An Innovative Methodology for the Design Automation of Low Power Libraries

Nicola Dragone, Carlo Guardiani, Roberto Zafalon and Pascal Meier[+]

STMicroelectronics, Central R&D 20041 Agrate - Italy

[+]Carnegie Mellon University, Pittsburgh, Pennsylvania.

Abstract - A new methodology for the design of low-power standard cell libraries is presented. The proposed approach addresses power consumption at various steps in the design flow, applying new design automation algorithms and incorporating innovative cell designs. CAD techniques are used to speed development of the library, allowing for quick analysis of power and delay characteristics, with subsequent feedback for redesign. The effectiveness of the proposed flow is demonstrated on several benchmark circuits implemented in a 0.35μm CMOS, 1.8 volt standard cell library designed using this methodology.

I - Introduction

Power consumption is becoming an increasingly important factor in IC design. While substantial power savings can be achieved by algorithmic and architectural optimization, the cell library fundamentally establishes the power characteristics of the final design. Cell library optimization has traditionally been targeted at minimizing area while meeting delay targets, but power has become a primary design concern.

We believe that the need to provide several libraries, addressing different kind of applications [1], coupled with the huge development time required for each of them, should induce IC makers to develop an efficient automated design flow. Such a flow could provide a competitive advantage by 1) reducing library design time, 2) achieving optimized libraries to meet performance goals. The design of a cell library affects power dissipation at several stages: at the system level, gate level as well as the physical design level. The gate level is concerned with power dissipation of each individual cell. In this context, the cell is designed to meet timing and area constraints while minimizing internal energy dissipation. At the system level one has to consider how the instantiated cells interact each other and how individual modifications affect the total power consumption of the design. The physical design level relies to the cell layout stage as well as the interaction with system level physical attributes. Furthermore, we have to consider that a cell library provides those basic logic functions extensively used by automatic synthesis tools [2]. An effective low power library design methodology must be able to generate a choice of cells suitable for a successful design synthesis mapping onto the given technology. The

[+] P. Meier is funded by ARPA contract DAAL 01-95-K-3527 and NSF contract MIP-9408457.

set of library functions and drive strengths can considerably impact the quality of the final implementation [3], both in speed and power dissipation. At the same time, the level of routing porosity and the physical structure of the cells affects system level characteristics such as area and length of wires, which also impact power dissipation. This paper presents a low-power cell library design flow, spanning all design phases from the logic design downto layout implementation, enhancing existing techniques with original CAD solutions. The library optimization process is addressed to achieve maximum benefit at macro-block level instead of just stacking the focus at the gate level. In this sense, while a single, stand-alone cell might show a sub-optimal behavior in terms of power consumption, this allows a larger power saving of the whole circuit [4].

The remainder of the paper is organized as follows. Section II describes the full design flow and presents some CAD solutions to the problem of library development. Section III discusses the circuit techniques which reduce power consumption. Particular emphasis is given to flip-flop internal architecture, usually a very critical cell in the library, and optimum transistor sizing. Section IV discusses physical implementation issues and automatic layout. Section V shows the synthesis results obtained by using a complete library developed with this methodology. Final conclusions are presented in Section VI.

II - Low Power Library Design Flow

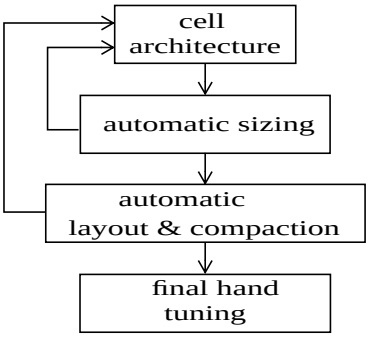


Fig. 1: Low power library design flow.

The design of a standard cell can be divided into three steps: cell architecture design, transistor sizing and physical design. The objective of a good methodology is to optimize the design goals and to minimize relooping through the design flow. The library design flow that we used is shown in figure 1.

The design process starts with the definition of the cell functionality and architecture (i.e transistor topology). By looking to a significative number of ASICs internally developed for several digital applications, we selected the core's functions most frequently used by the automatic synthesizer and identified their fanout statistics. The right choice of the cells can lead to a significant improvement of the final results, as reported in [6], where the implementation of a large Wallace Tree multiplier with boot recoding has been enhanced of about 25% in speed, by simply retargeting the synthesizer onto a better library, including a larger variety of two bit adders. Clearly not each kind of cell contributes the same to the total power consumption. For example, the cells belonging to the clock network (i.e buffers and flip-flops) and storage cells can be responsible for up to 50% of the total power dissipation of a logic circuit [5]. Therefore we devoted special attention to the realization of a low-power flip-flop, as described in following sections.

The design of the cell's topology is difficult to automate because it deeply relies on the experience of the designer, as well as on layout parasitic information, difficult to estimate at this early stage. One possible technique is transistor reordering [4] which can achieve reductions in delay and power, assuming to know the input signal arrival times. The main goal of automation is transistor sizing. In our flow we propose an approach based on genetic algorithms [7], in which the optimization parameters are the transistor widths and the target functions are the timing performance as well as the power consumption of the cell. Based on Response Surface Method (RSM), the optimization tool determines delay and power characteristics during the process. The most time consuming step of the cell design is traditionally the realization of the physical design. In our flow we have successfully integrated an automatic layout synthesizer [8], able to provide results whose quality is close to hand-crafted layout. However, final hand-optimization tuning is applied to achieve minimum delay and area. This flow provides two advantages over previous methodologies: a global speed-up of the whole design process is achieved by incorporating a high degree of automatization, and a thorough improvement in the quality of the cells is obtained.

III - Low Power Library Design Techniques

Power dissipation can be reduced by either lowering the energy involved in an operation, or reducing the amount of toggle on output pins. The techniques proposed in these chapters pursue both aims.

III.1 - Low Power Flip-Flops

When designing low power flip-flops, it is essential to minimize power consumption in all the working conditions of the cell. The most frequent condition is when the data input remains stable for several clock cycles and the flip-flop activity consists just in holding up the same output value, for long time.

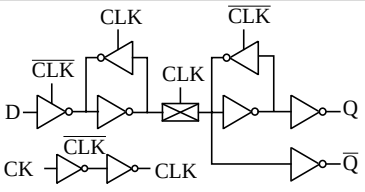


Fig. 2: Conventional D flip-flop.

Figure 2 shows a conventional D flip-flop, with a local buffer used in the generation of the two complementary clock phases for the sensitization of the master and slave structures. When the input D is stable, the local clock buffer burns power without doing anything useful. The easiest way to reduce this unwanted consumption is to ask the flip-flops to work with a single clock phase only. A second important source of power consumption are glitching and spurious transitions of D during the clock period. These transitions cause power consumption in the master block of the flip-flop. To reduce this energy dissipation, an approach would be to disable the master latch before the latching event.

Finally one has also to limit the power consumption of the flip-flop when data is actually latched: although this is the less frequent working configuration, it impacts the power consumption with the largest absolute value. In this case a power saving can be achieved by minimizing the internal capacitances and the short circuit currents via an optimal transistor sizing.

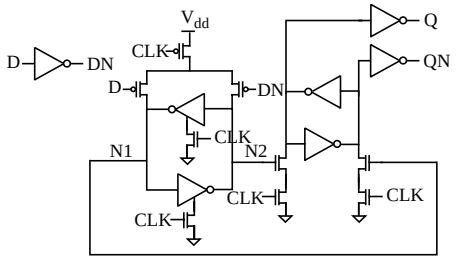


Fig. 3: Low power flip-flop architecture

In figure 3 we propose an original architecture for a low power flip-flop, which minimizes the overall power consumption. In this architecture only one clock phase is used, and the master structure is clocked to avoid power consumption when data is toggling and the clock is stable. The master structure is original because of the way the data is latched, based on different discharging velocity of the nodes N1 and N2, assigning the correct data value to the slave structure.

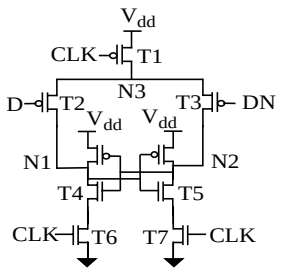


Fig. 4: Master structure of low power flip-flop

The master structure of the low power flip-flop presented in figure 3 is shown in figure 4. The key parameter for the correct operation of this circuit is N3 node capacitance (C3). Spurious D input toggling between clock transitions are eliminated by disabling the master latch when clock is stable low. However, with this solution a condition may result in which both nodes N1 and N2 in the master structure are at a high logic value. This can be considered a weak metastable state from which the flip-flop is able to recover by means of the charge stored in C_3. Let us suppose that the flip-flop has to latch a data at a low logic level, therefore the transistor T2 is on and the transistor T3 is off.

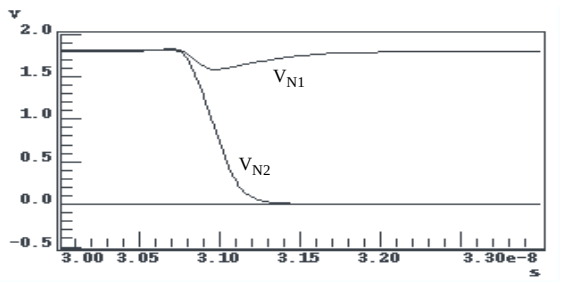


Fig. 5: Latching event in the master structure of figure 4.

When the clock goes high, the transistor T1 is off and the pull down networks T4-T6 and T5-T7 begin to discharge the capacitance at nodes N1 and N2 respectively. Node N2 has to be discharged more rapidly in order to prevent node N1 from being also discharged, and thus setting the correct latching of the data (see figure 5).

Therefore, the relative discharging speed of

the nodes N1 vs. N2 is the critical parameter. Table 1 shows the power consumption of the low power flip-flop (lp FF) and the standard flip-flop (std FF) on different stimuli configuration. As you can see, when data is stable and clock toggles we get the largest power saving (i.e.: 37X). This is a good candidate to save power since such an event happens during most of the FF's working life. The only condition in which the lp FF consumes more than the std FF is with the clock stable high and toggling data which is, indeed, extremely seldom for designs based on positive edge triggered FFs.

stimuli config.	lp FF	std FF	std FF/ lp FF
data stable clock @50MHz	0.19µW	7.16µW	37.6 X
data @25MHz clock stable low	1.26µW	3.5µW	2.77 X
data @25MHz clock stable high	2.24µW	0.96µW	0.42 X
data @25MHz clock @50MHz	16.5µW	21.2µW	1.28 X

Table 1: Power consumption of the two different flip-flops.

III.2 - Cell transistor sizing

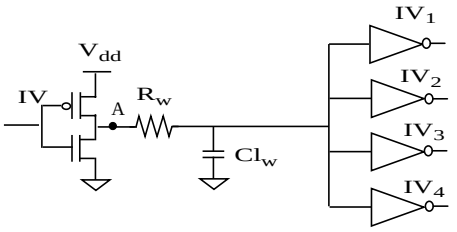


Fig. 6: Network for the analysis of the influence of transistor sizing on power consumption

Transistor sizes ultimately determine the area, timing performance and power consumption of the cell. While it is generally believed that power dissipation is proportional to the active area, it is not difficult to show how the energy can be reduced by adopting a non-minimal sizing for the transistors. In figure 6 an inverter 'IV' drives an RC modeled connection and four inverters as active loads.

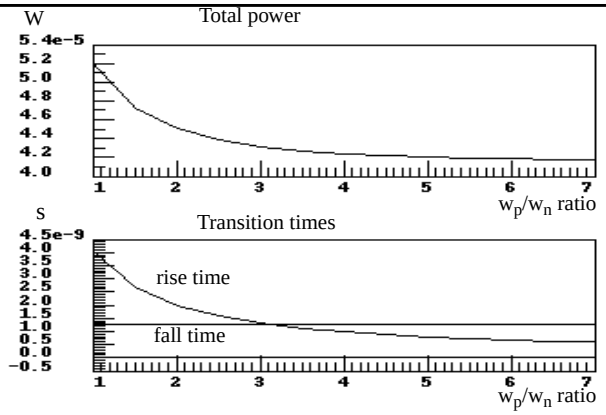


Fig. 7: Total power consumption and transition times at node A of the circuit shown in figure 6, varying W_P/W_N ratio.

Figure 7 shows the transition times at the output of inverter 'IV' and power consumption of the total network (including driver and active loads), varying the P/N ratio of the driving inverter, by keeping minimum the width of the NMOS and increasing the PMOS. By applying a fixed input transition time of 1ns, the curve of power consumption decreases, before reaching a nearly constant value. The explanation of such a behavior is on the contribution of the short cir-

cuit currents of the active loads: for minimum sizing (W_P/W_N=1), PMOS of inverter IV is too slow and the power consumption due to short circuit current through the loads is high; for larger size (W_P/W_N>3), the power is approximately a constant because the rise/fall time at the output of the inverter 'IV' (node A) don't diminish very much for higher ratios and the short circuit current of the loads is minimized. The objective of a good transistor sizing is to get close to the point of balanced rise-fall times while minimizing power consumption (a good W_P/W_N ratio for the 0.35μm technology under analysis has been calculated to be 2.8) Furthermore, this allows to optimize the synthesis process by avoiding oversized cells.

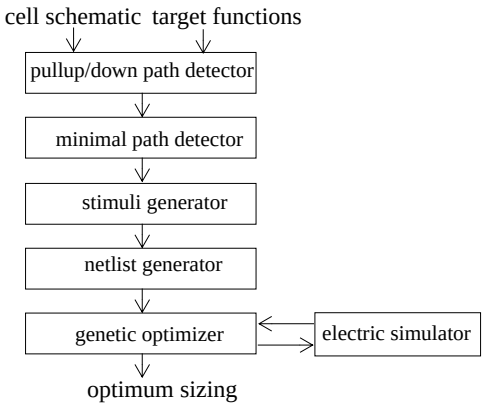


Fig. 8: Automatic transistor sizing flow

Clearly the optimum is also related to the electric network used in the analysis. For this reason it is important to make statistics on the average fanout and capacitive load of each kind of cell, when instantiated in real circuits. In our applications we found that 90% of the instantiated cells show a fanout between 1 and 4. In transistor sizing there are three main factors to consider in reducing power consumption: cell input capacitance, internal dissipation and balancing rise and fall times. For all of the cells it is important to keep balanced output slopes to reduce short circuit currents in the driven gates. Contemporarily, for multiple stage cells, it is possible to downsize input stages, so as to minimize input capacitance. Unfortunately this technique has some problems. First, it may introduce unbalanced delays. Therefore the automatic synthesizer might map the corresponding logic function onto cells with higher driving capability, resulting in an increased overall power consumption. Furthermore, small sizing of the input stages may lead to big short circuit currents of the internal stages.

To automate the transistor sizing process, we created an optimization procedure based on genetic algorithms. In this approach, outlined in figure 8, the optimization parameters are the widths of the transistors. The target functions are the timing performance and the power consumption of the cell. The main optimization process is based on an automatic topological pre-analysis of the cell's pull-up and pull-down networks, in order to identify the minimum number of paths to simulate and to cluster the transistors having identical width. Two useful characteristics of genetic algorithms are their ability to avoid local optima and their adaptability to parallel execution; in the core of the algorithm, we use distributed SPICE simulations to converge to the global optimum. For complexity reason, this approach has currently been applied only to the single and double-stage combina-

tional cells, representing more than 50% of the total cell library population. The results of the automatic sizing are very promising in terms of quality and CPU-time. The maximum distance in width for each transistor respect to the hand-sized cell, has been found to be less than 3% even in the most complex gate (AO20), where the CPU-time is shorter than 150 minutes. Use of a precise electric simulator within the optimization process allows us to take into account all the transistor parasitic contribution. Future work will address the development of an interconnection parasitics estimator [10] to get early information about their influence within the optimization process itself. In this way the process would actually become performance-driven, avoiding any back&forth relooping with the sizing procedure.

IV - Layout Optimization

Layout generation is often the most time-consuming process in a standard cell library design flow. Generating the layouts traditionally requires several expert physical designers who can only lay out a few cells per day. Furthermore, since this is one of the final steps in the design flow, subsequent modifications in earlier stages (for example, resizing of transistors) forces cell layouts to be re-done. Major modifications can significantly change cell geometries, necessitating complete redesign of the layout. Cell performance is highly influenced by parasitic capacitances, which must either be planned for a priori or dealt with incrementally, i.e. by reloops through the methodology.

Cell		pre-layout	post-layout	post/pre
IV: $Z=\overline{A}$	delay	0.43ns	0.44ns	2%
	power	0.71e-13W	0.73e-13W	3%
AO29: $Z=\overline{A \bullet B \bullet C + D \bullet E}$	delay	0.47ns	0.50ns	6%
	power	1.22e-13W	1.34e-13W	10%
FD1: flip-flop D	delay	1.24ns	1.82ns	47%
	power	6.61e-13W	10.24e-13W	55%

Table 2:Pre and Post layout delay and power figures.

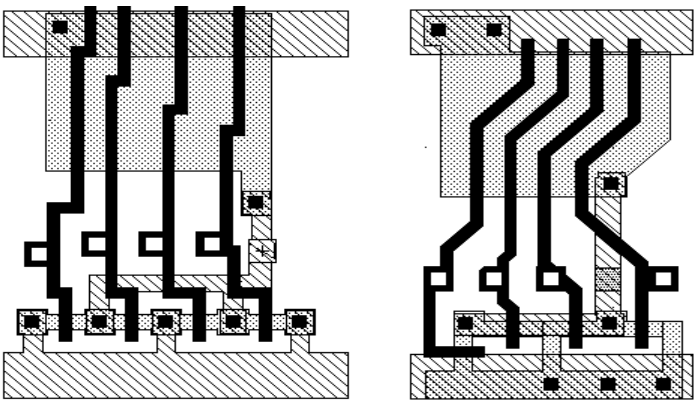
For some cells, pre-layout performance estimates can differ significantly from results obtained post-layout, especially in complex gates as shown in table 2. Therefore, our goal is to couple the design and layout phases, feeding back information generated from extracted layouts to the architecture and sizing steps.

IV.1 - Automatic layout generation

Automatic layout generation has been proposed as a rapid and efficient solution to this problem and several techniques have been proposed to arrive at computer-generated layouts (for an overview, see [9]). These have been effective in quickly generating logic in cases where minimum area-of-implementation is not the primary concern. In standard cell libraries, on the other hand, high reuse of each cell makes minimization of area a

very desirable goal. We tested the applicability of LAS [8] tool in our flow by comparing automatically generated layouts with existing hand-crafted layouts. Our experiments observed that a majority of the cells, in particular the smaller cells (< 20 transistors) can be laid out to within 4-6% of hand layout's area.

The remaining cells fall within 10-20% of hand layout, with a few cells being particularly hard to lay out (30-50% of hand layout's area). An interesting result was that even for cells whose width was much in excess of a hand layout result, the placement of transistors in nearly all cases matched the placement selected by the hand layout artist; this indicates that the tool is best used as a sort of "early layout engine", with hand layout and compaction being used as a cleanup stage. LAS was then applied to approximately 60 cells in the Low Power Library (see section V), which had initially been laid out by hand. These layouts took one designer 1.5 months to implement; the automatic layout generation tool performed these layouts in one hour. Layout quality is variable: LAS is limited in its use of 45 degree polygons; this problem can be alleviated by using 90 degree bent gates as a 'guide'. For example, the design on the left side of Figure 9 was generated automatically; the hand layout simply replaces 90 degree gates with 45 degree gates. Our experiences with this tool have shown that automatic cell generation performs very well on combinational cells, without needing much user input. Sequential cells require much more user involvement, with tool parameters being adjusted on a cell-by-cell basis.



LAS generated layout Hand generated layout

Fig. 9: Layout quality comparison

IV.2 - Feedback of parasitic capacitance information

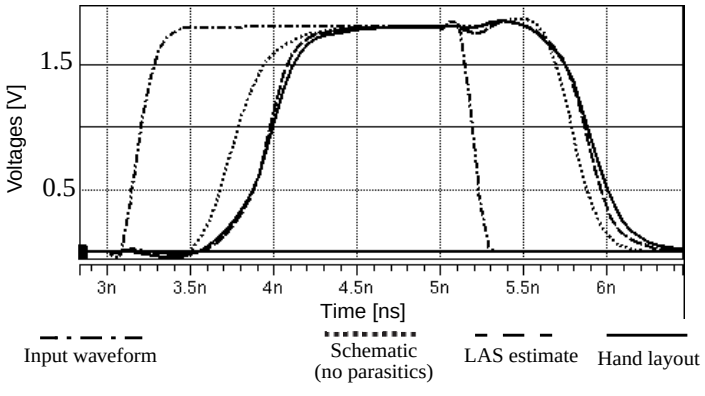


Fig. 10: Simulation estimates of parasitic effects

Parasitic capacitance can be a significant challenge in cell design, since parasitic effects are not easy to judge until the final layout has been extracted and simulated. In our methodology, two approaches are taken. First, based on earlier cell library designs, parasitic effects are estimated and taken into account during design of individual cells. The flip-flop mentioned in section

III is one such example. Second, LAS is used to efficiently generate layouts which are representative of the final design. It can be hard to determine whether the automatically generated layout is close to the best results achievable by hand, especially for complex cells. For such cells, the judgement of a layout expert is needed to assess the results of any automatic layout. For example, in Figure 10, we simulated three representations of a three-input XOR gate: the schematic, a LAS layout, and a hand layout. The LAS layout was slightly sub-optimal, resulting in higher parasitic capacitances (therefore a few delays are over or under-estimated). Nevertheless, this example allows us to see some estimates of the impact of layout parasitics.

V - LPL Library and synthesis results

The design methodologies presented above have been applied to our Low Power Library (LPL) project. This library has been realized in a 0.35µm minimum channel length CMOS process. LPL provides 90 cells, of which 70 combinationals and 20 sequentials, working at a nominal supply voltage of 1.8V. Several circuits have been synthesized with LPL and comparisons have been done with a general purpose library (we refer to it as General-Purpose Library, GPL). GPL is implemented in the same technology and provides 392 cells. This library is targeted at high performances and high density. Along the synthesis process, we have used Power Compiler [2] on a variety of test vehicles: a 24 bit multiplier with 15ns of critical path, an A/D converter's 63 to 6 bits encoder operating at 100MHz and an 8 bit micro-controller at 33MHz. Hereafter the synthesis results, as reported by Power Compiler, are shown in table 3.

	8 **bit µcontr. SEQUENCER**		**63 to 6 bit ENCODER**		**24 bits MULTIPLIER**	
	GPL	**LPL**	**GPL**	**LPL**	**GPL**	**LPL**
Cell Internal Power	378.0620 µW	148.5845 µW	2.0935 mW	598.6977 µW	13.9279 mW	9.5417 mW
Net Switching Power	340.1807 µW	231.768 µW	638.349 µW	540.4468 µW	11.9552 mW	9.9573 mW
Cell Leakage Power	293.263 pW	148.4970 pW	338.242 pW	72.5874 pW	1.4045 nW	645.9275 pW
Total Dynamic Power	718.2426 µW	380.3530 µW	2.7319 mW	1.1391 mW	25.8830 mW	19.4990 mW
data arrival time	3.88 ns	4.6ns	10.42 ns	12.0 ns	13.42 ns	14.78 ns
area	76644 μm^2	69126 μm^2	27342 μm^2	23004 μm^2	220680 μm^2	220411 μm^2
cells number	1121	1036	188	217	2081	2123

Table 3:Synthesis benchmarks. The Sequencer is a block of the whole 8 bit µcontroller

The LPL library always shows a power consumption much smaller than that obtained by mapping onto the GPL library. Maximum power reduction is achieved in the control logic where LPL saves about 60%. This is presumably due to the presence of a large number of sequential cells, where LPL achieve the most. In the full combinational logic, like the multiplier, the reduction is smaller, even if it holds about on 25%. Although the total area

is nearly the same, due to the higher porosity of the cells, leading to a lighter wire load, the net switching power is smaller in LPL. The loose of speed is never exceeding 20% in all of the synthesized blocks (the micro-controller have been fully synthesized), while in the bigger and more significant blocks it is close to 10%. Issues such as finding an optimal choice of logic functions to integrate in the library, the high routing density of certain designs, availability of levels of metal for signal routing, CAD tool limitations, etc., all impact the design of cells which are required to be low power and yet high performance. The LPL library has been shown to be suitable for automatic synthesis of low power applications.

VI - Conclusions

An innovative methodology for the design of low-power standard cell libraries has been presented. The methodology deals with power minimization issues at different levels (circuit, macroblock, and physical design) in a consistent manner, thus achieving significant power savings for synthesized logic modules. Several different techniques have been applied at the different levels. Some of them represent existing state-of-the art technologies, while other techniques are original to this work. The application of the proposed methodology to the development of a low-power library in a 0.35 μm CMOS technology at 1.8 V voltage supply showed that power savings up to 60% can be achieved with respect to other standard CMOS libraries.

References

[1] C. Fisher, R. Blankenship, J. Jensen, T. Rossman and K. Svilic, “Optimization of Standard Cell Libraries for Low Power, High Speed or Minimal Area Design,” IEEE CICC, May 1996.

[2] Power Compiler. “Synopsys Power Product Reference v1997.01,” Synopsys Inc., Mountain View, CA, Jan. 1997.

[3] K. Scott and K. Keutzer, “Improving Cell Libraries for Synthesis,” IEEE Custom Integrated Circuits Conference, May 1996.

[4] S. C. Prasad and K. Roy, “Circuit Optimization for Minimization of Power Consumption under Delay Constraint,” ACM/IEEE 1994 International Workshop on Low Power Design, 1994.

[5] R. Hossian, L. Wronsky, A. Albicki, “Low Power Design Using Double Edge Triggered Flip-Flops”, IEEE Trans. on VLSI Systems, Vol. 2, N. 2, Jun. 1994.

[6] P. Meier, R. Zafalon, “On the Synthesis-Oriented characteristics of high performance, deep-submicron CMOS VLSI cell libraries”, Euro-SNUG 97, Feb. 1997.

[7] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learnings. Reading, MA: Addison-Wesley, 1989.

[8] LAS, “Virtuoso Layout Synthesizer” Reference manual v4.4.1, Cadence Design systems, Santa Clara, CA, 1997

[9] Preas and M. Lorenzetti, eds., “Physical Design Automation of VLSI Systems,” Benjamin/Cummings Publishers, 1988, chapters 6,7.

[10]E. Malavasi, D. Pandini, “Optimum CMOS Stack Generation with Analog Constraints”, IEEE Trans. on CAD, pp. 107-122, Jan. 1995.