Low-Power Fanout Optimization Using Multiple Threshold Voltage Inverters

Behnam Amelifard
Department of EE-Systems
University of Southern California
Los Angeles, CA
(213) 740-9481
amelifar@usc.edu

Farzan Fallah
Fujitsu Laboratories of America
Sunnyvale, CA
(408) 530-4544
farzan@fla.fujitsu.com

Massoud Pedram
Department of EE-Systems
University of Southern California
Los Angeles, CA
(213) 740-4458
pedram@ceng.usc.edu

ABSTRACT

This paper addresses the problem of low-power fanout optimization with multiple threshold voltage inverters. Introducing splitting and merging conversions that preserve delay, power, and input capacitance, the fanout tree is converted to a set of inverter chains and for each chain the optimal sizes and threshold voltages are determined. Experimental results show that using this technique, the power dissipation of fanout tree is reduced by an average of 33% for a state-of-the-art CMOS technology.

Categories and Subject Descriptors

B.6.3 [Design Aids]: Automatic synthesis, Optimization.

General Terms

Algorithms, Design, Performance.

Keywords

Low-power design, Fanout optimization, Fanout tree, Buffer chain.

1. INTRODUCTION

In a VLSI design, it is often necessary to distribute a signal to several destinations under a required timing constraint at each destination. Furthermore, in practice, there may also be a limitation on the load that can be driven by the source signal. Fanout optimization is the problem of finding a buffer tree topology and sizing the buffers in this topology to satisfy the constraints.

The fanout optimization problem for libraries with discrete sizes had been proven to be NP-complete [1]. It is shown that using a buffer library with near-continuous sizes exponentially reduces the problem complexity [2, 3]. Several techniques have been proposed to address the fanout optimization problem using simplified delay models. Reference [3], for example, introduced transformations, namely "merging" and "splitting", which are used to convert any fanout tree to a set of inverter chains. Using the transformation introduced in [3], reference [4] proposed a logical effort-based fanout optimizer for area and delay which attempts to minimize the total buffer area under the required time and input capacitance.

Although much research has been done to address fanout optimization problem, to the best of our knowledge, there is no work on low-power fanout optimization which tries to reduce the total power dissipation of the fanout tree by utilizing two or more threshold voltages. The reminder of this paper is organized as follows. In Section 2 the delay and power model which will be used through the paper are described. Section 3 formulates the problem of fanout optimization for low power, while simulation results are given in Section 4. Section 5 concludes the paper.

2. DELAY AND POWER MODELS

2.1 Delay model

The delay model used in this paper is based on the concept of "logical effort" [5]. In this model, the delay of a gate is defined as:

$$d = (p + gh)\tau_0 \qquad (1)$$

where τ_0 is a conversion coefficient that characterizes the semiconductor process being used and converts the unit-less part, $p+gh$, to a time unit. Parameter p denotes the parasitic delay of the gate. The major contribution to the parasitic delay is the capacitance of the source/drain regions of the transistors that drive the output. Parameter g denotes the "logical effort" of the gate which depends only on the topology of the gate and its relative ability to produce output current. Finally, parameter h denotes the "electrical effort" of the gate and is defined to be the ratio of the output capacitance of the gate to its input capacitance. In effort-based technique, the value of g for an inverter is assumed to be 1 whereas for other gates this value is calculated based on their transistor-level topologies. The important point in logical effort is that parameters p and g are independent of the size of the gate. In fact, the only parameter that is affected by gate sizing is the electrical effort, h.

The concept of logical effort of a gate can be extended in order to handle multiple threshold technologies. It is shown in [7] that when the threshold voltage of a gate is changed, the new delay can be obtained from the alpha-power law [5] by the following equation,

$$d_{V_{T1}} = d_{V_{T0}} \frac{(1 - V_{t0}/V_{DD})^{\alpha}}{(1 - V_{t1}/V_{DD})^{\alpha}} \qquad (2)$$

where α is a technology parameter which is around 1.3 for short channel devices and 2 for long channel devices, V_{DD} is the supply voltage, V_{t0} is the nominal threshold voltage, $d_{V_{T0}}$ is the delay under this nominal threshold voltage, V_{t1} is an arbitrary threshold voltage, and $d_{V_{T1}}$ is the delay under this arbitrary threshold voltage. With this relation, the general equation of effort-based delay of a logic gate should be changed as follows,

$$d_v = \tau_0 \frac{p + gh}{(1 - v)^{\alpha}} (1 - v_0)^{\alpha} \qquad (3)$$

where v is the ratio of the threshold voltage to the supply voltage and v_0 is that ratio of the nominal threshold voltage to the supply voltage. The last term, i.e., $(1-v_0)^{\alpha}$, is a constant. For the sake of simplicity, we thus set $\tau_0(1-v_0)^{\alpha}$ to one, i.e., $d_v=(p+gh)/(1-v)^{\alpha}$.

2.2 Power model

Subthreshold leakage is the drain-source current of a transistor operating in the weak inversion region. In current CMOS technologies, the subthreshold leakage is much larger than the other leakage current components [9]. The subthreshold leakage current of a CMOS transistor can be expressed as follows [6],

$$I_{Leak} = A e^{q(V_{gs} - V_t + \eta V_{ds})/n'kT} \left(1 - e^{-qV_{ds}/kT}\right) \qquad (4)$$

where $A=\mu_0 C_{ox}(W/L_{eff})(kT/q)^2 e^{1.8}$, μ_0 is the zero bias mobility, C_{ox} is the gate oxide capacitance per unit area, W and L_{eff} denote the width and effective length of the transistor, k is the Boltzmann constant, T is the absolute temperature, and q is the electrical charge of an electron. In addition, V_t is the threshold voltage, η denotes the drain-induced barrier lowering (DIBL) coefficient, and $n'\geq 1$ is the slope shape factor of the transistor.

Let C_{in} denote the input capacitance of the transistor. Absorbing the effect of DIBL in a new coefficient A', assuming that V_{ds} of the OFF transistor is a few multiples of $kT/q\sim 26mV$, and noting that for a CMOS gate $W=C_{in}/(L_{eff}C_{ox})$, the leakage formula is simplified to,

$$I_{Leak} = A'C_{in}e^{-\lambda v} \qquad (5)$$

where $\lambda=qV_{DD}/n'kT$ is a constant and v is the (normalized) threshold voltage ($v=V_t/V_{DD}$).

Ignoring the short circuit power dissipation, we can write the total power consumption of an inverter with input capacitance of C_{in}, driving an output capacitance of C_L as,

$$P = fV_{DD}^2 C_L \chi + A'V_{DD}C_{in}e^{-\lambda v} \qquad (6)$$

where χ denotes the expected number of $0\rightarrow 1$ transitions at the output of the logic gate per clock cycle, C_L is the output capacitance, and f is the clock frequency. With fixed V_{DD}, f, and χ, we can simplify this equation to,

$$P = Q(C_L + RC_{in}e^{-\lambda v}) \qquad (6.a)$$

where $Q=fV_{DD}^2\chi$ and $R=A'/(fV_{DD}\chi)$.

3. FANOUT TREE DESIGN FOR LOW POWER

3.1 Fanout Optimization

By exploiting two transformations that (a) split a fanout tree to a set of inverter chains and (b) merge the chains back into a fanout tree, the low-power fanout optimization problem is performed in two steps. In the first step, the constraint on the input capacitance of the fanout tree is first converted into a set of input capacitance constraints for each inverter chain and then the inverter chain optimization problem is solved to minimize the power dissipation of the chain subject to its input capacitance and source-to-sink delay constraints. In the second step, the optimized results for each chain are merged to produce a fanout tree that satisfies the source-to-sink delay constraints, the total input capacitance constraint, and at the same time minimizes the total power dissipation in the fanout distribution tree. This is of course not a global optimum solution because of use of the two step process.

The split and merge transformations are depicted in Figure 1. Notice that all inverters have the same normalized threshold voltage, v, and electrical effort, h.
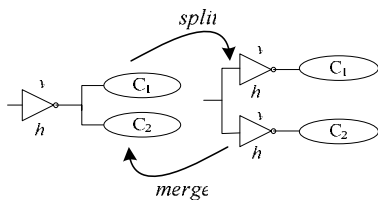

Figure 1. Merge/Split transformation.

Theorem T1: *The split/merge transformations preserve the delay, input capacitance, and power dissipation values of a fanout tree.*

Proof: Consider the split transformation. Before splitting, the delay through the inverter is $(p+h)/(1-v)^\alpha$, whereas the input capacitance is $(C_1+C_2)/h$. After splitting the original inverter to two inverters with equal electrical efforts of h and equal threshold voltages of v, the delay through the inverter in either branch will be $(p+h)/(1-v)^\alpha$, while the input capacitances will be C_1/h and C_2/h. Therefore, this transformation preserves the delay and input capacitance values. Furthermore, the total power dissipation of the fanout tree before the split transformation is equal to $Q(C_1+C_2+RC_{in}e^{-\lambda v})+P'$, where C_{in} is the input capacitance of the inverter which is equal to $(C_1+C_2)/h$ and P' is the power dissipation of the remaining circuits in the fanout branches. After splitting, the power dissipation of the tree will be equal to $Q(C_1+RC_1e^{-\lambda v}/h)+Q(C_2+RC_2e^{-\lambda v}/h)+P'$, which is again equal to the power dissipation before splitting. The proof for the merge transformation is similar and is omitted here. ■

3.2 Inverter Chain Optimization

When there is only one sink, the fanout tree is reduced to a chain of inverters between the source and sink and the fanout optimization problem becomes that of finding the sizes and threshold voltages of the inverters to satisfy a timing and input capacitance constraints while minimizing total power dissipation.
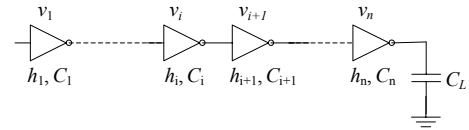

Figure 2. An inverter chain.

An inverter chain is shown in Figure 2. In this figure h_i's denote the electrical efforts of the inverters, C_i's are the input capacitances, and v_i's are the threshold voltages of the inverters.

The dynamic power dissipation of this inverter chain is given by,

$$P_{Dyn} = Q\sum_{i=1}^{n} C_i . \qquad (7)$$

Notice that in the above equation, we do not include the power dissipation needed to drive the final load capacitance, C_L. The reason is that this power dissipation term is fixed and does not change during the inverter chain optimization process.

The total leakage power dissipation of this chain is expressed by,

$$P_{Leak} = R\sum_{i=1}^{n} C_i e^{-\lambda v_i} . \qquad (8)$$

Since $C_i = C_L / \prod_{j=i}^{n} h_j$, the chain power dissipation is written as,

$$P_{chain} = QC_L \sum_{i=1}^{n} \frac{1}{\prod_{j=i}^{n} h_j}\left(1+\beta e^{-\lambda v_i}\right) \qquad (9)$$

where $\beta=R/Q$. The chain delay is expressed as,

$$d_{chain} = \sum_{i=1}^{n} \frac{p+h_i}{(1-v_i)^\alpha} . \qquad (10)$$

The goal is to find the number of inverters, n, h_i's, and v_i's so as to minimize the power dissipation while meeting both a timing constraint and an input capacitance constraint. For now, we assume that threshold voltages can take any value between a lower and an upper bound. So, ignoring the constant terms, the formulation of the inverter chain optimization problem is as follows.

Inverter Chain Optimization problem (ICO): Given an inverter chain as described in Figure 2, determine n, h_i's, and v_i's by solving the following mathematical program,

$$\begin{cases} Min \quad \sum_{i=1}^{n} \dfrac{1}{\prod_{j=i}^{n} h_j}\left(1+\beta e^{-\lambda v_i}\right) \\[4mm] st: \quad (i) \quad \sum_{i=1}^{n} \dfrac{p+h_i}{(1-v_i)^{\alpha}} \le T \qquad\qquad (11) \\[4mm] \quad (ii) \quad H \equiv \prod_{i=1}^{n} h_i \ge \dfrac{C_L}{C_{in}} \\[4mm] \quad (iii) \quad v_{min} \le v_i \le v_{max} \end{cases}$$

The first inequality ensures the total input to output delay of the inverter chain is no more than a delay budget, T. The second inequality ensures that the total effort of the chain, denoted by H, is greater than or equal to C_L/C_{in}, where C_{in} denotes the maximum allowed value of the input capacitance of the first inverter in the chain. Since H is equal to C_L/C_1, this constraint simply states that $C1 \le C_{in}$. The third constraint bounds the threshold voltages of the inverters between a minimum and a maximum value.

Theorem T2: *The ICO* problem *is a convex program.*

Proof: The objective function of ICO problem is the summation of convex functions; therefore, it is convex. On the other hand, the first inequality is a monotonically increasing function of both h_i's and v_i's, while the second inequality is a monotonically increasing function of h_i's; so, the ICO problem is a convex program. ∎

Lemma L1: *In the ICO problem, the total electrical effort, H, is maximized when all v_i's are equal to v_{min} and all h_i's are equal.*

Proof: The geometric mean of a number of positive numbers is less than or equal to their arithmetic mean. The equality holds if and only if all values are equal. From the first constraint it is seen that the maximum of summation of all h_i's happens when all v_i's are equal to v_{min}. In this case $\sum h_i = T(1-v_{min})^{\alpha} - np$ and the maximum value of $H = \prod h_i$ is $H_{max} = (T(1-v_{min})^{\alpha}/n-p)n$. ∎

The second constraint in (11) implies that H must be greater than or equal to C_L/C_{in}. Since Lemma L1 puts an upper bound on the maximum value that H can achieve, the only feasible inverter counts are those for which H_{max} is equal to or larger than C_L/C_{in}. This observation will be used in the next section to bound the number of inverters in the chain.

If only *m discrete threshold voltages* are available, then the ICO problem will be called *m*-VT ICO problem. Although the ICO problem can be solved by using standard mathematical program solvers, it is instructive and useful to consider the important case of the 2-VT problem as described in the next section.

3.3 Optimization with Two Threshold Voltages

Since each additional threshold voltage needs one more mask layer in the fabrication process which results in increasing the fabrication cost, in many cases, only two threshold voltages are utilized in the circuit [10]. At the same time, there are studies that show the benefit of having more than two threshold voltages is small [10]. So, in the following we concentrate on the problem of low-power fanout optimization when only two threshold voltages, namely v_L and v_H, are available. The results can be extended to handle more threshold voltages.

Theorem T3: *In the optimal solution of the 2-VT version of the ICO problem, the threshold voltages of inverters are non-decreasing: $v_{t1} \le v_{t2} \le \dots \le v_{tn}$*

This theorem states that in the optimal solution of the 2-VT fanout problem, all inverters with low threshold voltages are placed before the high threshold inverters. (The proof is long and is omitted here.) In light of this theorem we present an efficient algorithm for solving the 2-VT fanout optimization problem as described in Figure 3.

```
BestChain (C_in, C_L, T)
0. Begin
1.    (n_1*, n_2*)=FindSoln (T(1−v_min)^α−np)^n=C_L/C_in;
2.    n_1 = ⌊n_1*⌋ or ⌊n_1*⌋+1 ; (depending on polarity)
3.    n_2 = ⌊n_2*⌋ ; (pwr*, h̄*, v̄*) = (+∞, NULL, NULL) ;
4.    For n=n_1 to n_2 step 2
5.        For i=1 to n step 1  v̄(i) = v_L ; Endfor
6.        (h̄, pwr) = FVT(n, T, C_in, C_L, v̄) ;
7.        If h̄ = NULL  Continue; Endif
8.        If pwr < pwr*  (pwr*, h̄*, v̄*) = (pwr, h̄, v̄) ; Endif
9.        For m=n to 1 step -1
10.           v(m)=v_H ; (h̄, power) = FVT(n, T, C_in, C_L, v̄) ;
11.           If h̄ = NULL  Exit loop; Endif
12.           If pwr > pwr*  (pwr*, h̄*, v̄*) = (pwr, h̄, v̄) ; Endif
13.       Endfor
14.    Endfor
15.    Return(h̄*, v̄*)
16. End
```

Figure 3. Algorithm for 2-VT fanout optimization.

First, by using the result of Lemma L1, for a given C_{in}, C_L, and T, *FindSoln* finds the lower and upper bounds of n. Based on the polarity of the sink node, only even or odd numbers of inverters between these bounds are considered when searching for the optimum solution. For a given n, the *BestChain* algorithm attempts to solve the 2-VT ICO problem with all threshold voltages set to v_L. If there is no feasible solution, then the timing and/or input capacitance constraints are too tight. Otherwise, the algorithm goes through a number of iterations where, in each iteration, the threshold voltages of last m inverters in the chain are set to v_H. This process is repeated until we find m^* such that there exists a feasible solution to the 2-VT ICO with m^* inverters but not with m^*+1 inverters. Function *FVT* finds the optimum solution to the ICO problem with n stages and known threshold voltage values as captured by the assignment vector, $\vec{v}$. Since v_i's are set each time the *FVT* function is called, this optimization problem is a minimization of a posynomial function with posynomial inequality constraints which is solvable in polynomial time [7].

3.4 Building a Fanout Tree

To solve the tree fanout optimization problem using inverter chain fanout optimization, we need to address two issues. The first issue is the *input capacitance allocation* to different chains in a decomposed fanout tree. It was shown in [4] that this problem is NP-complete. The heuristic we use is similar to that of [4] and starts by allocating the minimum input capacitance required for each branch to have a feasible inverter chain solution. Next, the remaining total input capacitance is divided between the chains in proportion to the positive slopes of $H_{max,i}$ versus n_i for each branch i. The second issue to address is the assumption of the availability of a continuous-size inverter library. In reality, in the ASIC libraries the sizes of inverters are discrete. So the solution needs to be mapped onto one for the available inverters in the library. The problem when rounding the inverter sizes is that it may result in significant errors. To address this problem, reference [4] defined a constant ε and merged two inverters on different chains only if the difference between their electrical efforts was less than or equal to ε. In addition, two inverters are merged only if the rounding error after merging is smaller than the sum of the rounding errors of inverters before the merge operation. We adopt the same heuristic with the additional requirement that the two candidate inverters should also have the same threshold voltage. Merging is performed starting at the source of the signal, and proceeds toward sinks.

4. SIMULATION RESULTS

We performed simulations for a 70nm technology node. The supply voltage is 0.8V and the values of low and high threshold voltages for this technology node are 0.2 and 0.3, respectively. We compare the results of 2-VT fanout optimization with the results of 1-VT fanout optimization. Note that in 1-VT fanout optimization, the algorithm minimizes the total dynamic power. This indirectly minimizes the area of the tree. So the results of 1-VT fanout optimization are equivalent to the results of [4].

TABLE 1. COMPARISON BETWEEN 1-VT AND 2-VT ICO

Cir.	C_{in}	C_{out}	T	P	1-VT ICO		2-VT ICO		Pwr Red. (%)
					Pwr	*Ar*	*Pwr*	*Ar*	
I	1	1000	100	–	47.3	14.7	23.3	18.1	50.8
II	2	135	70	–	9.9	3.1	4.9	3.8	50.4
III	4	110	40	+	14.1	4.4	7.0	5.4	50.6
IV	3	500	60	+	39.3	12.1	19.2	14.9	51.1
V	0.1	500	80	+	33.7	10.4	18.9	12.8	44.0
VI	4	100	40	–	8.9	2.8	4.2	3.29	52.4
VII	0.4	300	60	–	26.7	8.3	13.5	10.5	49.5
IIX	2	2000	80	–	124.1	38.4	124.1	38.4	0.0
IX	2	100	50	–	11.3	3.5	5.7	4.4	49.6
X	5	1000	90	+	62.4	19.3	62.4	19.3	0.0

In the first set of experiments, we compare the efficiency of 2-VT version of ICO with the 1-VT version. Simulation results for a few random problems are shown in Table 1. In this table C_{in} is the maximum capacitance at the input of the inverter chain, C_{out} is the sink load, T is the required time at the sink, P is the polarity of the sink. The power dissipation, *Pwr*, and area of the chain, *Ar*, are shown for each version of the problem. The power dissipation of 2-VT ICO is on average 40% smaller than that of 1-VT version. The area of 2-VT ICO, however, is on average 18% larger than that of 1-VT version. The reason is that when the threshold voltages of some gates are raised, their sizes must be increased to satisfy the required time constraint. Notice that in this table and the following ones, $\tau_0(1-v_0)^\alpha$, QC_L and the parasitic delay of an inverter, p, have been normalized to one. Moreover, C_{in} and C_{out} are measured in arbitrary units. The area is defined as the total size of inverters.

In the second set of experiments, the fanout optimization problem is solved for a set of arbitrary circuits. Each circuit states a source and multiple sinks with capacitive load, required time, and polarity constraints specified for each sink. The specification of each circuit, including the maximum input capacitance (C_{in}), the number of sinks with positive and negative polarities ($p+$ and $p-$), the maximum and minimum required times of all sinks (T_{max} and T_{min}), and the maximum and minimum sink capacitances (C_{Lmax}, C_{Lmin}), are shown in Table 2. The resulting power and area of 1-VT and 2-VT versions of fanout optimization are reported in Table 3. It is seen that for a 70nm technology, 2-VT fanout optimization results in an average improvement of 33% in power dissipation.

For all the experiments in Tables 1 and 3, the minimization problems of our algorithm were solved using Matlab Optimization Toolbox 7.0.0. Note that in our problem setup and in the simulation results, we have ignored the interconnect power and delay cost. The reason is that we do the fanout optimization during logic synthesis and prior to generating layout. Therefore, the locations of the source and the sink are not known. It is thus reasonable to assume the expected values of delay and power dissipation per wire in the inverter chain or the fanout tree are nearly the same. This fixed contribution can, thus, be taken out of the problem formulation by adjusting the required time constraints on sinks and adding a constant term to the total power equation.

TABLE 2. SPECIFICATIONS OF THE TESTBENCHES

circuit	$P+$	$P-$	C_{in}	T_{max}	T_{min}	C_{Lmax}	C_{Lmin}
1	4	2	14	200	60	3000	200
2	3	4	11	100	40	1000	100
3	2	1	9	200	50	3000	100
4	6	4	24	180	40	2000	44
5	11	1	26	200	40	3000	64
6	4	1	11	90	35	1000	50
7	9	4	26	200	40	3000	64
8	4	3	17	100	45	1000	100
9	9	0	14	200	40	3000	64
10	4	1	12	90	40	1000	100

TABLE 3. SIMULATION RESULTS FOR 1-VT AND 2-VT FANOUT OPTIMIZATION

circuit	1-VT fan. opt.		2-VT fan. opt.		Pwr Red. (%)
	Pwr	*Ar*	*Pwr*	*Ar*	
1	304.0	78.3	201.2	90.1	33.8
2	228.7	58.9	92.6	72.9	39.5
3	147.4	38.0	138.7	38.9	5.9
4	561.0	144.5	276.3	192.6	30.7
5	412.4	106.2	306.9	117.8	25.6
6	155.2	40.0	90.8	46.3	41.5
7	638.6	164.4	368.6	193.9	42.3
8	408.4	105.2	197.7	128.1	41.6
9	343.7	88.5	250.8	99.9	27.0
10	158.0	40.7	64.8	50.2	39.0

5. CONCLUSION

This paper addressed the problem of low power fanout optimization with two threshold voltages. Using splitting and merging conversions that preserve delay, power, and input capacitance, the fanout tree was converted to a set of inverter chains and for each chain the optimal sizes and threshold voltages were determined. After that the results for chains were merged to generate results for the original tree. Experimental results demonstrated that using two threshold voltages, instead of one, can reduce the overall power dissipation of the fanout tree by an average of 33% for a 70nm CMOS technology.

REFERENCES

[1] Berman, C. L. , Carter, J. L., and Day, K. F. The fanout problem: from theory to practice. In *Advanced Research in VLSI: Proc. of the 1989 Decennial Caltech Conf.*, MIT Press, 1989, 69-99.

[2] Kodandapani, K. , Grodstein, J., Domic, A. and Touati, H. A simple algorithm for fanout optimization using high-performance buffer libraries. In *Proc. of ICCAD*, 1993, 466-471.

[3] Kung, D. S. A fast fanout optimization algorithm for near-continuous buffer libraries. In *Proc. 35th DAC.*, 1998, 352-355

[4] Rezvani, P. and Pedram, M. A fanout optimization algorithm based on the effort delay model. *IEEE Trans. on Computer-Aided Design*, 22, (Dec. 2003), 1671-1678.

[5] Sakurai, T. and Newton, A.R. A simple MOSFET model for circuit analysis. *IEEE Trans. Electron Device*, 38, (Apr. 1991), 887-894.

[6] De, V., *et al.* Techniques for leakage power reduction in Chandrakasan, A., *et al.*, *Design of High-Performance Microprocessor Circuits*. IEEE press, NJ, 2001, 46-62.

[7] Vaidya, P. M. A new algorithm for minimizing convex functions over convex sets. In *Proc. IEEE Foundations Comput. Sci.*, 1989, 332-337.

[8] Sundararajan, V. and Parhi, K. Low power synthesis of dual threshold voltage CMOS VLSI circuits. In *Proc. ISLPED*, 1999.

[9] Semiconductor Industry Association, International Technology Roadmap for Semiconductors, 2003 edition, http://public.itrs.net/.

[10] Sirvastava, A. Simultaneous Vt selection and assignment for leakage optimization. in *Proc. ISLPED*, 2003.