Minimizing Leakage Power in Sequential Circuits by Using Mixed $V_t$ Flip-Flops

Jaehyun Kim and Youngsoo Shin
Department of Electrical Engineering, KAIST
Daejeon 305-701, Korea

Abstract—Dual $V_t$ has been widely used to control leakage, while, at the same time, satisfying circuit performance. However, current approaches target the combinational circuits even though sequential elements, such as flip-flops and latches, contribute an appreciable proportion of the total leakage. The use of dual $V_t$ flip-flops is limited to circuits of large timing slack, because introducing high $V_t$ flip-flops in place of low $V_t$ ones yields abrupt change in timing. We propose mixed $V_t$ flip-flops, which are designed by using both low and high $V_t$, but in different transistors. Compared to low $V_t$ flip-flop, the mixed $V_t$ flip-flops exhibit increased delay, but either on setup time or on clock-to-Q delay but not on both, while their leakage is greatly reduced.

We extend the conventional sensitivity-based dual $V_t$ allocation algorithm to incorporate mixed $V_t$ flip-flops together with dual $V_t$ combinational gates. Experimental results show that an average leakage saving of 31% is achieved, compared to the use of dual $V_t$ on combinational subcircuits alone. The leakage of the flip-flops themselves is cut by 57% on average.

I. INTRODUCTION

Leakage current has been continuously growing to the point where it is now comparable to switching power. In recent nanometer CMOS technologies such as 90- and 65-nm, it is not uncommon to see leakage current being responsible for almost half of total power consumption [1]. Since, the leakage current is of little use other than being the index of circuit speed, it is important to minimize it. The leakage current consists of many components [2], but subthreshold leakage takes the largest proportion in most technologies.

There are many circuit techniques to suppress subthreshold leakage such as power gating and reverse body bias, which are two most popular ones [3]. However, these techniques involve substantial custom engineering for their implementation. For example, power gating, sizing current switch, designing retention flip-flops, employing custom power networks, and so on, need to be performed [4]. Moreover, their application to standard-cell-based designs involves significant customization of design flow due to the departure of their implementation from standard design flow. Static timing analysis specific to power gating [5] is an example.

On the other hand, the use of mixed $V_t$, mixed $T_{ox}$, or mixed $L_{gate}$ is transparent to designers, since they can be seamlessly integrated to most design tools, although the amount of leakage saving is much less than than that from power gating or reverse body bias. In addition, the leakage can be saved independent of mode of operation; it is saved only when circuits are in standby mode when power gating or reverse body bias is used.

A dual $V_t$ is especially popular for suppressing subthreshold leakage. A dual $V_t$ circuit utilizes low-threshold voltage ($V_t$) gates on timing-critical paths and high (or normal) $V_t$ gates on paths which are not critical to timing. Many algorithms for the use of dual $V_t$ have been proposed [6]–[9]. However, most algorithms proposed so far target the combinational portion of a circuit, although sequential elements such as flip-flops and latches are responsible for an appreciable proportion of the total leakage.

In order to see how important the leakage from flip-flops is, we took several circuits from ISCAS and ITC benchmarks, and simulated them with SPICE in 90-nm commercial technology model. Fig. 1(a) shows that the flip-flops contribute, on average, 43% to the total leakage of these circuits, which is already substantial. But, if dual $V_t$ is used in designing the combinational subcircuits [7], then the proportion of leakage in flip-flops goes up to 56% on average and can get as high as 63%.

Both transistor-level and gate-level mixed $V_t$ are used. In this paper, the former will be termed mixed $V_t$; the latter dual $V_t$, unless otherwise stated.
Instead of using dual $V_t$ only in the combinational subcircuits, which is the conventional approach, we may try to use high $V_t$ in some flip-flops (to the extent that the delay constraint of the circuit can be satisfied), and then apply a conventional dual $V_t$ to the combinational subcircuits. However, in this case, the proportion of combinational gates that can be assigned to high $V_t$ is likely to fall significantly because the original timing slacks, if present, are 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 can take advantage of high $V_t$ may not be very large, since assigning high $V_t$ to a flip-flop typically affects more than one of the timing paths in a circuit.

In this paper, we propose mixed $V_t$ flip-flops, in which we use both low and high $V_t$, but in different transistors. They are designed such a way that their footprint remains the same as the one of conventional flip-flop, in contrast to [10], which allows us to use them without perturbation in layout. We extend the conventional sensitivity-based dual $V_t$ allocation algorithm [7] to sequential circuits, which enables us to allocate dual $V_t$ to combinational gates and to employ mixed $V_t$ flip-flops in the same framework. We have compared the leakage of several benchmark circuits that are designed using the proposed methodology with the same circuits using traditional dual $V_t$ for the combinational subcircuits alone. The results show that we can reduce leakage by an additional 31% on average.

The remainder of the paper is organized as follows. In the next section, we discuss the design of mixed $V_t$ flip-flops, and present their leakage and timing characteristics. An algorithm for mixed $V_t$ allocation for sequential circuits is proposed in Section III. The experimental flow and the results with several benchmark circuits are presented in Section IV, and we draw conclusions in Section V.

II. MIXED $V_t$ FLIP-FLOPS

Fig. 2(a) shows an example (positive edge-triggered) D flip-flop with inverter and tristate inverter implementation, where low $V_t$ (LVT) is used for its implementation. Fig. 2(b) corresponds to the same flip-flop, but with high $V_t$ (HVT) for the implementation. The leakage of the two flip-flops (for each combination of input D and output Q) simulated with commercial 90-nm technology is shown in Table I. It can be readily seen that the leakage of HVT flip-flop is considerably lower than that of LVT one, as it must. However, the substitution of HVT flip-flop for LVT one in a circuit is abrupt in terms of timing as can be seen in Table II, since it affects all timing paths, which both setup time ($T_{su}$) and clock-to-Q delay ($T_{cQ}$) of the original flip-flop belongs to. Thus, the use of HVT flip-flop is limited only to those timing paths where large timing slacks are left even after optimizing combinational subcircuits through gate sizing, dual $V_t$, and so on.

Therefore, to achieve a smoother transition from using low $V_t$ to high $V_t$ in flip-flops, we design two more variants of flip-flops as shown in Fig. 2(c) and 2(d). This was motivated by two facts. First, although a flip-flop is physically a single gate, it has two components in timing viewpoint: master stage, which lies in the trailing end of circuit timing paths with its setup time; slave stage, which leads the circuit timing paths with its clock-to-Q delay. Second, mixed $V_t$ gate (e.g. nMOS in low $V_t$ and pMOS in high $V_t$ for an inverter), if it were to be needed, usually comes at a cost of increased layout area. Mixed $V_t$ (MVT) flip-flops, on the other hand, can be designed without any increase of area by careful use of high and low $V_t$ transistors.

In the first variant (Fig. 2(c)), which we call MVT-I flip-flop, we use high $V_t$ but only in those transistors that affect setup time. As shown in Table I, its leakage saving is not dramatic. However, the clock-to-Q delay remains almost unchanged as shown in Table II, implying that it can be substituted for flip-flops having slacks in D-input but not in Q-output. Its layout is shown in Fig. 3(c), and compared to those of low and high

---

---

**TABLE I**

<table>
<thead>
<tr>
<th>FF</th>
<th>DQ = 00</th>
<th>DQ = 01</th>
<th>DQ = 10</th>
<th>DQ = 11</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>LVT</td>
<td>460</td>
<td>450</td>
<td>469</td>
<td>342</td>
<td>415</td>
</tr>
<tr>
<td>HVT</td>
<td>47</td>
<td>51</td>
<td>50</td>
<td>48</td>
<td>49</td>
</tr>
<tr>
<td>MVT-I</td>
<td>323</td>
<td>373</td>
<td>404</td>
<td>278</td>
<td>345</td>
</tr>
<tr>
<td>MVT-II</td>
<td>80</td>
<td>84</td>
<td>75</td>
<td>73</td>
<td>78</td>
</tr>
</tbody>
</table>

**TABLE II**

<table>
<thead>
<tr>
<th>FF</th>
<th>Rising $T_{su}$</th>
<th>Falling $T_{su}$</th>
<th>Rising $T_{cQ}$</th>
<th>Falling $T_{cQ}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LVT</td>
<td>46.9</td>
<td>48.7</td>
<td>180.7</td>
<td>155.6</td>
</tr>
<tr>
<td>HVT</td>
<td>108.1</td>
<td>74.9</td>
<td>340.3</td>
<td>286.4</td>
</tr>
<tr>
<td>MVT-I</td>
<td>124.0</td>
<td>151.2</td>
<td>180.2</td>
<td>155.4</td>
</tr>
<tr>
<td>MVT-II</td>
<td>46.3</td>
<td>10.1</td>
<td>339.5</td>
<td>286.3</td>
</tr>
</tbody>
</table>

---

\[ V_t \text{ flip-flops in Fig. 3(a) and Fig. 3(b), respectively. The area remains constant, because three inverters in the master stage are localized in the original layout, which in turn guarantees enough space between high } V_t \text{ layer, shown in the figure as a thick rectangle, and adjacent poly lines.} \]

In the second variant of flip-flop (see Fig. 2(d)), we use high \( V_t \) in those transistors that affect clock-to-Q delay (i.e. slave stage), plus an inverter \( g_1 \) in the master stage and the two inverters \( g_3 \) and \( g_4 \), which internally generate \( \text{clk} \) and \( \overline{\text{clk}} \) from clock input (CK). Even though \( \text{clk} \) and \( \overline{\text{clk}} \) are delayed due to the use of high \( V_t \), the setup time is decreased rather than increased as shown in Table II. Fig. 4 shows waveforms that explain rising \( T_{su} \) and rising \( T_{\text{clk}} \) of MVT-II flip-flop. Note that the timing parameters of a flip-flop are measured with respect to its clock input (CK); the D-input can be captured in the master stage only after the late arrival of \( \text{clk} \) and \( \overline{\text{clk}} \), which are internally generated and thus lag behind CK (refer to \( T_1 \) and \( T_1' \) in Fig. 4). Therefore, the D-input is allowed to arrive later, which decreases the setup time. This also allows us to use high \( V_t \) for \( g_1 \) in the master stage, which translates into even decreased leakage current. The falling \( T_{su} \) is even more decreased (refer to Table II), because the falling D-input is captured only after \( \overline{\text{clk}} \) arrives at \( g_2 \), and \( \overline{\text{clk}} \) arrives earlier than \( \text{clk} \). The overall leakage of MVT-II flip-flop is reduced significantly (about 19% of LVT flip-flop on average), while we maintain the layout area as shown in Fig. 3(d).

### III. MIXED \( V_t \) ALLOCATION ALGORITHM

Once we have a netlist of sequential circuit, we need to determine, for each gate, its implementation in terms of \( V_t \).

We will develop an allocation algorithm for sequential circuits, which selects either high or low \( V_t \) for combinational gates and one of four implementations, introduced in the previous section, for flip-flops. The algorithm is based on a concept of “sensitivity” of gate [7], which determines a priority of each gate for allocation. The sensitivity expresses the amount of change in leakage and timing of a gate, if its \( V_t \) were to change. We will briefly overview the concept of sensitivity, extend it to mixed \( V_t \) flip-flops, and outline the overall allocation algorithm.

#### A. Sensitivity

Assume that all the gates in a netlist are initially in high \( V_t \). The sensitivity of a gate \( i \) is defined by [7]:

\[
S_i = \frac{\Delta I}{\Delta D} \quad \text{where} \quad \Delta I = I_{\text{low } V_t} - I_{\text{high } V_t}, \quad \Delta D = \sum_{\forall p \text{ covered by } i \text{ with } s_{ip} < 0 \text{ }} \text{MIN}(\Delta d_{ip}, |s_{ip}|)
\]

The numerator (\( \Delta I \)) is the amount of increase in leakage if low \( V_t \) were to be used for the gate. The denominator (\( \Delta D \)) indicates the amount of improvement in timing (\( \Delta d_{ip} \)). This is taken for all timing paths, which go through the gate \( i \), with negative slack. For those paths, the improvement beyond zero slack is discarded, because it is meaningless as far as the gate \( i \) is concerned.

**Example 1** Consider an example netlist shown in Fig. 5. We try to compute the sensitivity of NOR gate \( g_2 \). The leakage and the delay of \( g_2 \) in both high and low \( V_t \) is shown in the figure. Assume that arrival times at the inputs \((i_1 \text{ and } i_2)\) are 0, and required arrival times at the outputs \((o_1 \text{ and } o_2)\) are 40. For simplicity, we do not distinguish the phase of signals, i.e. rising or falling. The first timing path \((i_1-g_2-g_3-o_1)\) has a positive slack of 5 (40 – 35), so it is not included in \( \Delta D \) computation. The second timing path \((i_1-g_2-g_4-o_2)\) has a negative slack of –5, but the delay of \( g_2 \) can be improved by 8 (25 – 17) by using low \( V_t \) NOR gate. Since the improvement by 5 is enough for this path, we take \(|s_{ip}| = 5\) instead of \( \Delta d_{ip} = 8 \). The third path \((i_2-g_1-g_2-g_3-o_1)\) has zero slack. The fourth path \((i_2-g_1-g_2-g_4-o_2)\) has a negative slack of –10 (50 – 40); this path can be improved by 7 if low \( V_t \) is used for \( g_2 \), which
Fig. 6. An example netlist for sensitivity computation of flip-flops.

is still not enough to make the path positive. Thus, we take \( \Delta d_p = 7 \) as timing improvement. Therefore, the sensitivity of \( g_2 \) is computed as: \( S_2 = (220 - 20)/(5 + 7) = 16.67. \)

Once we compute the sensitivity of all the gates, we take the one with the lowest, and substitute low \( V_i \) gate for it. The sensitivities of the gates that are affected by this allocation are updated. The process continues until there are no negative slacks left in the netlist.

B. Sensitivity Computation of Flip-Flops

In order to extend the sensitivity-based allocation algorithm to sequential circuits, we need to define the sensitivity of a flip-flop. The sensitivity of a flip-flop \( i \) from one implementation to the other is defined by:

\[
S_i = \frac{\Delta I}{\Delta D}, \quad \text{where} \quad \Delta I = I_{\text{new}} - I_{\text{old}}, \quad \Delta D = \begin{cases} 
MIN(\Delta T_{su}, |s_{ip}|) & \text{if } p \text{ covered by } i \text{ with } s_{ip} < 0 \\
MIN(\Delta T_{c-q}, |s_{ip}|) & \text{if } p \text{ covered by } i \text{ with } s_{ip} < 0
\end{cases}
\]

where the initial implementation is one of HVT, MVT-I, or MVT-II; the target implementation is one of MVT-I, MVT-II, or LVT, while the two implementations are different. Note that we have two terms in the computation of timing improvement: one involved in setup time and the other in clock-to-Q delay. We again consider only timing paths with negative slack, which go through the flip-flop \( i \).

Example 2 Consider an example netlist shown in Fig. 6. We compute the sensitivity of a flip-flop \( g_6 \), when we try to use low \( V_i \) instead of the initial high \( V_i \). Assume that the circuit input \( i_1 \) has 0 as its signal arrival time, and the output \( o_1 \) has 70 as its required arrival time. Assume also that the clock cycle time is 70. There are two paths that lead to D-input: \( i_1 \rightarrow g_1 \rightarrow g_3 \rightarrow g_4 \rightarrow D \) and \( g_2 \rightarrow g_3 \rightarrow g_4 \rightarrow D \). Only the second path has a negative slack of \( -30 \) \((70 - 100)\); it can be improved by 5 in the setup time of \( g_6 \), which is then taken as timing improvement. There are two paths that launches at Q-output: \( g_2 \rightarrow g_3 \rightarrow g_4 \rightarrow D \). The first path is negative by \( -30 \) and the second by \( -25 \). Both can be improved by 20 in the clock-to-Q delay of \( g_6 \). Since it is not enough to compensate both negative slacks, we take 20 as timing improvements in both paths. The sensitivity of \( g_6 \) can be computed as: \( S_6 = (1000 - 90)/(5 + 20 + 20) = 20.22. \)

Starting from an HVT flip-flop, we compute three sensitivities: one for LVT flip-flop, and the other two for MVT-I and MVT-II flip-flops. The minimum of these is considered as a sensitivity of the initial HVT flip-flop. Unless the flip-flop is LVT, any allocation is considered intermediate to allow for further chance of optimization.

C. Overall Algorithm

Fig. 7 is a sketch of the overall allocation algorithm. The input to the algorithm is a technology-mapped netlist, which is represented as a graph \( G = (V, E) \), where \( V \) indicates a set of gates and \( E \) corresponds to a set of nets. All the gates are initially assigned high \( V_i \). Static timing analysis is run on the netlist to check if there are any paths with negative slacks (L3 and L4). We then compute the sensitivity of each gate (L5). Note that the sensitivity of combinational gate reflects the potential change from high \( V_i \) to low \( V_i \); the sensitivity of flip-flop corresponds to the minimum of three sensitivities: from HVT to LVT, from HVT to MVT-I, and from HVT to MVT-II. If all the paths have positive slack, then the procedure terminates. Otherwise, we repeat the allocation procedure, until there are no negative paths remained or all the gates are converted to low \( V_i \) (L6).

In the loop of allocation, we first select a gate with minimum sensitivity (L7). If it is a combinational gate, it is removed from a list of candidates (L9). In case of a flip-flop, if the minimum sensitivity is not from conversion to LVT, it has possibility of further allocation (i.e., it could be converted from MVT-I to LVT in later iteration). Thus, in this case, we keep the flip-flop in the candidate list (L8 and L9). We perform timing analysis and sensitivity computation for those gates that are affected by the gate, whose \( V_i \) is changed (L11 and L12).
### IV. EXPERIMENTAL RESULTS

We performed experiments on a set of sequential circuits taken from the ISCAS’89 and ITC’99 benchmarks. Each circuit was synthesized with SIS [11] and mapped into a gate library, which we built based on a commercial 90-nm technology. Technology mapping was done using a weighted sum of area and delay as the cost function, and gate sizing was performed during technology mapping.

In Table III, the second column shows the number of gates in the combinational subcircuit and the third column is the number of flip-flops. In the next three columns respectively, we see the leakage current of the combinational subcircuit, the sequential elements (flip-flops), and the sum of the two; these figures are for dual $V_t$ in the combinational subcircuit. Each circuit was simulated ten times with SPICE using ten different input vectors for the primary inputs (flip-flop states were determined after running the circuits with 100 random input vectors to assume stable states), and we take the average of them.

The seventh, eighth, and ninth columns show the leakage (as factors of the corresponding data under the heading dual $V_t$), when we use the mixed $V_t$ allocation algorithm in the previous section, which was implemented in SIS [11]. The leakage of flip-flops is cut by 57% on average, which is an understandable consequence of the distribution of flip-flops after allocation, as shown in Fig. 8. Note that, for five circuits (s298, s400, s838, b03, and b09), the use of high $V_t$ flip-flops is very limited, implying that straightforward extension of conventional dual $V_t$ allocation algorithm to these circuits will be less efficient. The leakage in the combinational logic remains largely unchanged. For the benchmark b03 and b09, we can see the decrease rather than increase of leakage implying that some paths have increased slack. This is partly due to the reduced falling setup time of MVT-II flip-flops (refer to Table II) and partly due to the heuristic nature of the algorithm. The saving in total leakage is 31% on average, up to 53% for s1423.

In order to test the efficacy of our heuristic algorithm, we also implemented an integer linear programming (ILP) based algorithm and compared the two for a subset of benchmarks, which are sufficiently small for ILP solver to output the optimal results within a few hours. We extended the traditional ILP formulation for dual $V_t$ allocation [9], in order to support mixed $V_t$ flip-flops. Fig. 9 shows the results, in which leakage is normalized against dual $V_t$ applied to combinational subcircuits alone. Surprisingly, the result of the proposed approach, which is based on a simple sensitivity, is very close to optimal.

In Fig. 10, we compare three different implementations of the benchmarks s838 and b12 (all low $V_t$, dual $V_t$ on combinational sub-circuits, which is a conventional approach, and our approach) in terms of cumulative slack (e.g. about 80% of nets have slacks less than 400ps in all low $V_t$ implementation of s838). From all low $V_t$ to dual $V_t$ implementation, we see the clear shift of histogram to the left, which implies increasing number of critical paths. However, the shift from dual $V_t$ (i.e. conventional approach) to our approach is marginal, which is desirable in manufacturing point of view. Note that the histogram that skews too much to the left implies many equally critical paths, which eventually harms the yield due to many uncertainty factors [13] during the design process.

![Fig. 8. Distribution of flip-flops after mixed $V_t$ allocation algorithm.](image)

### Table III

<table>
<thead>
<tr>
<th>Benchmark</th>
<th># Gates</th>
<th># FFs</th>
<th>Comb. ($\times$ nA)</th>
<th>SE (nA)</th>
<th>Total ($\times$ nA)</th>
<th>Mixed $V_t$ flip-flops + Dual $V_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s298</td>
<td>156</td>
<td>14</td>
<td>5</td>
<td>6</td>
<td>11</td>
<td>1.04 × 0.70 × 0.87</td>
</tr>
<tr>
<td>s400</td>
<td>205</td>
<td>21</td>
<td>7</td>
<td>8</td>
<td>15</td>
<td>1.00 × 0.76 × 0.87</td>
</tr>
<tr>
<td>s641</td>
<td>202</td>
<td>19</td>
<td>3</td>
<td>8</td>
<td>13</td>
<td>1.03 × 0.30 × 0.64</td>
</tr>
<tr>
<td>s838</td>
<td>412</td>
<td>32</td>
<td>9</td>
<td>13</td>
<td>22</td>
<td>1.05 × 0.58 × 0.77</td>
</tr>
<tr>
<td>s1423</td>
<td>736</td>
<td>74</td>
<td>19</td>
<td>31</td>
<td>50</td>
<td>1.00 × 0.16 × 0.47</td>
</tr>
<tr>
<td>s5378</td>
<td>1597</td>
<td>163</td>
<td>46</td>
<td>65</td>
<td>111</td>
<td>1.07 × 0.44 × 0.70</td>
</tr>
<tr>
<td>b03</td>
<td>168</td>
<td>30</td>
<td>8</td>
<td>12</td>
<td>20</td>
<td>0.91 × 0.46 × 0.64</td>
</tr>
<tr>
<td>b04</td>
<td>737</td>
<td>66</td>
<td>19</td>
<td>25</td>
<td>44</td>
<td>1.06 × 0.30 × 0.63</td>
</tr>
<tr>
<td>b07</td>
<td>519</td>
<td>49</td>
<td>16</td>
<td>19</td>
<td>35</td>
<td>1.03 × 0.34 × 0.64</td>
</tr>
<tr>
<td>b08</td>
<td>214</td>
<td>21</td>
<td>11</td>
<td>11</td>
<td>19</td>
<td>1.01 × 0.35 × 0.68</td>
</tr>
<tr>
<td>b09</td>
<td>184</td>
<td>28</td>
<td>8</td>
<td>11</td>
<td>19</td>
<td>0.93 × 0.63 × 0.75</td>
</tr>
<tr>
<td>b12</td>
<td>1243</td>
<td>119</td>
<td>36</td>
<td>48</td>
<td>84</td>
<td>1.07 × 0.43 × 0.65</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.01 × 0.43 × 0.69</td>
</tr>
</tbody>
</table>
V. CONCLUSION

Although in widespread use, the value of dual $V_t$ is limited, since it considers only the combinational portion of a circuit, even though the sequential elements contribute a non-negligible, and sometimes a significant, portion of the total leakage. We have proposed mixed $V_t$ flip-flops that exhibit increased delay, but either on setup time or on clock-to-Q delay but not on both, while their leakage is greatly reduced compared to low $V_t$ flip-flop. Moreover, this is achieved without any area overhead due to careful selection of transistors for high $V_t$ implementation. This concept is general and any kind of conventional flip-flop can be transformed to mixed $V_t$ flip-flop. We have presented a heuristic that substitutes mixed $V_t$ flip-flops for conventional flip-flops as well as allocating high or low $V_t$ to each combinational gate. An average saving of 31% of leakage was observed when this approach was compared to the use of dual $V_t$ alone.

ACKNOWLEDGMENT

This work was supported by Samsung Electronics.

REFERENCES


