Mixed-V_{th} (MVT) CMOS Circuit Design Methodology for Low Power Applications *

Liqiong Wei, Zhanping Chen, and Kaushik Roy
School of Electrical and Computer Engineering
Purdue University, W. Lafayette, IN 47907-1285

Yibin Ye and Vivek De
Intel Corp., Hillsboro, OR 97124-6461

Abstract

Dual threshold technique has been proposed to reduce leakage power in low voltage and low power circuits by applying a high threshold voltage to some transistors in non-critical paths, while a low-threshold is used in critical path(s) to maintain the performance. Mixed-V_{th} (MVT) static CMOS design technique allows different thresholds within a logic gate, thereby increasing the number of high threshold transistors compared to the gate-level dual threshold technique. In this paper, a methodology for MVT CMOS circuit design is presented. Different MVT CMOS circuit schemes are considered and three algorithms are proposed for the transistor-level threshold assignment under performance constraints. Results indicate that MVT CMOS design technique can provide about 20% more leakage reduction compared to the corresponding gate-level dual threshold technique.

1 Introduction

The increasing need for low power in portable computing and wireless communication systems is making design communities accept low voltage CMOS processes [1, 2]. With the lowering of supply voltage, the transistor threshold voltage (V_{th}) has to be scaled down to meet the performance requirements. Unfortunately, such scaling increases the subthreshold leakage current, thereby increasing leakage power.

Multiple-V_{th} design technique can be used to deal with the leakage problem in low power and high performance applications. Multi-Threshold-Voltage CMOS (MTCMOS) circuit technology was proposed by inserting high threshold devices in series to normal circuitry [3]. This technique is very effective for the standby leakage power reduction. But the large inserted MOSFETs increases the area and delay. For dual threshold design technique, a high threshold voltage can be assigned to some transistors in non-critical paths so as to reduce leakage current, while the performance is maintained due to the low threshold transistors in the critical path(s). Therefore, both high performance and low power can be achieved simultaneously. This technique has been demonstrated that leakage power can be reduced during both active and standby modes without any delay and area overheads [5]. Recently, a dual-V_{th} MOSFET process was developed [4], making the implementation of dual-V_{th} logic circuits more feasible.

However, due to the complexity of the circuits, not all the transistors in non-critical paths can be assigned a higher threshold voltage. Otherwise, some non-critical paths may become critical. In order to achieve the best leakage savings under performance constraints, algorithms for dual threshold assignment were presented in [5, 7]. But these algorithms only dealt with the circuits at the gate-level — the transistors within a gate were assumed to have the same threshold voltage.

For mix-V_{th} (MVT) CMOS circuits, the transistors within a gate can have different threshold voltages with certain process constraints. Therefore, more transistors can be assigned high-V_{th}, and hence, larger leakage current reduction can be achieved. In this paper, different MVT CMOS circuit schemes are introduced and several algorithms for MVT CMOS circuit design are presented. The efficiency of each algorithm is demonstrated by experiments on a 32-bit adder and some ISCAS benchmark circuits.

The paper is organized as follows. In Section 2, necessary definitions are introduced. Different MVT CMOS circuit schemes are proposed in Section 3. Section 4 describes three algorithms for MVT CMOS circuit design. Section 5 presents the implementation details and experimental results. Finally, conclusions are given in Section 6.

2 Preliminaries

Let us consider Figure 1. The logic gates are clearly marked in circles. Suppose gate G is the one being analyzed. GI_i and GO_j are the fanin and fanout gates of G, where i varies from 1 to the number of fanins (F_I) and j varies from 1 to the number of fanouts (F_O). Each fanin gate GI_i connects to a pair of transistors (p_i, n_i) in gate G for a standard CMOS implementation. Similarly, for each fanout gate, there are a pair of transistors (p_j, n_j) driven by gate G.

2.1 Transistor-level static timing analysis

Transistor-level static timing analysis is used in our algorithms. Each transistor has a propagation delay, which can

* Acknowledgment: This research is supported in part by DARPA (F33615-95-C-1625), NSF CAREER award (9501869-MIP), Semiconductor Research corporation (98-HJ-638), and Intel.

DAC 99, New Orleans, Louisiana

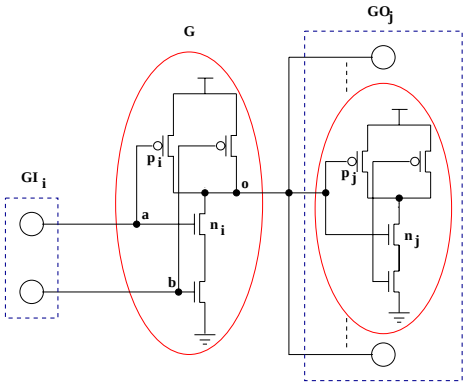

Figure 1: An example circuit schematic

be expressed by

$$t_d = t_{intrinsic} + t_{output}\ C_L \quad (1)$$

where $t_{intrinsic}$ and t_{output} are the intrinsic delay and the delay per unit load, respectively. They can be extracted from SPICE simulations [6]. C_L is the load capacitance, which is the sum of fanout gate capacitances. Increasing the threshold voltage will increase t_d. The difference between high-V_{th} delay and low-V_{th} delay is represented by Δt_d.

For the primary inputs and primary outputs, there are timing constraints. Each primary input (PI) has an arrival time. For the primary output (PO), there is a required time.

For each gate G, the arrival time at the input of G is the time when the signal propagates from the primary input to the input of G. The departure time of G is the sum of the arrival time and the delay of the corresponding transistor. Obviously, the arrival time is determined by the departure time of the corresponding fanin gate. There are two kinds of departure time. One is for the high-to-low transition at the output, denoted by $T_{lf}(G)$. The other corresponds to the low-to-high transition at the output, represented by $T_{lr}(G)$. T_{lr} and T_{lf} are determined by the p pull-up tree and the n pull-down tree of G, respectively. For standard CMOS circuits, they can be expressed by the following equations,

$$T_{lr}(G) = \max_i\{T_{lf}(GI_i) + t_d(p_i)\} \quad (2)$$

$$T_{lf}(G) = \max_i\{T_{lr}(GI_i) + t_d(n_i)\} \quad (3)$$

where i varies for all the fanins. If GI_i is a primary input, its delay is 0. Hence, the departure time equals to the arrival time.

2.2 Transistor delay slack

The slack is the amount by which a gate or a transistor can be slowed down without affecting the circuit performance. For a logic gate, the slacks of the p pull-up tree and the n pull-down tree are represented by S_p and S_n, respectively.

For a primary output PO, the slack is determined by the difference between the required time and the departure time of its fan-in gate. For any other gate G, S_p and S_n can be expressed by

$$S_p(G) = \min_j\{S_n(GO_j) + T_{lf}(GO_j) - T_{lr}(G) - t_d(n_j)\} \quad (4)$$

$$S_n(G) = \min_j\{S_p(GO_j) + T_{lr}(GO_j) - T_{lf}(G) - t_d(p_j)\} \quad (5)$$

where j varies for all the fanout of G. $S_n(GO_j)$ and $S_p(GO_j)$ are the slacks of the pull-down tree and pull-up tree for the fanout gate GO_j. $T_{lf}(GO_j) - T_{lr}(G) - t_d(n_j)$ and $T_{lr}(GO_j) - T_{lf}(G) - t_d(p_j)$ are the amounts by which the p pull-up tree and pull-down tree of gate G can be slowed down without affecting the departure time of fanout gate GO_j, respectively. $S_p(G)$ and $S_n(G)$ are taken as the minimum value over all the fanout gates so as to maintain the performance.

For each transistor pair (p_i, n_i) in gate G, their slacks can be represented as $s(p_i)$ and $s(n_i)$,

$$\begin{aligned} s(p_i) &= S_p(G) + (T_{lr}(G) - T_{lf}(GI_i) - t_d(p_i)) \\ &= \min_j s(n_j) + (T_{lr}(G) - T_{lf}(GI_i) - t_d(p_i)) \quad (6) \\ s(n_i) &= S_n(G) + (T_{lf}(G) - T_{lr}(GI_i) - t_d(n_i)) \\ &= \min_j s(p_j) + (T_{lf}(G) - T_{lr}(GI_i) - t_d(n_i)) \quad (7) \end{aligned}$$

where $S_p(G)$ ($S_n(G)$) is the slack of the p pull-up tree (n pull-down tree) of G, which is the minimal transistor slack over all the fanout NMOSFETs (PMOSFETs). The terms $(T_{lr}(G) - T_{lf}(GI_i) - t_d(p_i))$ and $(T_{lf}(G) - T_{lr}(GI_i) - t_d(n_i))$ are the amounts by which transistor p_i and n_i can be slowed down without affecting the departure time of gate G, respectively. If the threshold voltage is increased from a low-V_{th} to a high-V_{th}, t_d will increase by Δt_d, and therefore, the transistor delay slack will reduce by Δt_d. As long as the slack value is no less than 0, which means Δt_d is no larger than the slack value, the circuit performance is not degraded.

2.3 Transistor priority

From BSIM MOS transistor model [8], the subthreshold leakage current of a MOSFET can be modeled as

$$I_{sub} = \mu_0 C_{ox} \frac{W_{eff}}{L_{eff}} (\frac{kT}{q})^2 e^{1.8} e^{\frac{q}{n'kT}(V_{GS} - V_{th})} (1 - e^{\frac{-qV_{DS}}{kT}}) \quad (8)$$

where C_{ox} is the gate oxide capacitance per unit area. W_{eff} and L_{eff} are the effective channel width and the effective channel length, respectively. μ_0 is the zero bias mobility. n' is the subthreshold swing coefficient of the transistor.

For a low-V_{th} transistor, if its threshold voltage is increased to a high-V_{th} value, the leakage reduction is proportional to the effective channel width and the mobility. Therefore, we define the leakage reduction measure for transistor i as follows,

$$\Delta leak_i = W_{eff_i}\ \gamma_i \quad (9)$$

where γ is the normalized mobility, which is equal to $\frac{\mu_p}{\mu_n}$ and 1 for PMOS transistors and NMOS transistors, respectively. μ_p and μ_n are the hole mobility and electron mobility, respectively.

The leakage reduction measure of a dual-V_{th} circuit, denoted by $\mathcal{M}_{leak}$, is defined as

$$\mathcal{M}_{leak} = \sum_i \Delta leak_i \quad (10)$$

where the summation is taken over all the high-V_{th} transistors in the dual-V_{th} circuit. The larger the value of $\mathcal{M}_{leak}$, the more leakage reduction can be achieved for a dual-V_{th} circuit, compared to the corresponding single low threshold circuit.

For each transistor, larger $\Delta leak$ is preferable for larger leakage reduction. Consider the high-V_{th} delay and low-V_{th} delay difference (Δt_d). If it is small, a large number of

transistors can be assigned the high threshold under performance constraints, thereby leading to more savings in leakage power. In our analysis, we define the priority of transistor i as follows,

$$priority(i) = \frac{\Delta leak_i}{\Delta t_{d_i}} \tag{11}$$

Clearly, a transistor with a larger priority will result in more leakage reduction.

3 Mixed-V_{th} (MVT) CMOS Circuit Schemes

In this section, different mixed-V_{th} circuit topologies are presented. Let us first consider Figure 2 which illustrates a small part of a single-V_{th} circuit. Suppose that the transistors in the squares are the transistors in the critical paths, and hence, can have low-V_{th}. For the other transistors, a high-V_{th} can be assigned without degrading the performance.

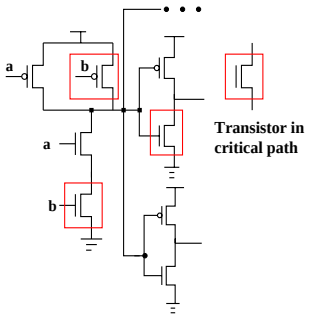

Figure 2: single-V_{th} circuit

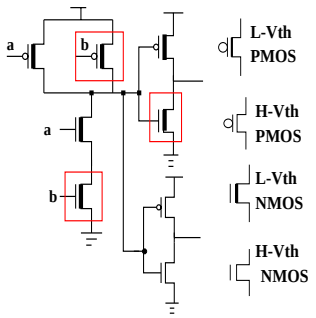

Figure 3: DVT scheme

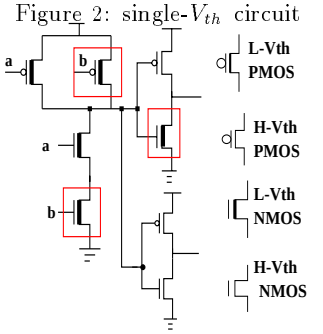

Figure 4: MVT1 scheme

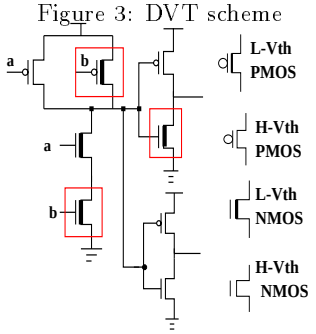

Figure 5: MVT2 scheme

Consider the gate-level dual-V_{th} (DVT) circuit. All the transistors within a gate have the same threshold voltage. The gates are either all high-V_{th} or all low-V_{th} gates. Figure 3 shows the gate-level dual-V_{th} scheme of the example circuit shown in Figure 2.

For mixed-V_{th} CMOS circuit, the transistors within a gate can have different threshold voltages with certain process constraints. There are two types of mixed-V_{th} CMOS circuit schemes that we consider. For type I scheme (MVT1), there is no mixed V_{th} in p pull-up or n pull-down trees. Figure 4 shows the example circuit in MVT1 scheme. For type II scheme (MVT2), mixed V_{th} is allowed anywhere except for the series connected transistors. The example circuit in MVT2 scheme is illustrated in Figure 5. The reason that transistors in a stack have the same threshold voltage is because of the process consideration. Suppose the transistor thresholds are controlled by channel doping. For the transistors in a stack, their channels are too close to each other, making it difficult to achieve distinct channel doping. Therefore, it is hard to get different thresholds for the transistors in a stack.

Obviously, MVT CMOS shows more opportunities for the high V_{th} assignment than the gate-level dual threshold circuit.

4 Algorithms for MVT Static CMOS Circuit Design

In this section, we will show how to develop a mixed-V_{th} (MVT) static CMOS circuit under performance constraints. Let us assume that the timing constraints for the primary inputs and primary outputs are given. There are two threshold voltages. The high-V_{th} is represented by V_{tH} and the low-V_{th} as V_{tL}. In order to achieve an optimal mixed-V_{th} static CMOS circuit, three transistor-level algorithms for the assignment of a high threshold to a single low V_{th} static CMOS circuit are proposed. The first one is an extension of the gate-level levelization-based back-tracing algorithm [5], where the transistors are traversed level by level from primary outputs. The second one is a priority selection algorithm, where the transistors are visited according to the priority values. The third one is priority-based back-tracing algorithm, which is the combination of the first two algorithms.

4.1 Back-tracing (BT) Algorithm

The first step in this algorithm is to levelize a circuit. The level of a primary input is defined to be 0. The level of a gate G, denoted by $l(G)$, can be calculated by

$$l(G) = 1 + \max_i l(GI_i) \tag{12}$$

where i varies for all the fanin of G and GI_i is the ith fanin of gate G. By determining $t_{intrinsic}$ and t_{output} values corresponding to V_{tH} and V_{tL} based on HSPICE simulations, the propagation delay of each transistor at V_{tH} and V_{tL} can be evaluated using equation 1, and the corresponding delay difference (Δt_d) can be easily calculated.

The next step is to assign dual threshold voltages to the transistors under performance constraints. All the transistors in the circuit are initially assumed to have the low threshold voltage. We forward-trace the circuit level by level from primary inputs to calculate the departure time of each gate using equations (2) and (3). Next, back-trace the circuit level by level from primary outputs to explore every gate G. The pull-up tree slack($S_p(G)$), pull-down tree slack ($S_n(G)$), and the slack of each transistor within G can be calculated by using the equations (4)-(7). For the gate-level dual-V_{th} (DVT) scheme, if Δt_d of all the transistors within G are no larger than their slack values, G is a high-V_{th} gate. For the mixed-V_{th} type I scheme (MVT1), if all the transistors in the pull-up (pull-down) tree of gate G satisfy the requirement that Δt_d are no larger than their slack values, the pull-up (pull-down) tree can be assigned V_{tH}. Let us consider the mixed V_{th} type II scheme (MVT2). For each transistor of gate G, if it is not a series connected transistor and its Δt_d is no larger than its slack value, this transistor can be assigned the high-V_{th}. For the series connected transistors, if the Δt_d of all the transistors in a series are no larger than their slack values, V_{tH} is assigned to all the transistors in the series. Otherwise, V_{tL} is maintained. After the threshold voltage assignment for each transistor within gate G, the propagation delay of each transistor within G is updated. Then the departure time of gate G, $S_p(G)$, and

$S_n(G)$, are recalculated. The pseudo-code of this procedure is shown below.

```
Backtracing algorithm () {
  Levelize the circuit
  Evaluate t_d of each transistor for V_tH and V_tL
  Calculate Δt_d of each transistor
  Assign V_tL to all the transistors
  Forward trace the circuit level by level
    Calculate the departure time of each gate
  Back-trace the circuit level by level to visit each gate G {
    Calculate S_p(G) and S_n(G)
    For each transistor (tr) within G
      Calculate s(tr)
    If DVT is selected {
      If all the transistors in G satisfy Δt_d ≤ slack
        G will be assigned V_tH
    }
    If MVT1 is selected {
      If all the transistors in pull-up tree of G satisfy Δt_d ≤ slack
        Pull-up tree of G will be assigned V_tH
      If all the transistors in pull-down tree satisfy Δt_d ≤ slack
        Pull-down tree of G will be assigned V_tH
    }
    If MVT2 is selected {
      For each transistor (tr) within G {
        If tr is not a series connected transistor {
          If Δt_d(tr) ≤ s(tr)
            tr can be assigned V_tH
        }
        Else if tr is in a series and not visited {
          If all the transistors in the series satisfy Δt_d ≤ slack
            All the transistors in the series are assigned V_tH
          Mark all the transistors in the series visited
        }
      }
    }
    Update the departure time of G,S_p(G) and S_n(G)
  }
}
```

For the backtracing (BT) algorithm, since each transistor is just visited once, the worst case run-time is $O(n)$, where n is the total number of transistors.

4.2 Priority Selection (PS) Algorithm

Priority selection algorithm is an exhaustive priority-based algorithm. The transistors are visited according to the priority values. After each visit, the transistor slacks are recalculated. The pseudo code of the priority selection algorithm for mixed-V_{th} type II scheme (MVT2) is outlined below.

```
Priority selection algorithm () {
  Levelize the circuit
  Calculate delay and priority of each transistor
  All the transistors are assigned V_tL and marked unvisited
  If the unvisited transistor number is not 0 {
    Forward-trace the circuit level by level
      Calculate the departure time of each gate
    Back-trace the circuit level by level
      Calculate the slack of each transistor
    Find max priority transistor (tr) from unvisited transistors
    If tr is not a series connected transistor {
      If Δt_d(tr) ≤ s(tr)
        tr can be assigned V_tH
      Mark tr as a visited transistor
    }
    Else if tr is in a series {
      If all the transistors in this series satisfy Δt_d ≤ slack
        All the transistors in this series can be assigned V_tH
      Mark all the transistors in this series as visited transistors
    }
  }
}
```

The first step is to levelize the circuit and calculate the delay and priority of each transistor. All the transistors are assumed to have V_{tL} and marked unvisited. The second step is to explore all the transistors in the circuit according to the transistor priority values. For each visit, the departure time of each gate can be evaluated by forward tracing the circuit level by level from primary inputs, and the slack of each transistor can be calculated by backtracing the circuit level by level from primary outputs. The transistor with the maximal priority from the unvisited transistors is then selected. By comparing the Δt_d of the transistor being visited with its slack value and considering the series connected transistors, the threshold voltage of this transistor can be determined. In order to avoid repeating assignment, this transistor is marked as a visited transistor.

For the priority selection (PS) algorithm, the circuit needs to be updated to re-calculate the transistor slack values after each transistor is visited. Therefore, the worst case run-time is $O(n^2)$.

4.3 Priority-Based Backtracing (PB) Algorithm

Priority-based backtracing algorithm combines the backtracing algorithm and the priority selection algorithm. During the initialization, the circuit is levelized; the delay and priority of all the transistors are calculated. Then, the transistors are put into m groups according to their priority values, where group 1 corresponds to the maximal priority group and group m is the group with the minimal priority. Next, from group 1 to m, backtracing is performed m times. During each backtracing, only the transistors in the selected group are considered. If $m = 1$, this algorithm is equivalent to the backtracing algorithm. If $m = n$, this is exactly the priority selection algorithm. The pseudo-code of the priority-based backtracing algorithm for mixed-V_{th} type II scheme (MVT2) is shown below.

```
Priority-based backtracing algorithm () {
  Levelize the circuit
  Calculate delay and priority of each transistor
  All the transistors are assigned V_tL and marked unvisited
  Transistors are divided into m groups based on priority values
  For group i from 1 to m {
   Forward trace the circuit level by level
    Calculate the departure time of each gate
   Back-trace the circuit level by level to visit each gate G {
    Calculate the slack of each transistor within G
    For each transistor (tr) within G{
      If tr is in group i{
        If tr is not a series connected transistor {
          If Δt_d(tr) ≤ s(tr)
            tr can be assigned V_tH
        }
        Else if tr is in a series and not visited before {
          If all the transistors in this series satisfy Δt_d ≤ slack
            All the transistors in this series are assigned V_tH
          Mark all the transistors in this series visited
        }
      }
    }
    Update the departure time of G,S_p(G) and S_n(G)
    }
  }
}
```

For priority-based backtracing (PB) algorithm, the transistors are divided into m groups. After each group is visited, the circuit is updated to re-calculate the transistor slacks. Hence, the worst case run-time is $O(mn)$.

5 Implementation and Results

The three algorithms described in Section 4 have been implemented in C under the Berkeley SIS environment. In this section, the results for a number of combinational circuits are presented. In our analysis, the threshold voltage and supply voltage of the original single low-V_{th} circuits are assumed to be around $0.2V$ and $1V$, respectively. The primary inputs are assumed to arrive simultaneously and the

timing constraints for primary outputs are determined by the critical path delay of the single low-V_{th} circuit.

5.1 Results for a 32-bit Adder

A well designed 32-bit static CMOS Kogg-Stone adder was investigated based on PathMill static timing analysis. The normalized active leakage power and standby leakage power at different V_{tH} are given in Figure 6 and Figure 7, respectively. The circuit temperature is assumed to be 110^oC and 25^oC for active mode and standby mode, respectively. Results show that there is an optimal V_{tH}, at which mixed-V_{th} design technique can provide nearly 20% more leakage power savings than the corresponding gate-level dual threshold technique. Suppose the group number (m) is 10 for the priority-based backtracing (PB) algorithm. For a HP workstation, the run-time of backtracing (BT) algorithm, priority-based backtracing (PB) algorithm, and priority selection algorithm are 3.8s, 4s, and 18s, respectively. Results indicate that the PB algorithm gives almost the same leakage savings as the PS algorithm, but the run-time is close to that of the BT algorithm.

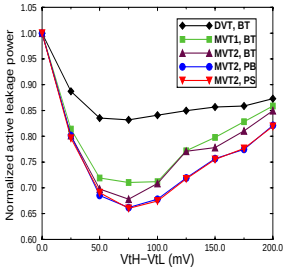

Figure 6: Active leakage

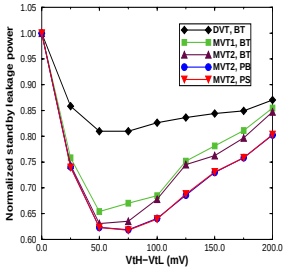

Figure 7: Standby leakage

Figure 8 gives the normalized total power of the mixed-V_{th} 32-bit adder at different V_{tH} and different primary input activities. The total power can be reduced by about 9% and 22% at max activity and 0.1max activity, respectively.

Figure 9 shows the path distributions of the 32-bit adder at single high-V_{th}, single low-V_{th}, and mixed dual-V_{th} conditions. Certainly, single high-V_{th} circuit has less leakage power, but the critical delay of single high-V_{th} circuit is 30% larger than that of single low-V_{th} circuit. Dual-V_{th} circuit has the same critical delay as the single low-V_{th} circuit. However, the delay values of the non-critical paths are increased by assigning the high threshold voltage to some transistors in non-critical paths.

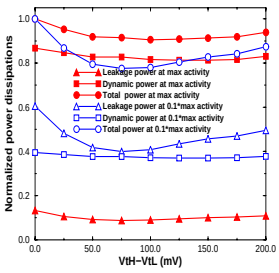

Figure 8: Total power

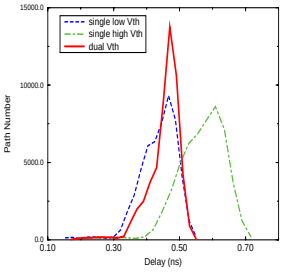

Figure 9: Path distribution

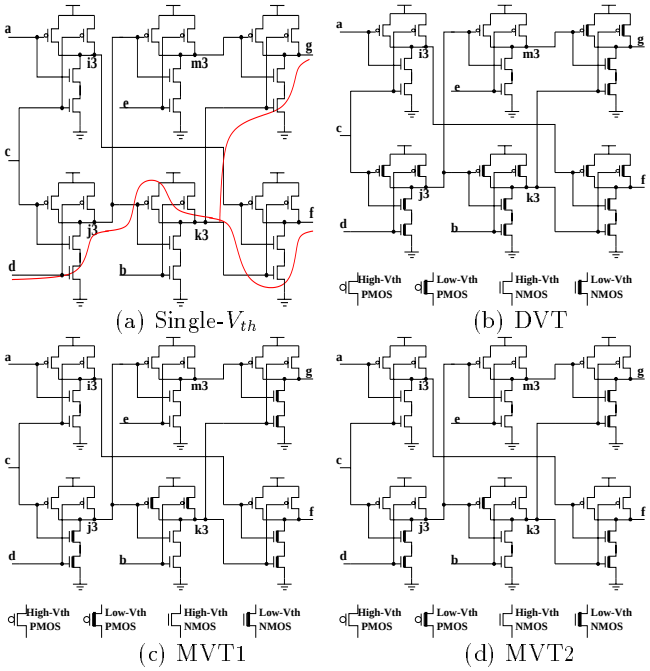

Figure 10: Benchmark C17 in different schemes

5.2 Results for ISCAS Benchmark Circuits

For the ISCAS benchmark circuits, technology-mapping was used to map the circuits to a library which contains NAND, NOR and INVERTER gates. Each type of gate has three different width implementations. In our analysis, V_{tL}, and V_{tH} are assumed to be $0.2V$ and $0.3V$, respectively. The supply voltage is 1V. The effective channel length is $0.32\mu m$ and the gate oxide thickness is $9.8nm$. The circuit temperature is assumed to be 110^oC. A delay look-up table based on HSPICE simulations and a leakage estimation technique which accurately models series connected transistors [9] have been used in our analysis.

Figure 10 illustrates the schematic of circuit C17(from the ISCAS benchmarks) in different schemes. Figure 10 (a) is the single low-V_{th} scheme. The two critical paths of C17 are identified. There are 12 NMOS transistors and 12 PMOS transistors. The effective channel width for each PMOS and NMOS transistor is $3\mu m$ and $1\mu m$, respectively. The leakage power dissipation for the single low-V_{th} circuit of C17 is $1.0\mu W$. The schematic of circuit C17 in gate-level dual-V_{th} (DVT) scheme, mixed-V_{th} type I scheme (MVT1), and mixed-V_{th} type II scheme (MVT2) are given in Figures 10 (b), (c), and (d), respectively. The numbers of high V_{th} PMOS transistors are 4, 10, and 11, while the numbers of high V_{th} NMOS transistors are 4, 6, and 6 for DVT, MVT1, and MVT2 schemes, respectively. The leakage power dissipations for C17 in DVT, MVT1 and MVT2 schemes are $0.726\mu W$, $0.366\mu W$, and $0.32\mu W$, respectively. Hence, the leakage savings for C17 in DVT, MVT1 and MVT2 schemes are 27.6%, 63.5%, and 68.1%, respectively, compared to the single low threshold scheme.

By using SIS command "map", circuits are mapped to a library targeting the minimal area, where the gates with the minimal width are preferred. Technology mapping can also be achieved using SIS command "map -n 1 -AFG" to achieve minimal delay. In the critical path, the gates with larger width are chosen, while the gates in the non-critical paths

Table 1: leakage power savings for ISCAS benchmark circuits mapped for area

Circuit Chosen	PI/PO #	FET #	DVT red.(%)	MVT1 red.(%)	MVT2 red.(%)
C432	36/7	836	63.4	81	82.5
C499	41/32	1892	53.4	58.7	61.4
C880	60/26	1398	81.5	83.3	85.1
C1355	41/32	070	55.8	62.5	66.4
C1908	33/25	2464	70.3	71.6	74.9
C2670	233/140	3360	80	82.9	84.4
C3540	50/22	4797	81.7	83.6	85
C5315	178/123	7708	79.7	80.9	82.5
C6288	32/32	9504	53	53	61.7
C7552	207/108	10846	79	80	81.9

Table 2: leakage power savings for ISCAS benchmark circuits mapped for delay

Circuit Chosen	PI/PO #	FET #	DVT red.(%)	MVT1 red.(%)	MVT2 red.(%)
C432	36/7	1056	37.5	52.6	59.2
C499	41/32	2136	22.5	36	41.6
C880	60/26	1546	65.9	71.3	74.9
C1355	41/32	2724	37.3	48.1	52.4
C1908	33/25	2986	40.8	47	53.5
C2670	233/140	3930	83.5	85.5	86.1
C3540	50/22	5440	55.7	63.9	68.9
C5315	178/123	9000	68.8	71.6	75.1
C6288	32/32	10630	20	24.3	37.4
C7552	207/108	12084	51.9	59.8	64.4

may have smaller width. Obviously, the circuit mapped for delay is more balanced than the circuit mapped for area.

Table 1 and Table 2 report the leakage power savings for ISCAS benchmark circuits which are mapped for area and delay, respectively. The backtracing algorithm is used and different circuit schemes, such as DVT, MVT1, and MVT2, are compared. More leakage reduction can be achieved for the circuits mapped for area because of the larger imbalance in slack. The leakage savings of MVT2 scheme are larger than those of MVT1 scheme. The mixed-V_{th} schemes provide more leakage savings than the corresponding gate-level dual threshold technique. For some benchmark circuits, the additional leakage savings can be more than 20%.

Table 3 shows the leakage power savings for different algorithms, such as backtracing (BT) algorithm, priority selection (PS) algorithm, and priority-based backtracing (PB) algorithm. MVT2 scheme is used and the circuits are mapped targeting the minimal delay. The CPU time is for a SUN UltraSPARC-II. Results indicate that PS algorithm shows more leakage savings, but also takes more CPU time. BT algorithm is the fastest one, but it gives less leakage saving than the other two algorithms. For PB algorithm, the group number (m) is set to be 10. The leakage savings are close to those of PS algorithm and the run-time is similar to that of BT algorithm.

6 Summary & Conclusion

In this paper, a mixed-V_{th} CMOS circuit design technique is presented and different mixed-V_{th} circuit techniques are introduced. Several algorithms for transistor level threshold assignment for mixed-V_{th} static CMOS circuit design style are proposed. A 32-bit adder was simulated based

Table 3: leakage power savings for different algorithms

Circuit Chosen	BT alg.		PS alg.		PB alg. (m=10)	
	%	CPU(s)	%	CPU(s)	%	CPU(s)
C432	59.2	0.03	62.9	2.83	62.7	0.08
C499	41.6	0.06	48.3	10.6	46.5	0.15
C880	74.9	0.04	81	7.0	81	0.10
C1355	52.4	0.07	60.8	22.4	58.6	0.20
C1908	53.5	0.10	63.3	26.2	62.5	0.21
C2670	86.1	0.11	86.5	55.4	86.4	0.28
C3540	68.9	0.14	76	95.94	75.1	0.41
C5315	75.1	0.24	84.5	302.0	83.4	0.70
C6288	37.4	0.50	56.1	337.47	53.1	1.03
C7552	64.4	0.40	75.6	591.66	74.2	1.08

on PathMill timing analysis. For ISCAS Benchmark circuits, a delay look-up table based on HSPICE simulations and a leakage estimation technique which accurately models transistor stacks have been used. Results indicate that the mixed-V_{th} CMOS design technique provides about 20% more leakage savings than the corresponding gate-level dual threshold technique.

Acknowledgment

The authors would like to thank V. Govindarajulu for the contribution in PathMill simulations.

References

[1] J. M. C. Stork, "Technology Leverage for Ultra-Low Power Information Systems", *Proceedings of the IEEE*, Vol.83, No.4, pp. 607-618, 1995.

[2] A. P. Chandrakasan, S. Sheng and R. W. Brodersen, "Low-Power CMOS Digital Design", *IEEE Journal of Solid-State Circuits*, Vol.27, No.4, pp.473, 1992.

[3] S. Mutoh, et al., "1-V Power Supply High-Speed Digital Circuit Technology with Multithreshold-Voltage CMOS", *IEEE Journal of Solid-State Circuits*, Vol.30, No.8, pp. 847-854, 1995.

[4] Z. Chen, C. Diaz, J. Plummer, M. Cao and W. Greene, "0.18um Dual Vt MOSFET Process and Energy-Delay Measurement", *IEDM Digest*, pp. 851, 1996.

[5] L. Wei, Z. Chen, K. Roy, M.C. Johnson, Y. Ye and V. De, "Design and Optimization of Dual Threshold Circuits for Low Voltage Low Power Applications", *IEEE Transactions on VLSI Systems*, Vol.7, No. 1, pp. 16-24, 1999

[6] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: a system perspective*, Addison-Wesley Publishing Company, pp. 221-223, 1992

[7] Q. Wang and S. Vrudhula, "Static Power Optimization of Deep Submicron CMOS Circuits for Dual Vt Technology", *International Conference on Computer-Aided Design*, pp. 490-494, 1998.

[8] B.J. Sheu, D.L. Scharfetter, P.K. Ko, and M.C. Teng, "BSIM: Berkeley Short-Channel IGFET Model for MOS Transistors", *IEEE J. Solid-State Circuits*, SC-22, No.4, pp. 558-566, 1987.

[9] M. Johnson, D. Somasekhar, and K. Roy, "Deterministic Estimation of Minimum and Maximum Leakage Conditions in CMOS Logic," *IEEE Transactions on Computer-Aided Design of IC's*, accepted for publication.