Scheduling and Binding for Low Gate Leakage NanoCMOS Datapath Circuit Synthesis

Saraju P. Mohanty
Computer Science and Engineering
University of North Texas
Box 311366, Denton, TX 76203.
Email: smohanty@cs.unt.edu

Elias Kougianos
Engineering Technology
University of North Texas
Box 310679, Denton, TX 76203.
Email: eliask@unt.edu

R. Velagapudi and V. Mukherjee
Computer Science and Engineering
University of North Texas
Box 311366, Denton, TX 76203.
Email: {rv0063,vm0058}@unt.edu

***Abstract*— With aggressive technology scaling, gate oxide tunneling current is emerging as a prominent component of power dissipation in nanoCMOS circuits. This paper presents a novel approach for reduction of tunneling current (gate leakage) during behavioral synthesis using simultaneous scheduling and binding of resources made of transistors of different gate oxide thicknesses. We provide a heuristic algorithm for optimizing allocation and utilization of resources while scheduling operations with the objective of reducing gate leakage. We selectively bind the off-critical operations to instances of functional units that have transistors of higher oxide thickness, and critical operations to the functional units of lower oxide thickness. We performed extensive experiments for several behavioral synthesis benchmarks using a $45nm$ technology library. For a time constrained approach we achieved a maximum reduction of 84.8%, while for a resource-time constrained approach the reduction is 75.8%.**

I. Introduction

In a short channel nanoCMOS, leakage has several forms, of which both reverse biased diode leakage and SiO_2 tunneling current flow are prominent during both active and sleep modes of the circuit [1]. As per the ITRS [2], high performance nanoCMOS circuits will require ultra low gate oxide thickness, thus making them susceptible to gate SiO_2 *direct tunneling* [3]. The probability of electron-hole tunneling is a strong function of height and thickness of the barrier. For a supply voltage V_{dd} and oxide thickness T_{ox}, tunneling current dissipation in a CMOS can be described as $I_{ox} \propto (\frac{V_{dd}}{T_{ox}})^2 exp\ (-\beta \frac{T_{ox}}{V_{dd}})$, where β is an experimentally derived factor [4], [5]. Thus, there are various options for reducing the tunneling current dissipation, which include decreasing supply voltage and/or increasing gate oxide thickness. While decreasing supply voltage continues to be a popular option for reduction of dynamic [6] as well as static power, the use of dual oxide thickness (Dual-T_{ox}) is currently being explored to reduce the tunneling current [7], [8]. It may be noted that the Dual-T_{ox} or Multi-T_{ox} (multiple thickness) approaches are in the same spirit as that of the Multi-V_{dd} [6], [9] and Multi-V_{Th} [10], [11] techniques and can be used in conjunction to provide a complete low power solution.

Behavioral level power reduction techniques have been largely confined to dynamic power and few deal with subthreshold leakage. However, many logic or transistor-level research works address reduction of gate oxide leakage. In [12] Khouri et. al. have proposed algorithms using dual-V_{Th} for subthreshold leakage analysis and reduction during behavioral synthesis. Gopalakrishnan et. al. in [13] also use a Multi-V_{Th} approach for reduction of subthreshold leakage during high-level synthesis. Mohanty et. al. [4] have introduced models and a scheduling algorithm for reduction of overall gate leakage of datapath circuits.

In this paper, we propose a resource-time constrained heuristic algorithm for simultaneous scheduling-binding for total gate leakage current reduction of nanoCMOS datapath circuits. We incorporate time constraints as a performance (or delay) trade-off factor and offer the user the choice of predetermining the performance of a circuit vis-a-vis the power requirements. The algorithm considers an unscheduled data flow graph (DFG), and schedules each of its nodes at appropriate control steps and simultaneously binds them to the best available resource so as to achieve the desired performance with minimum gate leakage. We have developed a pre-characterized datapath component library including functional, interconnecting and storage units for various gate oxide thicknesses. We assume that all transistors used in a resource have gate oxide of equal thickness, but the thicknesses of different resources may differ. The components with high T_{ox} have low tunneling current, but higher delay, thus making them more suitable for use in off-critical paths. While those with low T_{ox} are preferred in critical paths.

II. Multi-Oxide Thickness Component Library

At the top level our objective is to prepare a set of characterized cells of behavioral components for use in high-level synthesis. This necessitates the development of a seamless design environment that can make the transition between levels more transparent and enables the analysis of the various oxide tunneling currents at the behavioral level from the simulation data at the logic and transistor level. The three level hierarchical top-down approach is presented in Fig. 1.

We now assume that datapath components are constructed using universal logic gates, 2-input NAND, as NAND gates have minimal tunneling current compared to other logic gates [5], [14]. We also assume that there are a total of n_{total} NAND gates in the network of NAND gates constituting an n-bit functional unit, out of which n_{cp} number of NAND gates are in the critical path. In this model we do not consider the effect of interconnect wires and focus on the direct tunneling current

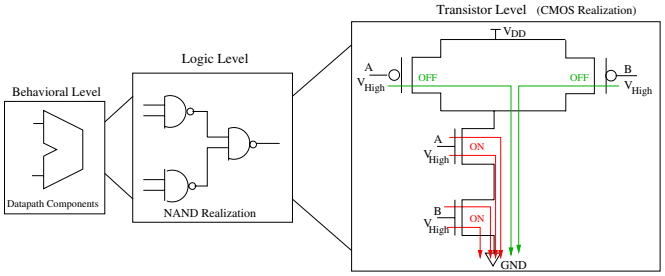


Fig. 1. Three levels of abstraction in which datapath components are realized using 2-input NAND gates. The transistor level diagram shows worst case tunneling current paths in the constituent NAND gates. The worst-case occurs when both the inputs are high i. e. $A = B = V_{High}$. In this case, the path of the current is from gate-to-source in the case of PMOS, whereas in the NMOS it is from gate to both drain and source. Thus, in the worst-case the total gate oxide tunneling current for a NAND gate is the sum of 6 different components as shown above. If the four possible states (00, 01, 10 and 11) have gate tunneling current ($I_{ox00}, I_{ox01}, I_{ox10}, I_{ox11}$), respectively, and assuming that all four states are equiprobable the average tunneling current of a 2-input NAND gate is $I_{ox_{NAND_i}} = \left(\frac{I_{ox\,00}+I_{ox\,01}+I_{ox\,10}+I_{ox\,11}}{4}\right)$. The gate oxide direct tunneling current is obtained by evaluating diffusion, channel and body components of the PMOS and NMOS devices from the SPICE model and summing them as: $\sum_{MOS_i}(|I_{gs_i} + I_{gd_i} + I_{gcs_i} + I_{gcd_i} + I_{gb_i}|)$. In summary, we account for the *tunneling current of both NMOS and PMOS devices for both their ON and OFF states.*

dissipation and propagation delay of the functional units only. This is because the phenomenon of oxide tunneling current is restricted to the active devices and does not have any influence on the power dissipation in the interconnects. We calculate the direct tunneling current ($I_{ox_{FU}}$) of a n-bit functional unit as,

$$I_{ox_{FU}} = \sum_{i=1}^{n_{total}} I_{ox_{NAND_i}}, \quad (1)$$

where, $I_{ox_{NAND_i}}$ is the *average gate oxide tunneling current* dissipation of the i^{th} 2-input NAND gate in the functional unit assuming all states to be equiprobable (refer to Fig. 1). Similarly, the propagation delay of an n-bit functional unit is,

$$T_{pd_{FU}} = \sum_{i=1}^{n_{cp}} T_{pd_{NAND_i}}. \quad (2)$$

We followed the approach presented in [5] for logic level characterization. For the present experiments we characterized a library of 16-bit datapath components following the structural descriptions from [15]. The results of the characterization for a $45nm$ technology are presented in Table I.

III. Simultaneous Scheduling and Binding

Behavioral synthesis involves various steps: compilation, transformation, datapath scheduling, resource allocation, operation binding, connection allocation and architecture generation. Resource or time constrained scheduling time stamps the variables and operations in the DFG so that the operations in the same group can be executed concurrently. While allocation fixes the number and types of resources to be used in the datapath, the binding process involves attaching operations to functional units and variables to memory units. With a simultaneous scheduling and binding approach we can achieve more flexibility while binding resources to operations. We state the research problem as follows:

Given an unscheduled DFG $G(V, E)$, it is required to find the scheduled DFG with appropriate resource binding such that the total tunneling current is minimized and resource constraints (silicon cost) and latency constraints (circuit performance) are satisfied.

We assume that the datapath is specified as a sequencing DFG, which is a directed acyclic DFG. Each vertex of the DFG represents an operation and each edge represents a dependency. Also, each vertex has attributes that specify the operation type. The delay of a control step is dependent on the delays of the functional unit, the multiplexer, and register. We assume that each node connected to the primary input is assigned two registers and one multiplexer while the inner nodes of the DFG have one register and one multiplexer. It is also assumed that the inputs to the algorithm are an unscheduled DFG, a resource constraint matrix and a delay trade-off factor (T_F) and the output is an RTL description that satisfies the resource constraint and tradeoff factor and minimizes the tunneling current.

We can then state the optimization problem as follows: Let V be the set of all vertices and V_{CP} be the set of vertices in the critical path from the source S of the DFG to the output or sink node D. For simplicity, we assume that the DFG has a single source node and a single sink node (the same approach can be extended to multiple source multiple sink graphs). The cost (resource constrained) and performance (latency constrained) driven tunneling current minimization problem can thus be formulated as:

$$Minimize \quad \sum_{v_i \in V} I_{ox}(v_i), \quad (3)$$

where, $I_{ox}(v_i)$ is the tunneling current consumed per sample node v_i of the DFG, such that the following resource and latency constraints, respectively, are satisfied:

$$\sum_{v_i \in V_P} D_i(v_i) \leq D_{CP} \quad [\forall\, v_i \in V_P \ \ (\text{where, } V_P \subset V)] \quad (4)$$

$$Allocated\left(FU_i\left(k, T_{ox}\right)\right) \leq Available\left(FU_i\left(k, T_{ox}\right)\right) \quad (5)$$

The constraints in Eqn. (4) ensure that the sum of all delays $D_i(v_i)$ is less than the critical path delay D_{CP}. The resource allocation is summarized in Eqn. (5), where the total allocation of the i^{th} resources (functional units) of type k and consisting of transistors of oxide thickness T_{ox} denoted as $(FU_i\,(k, T_{ox}))$ should be less than the total number of corresponding resources available.

We present the proposed heuristic algorithm in Fig. 2. The algorithm first performs a resource constrained ASAP schedule C_S and a resource constrained ALAP schedule C_L and identifies the set of nodes with zero (V_Z) and non-zero (V_{NZ}) mobility. With the available resources (R_{Avl}) we can achieve maximum tunneling current reduction when we assign resources with higher thickness (T_{ox_H}) to as many nodes as possible. Since the algorithm has no prior knowledge of the critical path, to begin with it assigns T_{ox_H} resources to zero mobility nodes. For each allowable clock cycle of the non-zero mobility nodes the algorithm identifies the clock cycle that can allocate a T_{ox_H} resource and schedules the node to that clock cycle and binds its operation with that T_{ox_H}

TABLE I

DATAPATH COMPONENT LIBRARY FOR VARIOUS GATE OXIDE THICKNESS FOR $45nm$ TECHNOLOGY

Datapath Components Library	Effective Gate Oxide Thickness (T_{ox})							
	$T_{ox} = 1.4nm$		$T_{ox} = 1.5nm$		$T_{ox} = 1.6nm$		$T_{ox} = 1.7nm$	
	$I_{ox}(\mu A)$	$T_{pd}(ns)$	$I_{ox}(\mu A)$	$T_{pd}(ns)$	$I_{ox}(\mu A)$	$T_{pd}(ns)$	$I_{ox}(\mu A)$	$T_{pd}(ns)$
Adder	1.765620	27.916601	0.686630	36.384698	0.305810	42.291999	0.138480	46.821900
Subtractor	1.973340	27.916601	0.767410	36.384698	0.340430	42.291999	0.155790	46.821900
Multiplier	23.622379	44.484201	9.185840	57.986599	4.131320	67.395603	1.869480	74.622100
Divider	36.397161	151.164796	14.153810	197.036105	6.364310	229.007296	2.885000	253.557994
Comparator	4.189020	35.860901	1.627140	46.744299	0.732790	54.329698	0.328890	60.149698
Register	1.402110	32.679299	0.542380	42.602398	0.242340	49.508801	0.109630	54.824400
Multiplexer	1.194390	1.581100	0.461600	2.066100	0.207720	2.405600	0.092320	2.657800

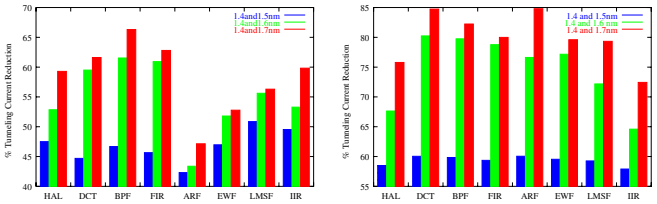
(01) Get resource constrained ASAP schedule C_S of the DFG.
(02) Get resource constrained ALAP cchedule C_L of the DFG.
(03) Fix the number of control steps as maximum of the ASAP and ALAP schedules.
(04) Initialize R_{Avl} matrix to given resource constraint.
(05) FOR each vertex $v_i \in G(V, E)$
(06) IF ($C_S[v_i]$ == $C_L[v_i]$) then
(07) $V_Z[i] = TRUE$. // Mark v_i zero mobile.
(08) ELSE
(09) $V_{NZ}[i] = TRUE$. // Mark v_i as nonzero mobile.
(10) FOR each vertex $v_i \in V_Z$ in ASAP order
(11) $C[v_i] = C_S[v_i]$. // v_i is ASAP time stamped
(12) IF ($FU_j\left(k, T_{ox_H}\right)$ is available in R_{Avl} for control step $C[v_i]$) then
(13) $R_{Bin}[k][T_{ox_H}][r_j]$ = Bind(v_i). // v_i needs type-k
(14) $R_{Avl}[C[v_i]][k][T_{ox_H}]$ $--$. // Resource allocated
(15) ELSE
(16) $R_{Bin}[k][T_{ox_L}][r_j]$ = Bind(v_i). // v_i needs type-k
(17) $R_{Avl}[C[v_i]][k][T_{ox_L}]$ $--$. // Resource allocated
(18) FOR each $v_i \in V_{NZ}$ in ASAP order
(19) FOR each possible cycle $c : C_S[v_i] \rightarrow C_L[v_i]$
(20) IF ($FU_j\left(k, T_{ox_H}\right)$ is available in R_{Avl} for c) then
(21) Schedule v_i in step c, $C[v_i] = c$. //
(22) $R_{Bin}[k][T_{ox_H}][r_j]$ = Bind(v_i). // v_i needs type-k
(23) $R_{Avl}[C[v_i]][k][T_{ox_H}]$ $--$. Break. // Allocated
(24) ELSE IF ($FU_j\left(k, T_{ox_L}\right)$ is available in R_{Avl} for c and v_i is not already scheduled)
(25) Schedule v_i in step c, $C[v_i] = c$.
(26) $R_{Bin}[k][T_{ox_L}][r_j]$ = Bind(v_i). // v_i needs type-k
(27) $R_{Avl}[C[v_i]][k][T_{ox_L}]$ $--$. // Resource Allocated
(28) Call Floyd's algorithm to determine V_{CP}.
(29) Calculate critical path delays $T_{CP_{ST}}$ and $T_{CP_{DT}}$.
(30) FOR each $c \in C$
(31) IF ($T_{CP_{DT}} > T_F \times T_{CP_{ST}}$) then // Time constraint
(32) Find all vertices in control step c.
(33) IF ($v_i \in V_{CP}$ and using T_{ox_H} resource) then
(34) IF ($v_j \notin V_{CP}$ and using T_{ox_L} resource) then
(35) Swap the resources of v_i and v_j.
(36) ELSE
(37) Assign T_{ox_L} resource from R_{Avl}.

Fig. 2. Simultaneous scheduling and binding heuristic

resource. If such a clock cycle is not available, then it allocates a T_{ox_L} resource in the ASAP clock cycle and schedules it to that clock cycle. Once the scheduling, allocation and binding is done, the critical path delay is calculated. If the critical path delay violates the trade-off factor then the algorithm tries to assign T_{ox_L} resources by swapping the resources of the critical path node and the node containing T_{ox_L} resources in that clock cycle. If there is no T_{ox_L} resource in that clock cycle the algorithm assigns a T_{ox_L} resource from the available resources.

IV. RESULTS AND CONCLUSIONS

The algorithm was implemented and tested with a number of benchmarks for various resource-time and time constraints. While calculating the gate leakage for single thickness, we used a nominal $1.4nm$ thickness. We considered resources of three dual thickness pairs of (i) $1.4nm - 1.5nm$, (ii) $1.4nm - 1.6nm$, and (iii) $1.4nm - 1.7nm$. For each benchmark and for each pair of dual thicknesses, we ran four sets of experiments. In the first experiment, we used a smaller number of T_{ox_H} resources and a higher number of T_{ox_L} resources. In the second experiment we used a higher number of T_{ox_H} resources as compared to the first experiment and in the third experiment we used a higher number of T_{ox_H} resources as compared to the second experiment. In the fourth experiment we relaxed the resource constraints to study the time constrained approach only. The percentage reduction in direct tunneling current is calculated as $\Delta I = \left(\frac{I_{ox\,ST} - I_{ox\,DT}}{I_{ox\,ST}}\right) * 100\%$. We estimate the critical path delay of the circuit as the sum of the delays of the vertices in the longest path of the DFG.



(a) Average percentage reduction time constrained approach for $T_F = 1.5$ with resource constraints

(b) Percentage reduction for time constrained approach for $T_F = 1.5$ without resource constraint

Fig. 3. Bar charts showing % reduction in gate oxide tunneling current for three cases (i)$1.4nm - 1.5nm$, (ii)$1.4nm - 1.6nm$ and (iii)$1.4nm - 1.7nm$

The results for various benchmarks for dual thicknesses for $1.4nm - 1.7nm$ are reported in Table II for both resource-time constrained and time constrained approaches. We observe that as T_F increases the reduction in gate leakage increases for all the benchmarks. The bar charts in Fig. 3 represent the experimental results for all the thicknesses pairs under consideration for a tradeoff factor of 1.5 and provide a comparative view of the approaches with and without resource constraint. This shows that for a particular performance requirement with resource constraint in effect there is a definite gain in the leakage reduction with increase in the thickness of the gate oxide. When compared with the resource constrained approach

TABLE II

TUNNELING CURRENT AND CRITICAL PATH DELAY OF CIRCUITS FOR VARIOUS BENCHAMARKS ($T_{ox_L} = 1.4nm$ AND $T_{ox_H} = 1.7nm$)

Bench -mark	Res Con	$I_{ox_{ST}}$ (μA)	$T_{CP_{ST}}$ (ns)	Gate Oxide Tunneling Current ($I_{ox_{DT}}$) and Percentage Reduction (ΔI)											
				T_F=1.0		T_F=1.1		T_F=1.2		T_F=1.3		T_F=1.4		T_F=1.5	
				$I_{ox}(\mu A)$	ΔI	$I_{ox}(\mu A)$	ΔI	$I_{ox}(\mu A)$	ΔI	$I_{ox}(\mu A)$	ΔI	$I_{ox}(\mu A)$	ΔI	$I_{ox}(\mu A)$	ΔI
H	1	86.14	329.64	160.25	18.2	160.25	18.2	156.04	20.4	151.83	22.5	151.83	22.5	127.67	34.8
A	2	86.14	329.64	90.68	53.7	90.68	53.7	86.46	55.9	82.25	58.0	82.25	58.0	58.10	70.3
L	3	86.14	329.64	86.30	55.9	86.30	55.9	82.09	58.1	77.87	60.2	77.87	60.2	537.29	72.6
(1)	∞	86.14	329.64	80.05	59.1	80.05	59.1	75.84	61.3	71.62	63.4	71.62	63.4	47.48	75.7
D	1	162.43	823.89	384.04	21.6	380.02	22.4	376.00	23.3	351.85	28.2	347.83	29.0	319.66	34.7
C	2	162.43	823.89	187.48	61.7	183.46	62.5	179.44	63.3	155.29	68.3	151.27	69.1	123.10	74.8
T	3	162.43	823.89	186.02	62.0	182.00	62.8	177.98	63.6	153.83	68.6	149.81	69.4	121.64	75.1
(2)	∞	162.43	823.89	139.18	71.6	135.16	72.4	131.14	73.2	106.99	78.1	102.97	78.9	74.80	84.7
B	1	152.57	689.43	282.03	31.2	278.00	32.2	273.98	33.1	249.84	39.0	245.62	40.1	237.38	42.1
P	2	152.57	689.43	141.45	65.5	141.45	65.5	131.30	67.9	113.09	72.4	103.13	74.8	99.11	75.8
F	3	152.57	689.43	139.35	66.0	135.32	67.0	113.09	72.4	107.15	73.8	84.72	79.3	77.92	80.9
(3)	∞	152.57	689.43	113.29	72.3	109.27	73.3	85.12	79.2	80.91	80.2	76.89	81.2	72.87	82.2
F	1	159.32	500.64	151.50	49.1	147.48	50.4	143.46	51.8	135.42	54.5	131.39	55.8	123.35	58.5
I	2	159.32	500.64	133.96	55.0	133.96	55.0	119.31	59.9	111.27	62.6	109.81	63.1	109.81	63.1
R	3	159.32	500.64	127.36	57.2	123.33	58.5	109.81	63.1	109.81	63.1	107.25	63.9	99.20	66.6
(4)	∞	159.32	500.64	87.71	70.5	83.69	71.9	79.67	73.2	71.62	75.9	67.60	77.3	59.56	79.9
A	1	162.43	831.33	383.30	22.4	379.27	23.3	375.25	24.1	351.10	28.9	343.06	30.6	316.33	36.0
R	2	162.43	831.33	317.75	35.7	317.75	35.7	293.60	40.6	269.45	45.5	253.38	48.7	242.72	50.9
F	3	162.43	831.33	289.60	41.4	285.57	42.2	281.55	43.0	257.41	47.9	242.72	50.9	225.21	54.4
(5)	∞	162.43	831.33	139.54	71.7	135.51	72.6	131.49	73.4	107.35	78.2	103.32	79.1	75.15	84.8
E	1	262.07	557.80	280.66	15.4	272.62	17.8	244.45	26.3	236.41	28.7	208.24	37.2	204.22	38.4
W	2	262.07	557.80	216.26	34.8	200.17	39.6	172.00	48.1	163.96	50.5	143.82	56.6	143.82	56.6
F	3	262.07	557.80	208.22	37.2	192.11	42.0	167.96	49.3	143.82	56.6	135.79	59.0	131.77	60.2
(6)	∞	262.07	557.80	150.76	54.5	142.72	56.9	114.55	65.4	106.51	67.9	78.34	76.3	74.32	77.6
L	1	145.82	474.44	236.54	16.1	232.51	17.6	232.51	17.6	208.37	26.1	184.22	34.7	176.18	37.5
M	2	145.82	474.44	188.24	33.3	188.24	33.3	164.09	41.8	139.94	50.4	113.21	59.8	113.21	59.8
S	3	145.82	474.44	161.50	42.7	161.50	42.7	137.35	51.3	113.21	59.8	86.47	69.3	86.47	69.3
(7)	∞	145.82	474.44	118.68	57.9	114.66	59.3	114.66	59.3	90.51	67.9	66.37	76.4	58.32	79.3
I	1	76.29	273.42	87.64	46.1	87.64	46.1	83.62	48.5	79.60	51.0	79.60	51.0	75.58	53.5
I	2	76.29	273.42	83.62	48.5	83.62	48.5	79.60	51.0	75.58	53.5	75.58	53.5	71.56	56.0
R	3	76.29	273.42	60.91	62.5	60.91	62.5	56.89	65.0	52.86	67.5	52.86	67.5	48.84	69.9
(8)	∞	76.29	273.42	56.89	65.0	56.89	65.0	52.86	67.5	48.84	69.9	48.84	69.9	44.76	72.4

the time constrained method shows an average reduction of almost 15% more across all benchmarks.

We presented simultaneous scheduling and binding of behavioral level elements utilizing functional units of dual oxide thickness. As can be seen from the results of the experiments carried out in this context, the dual oxide thickness methodology is highly effective in reducing the gate leakage. The choice of functional units is being made during scheduling and we are in the process of evaluating its impact on area, capacitance and dynamic power. Also we are exploring the option of replacing the heuristic based approach with more advanced optimization techniques. Finally, we aim to extend this work on gate leakage to a holistic solution to the entire spectrum of power dissipation at the behavioral level, considering Multi-V_{dd}, Multi-V_{Th}, and Multi-T_{ox}.

REFERENCES

[1] K. Roy, S. Mukhopadhyay, and H. M. Meimand, "Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits," *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305–327, February 2003.

[2] "Semiconductor Industry Association, International Technology Roadmap for Semiconductors," `http://public.itrs.net`.

[3] M. Depas, B. Vermeire, P. W. Mertens, R. L. V. Meirhaeghe, and M. M. Heyns, "Determination of Tunneling Parameters in Ultra-Thin Oxide Layer Poly-Si/SiO_2/Si Structures," *Elsevier Solid-State Electronics Journal*, vol. 38, no. 8, pp. 1465–1471, August 1995.

[4] S. P. Mohanty, V. Mukherjee, and R. Velagapudi, "Analytical Modeling and Reduction of Direct Tunneling Current during Behavioral Synthesis of Nanometer CMOS Circuits," in *Proc of the 14th ACM/IEEE Intl Workshop on Logic and Synthesis (IWLS)*, 2005, pp. 249–256.

[5] V. Mukherjee, S. P. Mohanty, and E. Kougianos, "A Dual Dielectric Approach for Performance Aware Gate Tunneling Reduction in Combinational Circuits," in *Proceedings of the 23rd IEEE International Conference of Computer Design (ICCD)*, 2005.

[6] S. P. Mohanty and N. Ranganathan, "A Framework for Energy and Transient Power Reduction during Behavioral Synthesis," *IEEE Transactions on VLSI Systems*, vol. 12, no. 6, pp. 562–572, June 2004.

[7] A. K. Sultania, D. Sylvester, and S. S. Sapatnekar, "Tradeoffs Between Gate Oxide Leakage and Delay for Dual T_{ox} Circuits," in *Proceedings of Design Automation Conference*, 2004, pp. 761–766.

[8] N. Sirisantana and K. Roy;, "Low-power Design using Multiple Channel Lengths and Oxide Thicknesses," *IEEE Design & Test of Computers*, vol. 21, no. 1, pp. 56–63, Jan-Feb 2004.

[9] M. Johnson and K. Roy, "Datapath Scheduling with Multiple Supply Voltages and Level Converters," *ACM Transactions on Design Automation of Electronic Systems*, vol. 2, no. 3, pp. 227–248, July 1997.

[10] K. Roy and R. Krishnammthy, "Design of Low voltage CMOS circuits : Tutorial Guide," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2001, pp. 3.2.1–3.2.29.

[11] P. Pant, R. K. Roy, and A. Chattejee, "Dual-Threshold Voltage Assignment with Transistor Sizing for Low Power CMOS Circuits," *IEEE Trans on VLSI Systems*, vol. 9, no. 2, pp. 390–394, April 2001.

[12] K. S. Khouri and N. K. Jha, "Leakage power analysis and reduction during behavioral synthesis," *IEEE Transactions on VLSI Systems*, vol. 10, no. 6, pp. 876–885, December 2002.

[13] C. Gopalakrishnan and S. Katkoori, "Knapbind: an area-efficient binding algorithm for low-leakage datapaths," in *Proceedings of 21st International Conference on Computer Design*, 2003, pp. 430–435.

[14] K. A. Bowman, L. Wang, X. Tang, and J. D. Meindl, "A Circuit-Level Perspective of the Optimum Gate Oxide Thickness," *IEEE Transactions on Electron Devices*, vol. 48, no. 8, pp. 1800–1810, August 2001.

[15] N. H. E. Weste and D. Harris, *CMOS VLSI Design : A Circuit and Systems Perspective*, Addison Wesley, 2005.