Low Power Gate-level Design with Mixed-V_{th} (MVT) Techniques

ABSTRACT

The reduction of leakage power has become an important issue for high performance designs. One way to achieve low-leakage and high performance designs is the use of multi-threshold techniques. In this paper, a new mixed-V_{th} (MVT) CMOS design technique is proposed, which uses different threshold voltages within a logic gate. This new technique allows the reduction of leakage power, while the performance stays constant. A set of algorithms is given assigning optimal distribution of gates. Results indicate that the new MVT approach can provide up to 40% leakage reduction by constant performance compared to dual-V_{th} (DVT) gate-level techniques.

1. INTRODUCTION

Off-state leakage current is a major technical problem for the semiconductor industry and a limiting factor in future microprocessor integration. In technologies below 90nm the percentage of leakage power is over 50% of total power, at upward tendency [Kim03]. The leakage is divided in two major parts, the sub-threshold leakage and the gate-oxide leakage. The sub-threshold leakage is caused by short channel effects and low threshold voltage, while the gate-oxide leakage is exponentially increasing with decreasing oxide thickness T_{ox}.

In each new technology the supply voltage decreases. This requires the threshold voltage being scaled down to meet the performance requirements. Unfortunately, sub-threshold leakage currents increase exponentially with decreasing threshold voltage. Multi-threshold CMOS (MTCMOS) techniques are an approach to deal with these conditions. MTCMOS techniques provide transistors with different threshold voltages [Kao00]. Transistors with a high-threshold voltage (HVT) are placed in non-critical paths, while low-threshold voltage transistors (LVT) are used in critical paths.

The approaches presented in [Sun99] and [Wei00b] detect the critical path and calculate a time slack for each gate. If the time slack is long enough, gates are changed to HVT-gates, which consist of high-threshold voltage transistors. A solution at transistor level is presented in [Wei99].

After timing analysis and evaluation of the transistor time slacks, every transistor obtains a priority, which depends on delay and leakage. The transistors with the highest priority are checked first, and if the slack is long enough they will be replaced by high-threshold voltage transistors. A solution for Multi-Threshold Differential Cascode Voltage Switch (MT-DCVS) is presented in [Che01].

This paper is organized as follows. In section 2 necessary definitions and information are introduced. Our approach is proposed in section 3. Section 4 the algorithm for MVT CMOS design are described. In Section 5 simulation results are presented and section 6 concludes this paper.

2. PRELIMINARIES

Multi-Threshold CMOS circuits

The performance of CMOS circuits depends on the supply voltage V_{DD}, the load C_L and the threshold voltage V_{th}:

$$t_{pd} \propto \frac{C_L V_{DD}}{(V_{DD} - V_{th})^{\alpha}} \tag{1}$$

where t_{pd} is the gate propagation delay, and α models short-channel effects [Sak90]. If the threshold voltage decreases, t_{pd} decreases too and the performance increases. Additionally, the leakage $P_{leakage}$ of a gate depends on V_{th}, too:

$$P_{leakage} \propto 10^{\frac{V_{th}}{S}} \tag{2}$$

where S is the subthreshold slope. If V_{th} decreases, the leakage increases exponentially. Resulting from (1) and (2), devices with high threshold voltage have a low performance and a low leakage, while devices with low threshold voltage have a high performance and a high leakage.

The basic concept of multi-threshold CMOS circuits is the use of high-V_{th} and low-V_{th} devices in a block flexibly [Kao00]. The concept relies on the observation that a circuits overall performance is often limited by a few critical paths. The devices along these critical paths are set to low-V_{th}, whereas the other devices are set to high-V_{th}. The performance can be improved significantly, while leakage current is kept within bounds.

Fabrication of Multi-threshold CMOS Transistors

Transistors with different threshold voltages can be achieved in several ways. The most commonly used technique is the adjustment of thresholds by ion implementation [Wei00a]. Dual thresholds require two additional masks. However, the low and high thresholds should be so far apart from each other to be distinguishable under process variation.

Another technique is the changing of body or back gate voltage [Ani03]. But for bulk silicon transistors triple well technology is required, because transistors cannot share the same well. The most interesting but also most complicated technique is the depositing of different gate-oxide thickness T_{ox} [Sir04]. Using larger gate-oxide thickness gives a transistor a higher threshold voltage. Further, a high T_{ox} increases not only threshold voltage. It reduces the gate leakage and the gate capacitance, too. Hence, dynamic und leakage power is reduced. figure 1 depicts the ratio of threshold voltage, doping concentration and oxide thickness [Sir04].

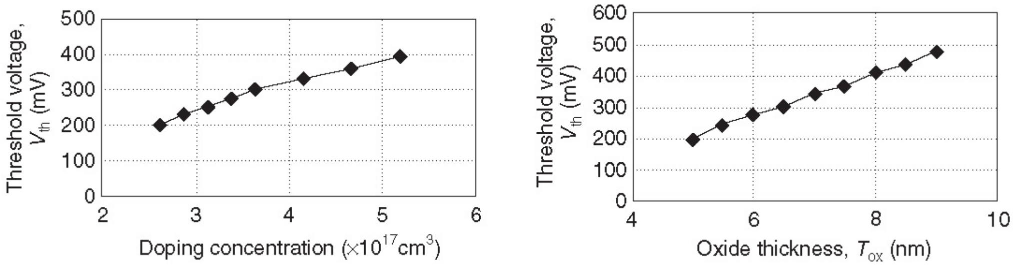


figure 1 V_{th} at different channel doping densities and oxide thicknesses [Sir04]

Gate Times

To describe the behavior of the gates in a design, we defined gate times. An example for the times of a NAND2 is depicted in figure 2. The *gate input time* $T_{in}(G_{in})$ describes the latest time when the input signal arrives at gate input G_{in}. $T_{in,max}(G)$ is the time when all input signals of gate G arrived. In figure 2, $T_{in,max}(G)$ results from the arrival time of input B. The *evaluation end time* $T_{eval}(G)$ of a gate describes the latest time, when the evaluation of the gate ends. $T_{eval}(G)$ results from the sum of $T_{in,max}(G)$ and the worst case delay of the gate $t_d(G)$:

$$T_{eval}(G) = T_{in,max}(G) + t_d(G) \tag{3}$$

The *gate delay slack* $S(G)$ is the amount by which a gate can be slowed down without affecting the design performance. A slack $S(G)$ of a gate is the difference of the time when the evaluation must be ready $T_{eval,max}(G)$, $t_d(G)$ and $T_{in,max}(G)$:

$$S(G) = T_{eval.max} - (t_d + T_{in,max}) \qquad (4)$$

$T_{eval,max}(G)$ results from all $T_{in,max}(G)$ of the gates, which are connected with the output signal of gate *G*. In the example, the slack of the NAND2 is determined by the arrival time of signal C, which is the input signal of the following AND2.

3. MIXED-V_{th} (MVT) CMOS CIRCUITS

Idea

The ambition of MVT-Gates is the reduction of leakage within a gate, without varying the performance. This will be achieved by replacing normal-V_{th} transistors with high-V_{th} and low-V_{th} transistors. Every optimization of a gate should not increase the worst case delay t_d. But a gate has different paths and a critical path, also. Each input vector generates an own evaluation delay. This affords, that transistors in noncritical paths can slow down. Usually, this is done by decreasing gate width *W*. The result is a reduction of P_{dyn}. But the requirements changed with increasing leakage. Actually, a design has to operate fast, if it is active. But in idle modes, the design should have low leakage.

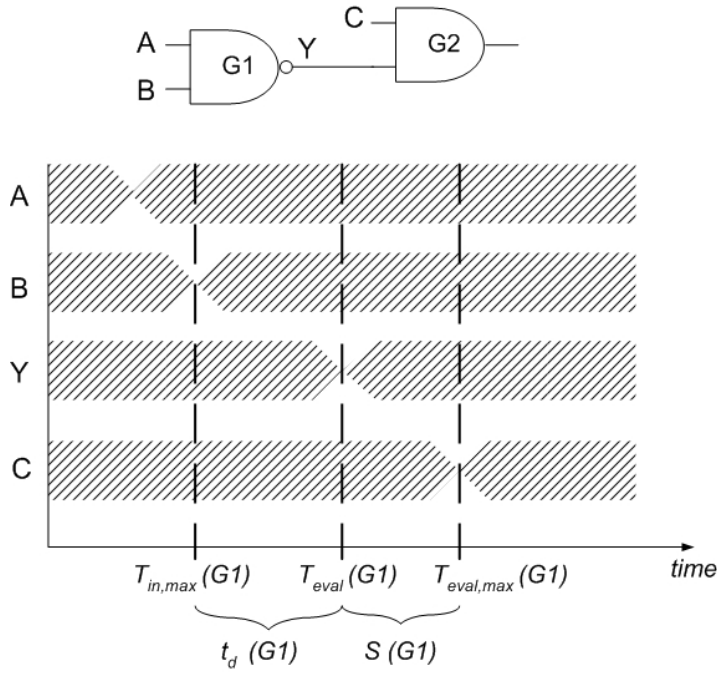
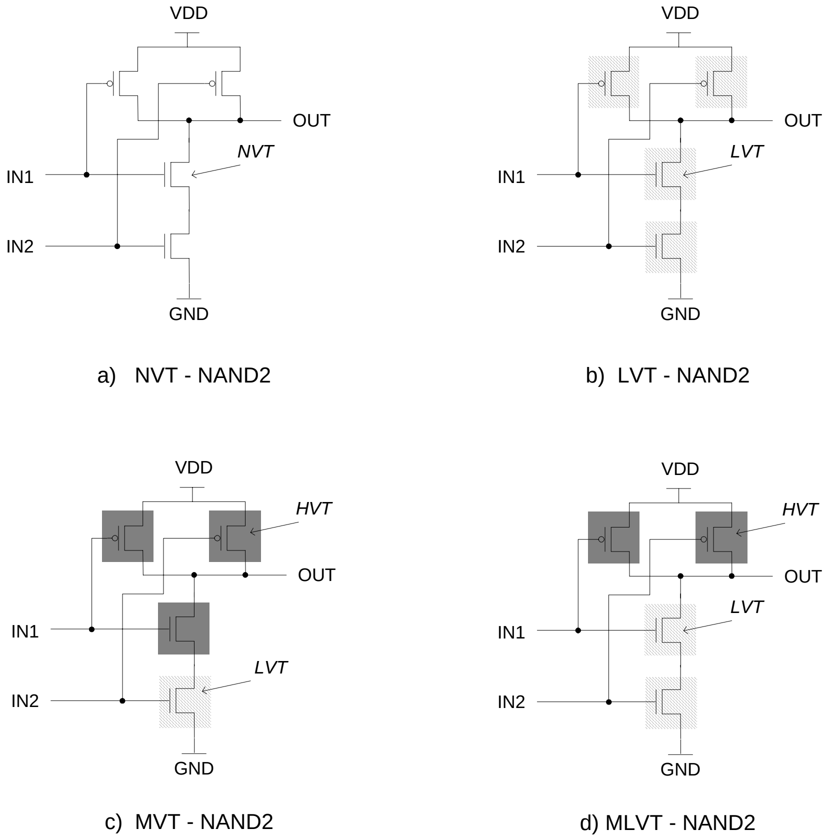


figure 2 Gate times

A lot of designs are most of the time in idle mode and in new technologies the leakage power has the same magnitude as dynamic power. Usually, the reduction of $P_{leakage}$ is more important than the reduction of P_{dyn}. Hence, in noncritical paths high-V_{th} transistors are used to reduce $P_{leakage}$, while critical paths consist of low-V_{th} transistors, which have a high leakage current. The stacked NMOS transistors are the critical path of the NAND2 in figure 3. Hence, the parallel PMOS transistors could slow down, as depicted in figure 3d).

The critical path mostly consists of stacked transistors, which must be low-V_{th} transistors. To reduce leakage in such a stack, transistors with different threshold voltages can be used. The delay will be the same, while the leakage at most of input states decreases. That means if the high-V_{th} transistors are non-conductive, the leakage is smaller than in a NVT-gate. If only the low-V_{th} transistor is non-conductive, the leakage is higher than in a NVT-gate. In a NAND2 that are 25% of all cases. This effect can be reduced by probability analysis, because the order of input signals is irrelevant in stacks. Hence, if signal probabilities are known, the signal with highest '*low*'-probability (NMOS-stack) or the highest '*high*'-probability (PMOS-stack) should connect to one of the high-V_{th} transistors. This optimization is depicted in figure 3c), where the NMOS stack consists of a high-V_{th} and a low-V_{th} transistor.



a) NVT - NAND2

b) LVT - NAND2

c) MVT - NAND2

d) MLVT - NAND2

figure 3 Different realizations of a NAND2 gate

Table 1 contains delay and leakage at different input vectors for a NAND2, which was simulated with 60nm BPTM models [BPTM]. We generated three types of transistors with different threshold voltages (180mV. 220mV, 250mV). The values were calculated from equation (1). The NVT-, HVT- and LVT-gates consists of normal-, low- and high-V_{th} transistors only. In MLVT-gates the non-critical path consists of high-V_{th} transistors, else low-V_{th} transistors are used (see figure 3d). The non-critical path in MVT-gates is the same as in MLVT-gates, while the critical path consists of low-V_{th} and high-V_{th} transistors (see figure 3c). The PMOS transistors have $W=1.2\mu m$ and the NMOS transistors $W=0.6\mu m$, but in the MVT-gate the high-V_{th} transistor has $W=0.7\mu m$. The temperature is 25°C and $V_{DD}=0.9V$.

There are two ways to use presented MVT technique. (I) All gates are replaced by MVT-gates or (II) after a timing analysis at gate level HVT, MLVT and MVT gates are inserted. Solution (I) is used if timing data of the gates are not available. The leakage is reduced, while the delay of the design is constant. Only a new gate library is necessary. If the worst case delay of the gates is known, solution (II) can be used. The result is a netlist with MLVT, MVT and HVT gates, depicts in figure 4b).

4. ALGORITHMS FOR OPTIMIZAION

In order to achieve an optimal mixed V_{th} design, a set of gate-level algorithms for timing analysis and assignment of HVT-, LVT- and MVT-gates are proposed. The first algorithm is used to generate the MVT-gates and needs three kinds of transistor-V_{th} types. Current technologies provide this [UMC, TSMC]. At first, the longest path in the *Power-Up-Network* (PUN) and *Power-Down-Network* (PDN) is estimated. Next, the transistors will be changed to high-V_{th} until the maximum delay is reached. In MVT-gates the stacks in the slowest Power-Network (PN) will be optimized additionally.

		NVT	LVT	HVT	MVT	MLVT
I_{leak} [nA]	V_{in}= 00	46	58	41	46	58
	V_{in}= 01	58	101	43	46	101
	V_{in}= 10	51	89	37	84	89
	V_{in}= 11	78	236	19	19	19
Sum [nA]		**233**	484	140	**195**	267
t_d [ps]		60	53	71	60	53

table 1 Leakage and delay at NAND2 gates

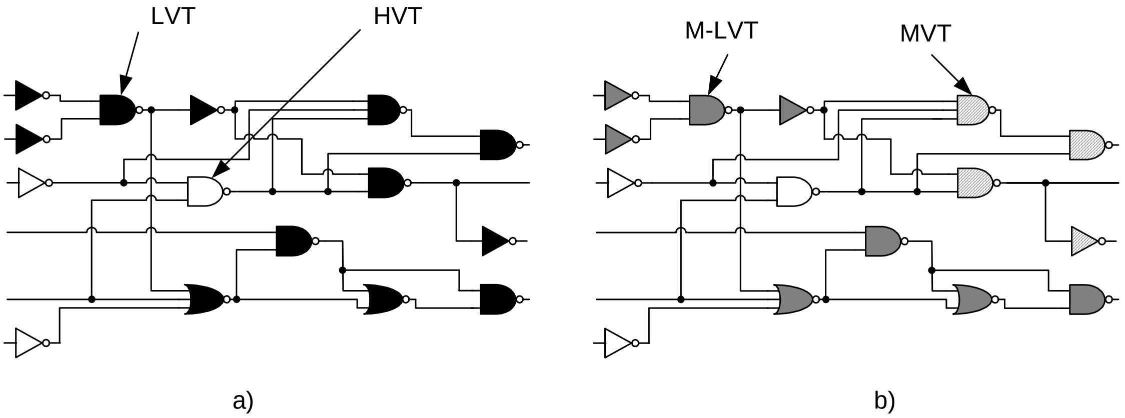


figure 4 Netlists of usual Dual-V_{th} design (a) and a MVT-design (b)

The first algorithm at gate-level evaluates the times of each gate. The netlist is traversed with a recursive function, whereas every gate on a net is scanned. If the time $T_{in}(G_i)$ of all gate inputs is known, $T_{eval}(G)$ will be calculated. The stop criterion is the achievement of the netlist inputs if all gates are scanned. The next algorithm on gate-level is the insertion of the LVT-gates. The optimization starts at the gate with the output signal of the design, which arrives at last. All gates, which are involved in this path, will change to LVT-gates until this path is the slowest. Then, the next path will be optimized, until the best delay of the design is reached. It is possible, that a path could be optimized more times. At last, the NVT-gate will be replaced by HVT-gates. An example for the resulting netlist is depicted in figure 4a. At this point usually used algorithms terminate [Anis03, Kar02, Wei99]. The last algorithm attends the insertion of MVT-gates by reanalyzing the design. For every LVT-gate the slack $S(G)$ is calculated. If this slack is higher then the delay $T_d(G)$ of a MVT-gate, the LVT-gate will be replaced by an MVT-gate.

```
Generate MVT-gates ()
BEGIN
  Estimate longest paths in PUN, PDN

  IF MLVT-gate
      Set all transistors to low-$V_{th}$

  $t_{PDN}$ = delay of longest path in PDN
  $t_{PUN}$ = delay of longest path in PUN
  $t_{max}$ = Max($t_{PDN}$,$t_{PUN}$)
  WHILE $t_{PDN} < t_{max}$
         Change one transistors of PDN to high-$V_{th}$
  WHILE $t_{PUN} < t_{max}$
         Change one transistors of PUN to high-$V_{th}$

  IF MVT-gate
     FOR every stack in slowest PN
        $t_{max}$ = delay of stack
        Set all transistors to low-$V_{th}$
        WHILE $t_{stack} > t_{max}$
            Change one transistor to high-$V_{th}$

END
```

```
Timing Analyses ()
BEGIN
  FOR all design_inputs
        Trace()


 Trace()
 BEGIN
    IF all next gates are visited from this gate
        IF gate_inputs = design_inputs
            QUIT
        ELSE
            Go back to gate before
            Trace()
    ELSE
        Go to next gate which was not visited from this
        gate
        IF all T_in of gate known
            Calculate T_eval(G)
            Trace()
        ELSE
            go back to last gate
            Trace()
 END
END
```

```
Insertion of LVT-gates ()
BEGIN
 T_max_old = Time of slowest  design_ output
 T_ max = 0
 WHILE T_max_old > T_max
 BEGIN
     T_max = T_max_old
     T_ max_path = T_max
     Go to gate with slowest design_output
     WHILE T_ max_path >= T_max AND inputs != design_inputs
      BEGIN
          Go to gate with slowest input
           IF gate is NVT-gate
                Change gate to LVT-gate
                Recalculate T_eval(G)
                T_ max_path = T_ max_path – spared time
      END
      Recalculate T_eval of all affected gates
      T_max = Time of slowest design_ output
   END

   FOR all gates
        IF gate = NVT-gate
       Change gate to HVT-Gate
END
```

```
Insertion of MVT-Gates ()
BEGIN
  FOR all design_outputs
        Trace()

 Trace()
 BEGIN
     IF all next gates are visited from this gate
          IF  gate_inputs = design_outputs
                QUIT
          ELSE
                Go back to last gate
                Trace()
     ELSE
          Go to next non-visited gate
          Calculate slack S(G)
          IF  S(G) >= $T_d(MVT)$
                Change LVT-gate to MVT-gate
          Recalculate T_eval of affected gates
          Trace()
 END
END
```

5. SIMULATION RESULTS

We simulated two kinds of designs. At first, we only used the new developed MVT-gates and simulated three ICSACS'85 designs [Han99]. Thereby, we generated designs, which consisted of only one gate type. Next, we optimized a 4Bit-adder design with proposed algorithms. The transistor models for the simulation based on the BPTM-models [BPTM] which we adapted. We simulated a set of input vectors and estimated the average values of the leakage currents.

Table 2 contained the results for the three ISCAS'85 designs. The analysis concludes that the leakage of the designs, which consist of MVT gates, is approximately 20% lower than the leakage in the NVT-designs, which have the same performance. The use of MLVT-gates can reduce the leakage down to 50% compared to a LVT-design.

The results of the optimized and simulated 4Bit Adder are listed in table 3. We simulated an adder, which consists of NVT-gates only. Next, we detected the critical paths and replaced corresponding NVT-gates with LVT-gates, while remaining gates were replaced by HVT-gates. We called the resulting design double-V_{th} (DVT). At last, we optimized the DVT-design and replaced some LVT-gates with MVT-gates, thereby the performance stayed the same. The remaining LVT-gates were changed to MLV-gates. The optimized designs were approximately 20% faster than the NVT-design, while the leakage of the MVT-design is about 20% lower than the leakage of the NVT-design. The leakage of the usually optimized DVT-design is approximately 40% higher than the leakage of the MVT-design.

	NVT - $I_{leakage}$	MVT - $I_{leakage}$	LVT - $I_{leakage}$	MLVT - $I_{leakage}$
c432	31uA	25uA	68uA	43uA
c1908	30uA	25uA	64uA	43uA
c6288	180uA	154uA	440uA	237uA

table 2 Average values of the leakage currents of simulated ISCAS’85 designs

	NVT	DVT	MVT
$I_{leakage}$	3.4uA	3.8uA	2.7uA
t_d	556ps	470ps	470ps

table 3 Average leakage and delay of simulated 4Bit-Adders

6. CONCLUSION

We proposed a Mixed-V_{th} (MVT) CMOS design technique to reduce the static power dissipation on gate-level. Thereby, the performance of the design stays constant or increases. We presented a set of algorithms for gate-level assignment of threshold voltage. To compare proposed Mixed-V_{th} technique with usually used techniques a 4Bit adder and three ISCAS’85 designs were simulated. The results indicate that MVT-designs reduce the leakage by about 20% compared to normal-V_{th} designs. The proposed algorithms provide about 40% more leakage savings than corresponding dual-V_{th} techniques, while the performance stays the same.

REFERENCES

Journals

[Kim03] N.S. Kim, T. Austin, D. Blaauw, T. Mudge, K. Flautner, J.S. Hu, M.J. Irwin, M. Kandemir, and V. Narayanan, *Leakage Current: Moore’s Law Meets Static Power*, in IEEE Computer, p. 68, no. 12 (2003).

[Kao00] J.K. Kao and A. Chandrakasan, *Dual-Threshold Voltage Techniques for Low-Power Digital Circuits,* in IEEE Journal of Solid State Circuits, p. 1009, no. 35 (2000).

[Han99] M. Hansen, H. Yalcin, and J. P. Hayes, *Unveiling the ISCAS-85 Benchmarks: A Case Study in Reverse Engineering*, in IEEE Design and Test, p. 72, no. 16 (1999).

[Sak90] T. Sakurai and A. Newton, *Alpha-Power Law MOSFET Model and its Application to CMOS Inverter Delay and other Formulas,* in IEEE Journal of Solid-State Circuits, pp. 584-594, no. 2 (1990).

[Sir04] N. Sirisantana and K. Roy, *Low-Power Design Using Multiple Channel Lengths and Oxide Thicknesses*, in IEEE Design and Test of computers, pp. 56-63, no. 1 (2004).

[Wei99] L.Wei, Z. Chen, K. Roy, M. Johnson, Y. Ye, and V.K. De, *Design and Optimization of Dual-Threshold Circuits for Low-Voltage Low-Power Applications*, in IEEE Trans. on VLSI Systems, pp.16, no. 1 (1999).

Conference Proceedings

[Che01] W.Chen, W.Hang, P.Kudva, G.D. Gristede, S.Kosonocky and R.V.Joshi, *Mixed Multi-Threshold Differential Cascode Voltage Switch (MT-DCVS) Circuit Styles and Strategies for Low Power VLSI Design*, ISLPED'01, California, USA (2001).

[Kar02] T.Karnik, Y. Ye, J. Tschanz, L. Wei, S. Burns, V. Govindarajulu, V. De, and S. Borkar, *Total Power Optimization by Simultaneous Dual-Vt Allocation and Device Sizing in High Performance Microprocessors*, in Proceedings of the 39th conference on Design automation, New Orleans, Louisiana (2002).

[Sun99] V.Sundararajan and K.Parhi, *Low Power Synthesis of Dual Threshold Voltage CMOS VLSI Circuits*, in Proceedings of the IEEE International Symposium on Low Power Electronics and Design, pp. 139-144 (1999).

[Wei99] L.Wei, Z.Chen, and K.Roy, *Mixed-*v_{th} *(MVT) CMOS Circuit Design Methodology for Low Power Applications*, in Proceedings of the 36th Design Automation Conference, pp. 430-435 (1999).

[Wei00a] L.Wei, K. Roy, and V.De, *Low Voltage Low Power CMOS Design Techniques for Deep Submicron ICs"* in Proceedings of the 13th Int. Conference on VLSI Design, pp. 24-29 (2000).

[Wei00b] L.Wei, K. Roy, and C. Koh, *Power Minimization by Simultanous Dual-Vth Assignment and Gate-sizing,* in Proceedings of the IEEE Custom Integrated Circuits Conference, pp. 413-416 (2000).

Books

[Ani03] M. Anis and M. Elmasry, in *Multi-Threshold CMOS Digital Circuits*, Kluwer Academic Publishers (2003).

[Vee00] H. Veendrick, in *Deep-Submicron CMOS ICs*, Kluwer Academic Publishers (2000).

Other

[BPTM] Berkeley Predictive Technology Model, *www-device.eecs.berkeley.edu/~ptm* (2002).

[UMC] United Microelectronics Cooperation, *www.umc.com*

[TSMC] Taiwan Semiconductor Manufacturing Company Ltd., *www.tsmc.com*