Integrating Dynamic Power Management in the Design Flow

António Mota, Nuno Ferreira, Arlindo Oliveira, José Monteiro
IST/INESC
{mota,nmlf,aml,jcm}@inesc.pt

Abstract

Power dissipation has recently emerged as one of the most critical design constraints. A wide range of techniques has already been proposed for the optimization of logic circuits for low power. Power management methods are among the most effective techniques for power reduction. These methods detect periods of time during which parts of the circuit are not doing useful work and shut them down by either turning off the power supply or the clock signal.

In this work, we describe the integration of dynamic power management tools in a design flow. The designer can thus easily apply these techniques to the design, evaluate the optimization achieved and decide if the changes are worthwhile to include in the final design.

We have used this design flow with a real project. An HDLC controller has been designed and these power management techniques have been applied.

Keywords: Power optimization, power management, design flow

INTRODUCTION

Power consumption has become a primary concern in the design of integrated circuits. Two independent factors have contributed for this. On one hand, low power consumption is essential to achieve longer autonomy for portable devices. On the other hand, increasingly higher circuit density and higher clock frequencies are creating heat dissipation problems, which in turn raise reliability concerns and lead to more expensive packaging.

In the last few years, research on techniques for low power at various levels of design has intensified. Techniques based on disabling the input/state registers when some input conditions are met have been proposed and shown to be

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35498-9_57
L. M. Silveira et al. (eds.), *VLSI: Systems on a Chip*
© IFIP International Federation for Information Processing 2000

among the most effective in reducing the overall switching activity in sequential circuits. The disabling of the input/state registers is decided on a clock-cycle basis and can be done either by using a register load-enable signal or by gating the clock. A common feature in these methods is the addition of extra circuitry that is able to identify input conditions for which some or all of the input/state registers can be disabled. This class of techniques is sometimes referred to as *logic level* or *dynamic power management.*

In this work, the objective was to test these power management techniques on a real project. First, some of the techniques were integrated in the design flow: from a high-level VHDL description the circuit is synthesized to logic level; tools can be applied at this level and an evaluation of the power reduction can be obtained; the power managed design can then be mapped to an FPGA and tested under normal operating conditions.

Using this design flow, a power managed HDLC controller has been designed. This circuit is part of an expansion board for a PC that implements the interface to an ISDN (Integrated Services Digital Network) line.

In Section 1, the power management techniques that have been integrated in the design flow are presented. The complete design flow is described in Section 2. A brief description of the circuit that has been designed is given in Section 3. Section 4 presents the experiments that were carried out and the power reductions obtained. A discussion about these results is given in Section 5.

1. DYNAMIC POWER MANAGEMENT

During normal operation of well designed CMOS circuits, power consumption is determined by the switching activity in the circuit (Chandrakasan et al., 1992). Under a generally accepted simplified model, the power dissipation at the output of a gate g in a logic circuit is given by:

$$P_g = \frac{1}{2} \cdot C_g \cdot V_{DD}^2 \cdot f \cdot N_g \tag{1.1}$$

where P_g denotes power, V_{DD} the supply voltage, and f the clock frequency. C_g represents the capacitance gate g is driving and N_g is the switching activity at the output of gate g, *i.e.*, the average number of gate output transitions per clock cycle. The product $C_g \cdot N_g$ is called *switched capacitance.*

Most power optimization techniques at different levels of abstraction target the minimization of the switched capacitance in the circuit (Devadas and Malik, 1995). So called power management techniques shutdown blocks of hardware for periods of time in which they are not producing useful data. Shutdown can be accomplished by either turning off the power supply or by disabling the clock signal. A system-level approach is to identify idle periods for entire modules and turn off the clock lines for these modules for the duration of the idle periods (Chandrakasan et al., 1992, Chapter 10). However, there still is no

real methodology for system level power management. It is up to the designer to devise a strategy for power management for a particular project.

In contrast, a few techniques have been proposed at the gate level, e.g., Alidina et al., 1994; Benini et al., 1996a; Benini et al., 1996b; Chow et al., 1996; Monteiro and Oliveira, 1998; Tiwari et al., 1995. These techniques are based on disabling the input/state registers when some input conditions are met, either by using a register load-enable signal or by gating the clock. In this situation there will be zero switching activity in the logic driven by input signals coming from the disabled registers. The main difference from system-level power management is that the shutdown of hardware is decided on every clock cycle, hence the name *dynamic power management.*

In order to decide whether to load or not new values into the registers, some extra logic has to be added to the original circuit. Naturally, this is redundant circuitry, increasing both area and power dissipation. In fact, the basic tradeoff in dynamic power management techniques is area for lower power consumption. The argument is that power is becoming the main constraint and that area is no longer critical.

Even if area is not a concern, this extra logic, which is active all the time, translates to additional power consumption. The power savings obtained by shutting down the registers must compensate for this overhead. The more complex this logic is, the larger the power overhead. Thus, on one hand, the more input conditions that are targeted for register shutdown, the larger the period of time during which the original circuit is being powered down. On the other hand, the larger the power penalty from the extra logic. In general, there is an optimum size for the extra logic and the goal of the different power management techniques at the logic level is to find this optimum.

The motto of logic-level power management is to have a small amount of logic that is active most of the time, but that is able to shutdown a much larger circuit during that time.

We have integrated two of these power management techniques in our design flow, which we briefly describe next.

1.1 PRECOMPUTATION

Precomputation, as proposed by Alidina et al., 1994 was one of the first logic-level shutdown methods proposed. In this method a simple combinational circuit (the precomputation logic) is added to the original circuit. Under certain input conditions, the precomputation logic disables the loading of all or a subset of the input registers. Under these input conditions, no power is dissipated in the portions of the original circuit with only disabled registers as inputs.

The basic architecture of this method is shown in Figure 1. A is the original combinational logic. Blocks g_1 and g_2 constitute the precomputation logic and are designed such that they are a function of a subset of the inputs to A. Power

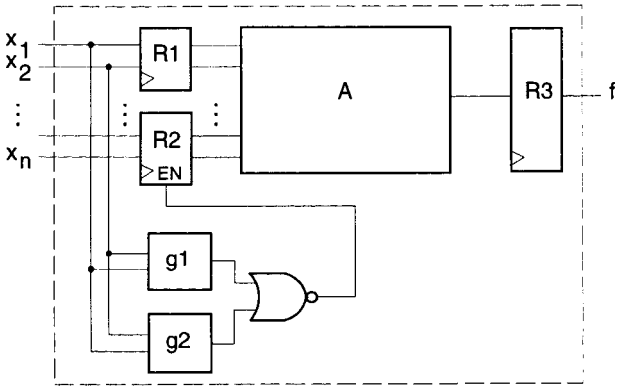

Figure 1 The precomputation architecture.

dissipation in the original circuit A is reduced when the outputs of either g_1 or g_2 evaluate to 1.

The choice and the number of inputs to use for the g_1 and g_2 functions is critical. The more inputs used, the highest the probability the precomputation logic will be active, thus disabling logic in block A. However, the size of the precomputation logic, a circuitry overhead that is active all the time, also increases, thus offsetting the gains obtained by disabling A a larger fraction of the time.

Once the number of inputs to the precomputation logic is fixed, the input selection is based on the probability that the outputs can be computed without the knowledge of a specific input, *i.e.*, the size of the observability don't-care set (ODC):

$$\mathrm{ODC}_i = f_{x_i} \cdot f_{\overline{x_i}} + \overline{f}_{x_i} \cdot \overline{f}_{\overline{x_i}} \tag{1.2}$$

Inputs with lowest $prob(\mathrm{ODC}_i)$ are selected to be in the precomputation logic.

1.2 FINITE STATE MACHINE DECOMPOSITION

Decomposition of finite state machines (FSMs) targeted for low power has been recently proposed by Chow et al., 1996 and Monteiro and Oliveira, 1998. The basic idea is to decompose the STG of the original finite state machine into two coupled STGs that together have the same functionality as the original machine. Except for transitions that involve going from one state in one sub-FSM to a state in the other, only one of the sub-FSMs needs to be clocked. The techniques described in Chow et al., 1996 and Monteiro and Oliveira, 1998 differ both in the way the partitioning of the states is performed and in the structure of the final circuit.

We have selected the technique of Monteiro and Oliveira, 1998 to integrate in our design flow. This technique follows the standard general decomposition

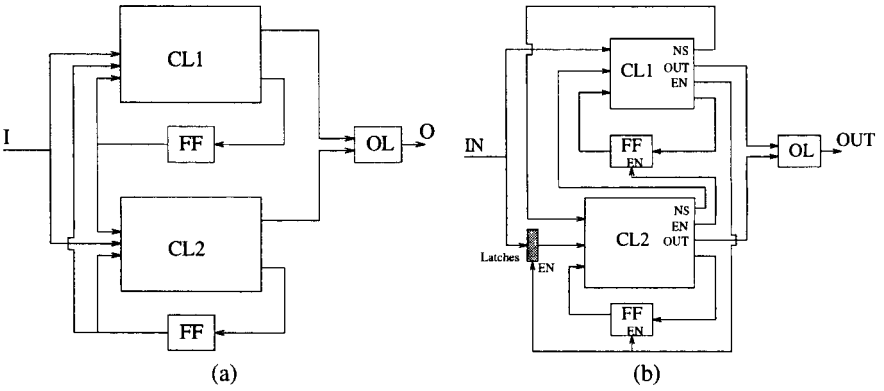

Figure 2 FSM general decomposition: (a) traditional, (b) with power management.

structure, which is shown in Figure 2(a). The selection of the states is such that only a small number is selected for one of the sub-FSMs. This selection consists in searching for a small cluster of states such that summation of the probability of transitions between states in the cluster is high and with a very low probability of transition to and from states outside of the cluster. The aim is to have a small sub-FSM that is active most of the time, disabling the larger sub-FSM. The reason for requiring a small number of transitions to/from the other sub-FSM is that this corresponds to the worst situation when both sub-FSMs are active.

The power optimized structure is shown is Figure 2(b). Each sub-FSM has an extra output that disables the state registers of the other sub-FSM. This extra output is also used to stop transitions at the inputs of the large sub-FSM. To avoid the area/power overhead incurred by adding latches, and since when this technique is effective the small sub-FSM is in operation most of the time, the inputs to the small sub-FSM are not filtered.

2. DESIGN FLOW

For the designers to be able to use dynamic power management tools, they have to be completely integrated in the design flow. We have done this for the two techniques described in the previous section. The design flow graph is depicted in Figure 3.

We start with a RTL description of the design in VHDL. We are using Synopsys to synthesize this VHDL description to gate-level. The power management techniques, precomputation and FSM decomposition, are applied at this level. However, these tools are available inside Berkeley's SIS package (Sentovich et al., 1992) and there is no common circuit description language accepted by

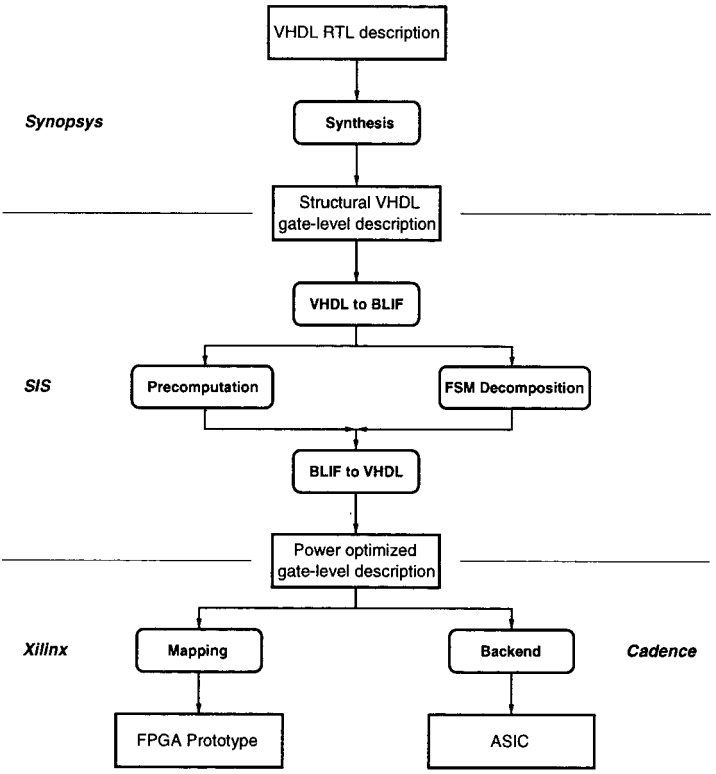

Figure 3 Design flow integrating the dynamic power management tools.

both frameworks. We have developed a translator between a mapped gate-level VHDL description and a mapped BLIF description. A translator in the opposite direction was also developed. The translation process involves mapping the gate-level description to a generic library that is common to both Synopsys and SIS.

Given the mapped BLIF description, the design is read into SIS and the power management tools can be applied. These tools are still not fully automatic in the sense that there are some parameters that the user has to specify. For precomputation, the number of inputs to use for the precomputation logic has to be specified. Thus, the designer may have to try different values in the search for the best possible power savings. Similarly, in the case of FSM decomposition, the number of states for the small machine has to be given. The typical behavior of the power estimates in this search is, as each of these values increase, the power reduces due to the increase of the fraction of time that the registers are stopped. After a certain value, the power starts to go up as the complexity of the extra circuitry required to generate the disabling signal increases significantly, offsetting the gains from the blocking of the registers.

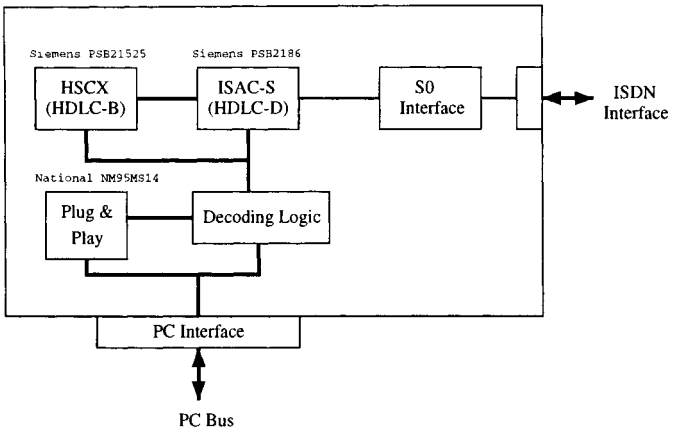

Figure 4 Schematic of the PCBIT board.

The power estimation tool in SIS permits the evaluation of the solution found by the power optimization tool. Translators from BLIF to SLS (van Genderen, 1989) and Spice formats have also been developed to allow more precise power estimates, at switch and electrical levels.

Once the designer has found the optimum in terms of power optimization, this circuit is again mapped to the common generic library and the VHDL description is obtained using the BLIF to VHDL translator. Again inside Synopsys, the circuit can be targeted for an ASIC or FPGA implementation. Depending on the target, the circuit is mapped to the corresponding library and the backend part of the process is performed using Cadence or Xilinx, respectively.

3. HDLC CONTROLLER

In this section is described the design that was used to demonstrate the integration of the dynamic power optimization tools in the design flow and their effectiveness when applied to a real design.

The PCBIT board is a PC expansion board currently being commercialized that implements an ISDN interface, allowing for a PC to be connected to an ISDN line. This board is being redesigned to a PC-Card format. One of the concerns is the power consumption of the new board. To this end, the plan is to integrate all the functionality of the board in a single ASIC and explore to the full extent the use of all power optimization techniques.

The results reported in this paper refer only to the part that implements the functionality of the Siemens PSB21525 chip. This circuit handles the layer 2 of the ISDN protocol (ITU, 1993), namely the HDLC protocol (High-level Data Link Control) for channels B1 and B2. On one side it interfaces using IOM-2 frames (Siemens, 1991) with the circuit that is responsible for layer 1.

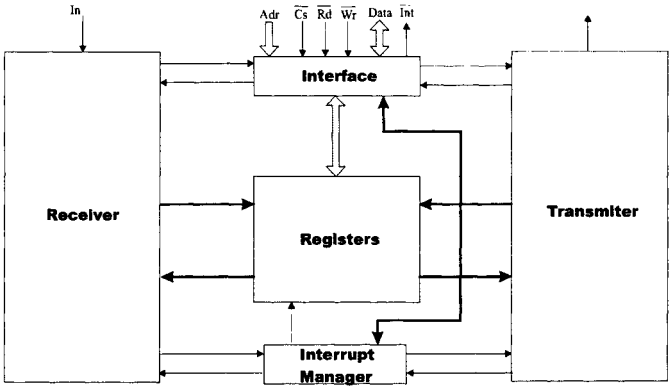

Figure 5 Block diagram of the demonstrator.

On the other side it is connected to the PC ISA bus. The block diagram of the circuit is depicted in Figure 5.

The two main modules in the circuit are the transmitter and the receiver. The transmitter receives data from the PC through a FIFO and constructs a IOM-2 frame to communicate with the layer 1 circuit. This process includes generating flags, bit-stuffing, and the frame check sequence (FCS). The receiver performs the reverse operation: decodes the information from the IOM-2 frame, writes the data to a FIFO and sends an interrupt to the PC to notify it that there is data available. The block diagrams for each of these modules are shown in Figure 6. Each is basically made of finite state machines that, with the help of counters, tracks the field of the IOM-2 frame that is being read or written. The FIFOs hold the data that has been received from the PC and is waiting to be processed by the circuit and vice-versa, data that has been read from the ISDN line and is waiting to be read by the PC. The power management techniques were applied mainly to the finite state machines in these circuits.

There are three other modules in the circuit of Figure 5. The interface module includes a decoder to generate the chip-select for the board and some latches for the interface with the PC bus. All the transfers to the PC bus are interrupt driven and this is handled by the interrupt-manager module. Finally, the registers shown in the figure are used to configure the board during system reset and to communicate with the ISA bus, namely by holding pending interrupts.

4. RESULTS

The circuit of the previous section was first described in VHDL and verified using logic simulation. It has been synthesized using Synopsys and mapped to a Xilinx FPGA, following the design flow presented in Section 2. This

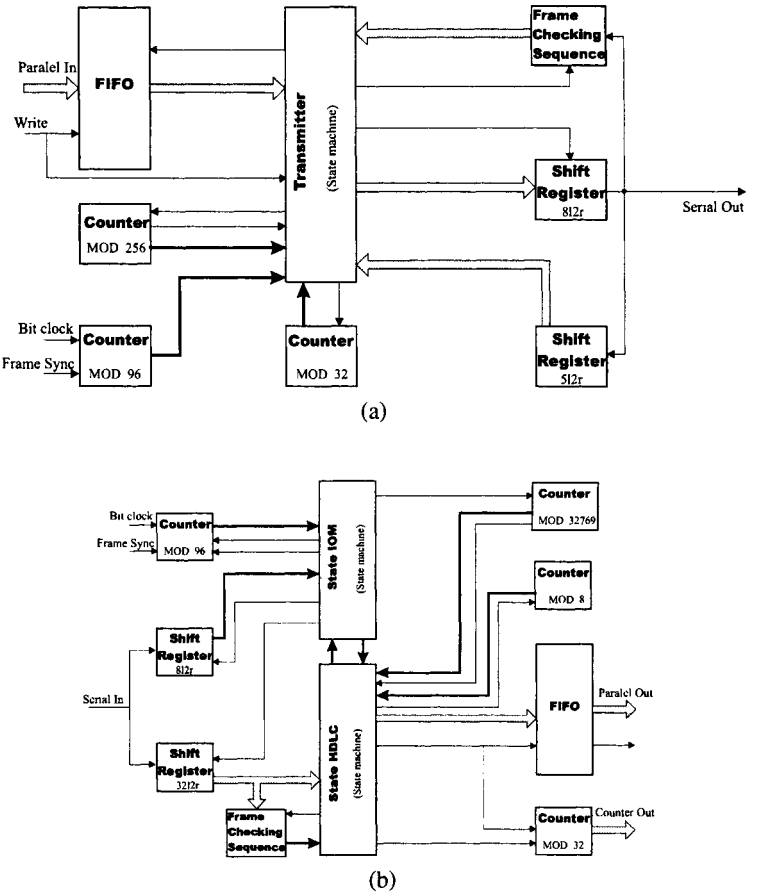

Figure 6 Block diagram: (a) transmitter, (b) receiver.

circuit was then tested under real operation by replacing the original Siemens PSB21525 circuit in the PCBIT board (see Figure 4) with the FPGA.

The power management techniques described in Section 1, precomputation and finite state machine decomposition, were applied separately to the finite state machines inside the transmitter and receiver modules. The FIFOs and counters shown in Figure 6 were not included in the FSMs since they make the state space much larger and increase significantly the number of transitions between states, two factors that necessarily limit the effectiveness of the power management techniques under test.

Table 1 Statistics for the circuits for which power management was applied.

FSM	PIs	POs	Registers	States
Trm	67	35	10	432
Rec	12	94	14	7,680

The statistics for the two FSMs are given in Table 1, namely, the number of primary inputs (PIs), number of primary outputs (POs), the number of registers and the number of states of the FSM.

Experiments with the power management technique based on FSM decomposition were carried out using a range of 1 to 10 for the number of states in the small FSM. All the solutions found by the tool led to an increase of the power consumption. Therefore this technique was not effective on these two test cases.

Precomputation was tested with 1 to 15 inputs in the precomputation logic. This technique was also not effective for the receiver. The results were not very good for the transmitter either: power reduction was only achieved when using 1 input for the precomputation logic and the reduction was a mere 0.1%.

Although these automatic tools found no good solution, still some power management for the circuit can be accomplished by inspection. The IOM-2 frame is composed of 12 windows of 1 byte each, as shown in Figure 7. The first two windows correspond to channels B1 and B2, respectively. Since this circuit only handles these channels, clock-gating can be used outside these windows to stop most counters in the circuit. Instead of adding additional logic, the `FSC` signal is being used as the gating signal. Figure 8 shows a switch-level simulation of the circuit using the gated-clock signal, `bcl_aux`, instead of the regular clock, `bcl`.

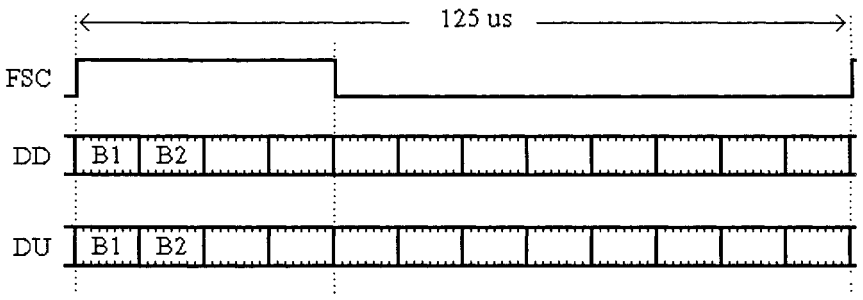

Figure 7 One frame of the IOM-2 protocol.

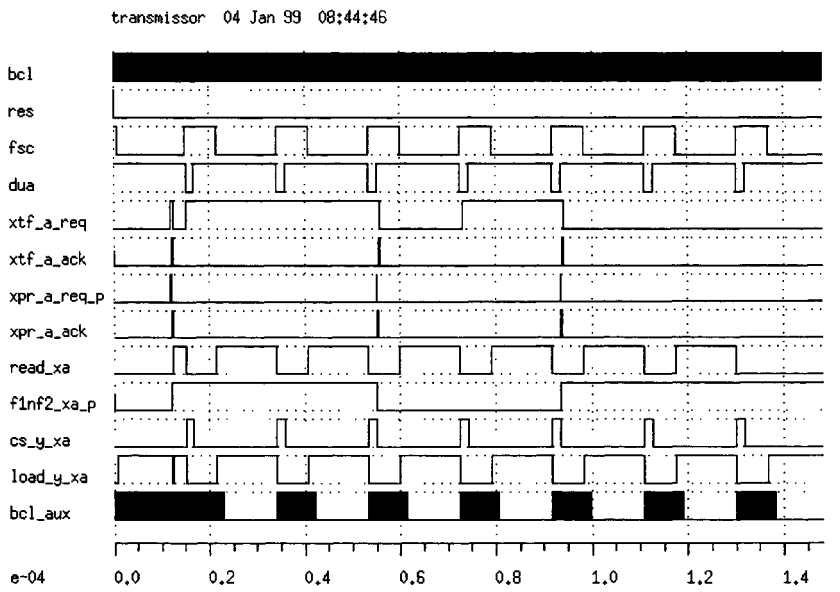

Figure 8 Simulation of the power managed version of the Transmitter.

Table 2 Statistics for the circuits for which power management was applied.

FSM	Original		FSM	Precomputation				Clock-gating			
	Tran.	Pwr	Decomp.	Tran.	% T	Pwr	% P	Tran.	% T	Pwr	% P
Trm	1,804	0.91	-	1,832	1.5	0.91	0.1	1,821	0.90	0.62	31.7
Rec	4,629	4.77	-		-			4,655	0.6	3.12	34.6

A summary of the results obtained is presented in Table 2. The power estimates in mW for the original and optimized circuit, together with the percentage variation are shown. These estimates were obtained using the switch-level simulator SLS (van Genderen, 1989). Similar statistics are given for the number of transistors in the circuit.

5. CONCLUSIONS

In this work, we have described a framework that integrates tools for dynamic power management in the design flow. The designer can thus easily experiment different techniques, evaluate their effectiveness and decide whether or not to include them in the final design.

The tools currently available in the design flow are precomputation and finite state machine decomposition. We have applied them on a real project, an HDLC controller. The typical circuits for which these techniques are applicable are controller-type sequential logic circuits. Hence, the finite state machines inside the HDLC circuit were selected as modules to be power managed. Regrettably, for this circuit, the technique based on finite state machine decomposition was not effective at all. No solution was found that reduced the power dissipation of the circuit. Moreover, even precomputation only achieved negligible power gains. Therefore, none of these techniques will be incorporated in the final design. Instead, a simple clock-gating mechanism will be implemented, with expected power savings around 30%.

Despite the negative results obtained with the automatic power management tools for this particular circuit, the integration of these techniques in the design flow makes them easy to use and, in general, worth the effort of applying them to power critical designs.

6. REFERENCES

Alidina, M., Monteiro, J., Devadas, S., Ghosh, A., and Papaefthymiou, M. (1994). Precomputation-Based Sequential Logic Optimization for Low Power. *IEEE Transactions on VLSI Systems*, 2(4):426–436.

Benini, L., Siegel, P., and Micheli, G. D. (1996a). Automatic Synthesis of Low-Power Gated-Clock Finite-State Machines. *IEEE Transactions on Computer-Aided Design*, 15(6):630–643.

Benini, L., Vuillod, P., Coelho, C., and Micheli, G. D. (1996b). Synthesis of Low-Power Partially-Clocked Systems from High-Level Specifications. In *9th International Symposium on System Synthesis*.

Chandrakasan, A., Sheng, T., and Brodersen, R. (1992). Low-Power CMOS Digital Design. *IEEE Journal of Solid-State Circuits*, 27(4):473–484.

Chow, S.-H., Ho, Y.-C., and Hwang, T. (1996). Low Power Realization of Finite State Machines – A Decomposition Approach. *ACM Transactions on Design Automation of Electronic Systems*, 1(3):315–340.

Devadas, S. and Malik, S. (1995). A Survey of Optimization Techniques Targeting Low Power VLSI Circuits. In *Proceedings of the* 32^{nd} *Design Automation Conference*, pages 242–247.

ITU (1993). *Digital Subscriber Signalling, System No. 1*. ITU-T Recommendation Q.921.

Monteiro, J. and Oliveira, A. (1998). Finite State Machine Decomposition for Low Power. In *Proceedings of the* 35^{th} *Design Automation Conference*, pages 758–763.

Sentovich, E., Singh, K., Moon, C., Savoj, H., Brayton, R., and Sangiovanni-Vincentelli, A. (1992). Sequential Circuit Design Using Synthesis and Optimization. In *Proceedings of the International Conference on Computer Design: VLSI in Computers and Processors*, pages 328–333.

Siemens (1991). *IOM-2 Interface Reference Guide*.

Tiwari, V., Ashar, P., and Malik, S. (1995). Guarded Evaluation: Pushing Power Management to Logic Synthesis/Design. In *Proceedings of the International Symposium on Low Power Electronics and Design*, pages 221–226.

van Genderen, A. (1989). SLS: An Efficient Switch-Level Timing Simulator Using Min-Max Voltage Waveforms. In *Proceedings of the VLSI'89 Conference*, pages 79–88.