Kawaguchi et al. 2000] and reverse body bias [Kuroda et al. 1996; Clark et al. 2004]; but the implementation of techniques involves substantial custom engineering. For example, the changes required for power gating [Kim et al. 2006] include the sizing of a current switch, the design of retention flip-flops, and a custom power network. Moreover, circuit-level power saving requires significant customization for designs based on standard cells, which involves a significant departure from standard design flow [Ohkubo and Usami 2006; Choi and Shin 2007].

In contrast, multi-V_t [Wei et al. 1998], multi-T_{ox} [Sultania et al. 2004], and multi-L_{gate} [Gupta et al. 2006] techniques are transparent to designers, since they can be seamlessly integrated with most design tools. These techniques save leakage in all modes of operation, and not just in standby, although much less leakage is saved than with power gating or reverse body bias. The multi-V_t technique is especially popular for suppressing subthreshold leakage. A multi-V_t circuit utilizes low-V_t gates on timing-critical paths, and gates with high-V_t on paths which are not critical to timing. Many algorithms have been proposed for the deployment of multi-V_t [Wei et al. 1998; Wang and Vrudhula 1998; Sirichotiyakul et al. 1999; Karnik et al. 2002; Ketkar and Sapatnekar 2002; Ho and Hwang 2004; Shah et al. 2005], but they all target the combinational portion of a circuit, even though sequential elements such as flip-flops are responsible for an appreciable proportion of the total leakage. There has also been an effort [Seomun et al. 2007] to combine multi-L_{gate} flip-flops with multi-V_t combinational gates, but the leakage saved is not significant and the scope of the allocation algorithm is limited since the flip-flops and combinational gates must be allocated separately.

We assessed the importance of the leakage from flip-flops by taking several circuits from benchmarks as well as from industrial designs, and simulating them with SPICE in a 65-nm commercial technology model. In Figure 1, the left-hand bars show the numbers of combinational gates and flip-flops, and demonstrate the numerical domination of combinational gates, while the bars on the right show that the flip-flops contribute disproportionately (43% on average) to the total leakage in these circuits.

This suggests that we might advantageously apply high-V_t to some flip-flops (to an extent that does not violate the delay constraint of the circuit), and then apply conventional multi-V_t to the combinational gates. But, if we took this approach, it is likely that the proportion of combinational gates that could

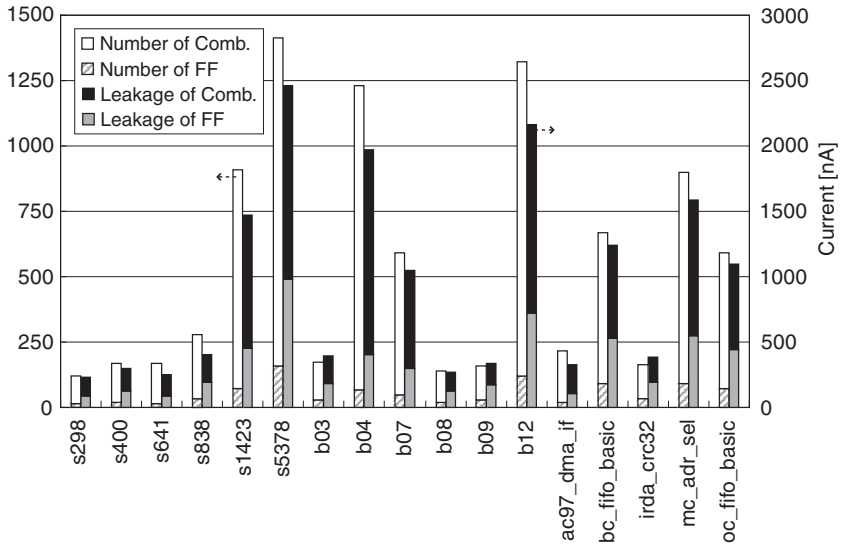

Fig. 1. The numbers of combinational gates and flip-flops (left bars), and distribution of leakage in combinational gates and flip-flops (right bars), for several benchmark circuits.

be assigned to high-V_t would be relatively small, because the original timing slacks, if any, would be absorbed by the increased setup guard time and propagation delay in the flip-flops that are assigned to high-V_t. Moreover, the number of flip-flops that would be able to take advantage of high-V_t might not be very large, since assigning high-V_t to a flip-flop typically affects several timing paths in a circuit.

In this article, we address the question of how to reduce the leakage in the sequential elements of a circuit while continuing to apply multi-V_t to its combinational gates. Our contribution can be summarized as follows.

—We propose two types of mixed-V_t flip-flops, which use both low- and high-V_t, but in different transistors. They are designed such a way that their footprint remains the same as that of a conventional flip-flop, so that they can be used without altering layout. However, either the setup time or the clock-to-Q delay is increased, but not both.

—We propose an allocation algorithm which determines the type (mixed-V_t, low-V_t, or high-V_t) of each flip-flop, and the type of V_t (low or high) for each combinational gate, within a single framework.

—The mixed-V_t flip-flops and the allocation algorithm are also extended to three V_ts (low, regular, and high).

—Extensive experimental results from the application of commercial 65-nm technology to several benchmark circuits as well as to industrial designs are used to assess both the proposed flip-flops and the allocation algorithm. Experiments are also conducted in the presence of die-to-die and within-die process variations.

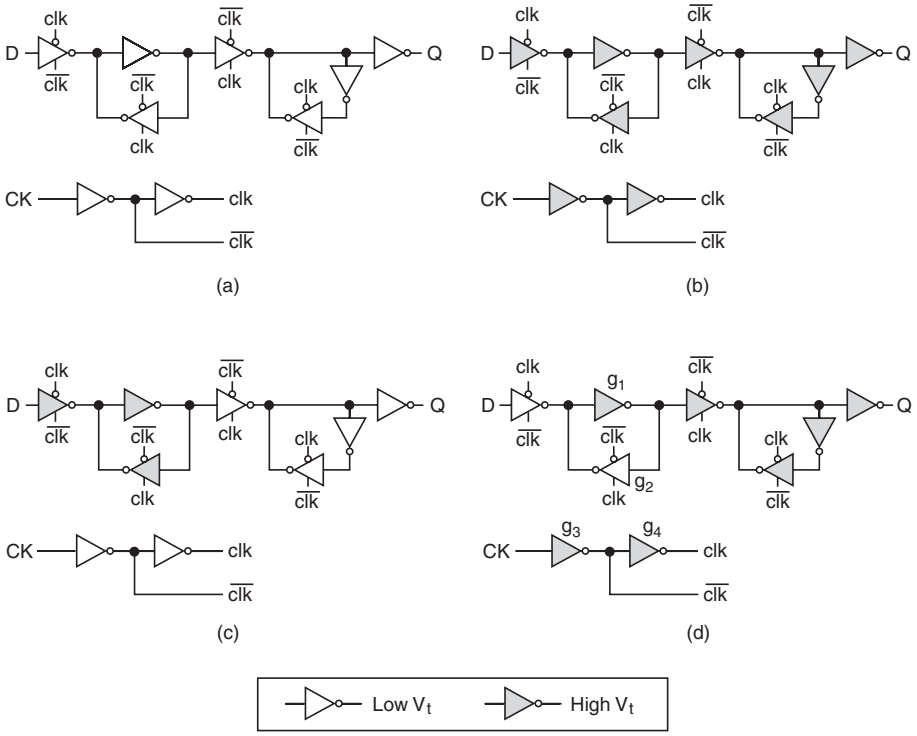

Fig. 2. (a) LVT flip-flop, (b) HVT flip-flop, (c) MVT-I flip-flop, and (d) MVT-II flip-flop.

The remainder of this article is organized as follows: in the next section, we discuss the design of mixed-V_t flip-flops, and their leakage and timing characteristics. An allocation algorithm is presented and experimental results are discussed in Section 3. The mixed-V_t flip-flops and the allocation algorithm are extended to three V_ts in Section 4, and we draw conclusions in Section 5.

2. MIXED-V_t FLIP-FLOPS

Figure 2(a) shows an example of a (positive edge-triggered) D flip-flop, implemented for low-V_t (LVT) as an inverter and a tristate inverter. Figure 2(b) shows the corresponding flip-flop implemented for high-V_t (HVT). The leakage of a 65-nm implementation of these two flip-flops was simulated for each combination of input D and output Q, and the results are shown in Table I. It can be readily seen that the leakage of the HVT flip-flop is considerably lower than that of the LVT version. However, substituting an HVT flip-flop for an LVT flip-flop can disrupt the timing of a circuit. Table II shows that such a change affects all the timing paths which include either the setup time (T_{su}) or the clock-to-Q delay ($T_{c\text{-}q}$) of the original flip-flop. Thus, the substitution of an HVT flip-flop is limited to those timing paths where large slacks remain, even after combinational subcircuits have been optimized by gate sizing, multi-V_t, and so on.

In order to achieve a smoother transition from a low-V_t flip-flop to some less leaky substitute, we designed the two new variant flip-flops shown in

Table I. Leakage Current of Low-, High-, and Mixed-V_t Flip-Flops

FF	Leakage current (pA)				
	DQ = 00	DQ = 01	DQ = 10	DQ = 11	**Average**
LVT	531	579	654	562	**582**
HVT	52	52	61	57	**56**
MVT-I	424	472	527	435	**464**
MVT-II	126	126	85	81	**105**

Table II. Timing Parameters of Low-, High-, and Mixed-V_t Flip-Flips

FF	Delay (ps)			
	Rising T_{su}	Falling T_{su}	Rising $T_{c\text{-}q}$	Falling $T_{c\text{-}q}$
LVT	23.6	16.2	61.5	71.8
HVT	**34.3**	**25.0**	**89.2**	**106.1**
MVT-I	**37.9**	**36.6**	61.7	71.8
MVT-II	22.4	10.7	**89.1**	**106.2**

Figures 2(c) and (d). The thrust of these designs follows from two observations. First, even though a flip-flop is physically a single gate, it has to be viewed as two components in terms of timing: a master stage, which is located in the end of circuit timing paths because of its setup time T_{su}; and a slave stage, which is located in the front of circuit timing paths because of its clock-to-Q delay $T_{c\text{-}q}$. Second, when a mixed-V_t gate (e.g., nMOS in low-V_t and pMOS in high-V_t for an inverter) is required, it usually comes at the cost of increased layout area. Mixed-V_t (MVT) flip-flops, however, can be designed without any increase in area by careful use of high- and low-V_t transistors.

The first variant, shown in Figure 2(c), we call an MVT-I flip-flop, and it only uses high-V_t in the transistors that affect T_{su}. As shown in Table I, its leakage saving is not dramatic. However, Table II shows that $T_{c\text{-}q}$ remains almost unchanged, which indicates that this design can be substituted for flip-flops which have slacks in their D-input but not in their Q-output. The layout of the MVT-I flip-flop is shown in Figure 3(c), and the layouts of low- and high-V_t flip-flops are shown for comparison in Figures 3(a) and (b). The area of the MVT-I is the same as that of the other flip-flops. This is achieved by localizing the two tristate inverters at high-V_t in the master stage to the left-hand side of the layout, while the inverter at high-V_t is located to its right. This still leaves enough space between the two high-V_t layers, which are shown in the figure as thick rectangles.

We call the second variant flip-flop, shown in Figure 2(d), an MVT-II flip-flop. It uses high-V_t in the transistors that affect $T_{c\text{-}q}$, in the inverter g_1 in the master stage, and in the two cascaded inverters g_3 and g_4, which generate clk and $\overline{\text{clk}}$ internally from the clock input (CK). Even though the use of high V_t delays clk and $\overline{\text{clk}}$, both the rising and falling setup times are actually reduced, as shown in Table II. The graph in Figure 4 compares the waveforms of D-input, clk, and Q-output for LVT and MVT-II flip-flops to explain the rising T_{su} and rising $T_{c\text{-}q}$. Note that the timing parameters of a flip-flop are measured with respect to its clock input (CK), which is marked on the x-axis of the graph. The

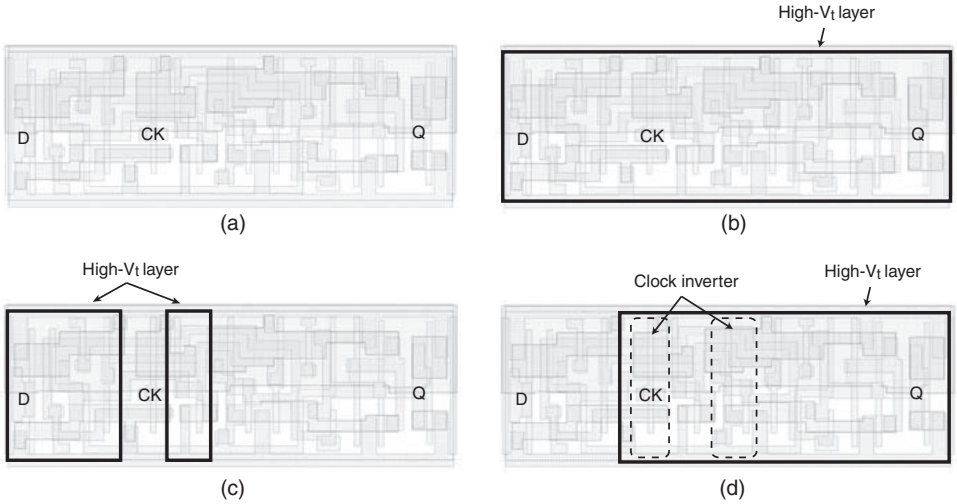

Fig. 3. Layouts of (a) LVT flip-flop, (b) HVT flip-flop, (c) MVT-I flip-flop, and (d) MVT-II flip-flop.

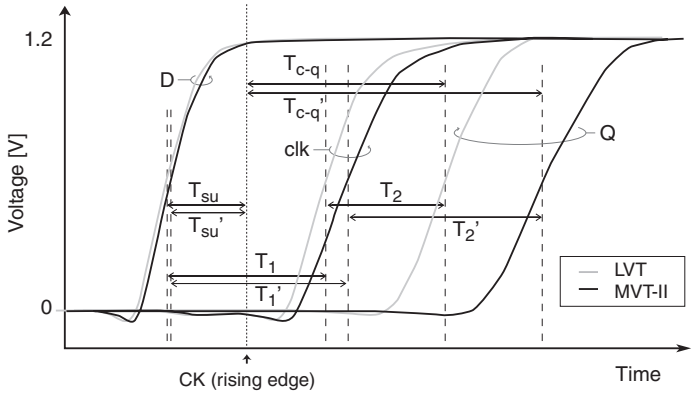

Fig. 4. Comparison of rising T_{su} and rising T_{c-q} waveforms for LVT and MVT-II flip-flops.

D-input can be captured in the master stage only after the late arrival of clk and $\overline{\text{clk}}$, which are internally generated and thus lag behind CK (in Figure 4, T_1 corresponds to the LVT and T_1' to the MVT-II). Since clk arrives later in the MVT-II (T_1') than in the LVT (T_1), the D-input is allowed to arrive later, which reduces T_{su} (marked T_{su}' in Figure 4). This also allows us to use high-V_t for gate g_1, even though it is in the master stage (see Figure 2(d)), further reducing the leakage current. The falling setup time T_{su} is even more reduced (see Table II) because the falling D-input is only captured after $\overline{\text{clk}}$ arrives at gate g_2, and $\overline{\text{clk}}$ arrives earlier than clk. The overall result is a significant reduction in the leakage of an MVT-II flip-flop, which is 82% less than that of an LVT flip-flop, as shown in Table I, even though the layout area remains unchanged, as shown in Figure 3(d).

3. ALLOCATION ALGORITHM OF TWO V_TS

Once we have a netlist of the sequential circuit, we need to determine how each gate will be implemented. The allocation algorithm for sequential circuits that we will go on to present selects either high- or low-V_t for all the combinational gates, and one of the four implementation types (LVT, HVT, MVT-I, and MVT-II) introduced in the previous section, for flip-flops. Our algorithm is based on the concept of the sensitivity of a gate, which we use to determine the priority of each gate for allocation. This sensitivity is the change of leakage that would be caused by a change of implementation, divided by the change in timing of the whole circuit. We now expand on this concept.

3.1 Sensitivity

We can perform a static timing analysis (STA) on a gate-level netlist of a sequential circuit to obtain the slack in each net. These slacks are continuous along the paths that are more critical, and discontinuous from noncritical inputs to an output of a gate, from a multifanout net to noncritical fanouts, and across flip-flops. We will denote a set of successive nets with continuous slack as N_i, and the value of that slack as $S(N_i)$. All the nets in the netlist can thus be grouped based on slack continuity, which we denote as $\mathcal{N} = \{N_1, N_2, \dots\}$. A set of net groups with continuous negative slacks can then be written $\mathcal{N}_- = \{N_i \in \mathcal{N} | S(N_i) < 0\}$.

Example 1. Consider an example netlist shown in Figure 5(a). The number inside each gate indicates its delay, and the timing parameters of the flip-flops are also given. Let us assume that the clock period is 70, the signal arrival times (ATs) at the primary inputs i_1 and i_2 are both 0, and the required arrival time (RAT) at the primary output o_1 is 70. For simplicity of presentation, we will not distinguish between rising and falling signal phases, and we will assume an equal delay for each pair of input and output pins of a gate. The slack of each net, computed by performing STA, is shown in the figure, from which we can identify the set of net groups with continuous slacks $\mathcal{N} = \{N_1, N_2, N_3, N_4, N_5, N_6, N_7\}$;

$$
\begin{array}{ll}
N_1 = \{n_1, n_2, n_3, n_7, n_8, n_9\}, & S(N_1) = -30, \\
N_2 = \{n_4, n_5\}, & S(N_2) = -25, \\
N_3 = \{n_6\}, & S(N_3) = 0, \\
N_4 = \{n_{10}\}, & S(N_4) = -20, \\
N_5 = \{n_{11}, n_{12}, n_{13}\}, & S(N_5) = -25, \\
N_6 = \{n_{14}, n_{15}\}, & S(N_6) = -10, \\
N_7 = \{n_{16}\}, & S(N_7) = 20,
\end{array}
$$

and corresponding set of net groups with continuous negative slacks $\mathcal{N}_- = \{N_1, N_2, N_4, N_5, N_6\}$.

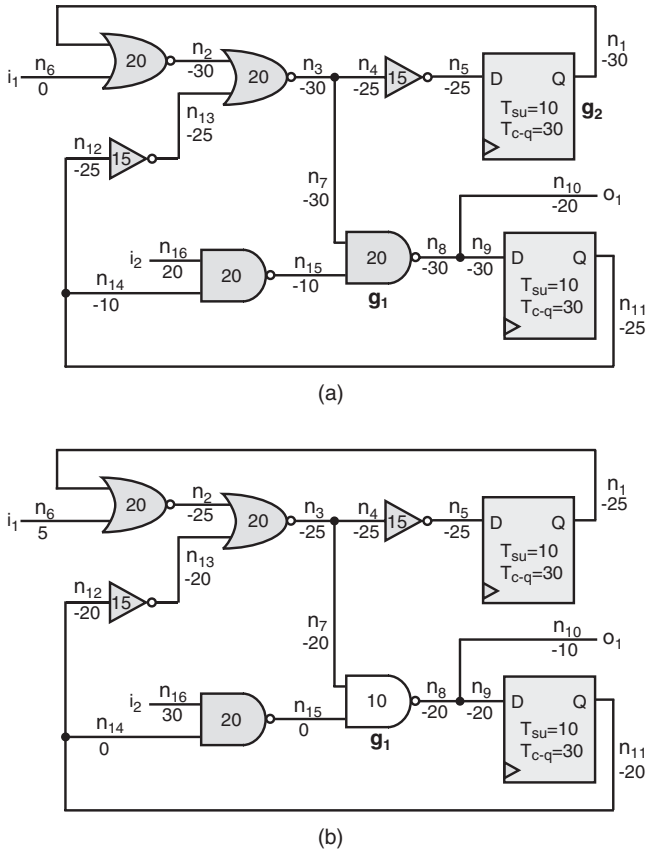

Fig. 5. An example netlist: (a) initial netlist and (b) sensitivity calculation for gate g_1.

The sensitivity of a gate i to a change of implementation from its current type t_c to a new type t_n can be defined as:

$$S_i = \frac{\Delta I_i}{\Delta D_i}, \quad \text{where } \Delta I_i = I_i(t_n) - I_i(t_c), \qquad (1)$$
$$\Delta D_i = \sum_{N_j \in \mathcal{N}'_-} S(N_j) - \sum_{N_j \in \mathcal{N}_-} S(N_j),$$

where ΔI_i is the increase in leakage. $\mathcal{N}'$ is a set of net groups with continuous slacks when gate i is implemented in a new type t_n; and the same set is denoted by $\mathcal{N}$ when gate i is implemented in a current type t_c. Note that both terms in the expression for ΔD_i are summed over net groups of negative slacks. Therefore ΔD_i indicates the extent of the potential improvement in overall timing of a circuit, as regards negative slacks. Note also that computing ΔD_i involves an incremental STA. New AT is propagated from i toward the gates within its fanout cone and new RAT is propagated toward the gates within its fanin cone; propagation stops when there is no change in AT or in RAT. Similar cost functions have been introduced [Sirichotiyakul et al. 1999; Karnik et al. 2002]

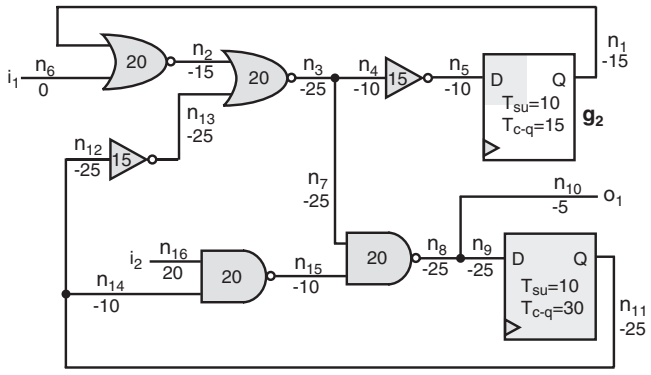

Fig. 6. Sensitivity calculation for a flip-flop g_2.

to select threshold voltages in combinational circuits of the transistor level, but in this case the improvement in timing is only calculated locally, from the timing arcs of a gate to which a transistor belongs [Sirichotiyakul et al. 1999], or from the timing paths that include a transistor [Karnik et al. 2002].

Example 2. Assuming that all the gates in Figure 5(a) are implemented in high-V_t, we compute the sensitivity of NAND gate g_1 when we try to change its implementation to low-V_t. Suppose its leakage current increases from 20 to 200, so that $\Delta I_1 = 180$, while its delay decreases from 20 to 10, as shown in Figure 5(b). The slacks of the nets that are affected by this change are recalculated, and the results are shown in Figure 5(b). We can see from Figure 5(b) that there are four net groups with continuous negative slacks, -25, -20, -20, and -10, whereas there were five groups in the initial netlist of Figure 5(a), which were -30, -25, -20, -25, and -10, as explained in Example 1. Therefore, $\Delta D_1 = (-75) - (-110) = 35$, and $S_1 = \frac{I_1}{D_1} = \frac{180}{35} = 5.14$.

Example 3. We now try to compute the sensitivity of a flip-flop g_2 if its potential implementation is changed from high-V_t (Figure 5(a)) to MVT-I (Figure 6). We will assume that its leakage current increases from 50 to 500, so that $\Delta I_2 = 450$, and that its T_{c-q} decreases from 30 to 15, while its T_{su} is unchanged. The slacks in the nets that are affected by this change are recalculated, with the results shown in Figure 6. There are now five net groups with continuous negative slacks, -15, -25, -10, -10, and -5, and the sum of these slacks is -65. Therefore $\Delta D_2 = (-65) - (-110) = 45$, and $S_2 = \frac{450}{45} = 10$.

If all the gates in a netlist are initially in high-V_t, we can only determine the sensitivity of a combinational gate to a potential change of implementation to low-V_t. But a flip-flop might be changed to low-V_t, to MVT-I, or to MVT-II; the sensitivity of a flip-flop at high-V_t is its lowest sensitivity to any of these three possible changes.

Once we have computed the sensitivities of all the gates, we select the gate with minimum sensitivity and change its implementation type appropriately. If the selected gate is a combinational gate, the change (to low-V_t) is final; but

Algorithm *Allocate_Two_V_ts*
Input: a netlist of gates $G = 1, 2, \ldots$ and timing constraints
Output: the same netlist, with implementation type of each gate
L1 Set high-V_t for all gates $i \in G$
L2 Initialize a list of candidate gates for allocation $L = \{G\}$
L3 Combinational gate types = {HVT, LVT}
L4 Flip-flop types = {HVT, {MVT-I, MVT-II}, LVT}
L5 *Sensitivity_Allocate* (L)

Function *Sensitivity_Allocate* (L)
L6 Perform static timing analysis
L7 $\mathcal{N}_-$ = a set of net groups with continuous negative slacks
L8 Compute sensitivity of each gate $i \in L$
L9 **while** $\mathcal{N}_- \neq \phi$ **do**
L10 g_{min} = gate of minimum sensitivity
L11 **if** $g_{min} = \phi$ **then** return fail
L12 **if** (g_{min} = combinational gate) or [($g_{min} = FF$) and (sensitivity is to LVT)] **then**
L13 $L = L - \{g_{min}\}$
L14 Change implementation type of g_{min}
L15 Perform incremental timing analysis and update $\mathcal{N}_-$
L16 Update sensitivities of gates in L affected by g_{min}

Fig. 7. Pseudocode of the allocation algorithm.

if it is a flip-flop and it is not changed to low-V_t, so that it is now either MVT-I or MVT-II, its implementation may be further changed to a type that has not been tried already in later iterations of the allocation algorithm.

3.2 Allocation Algorithm

Figure 7 is a sketch of the allocation algorithm using two V_ts. The input to the algorithm is a technology-mapped gate-level netlist with timing constraints (arrival times at primary inputs, required arrival times at primary outputs, and the clock period). All the gates are initially assigned to high-V_t (L1), and are considered to be candidates for allocation (L2). Only the conversion from high-V_t (HVT) to low-V_t (LVT) is considered for combinational gates; we consider two types of implementation for combinational gates (L3). For flip-flops, we consider the conversion from HVT to MVT-I or MVT-II, from HVT to LVT, and from MVT-I or MVT-II to LVT; we consider four types of implementation for flip-flops (L4). Once flip-flop is converted to LVT, it is removed from further consideration; if HVT flip-flop is converted to MVT-I or MVT-II, however, it could be converted to LVT in later iterations. A static timing analysis is run on the netlist and a set of net groups with continuous negative slacks are examined (L6 and L7). We then compute the sensitivity of each gate (L8), and select the gate with the lowest, which we call g_{min} (L10). If the selected gate is a combinational gate (L12), it is changed to low-V_t (L14) and removed from the list of candidates for further allocation (L13). If it is a flip-flop and it is being changed to LVT (L12), the change is made and the flip-flop is removed from the list; otherwise it is changed to MVT-I or MVT-II, and it remains in the list to be considered during subsequent iterations. We then perform incremental timing analysis (L15), and update the sensitivities of the gates that are affected by the change

Table III. Benchmark Circuits

Name	# Gates	# FFs	Clock period (ps)
s298	106	14	780
s400	147	21	880
s641	152	14	1020
s838	245	32	2750
s1423	836	74	1880
s5378	1253	160	1160
b03	141	30	930
b04	1163	66	1940
b07	542	49	1690
b08	118	21	1080
b09	131	28	940
b12	1204	119	1520
ac97_dma_if	198	18	400
ac97_prc	114	29	520
ac97_rf	727	157	710
ac97_soc	132	32	720
bc_fifo_basic	578	89	1170
irda_crc32	133	32	630
irda_data_ctrl	401	39	980
irda_fir_flag_det	201	37	1200
irda_reg	554	92	730
mc_adr_sel	809	90	1030
mc_cs_rf	532	74	780
mc_obct	348	56	660
mc_refresh	158	21	830
oc_fifo_basic	515	74	940

in g_{min} (L16). This procedure is repeated until no negative slacks are left in the netlist (L9). If we still have negative slacks but the list is empty (L11), the procedure terminates in failure.

3.3 Experimental Results

We performed experiments on a set of sequential circuits taken from the ISCAS [Brglez et al. 1989] and ITC benchmarks [Corno et al. 2000]. We also included circuits extracted from several open cores[1] including an audio codec and a cryptography core. The first three columns of Table III give the name, the number of combinational gates, and the number of flip-flops of each circuit; and the last column is the clock period, which we assume corresponds to the critical path delay of each circuit when all gates are in low-V_t, which is the fastest clock period for that circuit. The ATs of primary inputs are assumed to be 0, and the RATs of primary outputs are assumed to be equal to the clock period.

Each circuit was synthesized with SIS [Sentovich et al. 1992] and mapped into a gate library, which we based on 65-nm commercial technology. Technology mapping was performed using a weighted sum of area and delay as the

[1]http://www.opencores.org

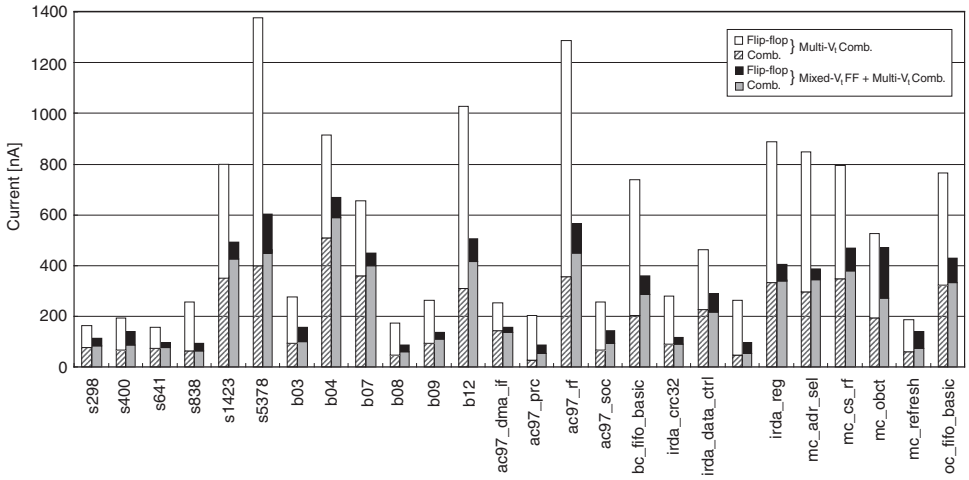

Fig. 8. Comparison of leakage current between conventional multi-V_t on combinational gates (left-hand bars), and our method (right-hand bars).

cost function, and gate sizing was included in the technology mapping process. The technology-mapped netlist was read into our allocation algorithm, which was implemented in SIS. As a reference of comparison, we allocates multi-V_t to combinational gates alone while flip-flops remain in low-V_t, which we call conventional allocation.

3.3.1 *Effectiveness of Proposed Method.* Experimental results are illustrated in Figure 8, in which the left-hand bar for each circuit is the leakage corresponding to the netlist obtained by the conventional method, and the right-hand bar is the leakage with our method. Each bar has two components: the leakage from combinational gates and that from flip-flops. The leakage current was obtained by simulating each circuit with SPICE, which was repeated hundred times using different random input vectors for the primary inputs and the results were averaged.

Our method leads to an increase in the leakage from combinational gates, because some of the slacks, which are used exclusively by combinational gates in the conventional method, are now shared between combinational gates and flip-flops. However, in the ac97_dma_if and irda_data_ctrl, there is a reduction in leakage from the combinational gates, due to their significant use of MVT-II flip-flops (see Figure 9), which have a smaller falling T_{su} than normal low-V_t flip-flops (see Table II). Our method reduces the leakage from the flip-flops by 78% on average, which is an understandable consequence of the new distribution of flip-flop types achieved by our allocation algorithm, as shown in Figure 9. Overall, our method reduces the total leakage by on average of 42% (minimum saving of 11% for mc_obct and maximum saving of 64% for s838).

Timing constraint affects the result of allocation. Figure 10 shows leakage current from flip-flops and combinational gates of five example circuits while we vary their clock periods. Overall leakage decreases as clock period increases

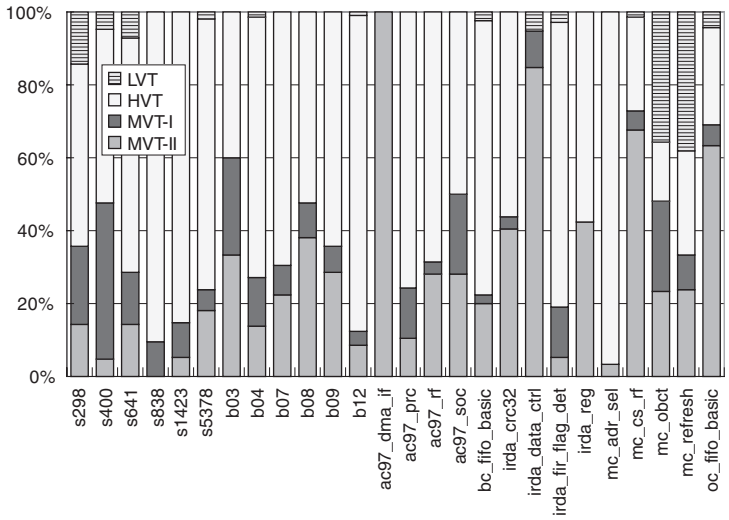

Fig. 9. Distribution of flip-flops after running the allocation algorithm.

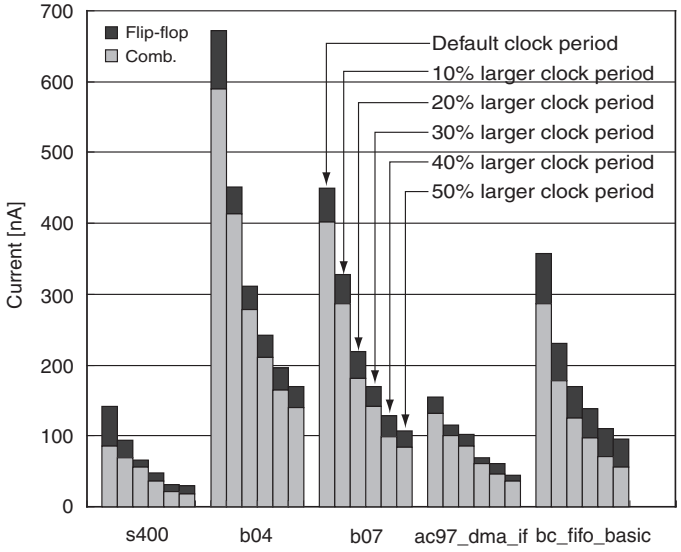

Fig. 10. Distribution of leakage current with varying clock period.

since more gates can be mapped to high-V_t. The proportion of leakage from flip-flops is kept small due to the use of mixed-V_t flip-flops.

The algorithm in Figure 7 starts from all high-V_t gates. We could start from all low-V_t gates instead and gradually convert gates to high-V_t. The definition of sensitivity (1), however, should be refined; ΔI_i is now the decrease in leakage and ΔD_i now represents the amount of potential deterioration in overall timing of a circuit. We thus have to select a gate of maximum sensitivity. Applying this strategy of algorithm yields on average of 5% increase of leakage.

Table IV. Comparison of Leakage Current After Monte Carlo Simulation of Example Circuits. The Mean is Denoted by μ and the Standard Deviation by σ

	(Mixed-V_t FF + Multi-V_t Comb.) / (Multi-V_t Comb.)						
	Det.	Stat. (SS)		Stat. (NN)		Stat. (FF)	
Benchmark	(SS)	μ	σ	μ	σ	μ	σ
s298	0.71	0.71	0.76	0.68	0.77	0.69	0.84
s400	0.73	0.76	0.93	0.73	0.95	0.74	0.94
b08	0.49	0.50	0.52	0.44	0.50	0.44	0.56
b09	0.53	0.53	0.52	0.47	0.52	0.47	0.56
ac97_prc	0.44	0.44	0.46	0.38	0.47	0.40	0.53
irda_crc32	0.41	0.40	0.35	0.34	0.35	0.35	0.40

3.3.2 *Statistical Considerations.* The variation of V_t increases as devices get smaller. It is reported [Chiang and Kawa 2007] that the standard deviation of threshold voltage is about 6% of its mean for 180-nm technology, but about 11% for 65-nm technology. Since leakage current is significantly affected by process variation, it is important to assess leakage current from a statistical point of view. We took two netlists, one produced by our method and the another produced by the conventional method, for each of six small example circuits, and simulated them with SPICE, using a Monte Carlo method to obtain the distribution of leakage resulting from within-die process variations. The ratios of the mean (μ) and standard deviation (σ) of the two distributions are shown in the third and fourth columns of Table IV, respectively. The ratio of the leakages calculated deterministically without allowing for within-die process variation, which corresponds to Figure 8, is given in the second column for comparison. The ratios in the second and third columns are quite similar, showing that our method remains effective in the presence of within-die process variation. Additionally, our method reduces σ for all six circuits due to less use of low-V_t devices, suggesting an improvement in the predictability of leakage.

The netlists (both from the conventional and our method) were obtained assuming the slow process corner (SS); the second, third, and fourth columns also correspond to SS. We performed the Monte Carlo simulation using the same netlists, but this time in the nominal (NN) and fast (FF) process corners to simulate die-to-die process variations. The ratios of μ and σ in these other two process corners are shown in the last four columns of Table IV. Comparing the ratios of μ in three process corners (the third, fifth, and seventh columns) suggests that our method saves more leakage in more leaky process corners (NN and FF), which is because of less use of low-V_t devices.

A second statistical experiment involved assessing the probability density function (PDF) of the critical path delay after applying our method. We implemented a statistical static timing analysis (SSTA) engine [Liou et al. 2001] on SIS [Sentovich et al. 1992]. The delay of each gate was modeled as a discrete PDF, with threshold voltages assumed to follow a normal distribution with a standard deviation of 20 mV. SSTA was then run on the netlist with all gates in low-V_t, the netlist from applying conventional multi-V_t to the combinational gates, and the netlist resulting from our method, and then the delay PDFs were

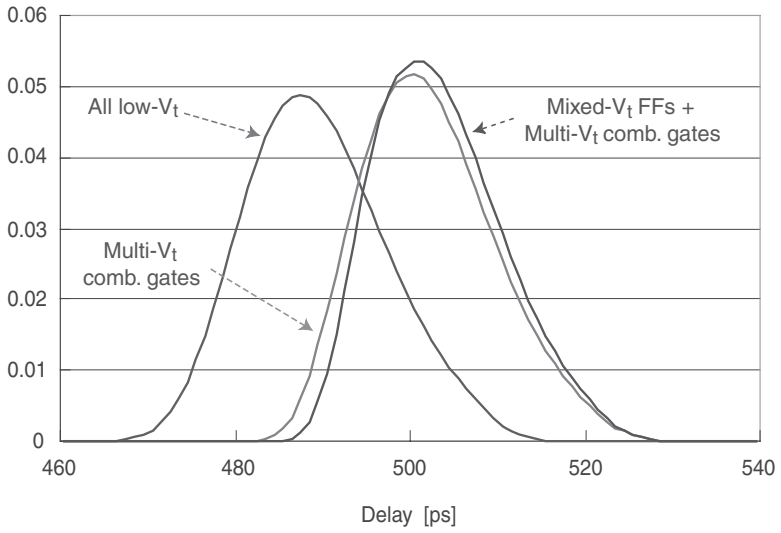

Fig. 11. Probability density functions of critical path delay for three netlists of the circuit b12.

derived for all primary outputs. For each netlist, we combined the PDFs of primary outputs by finding their statistical maximum [Liou et al. 2001] to derive a single PDF of critical path delay. Figure 11 compares the PDFs of the three netlists for circuit b12. The PDF corresponding to the application of multi-V_t to the combinational circuit is clearly shifted to the right of the PDF for the circuits where everything is low-V_t, which is the fastest. This indicates a drop in the timing yield, which is the probability that a circuit will satisfy a given timing constraint. The PDF of the netlist from our method is further shifted to the right, but not much more than occurs when the conventional method is used to apply multi-V_t to the combinational gates, implying a small sacrifice in timing yield, for a large saving in overall leakage, when our method is employed.

4. EXTENSION TO THREE V_TS

Many modern semiconductor processes support three V_ts: low-V_t (LVT), regular-V_t (RVT), and high-V_t (HVT). They are usually used in pairs: LVT and RVT for high-performance applications, or RVT and HVT for low-power applications, are the preferred combinations. However, at the cost of additional masks, three V_ts allow more flexibility in making a trade-off between leakage and delay, and are used in some high-performance circuits such as microprocessors [Yamashita et al. 2000; Geissler et al. 2002; Clabes et al. 2004; Ito et al. 2007]. In this section, we extend our mixed-V_t flip-flops and allocation algorithm to three V_ts.

4.1 Mixed-V_t Flip-Flops

When we designed MVT-I and MVT-II for two V_ts, we allocated the same V_t to a group of internal gates in each flip-flop (see Figure 2(c) and (d)): one group

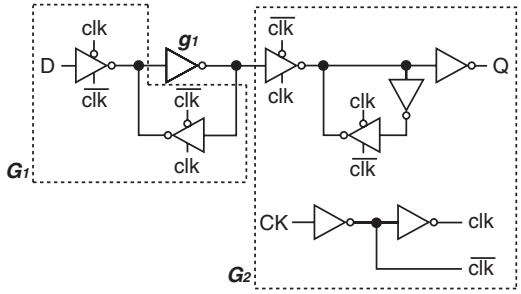

Fig. 12. Groups of gates G_1 and G_2 with the inverter g_1 for designing mixed-V_t flip-flops based on three V_ts.

Table V. Mixed-V_t Flip-Flops Based on Three V_ts

Name of FF	V_t type		
	Group G_1	Group G_2	Inverter g_1
LL	LVT	LVT	LVT
LR	LVT	RVT	LVT
LH	LVT	HVT	HVT
RL	RVT	LVT	LVT
RR	RVT	RVT	RVT
RH	RVT	HVT	HVT
HL	HVT	LVT	HVT
HR	HVT	RVT	HVT
HH	HVT	HVT	HVT

affects T_{su} and the other affects $T_{c\text{-}q}$. This grouping keeps the footprint of the flip-flops constant. The inverter g_1 belongs to the first group in an MVT-I but to the second in an MVT-II, so we separate g_1 from both groups. We can use the same grouping to design mixed-V_t flip-flops based on three V_ts, as shown in Figure 12.

Depending on the types of V_t that are assigned to groups G_1 and G_2, there are nine possible types of mixed-V_t flip-flops, which are listed and named in Table V. The last column of this table gives the type of V_t assigned to the inverter g_1. In the flip-flops that use low-V_t for group G_1 (LL, LR, and LH), the V_t for g_1 is selected so that the T_{su} of the LR and LH is not larger than that of the LL (see Figure 13(a)). This makes the LR and LH more like the MVT-II. Similar considerations apply to the group containing the RL, RR, and RH types, which use regular-V_t for G_1, and to the group containing the HL, HR, and HH types, which receive high-V_t for G_1 (see Figure 13(a)).

If we compare the LL, RL, and HL types, which are the flip-flops that receive low-V_t in group G_2, the RL and HL types, which are more like the MVT-I, have a higher T_{su} than the LL, but almost the same $T_{c\text{-}q}$ (see Figure 13(a)). The same holds for the groups containing the LR, RR, and HR, and the LH, RH, and HH types. Figure 13(b) shows the leakage current of all nine mixed-V_t flip-flop. The flip-flops of types LL, RL, and HL are the most leaky since they employ low-V_t in group G_2, which has more gates than G_1.

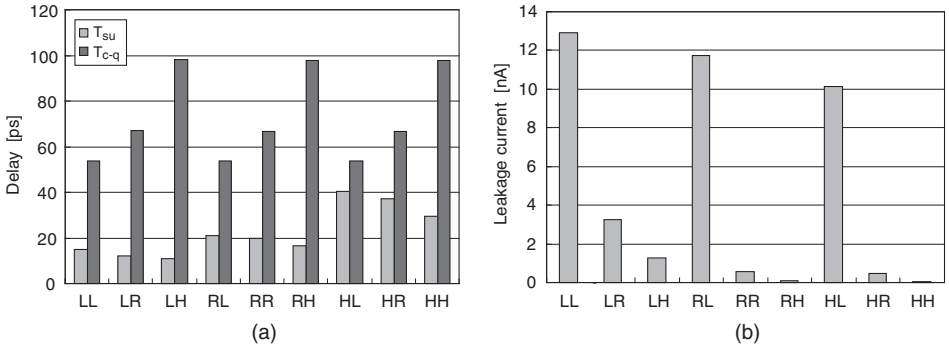

Fig. 13. Mixed-V_t flip-flops based on three V_ts: (a) timing parameters and (b) leakage current.

Algorithm *Allocate_Three_V_ts*
Input: a netlist of gates $G = 1, 2, \ldots$ and timing constraints
Output: the same netlist, with implementation type of each gate

Phase-I:
L1 Set all gates to regular-V_t $i \in G$
L2 Initialize a list of candidate gates for allocation $L = \{G\}$
L3 Combinational gate types = {RVT, LVT}
L4 Flip-flop types = {RR, {RL, LR}, LL}
L5 *Sensitivity_Allocate* (L)

Phase-II:
L6 Convert combinational gates: RVT → HVT
L7 Convert flip-flops: RR → HH, RL → HL, LR → LH
L8 $L = \{G\}$
L9 **forall** $i \in L$ **do**
L10 **if** (i is a combinational gate and i is LVT) or (i is a flip-flop and i is LL) **then**
L11 $L = L - i$
L12 Combinational gate types = {HVT, RVT}
L13 Flip-flop types = {LH, ϕ, LR}, {HL, ϕ, RL}, {HH, {RH, HR}, RR}
L14 *Sensitivity_Allocate* (L)

Fig. 14. Pseudocode of the allocation algorithm using three V_ts.

4.2 Mixed-V_t Allocation Algorithm

In this allocation, combinational gates can be at HVT, RVT, or LVT, and flip-flops can take one of the nine implementation types listed in Table V. The allocation algorithm, shown in Figure 14, consists of two phases. In the first phase (L1 to L5), we use only regular-V_t and low-V_t, so as to satisfy the timing constraints while we maximize the use of regular-V_t. We first set all the gates, including the flip-flops, to regular-V_t (L1), so that the combinational gates are all type RVT and the flip-flops are all type RR. Combinational gates can either be of type RVT or LVT (L3); and flip-flops are allowed to be RL, LR, and LL, as well as the initial RR type (L4). Note that RL and LR are respectively similar to MVT-I and MVT-II, in this phase of allocation, and LL is the final type that flip-flops can have. We then call the procedure *Sensitivity_Allocate* (see Figure 7).

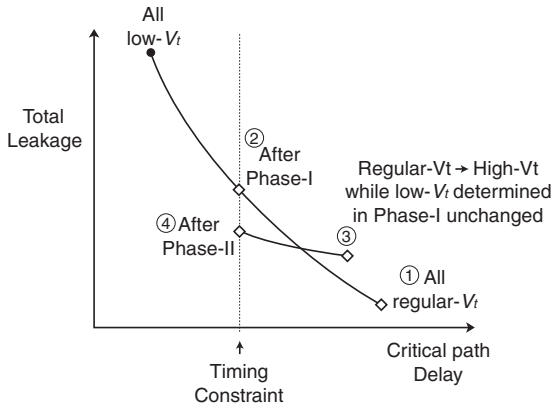

Fig. 15. Concept of the allocation algorithm using three V_ts. The circled numbers indicate the order of algorithm progress.

Starting from the netlist obtained in the first phase, the second phase (L6 to L14), allocates high-V_t and regular-V_t, using high-V_t as widely as possible, so as to reduce the leakage current further. However, the allocation of low-V_t made in the first phase remains unchanged. This process starts reassigning all the combinational gates and flip-flops which were assigned to regular-V_t in the first phase to high-V_t (L6, L7). The resulting netlist violates the timing constraints which were met at the end of the first phase (L5). As we have already mentioned, the combinational gates that were assigned to low-V_t, and the flip-flops that were assigned to LL in the first phase, are not considered for further allocation in this second phase (from L8 to L11). The candidate combinational gates in the list L are of types HVT and RVT (L12). If flip-flops of types LH and HL are selected, they can respectively be converted directly to LR and RL types. HH flip-flops can be converted to RH and HR, which are similar to MVT-I and MVT-II in this phase of allocation, as well as to type RR (L13). The procedure *Sensitivity_Allocate* is called again (L14).

The effect of the allocation algorithm on leakage and delay is shown diagrammatically in Figure 15, which depicts successive design points. All gates are initially in regular-V_t (1). In the first phase, we explore the design points on the curve spanned by mixed use of regular-V_t and low-V_t. At the end of the first phase, we reach the design point, where timing constraints are satisfied (2). If we convert all regular-V_t to high-V_t while we keep low-V_t determined in the first phase, our design point shifts to the right (3) as shown in Figure 15, where we have less leakage but timing constraints are violated again. In the second phase, we then explore the design points but in different curve spanned by mixed use of high-V_t and regular-V_t, while low-V_t determined in the first phase unchanged. Note that the slope of the curve in the second phase (point 3 to point 4) is less steep, due to the exponential dependency of leakage on threshold voltage, allowing us to satisfy the timing constraints with a lower leakage (4) compared to exploring design points in the first phase alone.

Table VI. Comparison of Leakage Current Resulting from the Allocation of Two V_ts and Three V_ts

Name	Two V_ts	Three V_ts	Saving (%)
s298	1238	1139	8.0
s400	1193	1114	6.6
s641	1098	1011	7.9
s838	1204	880	26.9
s1423	4299	3842	12.9
s5378	5194	4096	21.1
b03	2043	1773	13.2
b04	6410	5684	11.3
b07	4660	4255	8.7
b08	603	438	27.3
b09	1437	1374	4.4
b12	3571	2651	25.8
ac97_dma_if	1988	1945	2.2
ac97_prc	603	499	17.2
ac97_rf	5140	4435	13.7
ac97_soc	1296	1225	5.5
bc_fifo_basic	3635	3043	16.3
irda_crc32	1303	1072	17.7
irda_data_ctrl	2986	2626	12.1
irda_fir_flag_det	812	627	22.8
irda_reg	4594	3861	16.0
mc_adr_sel	2789	2102	24.6
mc_cs_rf	5988	5243	12.4
mc_obct	3094	3009	2.8
mc_refresh	1222	1160	5.1
oc_fifo_basic	4665	4160	10.8
Average			13.6

4.3 Experimental Results

The allocation algorithm shown in Figure 14 was implemented in SIS, and the circuits shown in Table III were again used for experiments. We assessed the effectiveness of three V_ts by comparing the leakage current for the netlist obtained with the allocation algorithm based on two V_ts (Figure 7) with that for the netlist obtained with the allocation algorithm based on three V_ts (Figure 14), which is shown in Table VI. Note that the two V_ts in the former case correspond to low-V_t and regular-V_t; using low-V_t and high-V_t yields 92% more leakage and so was not considered, and a combination of regular-V_t and high-V_t was not considered either, because the timing constraints of these circuits can never be met without using low-V_t. Using three V_ts yields an average additional leakage saving of 13.6% compared to two V_ts, as shown in Table VI.

The distribution of flip-flops after running *Allocate_Three_V_ts* is shown in Figure 16. Monte Carlo simulation of two netlists, one obtained after running *Allocate_Two_V_ts* and the other after running *Allocate_Three_V_ts*, were performed to assess leakage under process variations. The ratios of the mean (μ) and standard deviation (σ) of the two leakage distributions at three different process corners are shown in Table VII; the ratio of the leakages corresponding to Table VI is shown in the second column for comparison.

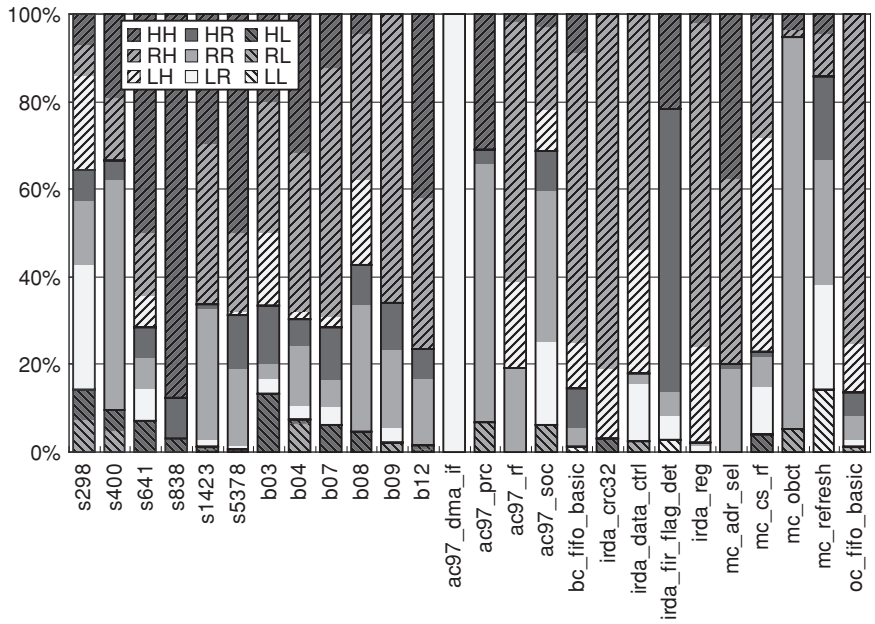

Fig. 16. Distribution of flip-flops after running *Allocate_Three_V_ts*.

Table VII. Comparison of Leakage Current After Monte Carlo Simulation of Example Circuits. The Mean is Denoted by μ and the Standard Deviation by σ

	Three-V_ts / Two-V_ts						
	Det.	Stat. (SS)		Stat. (NN)		Stat. (FF)	
Benchmark	(SS)	μ	σ	μ	σ	μ	σ
s298	0.92	0.93	0.94	0.93	0.94	0.93	0.93
s400	0.93	0.96	0.96	0.95	0.95	0.93	0.93
b08	0.73	0.84	0.83	0.80	0.78	0.76	0.73
b09	0.96	0.98	0.99	0.97	0.98	0.96	0.98
ac97_prc	0.83	0.88	0.92	0.87	0.90	0.85	0.87
irda_crc32	0.82	0.84	0.81	0.81	0.75	0.77	0.68

5. CONCLUSION

Although it is in widespread use, the value of the current multi-V_t approach is limited, since it considers only the combinational parts of a circuit, even though the sequential elements contribute a proportion of the total leakage which is not negligible, and is sometimes significant. We have proposed mixed-V_t flip-flops, which have a substantially lower leakage than conventional low-V_t flip-flops, at the cost of an increase in delay, either in the setup time or in the clock-to-Q delay but not in both. This is achieved without any increase in area, due to the careful selection of transistors for the high-V_t implementation. This concept is general and any kind of conventional static flip-flop could be transformed to a mixed-V_t flip-flop. We have also presented a heuristic algorithm that substitutes mixed-V_t flip-flops for conventional flip-flops as well as allocating high- or low-V_t to

each combinational gate. In addition, the mixed-V_t flip-flops and the allocation algorithm were both extended to the use of three V_ts.

REFERENCES

BRGLEZ, F., BRYAN, D., AND KOZMINSKI, K. 1989. Combinational profiles of sequential benchmark circuits. In *Proceedings of the International Symposium on Circuits and Systems*. 1929–1934.

CHIANG, C. AND KAWA, J., Eds. 2007. *Design for Manufacturability and Yield for Nano-Scale CMOS*. Springer.

CHOI, B. AND SHIN, Y. 2007. Lookup table-based adaptive body biasing of multiple macros. In *Proceedings of the International Symposium on Quality Electronic Design*. 533–538.

CLABES, J. ET AL. 2004. Design and implementation of the POWER5 microprocessor. In *Proceedings of the IEEE International Solid-State Circuits Conference*. 56–57.

CLARK, L. T., MORROW, M., AND BROWN, W. 2004. Reverse-body bias and supply collapse for low effective standby power. *IEEE Trans. VLSI Syst. 12*, 9, 947–956.

CORNO, F., REORDA, M., AND SQUILLERO, G. 2000. RT-level ITC'99 benchmarks and first ATPG results. *IEEE Des. Test Comput. 17*, 3, 44–53.

FRIEDRICH, J., MCCREDIE, B., JAMES, N., HUOTT, B., CURRAN, B., FLUHR, E., MITTAL, G., CHAN, E., CHAN, Y., PLASS, D., CHU, S., LE, H., CLARK, L., RIPLEY, J., TAYLOR, S., DILULLO, J., AND LANZEROTTI, M. 2007. Design of the Power6 microprocessor. In *Proceedings of the IEEE International Solid-State Circuits Conference*. 96–97.

GEISSLER, S. ET AL. 2002. A low-power RISC microprocessor using dual PLLs in a 0.13μm SOI technology with copper interconnect and low-k BEOL dielectric. In *Proceedings of the IEEE International Solid-State Circuits Conference*. 148–149.

GUPTA, P., KAHNG, A. B., SHARMA, P., AND SYLVESTER, D. 2006. Gate-length biasing for runtime-leakage control. *IEEE Trans. Comput.-Aid. Des. 25*, 8, 1475–1485.

HO, Y. AND HWANG, T. 2004. Low power design using dual threshold voltage. In *Proceedings of the Asia South Pacific Design Automation Conference*. 205–208.

INUKAI, T., TAKAMIYA, M., NOSE, K., KAWAGUCHI, H., HIRAMOTO, T., AND SAKURAI, T. 2000. Boosted gate MOS (BGMOS): device/circuit cooperation scheme to achieve leakage-free giga-scale integration. In *Proceedings of the Custom Integrated Circuits Conference*. 409–412.

ITO, M. ET AL. 2007. A 390MHz single-chip application and dual-mode baseband processor in 90nm triple-Vt CMOS. In *Proceedings of the IEEE International Solid-State Circuits Conference*. 274–275.

KARNIK, T., YE, Y., TSCHANZ, J., WEI, L., BURNS, S., GOVINDARAJULU, V., DE, V., AND BORKAR, S. 2002. Total power optimization by simultaneous dual-Vt allocation and device sizing in high performance microprocessors. In *Proceedings of the Design Automation Conference*. 486–491.

KAWAGUCHI, H., NOSE, K., AND SAKURAI, T. 2000. A super cut-off CMOS (SCCMOS) scheme for 0.5-V supply voltage with picoampere current. *IEEE J. Solid-State Circ. 35*, 10, 1498–1501.

KETKAR, M. AND SAPATNEKAR, S. S. 2002. Standby power optimization via transistor sizing and dual threshold voltage assignment. In *Proceedings of the International Conference on Computer Aided Design*. 375–378.

KIM, H.-O., SHIN, Y., KIM, H., AND EO, I. 2006. Physical design methodology of power gating circuits for standard-cell-based design. In *Proceedings of the Design Automation Conference*. 109–112.

KURODA, T., FUJITA, T., MITA, S., NAGAMATSU, T., YOSHIOKA, S., SUZUKI, K., SANO, F., NORISHIMA, M., MUROTA, M., KAKO, M., KINUGAWA, M., KAKUMU, M., AND SAKURAI, T. 1996. A 0.9-V, 150-MHz, 10-mW, 4 mm^2, 2-D discrete cosine transform core processor with variable threshold-voltage (VT) scheme. *IEEE J. Solid-State Circ. 31*, 11, 1770–1779.

LIOU, J.-J., CHENG, K.-T., KUNDU, S., AND KRSTIC, A. 2001. Fast statistical timing analysis by probabilistic event propagation. In *Proceedings of the Design Automation Conference*. 661–666.

MUTOH, S., DOUSEKI, T., MATSUYA, Y., AOKI, T., SHIGEMATSU, S., AND YAMADA, J. 1995. A 1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS. *IEEE J. Solid-State Circ. 30*, 8, 847–854.

OHKUBO, N. AND USAMI, K. 2006. Delay modeling and static timing analysis for MTCMOS circuits. In *Proceedings of the Asia South Pacific Design Automation Conference*. 570–575.

ROY, K., MUKHOPADHYAY, S., AND MAHMOODI-MEIMAND, H. 2003. Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits. *Proc. IEEE 91*, 2, 305–327.

SENTOVICH, E. M., SINGH, K. J., LAVAGNO, L., MOON, C., MURGAI, R., SLDANHA, A., SAVOJ, H., STEPHAN, P. R., BRAYTON, R. K., AND VINCENTELLI, A. S. 1992. SIS: a system for sequential circuit synthesis. Tech. rep. UCB/ERL M92/41.

SEOMUN, J., KIM, J., AND SHIN, Y. 2007. Skewed flip-flop transformation for minimizing leakage in sequential circuits. In *Proceedings of the Design Automation Conference*. 103–106.

SHAH, S., SRIVASTAVA, A., SHARMA, D., SYLVESTER, D., AND BLAAUW, D. 2005. Discrete Vt assignment and gate sizing using a self-snapping continuous formulation. In *Proceedings of the International Conference on Computer Aided Design*. 704–711.

SIRICHOTIYAKUL, S., EDWARDS, T., OH, C., ZUO, J., DHARCHOUDHURY, A., PANDA, R., AND BLAAUW, D. 1999. Stand-by power minimization through simultaneous threshold voltage selelction and circuit sizing. In *Proceedings of the Design Automation Conference*. 436–441.

SULTANIA, A. K., SYLVESTER, D., AND SAPATNEKAR, S. S. 2004. Tradeoffs between gate oxide leakage and delay for dual Tox circuits. In *Proceedings of the Design Automation Conference*. 761–766.

WANG, Q. AND VRUDHULA, S. 1998. Static power optimization of deep submicron CMOS circuits for dual Vt technology. In *Proceedings of the International Conference on Computer Aided Design*. 490–496.

WEI, L., CHEN, Z., JOHNSON, M., ROY, K., AND DE, V. 1998. Design and optimization of low voltage high performance dual threshold CMOS circuits. In *Proceedings of the Design Automation Conference*. 489–494.

YAMASHITA, T. ET AL. 2000. A 450MHz 64b RISC processor using multiple threshold voltage CMOS. In *Proceedings of the IEEE International Solid-State Circuits Conference*. 414–415.

Received January 2009; revised June 2009, July 2009; accepted September 2009