

Design and Optimization of a High Performance Low-Power CMOS Flex Cell

M. Sreedevi and P. Jenopaul
Anna University, Tamil Nadu, India

Abstract: A high performance low-power flex cell is being introduced. With increasing circuits' complexity and demand to use portable devices, power consumption is one of the most important parameters these days. The close coupling between the clustering and mapping process is key to success of this design optimization technique. Specifically, the mapping process is tailored to choose from a variety of techniques that can be used to create new flex cells based upon the inputs, it receives from the clustering process. Such mapping techniques can include time-tested methods of gate sizing and transistor sizing as well as techniques typically found in manual design flows, e.g., creation of new transistor-level implementation of the function of a given cluster of standard cells. Simulation results are performed by HSPICE based on 0.18 μm CMOS technology shows that the new circuit has the lowest power-delay product over a wide range of voltages among several low-power flex cells of different CMOS logic styles.

Key words: Low-power, low-voltage, flex cells, voltaged, cells, India

INTRODUCTION

Most of the VLSI applications such as digital signal processing, image and video processing and microprocessors, extensively use arithmetic operations. Addition, subtraction, multipliers, filters and multiplication are examples of the most commonly used operations. The flex cell is the building block of all these modules. Thus, enhancing its performance is critical for enhancing the overall module performance. Recently, building low-power VLSI systems has emerged as highly in demand because of the fast growing technologies in mobile communication and computation (Jain and Brodersen, 1996). The battery technology does not advance at the same rate as the microelectronics technology. There is a limited amount of power available for the mobile systems. So, designers are faced with more constraints: high speed, high throughput, small silicon area and at the same time, low-power consumption. So, building low-power, high performance flex cells are of great interest. A structured approach for designing and analyzing, a flex cell is based on decomposing it into smaller modules. Each of these modules is implemented, optimized and tested separately. Several flex cells are composed by connecting these modules. The goal of this study is designing a low-voltage and so, low-power flex cell. The inputs to this mapping process can include: A set of structural net lists composed of standard cells, otherwise designated clusters. A set of performance constraints for each individual cluster. The set of clusters and the set of performance constraints for the clusters are identified by

a clustering process that precedes the mapping process (Shams *et al.*, 2002). The clustering step, essentially partitions the output of a conventional logic synthesis tool either using heuristics to guide the partitioning process or using a systematic search procedure (Karayiannis and Tragoudas, 1995).

CMOS FLEX CELLS

The mapping process of the flex cell and its building blocks are shown in Fig. 1. In addition, various circuits have been proposed for each module. Most of the existing mapping algorithms are geared towards working with very simple objectives such as minimizing transistor count. Moreover, many of the methods suffer from relatively high computational complexity. The optimization criteria and design requirements for the generated flex-cells are not static but are varied and complex even across different parts of a given IC design.

Consequently, practical transistor net list generation processes need to start with the invocation of a plurality of algorithms to generate multiple flex-cells that may ultimately be used in the target design. It has only one critical input, namely input a. In this context, a critical input denotes an input such that the delay from this input to the output of the cell limits, the overall performance of the cell. A flex cell is generated by the mapping process and the performance improvement that would result from replacing the cluster by the flex-cell. Flex-cell based optimization can be used to significantly reduce the total number of place able instances in a design as well as the

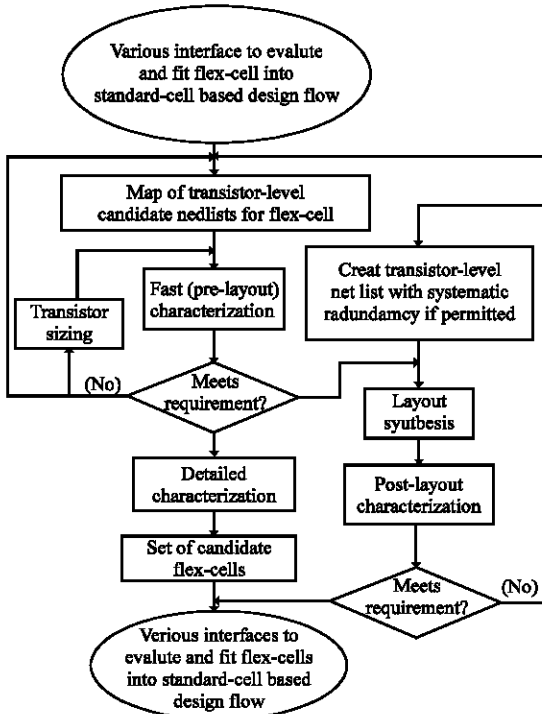


Fig. 1: Mapping process of the flex cell

total number of interconnects between instances in a design (Morrissey, 1996). Such reduction in turn, benefits the final design quality in a multitude of ways including higher quality results during place-and-route phase and potentially improved signal integrity and noise problems due to fewer interconnects between cells (Goes *et al.*, 2005).

AUTOMATED FLEX CELL BASED DESIGN

The impact of a change in transistor topology and transistor sizing on the performance of a flex-cell is complex. Various combinations of choices made in the above processes may result in a large set of candidate design specific cells. As a result, the mapping process typically includes a selection step. The selection step begins with ranking the candidate flex-cells, using a sophisticated cost function that evaluates the quality of each design-specific cell, measured using various appropriate target metrics such as input-to-output delay through the design-specific cell, number of transistors, stack-depth (i.e., length of a path through N or P-transistors), input load capacitance, output drive strength, etc. (Yelamarthi and Chen, 2008). A limited number of candidate flex-cells from the top of the rank-ordered list are then chosen for use in the overall design optimization loop. In a simplified optimization scheme, the flex-cell

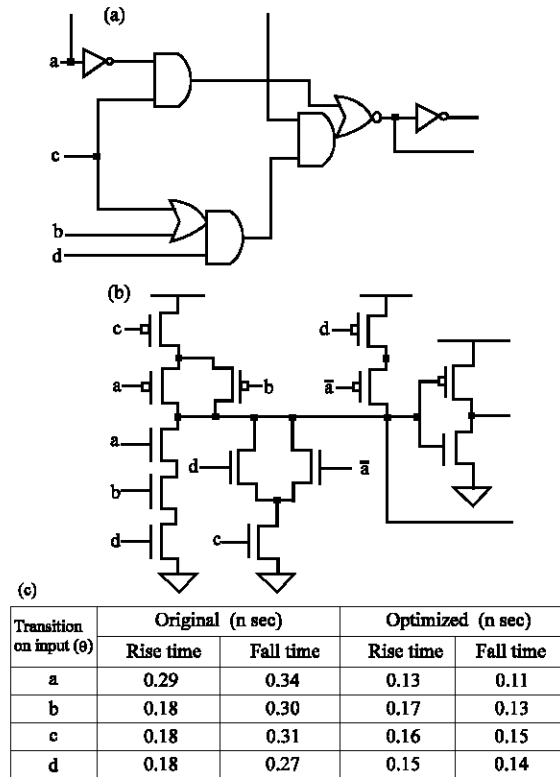


Fig. 2: a) Original cluster of standard size; b) Flex cell created by

selection process can be greedy or iterative in nature. Other sophisticated search schemes may be employed in flex-cell selection process including linear programming, dynamic programming, branch-and-bound search technique, etc. or some combination thereof to achieve near-optimal design of the flex-cells. Although, the previous description has been implicitly focused on the static CMOS family of logic circuits, the mapping process described previous is broadly applicable for creation of the NMOS or the PMOS networks individually.

If the target IC design implementation calls for using another family of MOS circuit design including various forms of dynamic CMOS. Figure 2a shows the original cluster of standard size, Fig. 2b shows the flex created by mapping process and Fig. 2c shows the performance improvement created by flex cell.

The addition of such constraints will decrease the potential of any matches in a specific library of cells (standard-cells, flex-cells or mix). For example, Fig. 3 shows a modification to the matching problem of with added constraints on the rise and fall timing. The desired signature could consist of a sorted set of rise and fall times since such a sorted list is independent of any

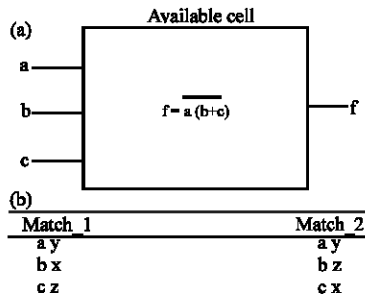


Fig. 3: Two functional matches in a cell

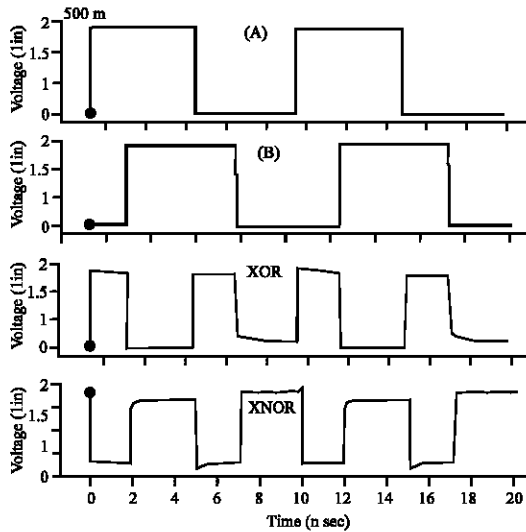


Fig. 4: Waveforms of the outputs of different matches of flex cell

permutations on input pins. It is also possible to combine rise and fall times and construct signatures comprised of sorted lists of functions of rise and fall times that also remain independent of any permutation (Shabtai *et al.*, 2010) or potential complementation's at the inputs of the library cells. In order to account for variances on constraints such signatures can use one or more fixed corners within the range of variability, e.g., the worst possible rise time, the worst possible fall time, etc. Once the fixed corners are chosen, similar lists can be constructed with the sorted rise and fall times for the existing library cells and the signatures for the library cells can be matched against the signatures for any potential feasible matches. The waveforms of different matches of the output is shown in Fig. 4. The feasibility is checked by ensuring that for every target constraint value on a pin (e.g., rise time), there exists at least one value on an actual pin that satisfies the constraint. Automated cell layout synthesis plays a key role in closing the loop with respect to creation of the actual layouts of the flex-cells that are

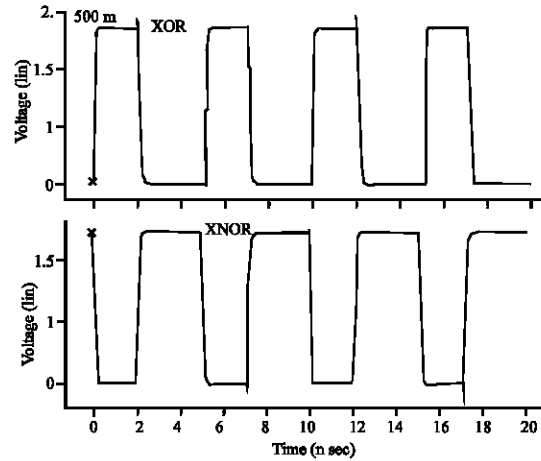


Fig. 5: Improved flex cell and its output waveforms

designed as transistor-level net lists during the mapping process in Fig. 1. Layout synthesis takes as input, the transistor-level net lists of the flex-cells, various fabrication process technology parameters including layout design rules, desired standard cell architecture parameters including cell height, number of tracks, well and implant specifications, etc. and creates the detailed transistor layout-polygons that will be eventually fabricated on silicon substrate. Layout synthesis commonly includes further tuning of the sizes of the transistors in the flex-cells, especially to ensure that the timing characteristics of the flex cells, post-layout, closely match the desired timing characteristic passed to layout synthesis as input.

The improved flex cell and its output waveform is shown in Fig. 5. A key objective for layout synthesis step is compatibility with standard cell library blocks such that the flex cells created can be mixed seamlessly with the predefined standard cells that are used in the rest of design from a layout point of view. The compatibility of the flex cells and the standard cells at the layout level, enables the final IC design to be highly customized (i.e., using flex-cells) and yet stay flexible enough to use standard-cells where possible and/or desired. It is also interesting to note that the use of carefully controlled cell-layout.

CHARACTERIZATION OF FLEX CELL

The characterization of flex cell is shown in Fig. 6. As minimum feature size of fabrication processes have progressed to 0.18 μm and smaller, it has become virtually impossible to create designs, especially high-performance designs without incorporating detailed physical design information into the synthesis and optimization process.

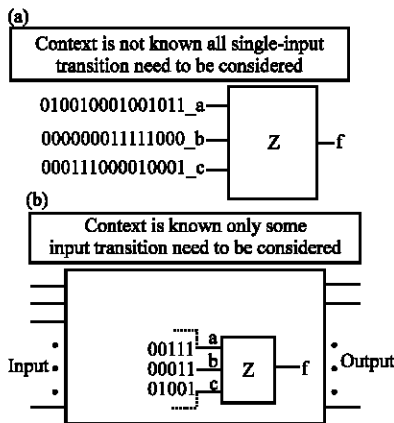


Fig. 6: Characterization of flex cell

The dominant factor guiding this development of course is the greater role played by interconnect delays in determining the overall delay of critical paths in a design. Flex-cell based optimization is no exception to this trend. The flex-cell based design must take into account actual wire delays, loads and slew degradation differences between different parts of the same interconnect net-or good estimates there of derived from physical design knowledge.

The local optimization steps including clustering and mapping must also take into the impact of wire delays, loads and slew degradation profiles of nets of interest. Various intermediate steps can be taken as flex-cell based optimization transitions from traditional wire-load model based load computation to physical design based load computation. A necessary step is to understand placement of the cells-standard-cells and flex-cells and to estimate the wire lengths of individual nets, utilizing various well-known parameters like half-perimeter, number of terminals on the net, fraction of bounding box covered by cells, fraction of bounding box occupied by blockages.

At high utilization or in the presence of high congestion, more detailed routing information become essential to allow accurate estimation of the delays and loads that need to be taken into account by various stages of the flex-cell based optimization tool. Key to the incorporation of such physical design data into a flex-cell based optimization process is the use of fast incremental placement algorithms. Such incremental placement algorithms can be based on well known techniques like quadratic placement, force-directed placement, etc. or some appropriate combination of multiple placement techniques. Important issues that need careful attention for effective use of incremental placement execution time of the incremental placement algorithm quality of result as

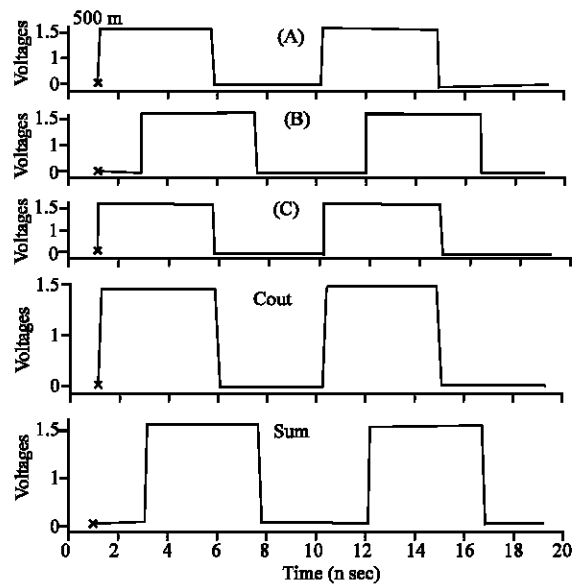


Fig. 7: Snapshots of waveforms at 1.8 V and 100 MHz

measured by correlation to final placement that will be generated by whatever place-and route tool is to be used for actual layout. A practical solution may need to make a variety of trade-offs to achieve the desired speed, potentially at the cost of some quality degradation. Such trade-offs may include: Relaxing requirements to generate legal placements, i.e., allowing placements generated to have some design-rule violations like cell overlap, Invoking incremental placement after a set of optimization steps are completed as opposed to invoking incremental placement after every change made during optimization Using simpler algorithms like force-directed placement as opposed to more sophisticated placement techniques.

Simulation results: Simulation results are performed by HSPICE based on 0.18 μm CMOS technology. The supply voltages range from 0.5-3.3 V. The operating frequency is 100 MHz for supply voltages >1 V and 10 MHz for supply voltages <1 V. The snapshot of the waveforms at 1.8 V is shown in Fig. 7. The power and delay plotted against supply voltages of the five flex cells are shown in Fig. 8a, b. Simulation results show that the function at low voltage. The lowest voltage that it can work at 100 MHz is 1.8 V. The excessive power and delay attributed to the threshold voltage drop problem and the poor driving capability of some internal nodes at input combinations that create non full swing transitions. The speed of the 14 T decreases faster with the supply voltage than other cells. This circuit does not work at supply voltages <0.8 V at 10 MHz. Simulation results also show that the new cell together can work reliably at low supply voltages

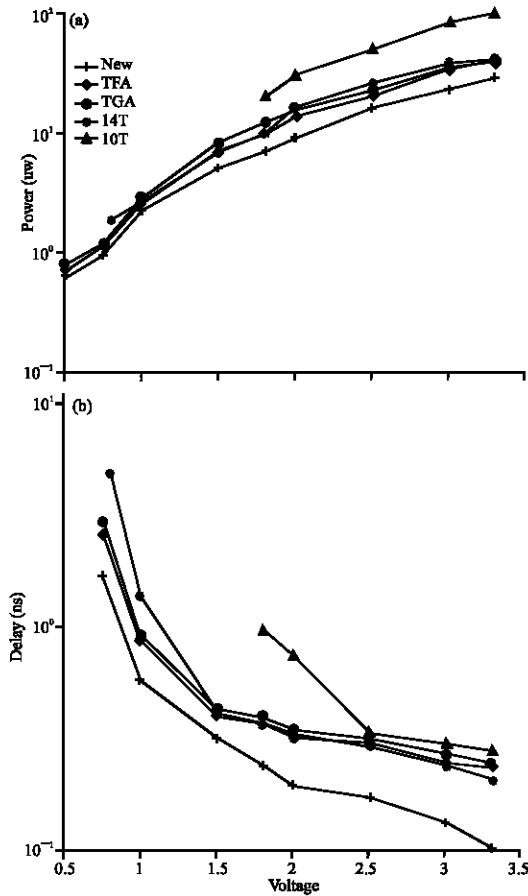


Fig. 8: a) Compared power result of flex cell; b) Compared delay result of flex cell

<0.5 V at 10 MHz. Although, it has lesser transistor count, additional buffers are required at each output to boost up its drivability which increases their short circuit and switching power. As can be seen, the new flex cell is the most energy efficient cell.

CONCLUSION

A high performance low-power flex cell is capable of operating <0.5V has been presented. The new flex cells were implemented that uses only different clusters. Proposed flex cells were tested in various conditions such as various output loads, supply voltages and input frequencies and they show good power consumption and

performance. The proposed circuits are suitable for arithmetic circuits and other VLSI applications such as digital signal processing, image and video processing and microprocessors, extensively use arithmetic operations. Addition, subtraction, multipliers, filters and multiplication are examples of the most commonly used operations. With very low power consumption and a very high performance. Based on the simulation results, it has been culminated that the new proposed circuits have good output signal levels, consume less power. The new circuit is the most energy efficient cell compared to several recently proposed circuits.

REFERENCES

Goes, J., N. Paulino, H. Pinto, R. Monteiro, B. Vaz and A.S. Garcao, 2005. Low-power low-voltage CMOS A/D sigma-delta modulator for bio-potential signals driven by a single-phase scheme. *IEEE Trans. Circuits Syst.*, 52: 2595-2604.

Jain, R. and R.W. Brodersen, 1996. VLSI in mobile communications. *Proceedings of the 9th International Conference on VLSI Design*. Jan. 3-6, Bangalore, India, pp: 12-18.

Karayiannis, D. and S. Tragoudas, 1995. A new approach for clustering network modules for delay minimization. *Proceedings of the 1995 International Symposium and Workshop on Systems Engineering of Computer Based Systems*. March 6-9, Tucson, AZ. USA, pp: 325-332.

Morrissey, J.M., 1996. Reduction filters for minimizing data transfers in distributed query optimization. *Can. Conf. Electr. Comput. Eng.*, 1: 198-201.

Shabtai, A., E. Menahem and Y. Elovici, 2010. F-Sign: Automatic, function-based signature generation for malware. *IEEE Trans. Syst., Man Cybern.*, PP: 1-15.

Shams, A.M., T.K. Darwish and M.A. Bayoumi, 2002. Performance analysis of low-power 1-Bit CMOS full adder cells. *IEEE Trans. VLSI Syst.*, 10: 20-29.

Yelamarthi, K. and C.I.H. Chen, 2008. Process variation aware timing optimization through transistor sizing in dynamic CMOS logic. *Proceedings of the 9th International Symposium on quality electronic design. (ISQED'08)*, Quality Electronic Design, pp: 143-147.