Dual-K Versus Dual-T Technique for Gate Leakage Reduction : A Comparative Perspective

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

Ramakrishna Velagapudi
Computer Science & Engineering
University of North Texas
Denton, TX 76203.
Email:rv0063@unt.edu

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

Abstract

As a result of aggressive technology scaling, gate leakage (gate oxide direct tunneling) has become a major component of total power dissipation. Use of dielectrics of higher permittivity (Dual-K) or use of silicon dioxide of higher thicknesses (Dual-T) are being considered as methods for its reduction. This paper presents a comparative view of dual dielectric and dual thickness low leakage design techniques from a behavioral synthesis perspective. An algorithm is presented for the gate leakage current reduction that does simultaneous scheduling, allocation and binding during behavioral synthesis while accounting for process variations. The algorithm minimizes the gate leakage for given time constraints. We performed experiments for a number of benchmark circuits using a $45nm$ CMOS technology datapath library. We obtained gate leakage reduction as high as 95% for the dual-K (SiO_2 and Si_3N_4) and 91% for the dual-T ($1.4nm$ and $1.7nm$) approaches. It is observed that the dual-K approach outperformed the dual-T approach for all benchmark circuits.

1 Introduction

The advent of portable communication and computing services has promoted a great deal of interest in both commercial and research areas. Due to scaling and the consequent increase in leakage components in both active and sleep states power consumption of CMOS circuits has become an issue of major concern. As the systems become faster and smaller in size, the leakage power dissipation tends to increase exponentially [6, 1]. According to the ITRS, a high performance CMOS device will require gate oxide thicknesses of $0.7nm - 1.2nm$, thus leading to gate leakage due to carrier tunneling through the ultra thin layer of gate oxide [11]. Various options need to be explored for reduction of active leakage power dissipation in the advent of nanoscale CMOS technologies.

Tunneling current dissipation in a CMOS with a supply voltage V_{dd} and effective gate oxide thickness T_{ox}, is given by [6, 3], $I_{gate} \propto \left(\frac{V_{dd}}{T_{ox}}\right)^2 exp\left(-k\frac{T_{ox}}{V_{dd}}\right)$, where, k is an experimentally derived factor. This gives rise to two options for reduction of gate leakage: reduction of supply voltage and/or increase of the gate oxide thickness. The popular option of scaling down of supply voltage [8] continues to play its role in the reduction of dynamic power as well as leakage power, but it is not sufficient to control the exponential growth of gate leakage. Increase in the gate oxide thickness leads to an increase in propagation delay and area. Thus, the use of multiple gate oxide thicknesses can serve as a leakage current, performance and area trade-off. Recently, the "dual dielectric" approach is being introduced for logic level gate leakage reduction, where, SiO_2 is selectively replaced with high-K materials such as SiON, Si_3N_4, etc. In this paper we explore the three dimensional design space (gate leakage, performance, area) for reduction of gate direct tunneling current during behavioral synthesis considering (1) dual-K, and (2) dual-T.

2 Related Works and Our Contributions

Research in leakage optimization at the behavioral level has not been as intensive as in dynamic power reduction. In [5], Khouri and Jha proposed a dual-V_{Th} technique for subthreshold leakage analysis and reduction that targets the least used modules as the candidates for leakage optimization. Gopalakrishnan and Katkoori in [4] also use the multi-V_{Th} approach for reduction of subthreshold current and proposed binding algorithms for power, delay, and area trade-off. Mohanty et. al. [7] have presented analytical models and a datapath scheduling algorithm for reduction of gate leakage current.

We explore the dual dielectric (dual-K, SiO_2 and Si_3N_4) and dual thickness (dual-T, Low-T_{ox} and high-T_{ox}) approaches for total gate leakage reduction of datapath circuits. We minimize the gate tunneling current of a datapath circuit with a given time constraint using a simulated annealing based algorithm that performs simultaneous scheduling, allocation and binding. The algorithm considers multicycling and chaining based datapath to reduce delay penalty. It takes process variations into account while optimizing gate leakage. In the sub-$65nm$ CMOS technology the gate oxide thickness is very small, approximately $1.2nm$. However, a monolayer of SiO_2 is approximately $0.2nm$. Thus, a layer of SiO_2 misplacement can cause significant variation in the effective T_{ox}, and resultant gate leakage, and hence this effect needs to be accounted for. We also explore the use of dual-K (SiO_2 and Si_3N_4) approach as alternative to dual-T approach. We also present analytical functions for the calculation of gate leakage, delay and area of nano-CMOS based architectural units.

3 Low Gate Leakage Datapath Library

In this section we present a datapath component library that will be used during behavioral synthesis. We constructed functional units (adder, subtractor, multiplier, divider, comparator, register and multiplexer) using NAND logic gates. We chose the NAND gate for two reasons: it is a universal gate and its gate leakage is minimal compared to other gates [10]. We used BPTM BSIM4 model for simulation to estimate gate current (I_{gate}) and propagation delay (T_{pd}). On the other hand, due to unavailability of silicon data we used an analytical formula for area calculations.

We followed the logic level characterization methodology from [10]. If the four possible states (00, 01, 10 and 11) have gate tunneling current ($I_{00}, I_{01}, I_{10}, I_{11}$), respectively, and assuming that all four states are equiprobable, the average gate tunneling current of a 2-input NAND gate is $I_{gate_{NAND_i}} = \left(\frac{I_{00}+I_{01}+I_{10}+I_{11}}{4}\right)$. The gate oxide direct tunneling current is obtained by evaluating diffusion, channel and body components for a PMOS or NMOS device from the BPTM and summing them as: $\sum_{MOS_i}(|I_{gs_i} + I_{gd_i} + I_{gcs_i} + I_{gcd_i} + I_{gb_i}|)$ to account for the ON as well as the OFF state gate leakage for both NMOS and PMOS devices. The area of a NAND gate is calculated using the following expression [2]:

$$A_{NAND} = k_{inv}\left(1 + 4\left(n_{in} - 1\right)\sqrt{\frac{AR_{NAND}}{k_{inv}}}\right) \times \left(1 + \frac{\left(\frac{W_{NMOS}}{f} - 1\right)(1+\beta_{NAND})}{\sqrt{k_{inv} AR_{NAND}}}\right).$$

Here, W_{NMOS} is the width of the NMOS, f is the minimum feature size for a technology, k_{inv} is the area of minimum size inverter with respect to f^2, AR_{NAND} is the aspect ratio of NAND gate, n_{in} is the number of inputs, and β_{NAND} is the ratio of PMOS width to NMOS width.

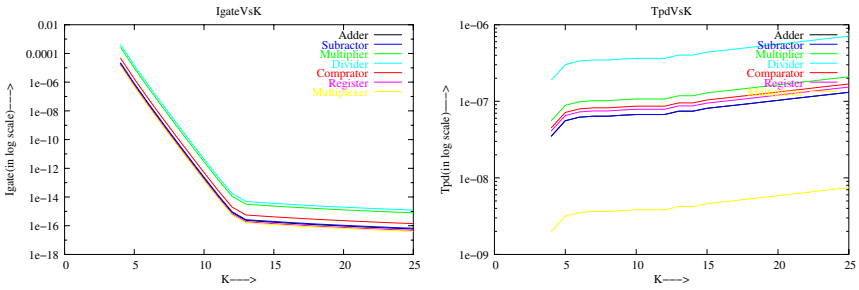
We assume that in a n-bit functional unit there are total n_{total} NAND gates out of which n_{cp} are in the critical path. We calculate the direct tunneling current ($I_{gate_{FU}}$) of a n-bit functional unit as $I_{gate_{FU}} = \sum_{i=1}^{n_{total}} I_{g_{NAND_i}}$, where $I_{g_{NAND_i}}$ is the *average gate oxide tunneling current* dissipation of the i^{th} 2-input NAND gate in the functional unit, assuming all states to be equiprobable. Similarly, the propagation delay and silicon area of a n-bit functional unit are $T_{pd_{FU}} = \sum_{i=1}^{n_{cp}} T_{pd_{NAND_i}}$ and $A_{FU} = \sum_{i=1}^{n_{total}} A_{NAND_i}$, respectively. In this model we do not consider the effect of interconnect wires and focus on the direct tunneling current dissipation and propagation delay of the active units only. The phenomenon of oxide tunneling current is restricted to the active devices and does not contribute to power dissipation in the interconnect.

Analytical Functions in terms of K and T_{ox}: In order to facilitate the optimization during behavioral synthesis we need to describe the characterization data obtained as functions of K and T_{ox}. We used the calculated and simulated data to fit different analytical functions for gate leakage, area and delay in terms of dielectric constants and oxide thickness. The results are presented in Table 1 and Table 2, respectively. Curve fitting results for the leakage current and delay in terms of the dielectric constant is shown in Fig. 1 and the leakage current, delay and area in terms of oxide thickness is shown in Fig. 2. The functions obtained have a correlation coefficient of approximately 0.99. Thus, the curves faithfully represent the data.

4 Gate Leakage Optimization

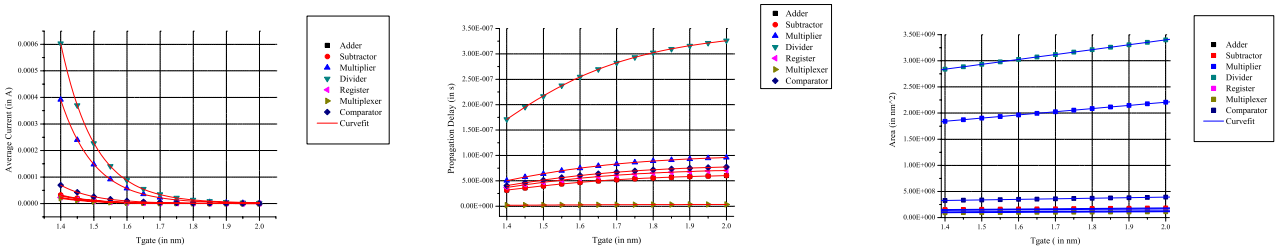
In this section we present a simulated annealing approach based on the algorithm in [9] that performs simultaneous scheduling, allocation and binding and minimizes the gate leakage current. Given a reduction mechanism (either dual-K or dual-T) and time constraint we need to determine an RTL implementation that has minimum leakage current. In both approaches, in order to maximize the leakage reduction we need to ensure that every node can be scheduled in such a way that a less leaky resource can be assigned to non critical resources so that the total delay is not effected.

Simulated annealing algorithms borrow ideas from Materials Science. Annealing is the process of heating and cooling a material slowly until it crystalizes. The atoms of this material have higher energies at very high temperatures. This gives the atoms a great deal of freedom in their ability to reconstruct themselves. As the temperature decreases the energy of the atoms decrease. Analogous to the annealing process, the mobility of nodes in a DFG (data flow graph representing data path circuit) is dependent on the total available resources. Here the nodes of a DFG are analo-



(a) Gate Leakage (b) Propagation Delay

Figure 1. Variation of gate leakage and propagation delay of different architectural units with K.



(a) Gate Leakage (b) Propagation Delay (c) Area

Figure 2. Variation of gate leakage, delay and area of different architectural units with T_{ox}.

Table 3. Notations used in Description

$V\{v_i\}$	Set of nodes or vertices in $UDFG$
LRM	The Leakage Reduction Mechanism used. It can be either dual-T_{ox} or dual-K approach.
$C[v_i]$	Final time stamp of a vertex v_i
$FU_j\ (m, T_{ox})$	j^{th} resource of type m and thickness T_{ox}
RH	Higher leaky resource. Can be either a T_{ox_L} or a K_L resource.
RL	Less leaky resource. Can be either a T_{ox_H} or a K_H resource.
T_{ox_H}	Higher gate oxide thickness
T_{ox_L}	Lower gate oxide thickness
K_H	Higher dielectric
K_L	Lower dielectric
$R_{Avl}[..c..][..k..][..T_{ox}..]$	Availability matrix; c is any clock cycle
DT_F	Performance or delay trade-off factor
$T_{CP_{ST}}$	Critical path delay for single oxide thickness
$T_{CP_{DT}}$	Critical path delay for dual oxide thickness
$I_{gate_{ST}}$	Tunneling current for single oxide thickness
$I_{gate_{DT}}$	Tunneling current for dual oxide thickness
A_{ST}	Total Area for single oxide thickness
A_{DT}	Total Area for dual oxide thickness
S	Scheduled DFG with resource binding

gous to the atoms and temperature is analogous to the total number of available resources. The mobility of the nodes (chance for assigning a higher oxide thickness resource or higher dielectric resource) is dependent on the total number of available less leaky resources. We apply the annealing principle to our problem and explore the trade-offs between power, performance and area.

The input to this algorithm is a DFG, input data streams, gate leakage reduction mechanism (either dual-K or dual-T) and the output is an RTL description with gate leakage. The algorithm is presented in Fig. 3 and the notations used in the algorithm are presented in Table 3. It starts with the ASAP schedule and assigns RH (leaky resources, lower thickness resources for dual-T approach and lower dielectric for dual-K approach) to all the operations. This is done by the function Allocate_Bind. The total leakage is determined as the sum of leakages of all the allocated resources, so the minimum number of resources required for the schedule is determined and allocated. Once the execution in a clock cycle is finished all the resources are assumed to be in the ready state before running the next clock cycle.

In the outer loop during each iteration the number of less leaky resources RL (higher thickness resources for dual-T approach and higher dielectric constant for dual-K approach) is decreased, which restricts the mobility of the nodes. The algorithm attempts to find an RTL that has minimum leakage for a given number of available resources. In the inner loop during each iteration a neighborhood solution is generated (Fig. 4). If this solution has less leakage than the current solution, the neighborhood solution is made the current solution. This way the algorithm converges to a solution that has minimum leakage. In generating a neighborhood solution we randomly select a node and check if a less leaky resource can be assigned in all possible clock

Table 1. Gate Leakage and Delay as Analytical Functions of K

Functional Unit	$I_{gate}(\mu A) = A * e^{\left(\frac{-K}{\alpha}\right)} + I_{gate_0}$			$T_{pd}(ns) = \begin{cases} A_2 + \frac{A_1 - A_2}{\left(1+e^{\left(\frac{K-K_0}{dK}\right)}\right)}, & 2.5 \leq K < 6 \\ C * e^{\left(\frac{-K}{\alpha}\right)} + T_{pd_0}, & 6 \leq K < 30 \end{cases}$					
	I_{gate_0}	A	α	A_1	A_2	K_0	β	α	T_{pd_0}
Adder	-1.20	2.53	0.36	5.06	63.22	4.02	0.47	-10.63	0.37
Subtractor	-1.34	2.83	0.36	5.06	63.22	4.02	0.47	-10.63	0.37
Multiplier	-16.10	33.81	0.36	8.07	100.74	4.02	0.47	-10.63	0.37
Divider	-24.80	52.08	0.36	27.43	342.32	4.02	0.47	-10.63	0.37
Register	-0.95	2.00	0.36	5.93	74.01	4.02	0.47	-10.63	0.37
Multiplexer	-0.81	1.70	0.36	0.28	3.59	4.02	0.47	-10.63	0.37
Comparator	-2.85	5.99	0.36	6.50	81.21	4.02	0.47	-10.63	0.37

Table 2. Gate Leakage, Delay and Area as Analytical Functions of T_{ox} (in nm)

$I_{gate}(\mu A) = A\, e^{(-T_{ox}/\alpha)} + B$				$T_{pd}(ns) = \frac{(A_1 - A_2)}{(1+(T_{ox}/\beta)^\gamma)} + A_2$				$A(nm^2) = \alpha T_{ox} + \beta$	
Unit	α	A	B	A_1	A_2	β	γ	α	β
Adder	0.10	24.82	0.08	-7.37	64.47	1.36	7.24	0.45×10^8	0.74×10^8
Subtractor	0.10	27.76	0.09	-7.37	64.47	1.36	7.24	0.50×10^8	0.83×10^8
Multiplier	0.10	331.62	1.09	-11.75	102.74	1.36	7.24	6.07×10^8	9.92×10^8
Divider	0.10	510.89	1.68	-39.93	349.12	1.36	7.24	9.35×10^8	15.28×10^8
Register	0.10	19.70	0.06	-8.63	75.48	1.36	7.24	0.36×10^8	0.58×10^8
Multiplexer	0.10	16.77	0.05	-0.41	3.66	1.36	7.24	0.30×10^8	0.50×10^8
Comparator	0.10	58.83	0.19	-9.47	82.82	1.36	7.24	1.07×10^8	1.76×10^8

Simulated_Annealing_Algorithm(UDFG, DTF, LRM)
(01) Initial Temperature $\leftarrow t_o$
(02) Available Resources $\leftarrow \infty$
(03) While there exists a schedule with available resources.
(04) i = Number of iterations
(05) Perform resource constrained ASAP
(06) Perform resource constrained ALAP
(07) Initial Solution $\leftarrow$ ASAP Schedule
(08) $S \leftarrow$ Allocate_Bind()
(09) Initial Cost $\leftarrow$ Cost(S)
(10) While ($i > 0$)
(11) Generate a random thicknesses in range of $(T_{ox} - \delta T_{ox}, T_{ox} + \delta T_{ox})$
(12) Generate random transition from S to S^*
(13) $\Delta_I \leftarrow$ Tunneling(S) $-$ Tunneling(S^*)
(14) if($\Delta_I > 0$) then $S \leftarrow S^*$
(15) else if($e^{\Delta_I / t}$ >random[0,1)) then $S \leftarrow S^*$
(16) $i \leftarrow i - 1$
(17) end While
(18) Decrement available resources
(19) $t \leftarrow \alpha \times t$
(20) end While
(21) return S

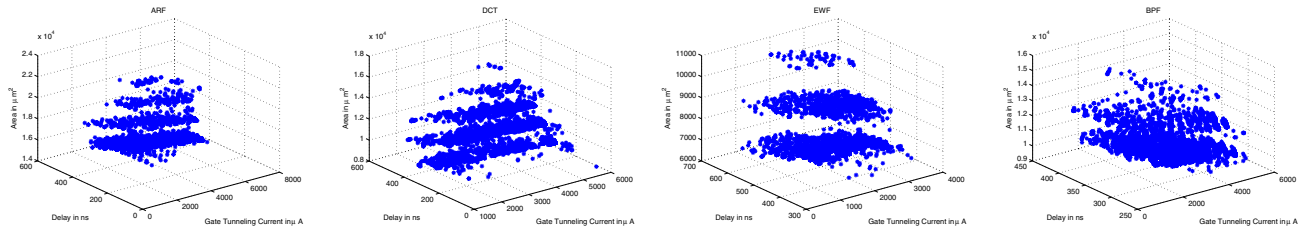
Figure 3. Simulated Annealing based algorithm for minimizing the cost function.

cycles and that it satisfies a time constraint. Each time a different resource is assigned a random thickness in the range of $(T_{ox} - \delta T_{ox}, T_{ox} + \delta T_{ox})$ to take process variation into account. Assuming a monolayer misplacement of SiO_2, δT_{ox} is approximately 15%.

For calculating the total delay of the circuit for a single cycle case we used the critical path delay. While generating a random neighborhood solution for a dual dielectric approach the algorithm ensures that the nodes in the critical path are not assigned a higher K resource, In the case of dual thickness, the algorithm ensures that the nodes in the critical path are not assigned a higher T_{ox} resource. For multicycling, the total delay of the circuit is calculated as the product of total number of control steps and the maximum delay of any resource in the circuit. Assigning higher thickness/dielectric resources will increase the delay which can be compensated using chaining and multicycling. Using multicycling or chaining alone may not be enough. While multicycling increases the number of control steps there were only few operations for which chaining can be implemented. The idea behind using both multicycling and chaining is to ensure that the execution of any operation that is ready (all its predecessors finished execution) and has a resource available will start execution.

5 Experimental Results

We implemented the overall design flow using C under UNIX environment and integrated it into the high-level synthesis framework in [8]. We ran the algorithm with various benchmark circuits [8] for both dual-K and dual-T approaches. For the dual thickness approach we calculated the tunneling current for single thickness of $1.4nm$ as the base case value from the BSIM4.4.0 model. Similarly, for the dual dielectric approach we calculated the base case tunneling current using SiO_2 as the dielectric. We considered a



(a) (b) (c) (d)

Figure 5. Gate leakage - propagation delay - area design space exploration for single cycle operation.

Table 4. Experimental Results

	D	Dual-T (K = 3.9, SiO_2)			Dual-K (T_{ox} = 1.4nm)		
	T	$I_{gate_{DT}}$	$T_{CP_{DT}}$	ΔT	$I_{gate_{DK}}$	$T_{CP_{DK}}$	ΔT
	F	(μA)	(ns)		(μA)	(ns)	
	Base Case : I_{gate} = 6618.2, T_{CP} = 308.9						
A	1.0	1628.1	308.9	75.3	1321.7	308.9	80.0
R	1.1	1600.5	329.4	75.8	1321.7	308.9	80.0
F	1.2	1231.5	362.2	81.3	930.1	360.4	85.9
	1.3	890.2	374.4	86.5	930.1	360.4	85.9
	1.4	890.2	374.4	86.5	871.5	424.9	86.8
	Base Case : I_{gate} = 5222.4, T_{CP} = 290.1						
B	1.0	1215.8	290.1	76.7	969.8	290.1	81.4
P	1.1	1184.9	310.7	77.3	969.8	290.1	81.4
F	1.2	815.9	343.5	84.3	578.1	341.7	88.9
	1.3	788.3	364.0	84.9	574.7	373.9	88.9
	1.4	754.1	405.2	85.5	512.6	406.2	90.1
	Base Case : I_{gate} = 5941.6, T_{CP} = 308.9						
D	1.0	1644.2	308.9	72.3	1380.3	308.9	76.7
C	1.1	1589.0	308.9	73.2	1351.0	308.9	77.2
T	1.2	1330.5	341.7	77.6	1351.0	308.9	77.2
	1.3	1330.5	341.7	77.6	1047.3	360.4	82.3
	1.4	1275.2	362.2	78.5	959.4	392.7	83.8
	Base Case : I_{gate} = 3895.4, T_{CP} = 498.4						
E	1.0	1636.2	498.4	57.9	1497.5	498.4	61.5
W	1.1	1267.2	531.2	67.4	1468.2	530.6	62.3
F	1.2	870.6	584.5	77.6	1076.6	582.2	72.3
	1.3	815.4	646.2	79.0	684.9	633.7	82.4
	1.4	815.4	646.2	79.0	684.9	633.7	82.4
	Base Case : I_{gate} = 3572.9, T_{CP} = 303.0						
F	1.0	796.7	282.4	77.6	626.3	303.0	82.4
I	1.1	769.1	323.5	78.4	626.3	303.0	82.4
R	1.2	741.5	344.1	79.2	234.7	354.5	93.4
	1.3	400.1	356.3	88.7	205.3	386.8	94.2
	1.4	372.5	418.0	89.5	176.0	419.0	95.0
	Base Case : I_{gate} = 2543.5, T_{CP} = 163.8						
H	1.0	946.9	163.8	62.7	848.9	163.8	66.6
A	1.1	946.9	163.8	62.7	848.9	163.8	66.6
L	1.2	916.0	184.4	63.9	816.1	196.0	67.9
	1.3	578.0	196.6	77.2	816.1	196.0	67.9
	1.4	547.1	217.1	78.4	457.3	215.3	82.0
	Base Case : I_{gate} = 2075.5, T_{CP} = 145.0						
I	1.0	571.9	145.0	72.4	479.6	145.0	76.8
I	1.1	571.9	145.0	72.4	479.6	145.0	76.8
R	1.2	544.3	165.6	73.7	479.6	145.0	76.8
	1.3	203.0	177.8	90.2	450.3	177.3	78.3
	1.4	175.3	198.4	91.5	88.0	196.6	95.7
	Base Case : I_{gate} = 3759.4, T_{CP} = 277.3						
L	1.0	1435.0	244.5	61.8	1292.3	277.3	65.6
M	1.1	1407.4	297.9	62.5	1292.3	277.3	65.6
S	1.2	1038.4	330.6	72.3	900.7	328.8	76.0
	1.3	697.0	342.8	81.4	900.7	328.8	76.0
	1.4	641.8	384.0	82.9	509.0	380.4	86.4
	Overall Average Reductions			76.8			79.5

Generate_Neighborhood(S, DTF, $CellLibrary$, LRM)
(01) Select a random vertex $v_i \in V$
(02) FOR each possible cycle $c : C_S[v_i] \rightarrow C_L[v_i]$
(03) IF (FU_j (m, RL) is available in R_{Avl} for c) then
(04) Schedule v_i in step c, Assign c to $C[v_i]$
(05) $R_{Bin}[m][RL][r_j]$ = Bind(v_i) /*v_i needs type-m*/
(06) Decrement $R_{Avl}[C[v_i]][m][RL]$ /*Allocated*/
(07) Increment $R_{Avl}[C[v_i]][m][RH]$/*RH resource freed*/
(08) TotalDelay=0 /*Initialize Delay*/
(09) While($\forall$ $v_i \in V$ execution of v_i is not done)
(10) For each $v_i \in V$
(11) If(All predecessors of v_i finished execution and v_i has not yet started execution and required resource is available) then
(12) start executing v_i
(13) For each $v_i \in V$
(14) If(v_i started execution and not yet finished)
(15) var=v_i, break /*var–node to be executed*/
(16) Increment totaldelay by Delay(var, FU_j (m, RL))
(17) For each $v_i \in V$
(18) If(v_i started execution and not yet finished)
(19) Execute v_i for a period of Delay(var,FU_j (m, RL))
(20) If(v_i finished execution))
(21) then mark v_i as completed /*executed*/

Figure 4. Algorithm to generate random transition of a Multi-cycling and chaining datapath. Algorithm to single cycle datapath is a special case of the multi-cycling situation.

dual dielectric pair SiO_2(K=3.9) - Si_3N_4(K=7) for dual-K approach (the gate oxide thickness for both K_H and K_L resources is a constant $1.4nm$). For dual-T approach we considered thicknesses $1.4nm - 1.7nm$. To start with, we assumed an infinite number of RL and RH resources and during each iteration we decreased the number of RL resources. We performed the experiments for both multicycling and chaining based datapath as well as single cycle datapath circuits.

The results take into account the gate leakage, area and propagation delay of functional units, interconnect units, and storage units present in the datapath circuit. The subscripts ST and DT stand for single thickness and multiple thickness, respectively. For a dual-T approach the percentage reduction in direct tunneling current is calculated as

$\Delta I = \left(\frac{I_{gate_{ST}} - I_{gate_{DT}}}{I_{gate_{ST}}}\right) * 100\%$ and the percentage area overhead is calculated as $\Delta A = \left(\frac{-(A_{ST} - A_{DT})}{A_{ST}}\right) * 100\%$. For a dual dielectric approach reduction in direct tunneling current is calculated as $\Delta I = \left(\frac{I_{gate_{SK}} - I_{gate_{DK}}}{I_{gate_{SK}}}\right) * 100\%$. 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 for single cycle case and number of control steps times the slowest delay resource for multicycling-chaining case. The delay trade-off factor (DTF) is used to provide various time constraints for our experiments. T_{CP} is used to denote the critical path delay of the circuit.

For a dual-T we had the area estimate and we obtained a number of design alternatives in the 3-D space during each iteration of our algorithm. We plotted the design space for some benchmarks as shown in Fig. 5. We observed that the extent to which tunneling current reduction takes place increases as the number of available T_{ox_H} resources increase. We selected the design so as to minimize tunneling current. The results for single cycle datapath circuits for various benchmarks for dual thickness technique for $1.4nm - 1.7nm$ and a dual dielectric technique for $3.9 - 7$ are reported in Table 4. For the dual-K approach the reduction in tunneling current for all the benchmarks ranges from 57.9% to 91.5% for an area penalty ranging from 6.6% to 34.8% for different delay trade-off factors considered in the experiment. For the dual dielectric approach it ranges from 61.5% to 95.7%. It can be seen from the results table that the area penalty increases with tunneling current reduction for a dual thickness approach. The variations in area with the dual-K approach is due to the effect of allocating more number of resources.

For multicycling and chaining datapath with a dual-T approach we observed gate leakage reduction ranging from 30.5% to 91.1% for an area penalty ranging from 4.80% to 28.67% for different delay trade-off factors. One significant observation is that there is a drastic reduction in delay compared to single cycle operation. The design space exploration for multicycling and chaining datapath is very much similar to the single cycle case.

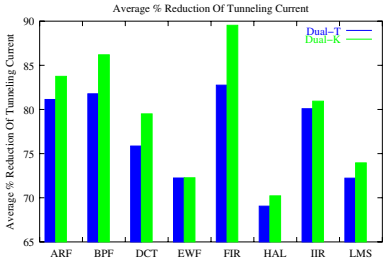


Figure 6. Dual$K$ Vs dual$T$ for Benchmarks.

The comparison between a dual-K and a dual-T approach is shown in Fig. 6. We observed that a dual-K approach gives better tunneling reduction compared to the dual-T approach for all benchmark circuits.

6 Conclusions

In this paper we presented a comparison of dual thickness and dual dielectric approaches for reduction of tunneling current at behavioral level using simultaneous scheduling and binding of functional units. The method of using dual dielectric is proven to be more productive than the dual thickness approach. The results achieved with the proposed method outperformed other behavioral level leakage reduction works available in the literature in terms of percentage reduction. Further exploration of these techniques in addition to the use of dual-K along with dual-T_{ox} is being investigated for future implementations. The ultimate objective is to extend the work on tunneling current to provide a broader solution to the problem of power dissipation in all its forms at the behavioral level.

References

[1] Semiconductor Industry Assoc, International Tech Roadmap for Semiconductors. `http://public.itrs.net`.

[2] K. A. Bowman, et. al. A Circuit-Level Perspective of the Optimum Gate Oxide Thickness. *IEEE Transactions on Electron Devices*, 48(8):1800–1810, August 2001.

[3] A. Chandrakasan, W. Bowhill, and F. Fox. *Design of High-Performance Microprocessor Circuits*. IEEE Press, 2001.

[4] C. Gopalakrishnan and S. Katkoori. Knapbind: an area-efficient binding algorithm for low-leakage datapaths. In *Proceedings of 21st ICCD*, pages 430–435, 2003.

[5] K. S. Khouri and N. K. Jha. Leakage power analysis and reduction during behavioral synthesis. *IEEE Transactions on VLSI Systems*, 10(6):876–885, December 2002.

[6] N. S. Kim, T. Austin, D. Blaauw, T. Mudge, K. Flautner, J. S. Hu, M. J. Irwin, M. Kandemir, and N. Vijaykrishnan. Leakage Current - Moore's Law Meets Static Power. *IEEE Computer*, pages 68–75, December 2003.

[7] S. P. Mohanty and E. Kougianos. Modeling and Reduction of Gate Leakage during Behavioral Synthesis of NanoCMOS Circuits. In *Proceedings of the 19th IEEE International Conference on VLSI Design (VLSID)*, 2006.

[8] S. P. Mohanty and N. Ranganathan. A Framework for Energy and Transient Power Reduction during Behavioral Synthesis. *IEEE Trans. on VLSI Systems*, 12(6):562–572, June 2004.

[9] S. P. Mohanty, R. Velagapudi, and E. Kougianos. Physical-Aware Simulated Annealing Optimization of Gate Leakage in Nanoscale Datapath Circuits. In *Proc of 9th IEEE Intel Conf on Design Automation and Test in Europe*, 2006.

[10] V. Mukherjee, S. P. Mohanty, and E. Kougianos. A Dual Dielectric Approach for Performance Aware Gate Tunneling Reduction in Combinational Circuits. In *Proc. of 23rd IEEE Intl. Conf. on Computer Design*, pages 431–436, 2005.

[11] A. K. Sultania, D. Sylvester, and S. S. Sapatnekar. Tradeoffs Between Gate Oxide Leakage and Delay for Dual T_{ox} Circuits. In *Proceedings of DAC*, pages 761–766, 2004.