A Design Methodology for High-Performance and Low-Leakage Fixed-Point Transpose FIR Filters

Dimitris Bekiari, Sotiris Xydis, George Economakos and Kiamal Pekmestzi
Department of Electrical and Computer Engineering, National Technical University of Athens
Iiron Polytechnieiou 9, 157 80, Zografi, Athens, Greece
phone: +30 210 7722500, e-mail: {mpekiaris, sxydis, geconom, pekmes}@microlab.ntua.gr
web: www.microlab.ntua.gr

Abstract—This paper addresses the low leakage implementation of fixed-point transpose FIR filters, considering dual-Vth CMOS standard-cell libraries. Specifically, we introduce a design flow, based on a novel two-level selection algorithm, which replaces low-Vth Multiplication-Addition units by their high-Vth counterparts, taking into account the timing slack of each unit and the word-level binary representation of the units’ coefficients. The proposed methodology is evaluated on an 8-tap and a 16-tap transpose FIR filters. Post-layout power results demonstrate leakage improvements ranging from 6.69-25.85% for several clock period constraints, compared to the low-Vth FIR implementations. Also, reduction of up to 12.35% is measured in overall power dissipation.

I. INTRODUCTION AND RELATED WORK

The advances in CMOS process scaling have revealed new design challenges in the field of low-power VLSI design. Specifically, as the shrinking of transistor dimensions and the down-scaling of voltage, Vth tends to reduce the impact of dynamic power in deep submicron technologies, the need for planning strategies regarding leakage power is emerging. At the same time, the threshold voltage, Vth, is also reduced, to maintain performance, therefore enhancing the role of static dissipation. So, the various leakage factors [1] comprise a significant portion of total power in modern standard CMOS technologies. The importance of leakage in deep submicron regime is depicted at the reports of ITRS roadmap [2] and also at recent research activities modeling standby current [3][4].

In general, the main leakage factors are the subthreshold current (Isub), which is the dominating one at the submicron technology nodes, the gate-induced drain and source leakage (GIDL, GISL) [1], the gate tunneling [1][3][4], rising from 45nm and below, and finally, the p-n junction band-to-band tunneling (BTBT) current [1][3][4]. Considering an nMOS transistor [1], subthreshold leakage rises when the device operates at the cut-off region (Vge < Vth) and it is affected by the parameters depicted at the following equation:

\[ I_{sub} = \mu_0 C_{ox} \frac{W}{L} v_t e^{\frac{(V_{gs}-V_{th})}{\eta V_{th}}} (1 - e^{-\frac{V_{gs}}{\eta V_{th}}}) \]  

In (1), \( \mu_0 \) is the electron mobility, \( C_{ox} \) is the gate capacitance per unit area, \( W/L \) is the ratio of the transistor’s width to length, \( v_t \) is the thermal voltage and \( n \) is the subthreshold slope coefficient [11]. The above equation demonstrates that Isub is exponentially dependent on both \( V_{th} \) and \( V_{ds} \). Thus, most of the leakage-aware design techniques use high-Vth components [9][10][11][12], or dynamically reduce \( V_{th} \) in [5][6][7][8].

The most common approach for the tuning of \( V_{th} \) is the Multi-Threshold CMOS (MTCMOS) technique, where a high-Vth sleep transistor is in-series connected with the low-Vth circuit [5][6], in order to take advantage of the stacking effect [7]. This methodology requires process modifications, due to the different \( V_{th} \) levels needed. In [8], this drawback is tackled through power gating, where sleep transistors of regular, low threshold voltage are distributed at each layout row. However, these techniques are not easily applicable at both the RTL and the gate-level, as they require placement information.

On the contrary, multi-Vth design strategies do not require layout feedback. Even if they were initially applied at the physical and the gate level, by assigning to high-Vth the non-critical paths [9][10], leakage gains are limited by the small number of available paths. However, such strategies are more promising at the RTL and the behavioral level, where the selection is performed between coarse components. In [11], low-Vth datapath components are replaced by high-Vth ones, based on maximum-weight-independent-sets, while in [12], an heuristic is suggested for replacement, based on modules’ leakage lookup-tables and on the cycles these stay inactive.

Regarding transpose FIR filters, most of the works in the literature target dynamic rather than static power, by using subexpression sharing techniques, suitable for fixed-point structures [13][15]. Also, [14] addresses a process variation-aware transpose FIR design flow, based on a similar concept.

Thus, this work attempts to fill a gap in the specific field, by introducing a design planning methodology, targeting leakage reduction with no performance degradation. Specifically, the proposed flow addresses the replacement of fast, high-leakage low-Vth Multiplication-Addition units of the transpose fixed-point FIR scheme by the slow, low-leakage high-Vth ones,
taking into account the interdependence between the timing slack and the binary structure of each unit’s coefficient. By this way, we achieve leakage power reduction even under the minimum clock period constraint, obtained by low-\(V_{th}\) implementation. Therefore, performance is not sacrificed for leakage merits. The proposed methodology is compatible with industrial tool flows. Post-layout experimental results of both the 8-tap and the 16-tap filter designs show that we achieve up to 25.85% improvement in leakage, along with overall power gains up to 12.35%, compared to the low-\(V_{th}\) implementation.

The outline of the paper starts introducing the basic concepts that motivated our approach and continues with the third section, which illustrates the proposed flow. The results evaluating our methodology are presented in the fourth section, while the paper concludes with hints for future work.

II. MOTIVATION AND BASIC CONCEPTS

Mixed-\(V_{th}\) standard-cell design methodologies consider a given timing constraint, usually relaxed, to perform the component replacement. However, as this constraint becomes more and more strict, fewer paths in the circuit can be selected for low-leakage implementation. This is a severe drawback for DSP datapaths, where performance bounds should not be violated, while static power savings should be ensured in both quiescent and active periods.

The transpose FIR filter [15] is a typical DSP example. Its architecture is based on a Multiplication-Addition (MA) unit, which comprises the longest path in the circuit. An N-tap transpose filter structure is depicted in Fig. 1, where the input data signal, \(x_n\), is multiplied with \(N\) coefficients and the resulting products are added in a pipelined manner.

For each MA unit and therefore, for each coefficient, there is a corresponding register-to-register path. Therefore, the circuit’s critical path delay is strongly dependent on the coefficient structure, which is defined by the number of zero bits found in its word-level binary representation.

So, let us consider the dot diagram of a 4x4-bit multiplication operation, shown in Fig. 2. In Fig. 2(a), the input data signal, \(x_n\), is multiplied by coefficient \(c=1111\), while in Fig. 2(b), the signal \(x_n\) is multiplied by \(c=1101\). It can be seen that, in the later case, a timing slack is generated, because of the existence of one zero coefficient bit. Such a slack is produced not only in array multipliers, as shown in Fig. 2, but also in Wallace tree and Parallel-Prefix structures. Moreover, this slack exists independently of the timing constraint, so it can be exploited even at the filter’s maximum performance.

The possibility to take advantage of the coefficient’s word-level binary representation for static power reduction forms the motivational observation of the proposed low-leakage design planning methodology for transpose fixed-point FIR filters. In brief, our approach takes into account both the fixed coefficients’ structure and the power-delay tradeoff found into dual-\(V_{th}\) standard cell libraries, to guide the replacement of low-\(V_{th}\) MA units with the corresponding high-\(V_{th}\) ones, thus reducing the leakage power of the transpose FIR architecture. The following section presents in detail the proposed methodology and the design flow that implements it.

III. DESIGN FLOW

The introduced top-down design flow targets the leakage reduction of transpose fixed-point FIR filters. It is fully compatible with industrial EDA tools [16][17] and it can be easily implemented as far as a dual-\(V_{th}\) cell library is available.

The overall design flow, from RTL to GDSII (layout), is depicted at Fig. 3. The Synopsys Design Compiler (DC) [17] and Cadence SoC Encounter [16] tools have been used for synthesis and place-and-route, respectively. Each step of the proposed flow has been annotated in Fig. 3. In more detail:

*Step 1:* At this step, the selection of the MA units to be replaced is performed. In a first level, the filter’s VHDL description is fed to the toolchain depicted at Fig. 3. Then, it is synthesized, in respect to the selected clock period, which is the flow’s second input and comprises the timing constraint. At this step, we linked only the low-\(V_{th}\) standard-cell library, following a top-down synthesis strategy.

*Step 2:* Static Timing Analysis (STA) is performed on the synthesized low-\(V_{th}\) gate-level FIR netlist, in order to extract the register-to-register paths, one for each MA unit (Fig. 1).

*Step 3:* At this step, the selection of the MA units to be implemented by high-\(V_{th}\) cells is performed. In a first level, the selection is guided by the available timing slack of each MA. Thus, the MA units with zero timing slack are early pruned. The rest of the MA units are candidates for replacement. According to the analysis provided in Section II, it was expected to find significant timing slack differences among MA units. Nevertheless, at the post-synthesis level, we observed that MA units with few zero coefficient bits had almost equal slack with units, whose coefficients have most of their bits equal to zero. This happens due to the load balancing that the synthesis tool performs to the filter’s existing paths, in order to achieve the minimum possible area, while satisfying the clock period constraint at the same time.

However, in each MA unit, there can be an extra timing slack available, depending on the number of zero bits of the fixed coefficients (Section II). In order to exploit this extra slack, the MA selection is guided to a second level now, by...
the number of coefficients’ zero bits. By this way, we limit the selection among MA units that combine manageable timing slack with sufficient number of zero bits, in order to avoid timing violations, due to the insertion of high-$V_{th}$ cells.

After extensive experiments, a percentage value of 60% (percentage of the coefficient’s bits equal to zero) has been considered adequate as the threshold, above which the MA units can be assigned to high-$V_{th}$ cells. This threshold is put for any clock period constraint and, although conservative, it guarantees leakage savings without violation, even at the most strict timing bound. The selection algorithm is given below:

```c
if (zero_bits(i) for (i=0; i<N; i++) { // N: the number of filter taps and of MAs if (slack(i) > 0) {
    if (zero_bits(i) ≥ 0.6 * coeff_bitwidth)
        Assign_high_Vth(MA);
else
    Assign_low_Vth(MA);
}
```

Step 4: In the initial Verilog netlist of the filter, we replace the low-$V_{th}$ cells of the selected MA units by the corresponding low leakage, high-$V_{th}$ ones. As these cells are slower, we re-synthesize separately each of the selected units under the given timing constraint, by linking only the high-$V_{th}$ library.

Step 5: The produced timing-accurate high-$V_{th}$ netlists of the selected MA units are inserted into the filter design, replacing the initial low-$V_{th}$ implementations. Then, the filter is linked with the two cell libraries, to check for possible errors. If these do not exist, the mixed-$V_{th}$ netlist is driven to the physical implementation stage.

Step 8: The routed design is RC-extracted, to obtain the parasitics required for the post-layout STA. Step 9 performs RC-extraction aware STA. In case of timing violations, continuous timing-driven post-routing optimization iterations are employed, until the constraint is satisfied.

Step 9: We generate the Standard Delay Format (SDF) file from the final, violation-free layout of Step 8 and we back-annotated it to ModelSim, for the post-layout simulation of the filter’s Verilog netlist. Through simulation, we obtain the switching activity information, required for power analysis.

Step 10: The produced switching activity information file is driven to Encounter Power Analysis tool, to extract the power results, under nominal operating conditions, which are defined by the dual-$V_{th}$ library. The final outputs are the reports of leakage and overall power, presented in the following section.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

We evaluated our methodology by comparing the mixed-$V_{th}$ 8-tap and 16-tap transpose FIR designs with their low-$V_{th}$ counterparts, in terms of leakage and overall power, based on the 1.0V Faraday dual-$V_{th}$ 90nm CMOS standard-cell library. In the experiments, the filter used is given in [19], while input data is a 16-bit signed number. The MA units are implemented with the Synopsys DesignWare 2’s complement Parallel-Prefix multipliers and fast carry-lookahead adders [18].

The compared designs are implemented under both very strict, as well as relaxed timing constraints, to better demonstrate our approach. The harder timing bound corresponds to the filter’s maximum performance and thus, to the minimum possible clock period. This lower bound is obtained by implementing the filter only with low-$V_{th}$ cells and it is determined by performing post-layout STA, in SoC Encounter environment. Similarly, the most relaxed clock constraint is found by using only high-$V_{th}$ cells. Thus, the range of clock periods in our experiments is between the total low-$V_{th}$ and the total high-$V_{th}$ implementation bounds. For the filter used in the experiments, the minimum clock period (i.e. maximum performance) was 3.4ns, giving 0.06mW in leakage and 1.11mW in total power. For the 16-tap case, it was 3.8ns, resulting into 0.13mW of static and into 1.6mW of overall power.

Given this range, the leakage power results are illustrated in Fig. 4 and Fig. 5, for both the 8-tap and the 16-tap filters. At the post-layout simulation, performed in ModelSim, the compared designs operated in equal active and quiescent periods. During runtime, we fed them with 100,000 random input data samples, through a Verilog testbench. For the idle periods, both the clock and the input data inputs were “frozen” through the testbench, so that only static power is consumed.

The results depicted at the following diagrams show that the proposed dual-$V_{th}$ approach achieves significant gains in leakage power. According to the proposed selection algorithm, 50% of the MA units have been assigned to high-$V_{th}$ in both FIR case studies. In perspective, for the 8-tap scheme, we have 10.2% reduction in leakage at the highest performance,
while the maximum reduction of 25.8% is achieved at 2.3ns. In the 16-tap scheme, the results demonstrate an improvement in leakage power dissipation ranging from 6.69%, at 1.7ns, to 23.8%, at 2.8ns.

From Fig. 4 and Fig. 5, it is shown that leakage reduction is decreased, as the clock period increases, because cells with lower driving strength and thus, with less static power, are used at more relaxed constraints. For both the 8-tap and the 16-tap cases, the static power gains achieved, compared to the low-$V_{th}$ strategy, rise sharply over 2ns. On the contrary, leakage savings are smaller at shorter clock periods, as larger cells are used, to satisfy the specific timing constraint.

VI. CONCLUSION AND FUTURE WORK

In this work, we presented a design planning methodology for leakage optimization of fixed-point transpose FIR filters, based on a dual-$V_{th}$ 90nm CMOS standard-cell library. Future work will focus on more computational-intensive DSP cores and also to coefficient transformations, in order to increase the number of MA units assigned to low-leakage implementation.

REFERENCES

[17] Synopsys DesignWare Building Block IP User Guide”, SOLD.
[18] Synopsys DesignWare FIR filter, “DW_fir Data Sheet”, SOLD.