

Design of Low Power FSM Using Verilog in VLSI

Himani Mittal, Dinesh Chandra, and Arvind Tiwari

J.S.S. Academy of Technical Education,
Noida, India
himanimit@yahoo.co.in

Abstract. With day by day increase in integration density of CMOS technology the concern for area usage for VLSI circuits is increased, giving more importance to timing and power dissipation constraints. Controllers are running continuously so critical for power (while parts of the data path may be shut down), and for timing because the delay through the controller may constrain the delay through the data path.

In this work we propose a procedure for the decomposition of a network of interacting FSMs starting from a single state-table specification. The straightforward single machine implementation is called *undecomposed FSM*. We call *decomposed FSM* the interacting FSM implementation. The sub-machines in the decomposed FSM communicate through a set of l interface signals. The decomposed implementation has low power dissipation because *one single sub-machine is clocked* at any given time and it controls the outputs values while all other sub-machines are idle: they do not receive the clock signal and dissipate little power. When a sub-machine terminates its execution, it sends an *activation signal* to another sub-machine which takes control of the computation, then it de-activates itself.

Keywords: Heuristic Approach, Mealy and Moore Machines, STM (State Transition Matrix), ULP (Ultra Low Power), Power saving, Entropy, State Codes.

1 Introduction

Power consumption has become a major design parameter in the project 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 static CMOS circuits, the probabilistic switching activity of nodes in the circuit is a good measure of the average power dissipation of the circuit. Methods that can efficiently compute the average switching activity, and thus power dissipation, in CMOS combinational [10] and sequential [9] circuits have been developed. In this work, we are concerned with the problem of optimizing logic-level sequential circuits for low power. This problem has received some attention recently. Several techniques for state assignment have been presented which aim at

reducing the average switching activity of the present state lines, and consequently of the internal nodes in the combinational logic block (see for example [7]). Retiming has also been tailored so that the distribution of the registers within the logic block minimizes the total amount of glitching in the **sequential circuit** [5]. Techniques based on disabling the input/state registers when some input conditions are met have been proposed. The most effective in reducing the overall switching activity in sequential circuits [1], [2], [3]. The disabling of the input/state registers is decided on a clock-cycle basis and can be done either by using **B**register load-enable signal or by gating the clock. A common feature of these methods is the addition of extra circuitry that is able to identify input conditions for which some or all of the input/state register: can be disabled. In this situation there will be zero switching activity in the logic driven by input signals coming from the disabled registers. This class of techniques is sometimes referred to as *dynamical power management*. The method we propose in this paper falls into this class of techniques. We use finite state machine (FSM) decomposition to obtain the conditions for which a significant part of the registers in the circuit can be disabled. The original FSM is divided into two sub-FSM where one of them is significantly smaller than the other. Except for transitions that involve going from one state in one sub-machine to *i*state in the other, only one of the sub-machines needs to be clocked. By selecting for the small sub-FSM a cluster of states in which the original FSM has a high probability of being in, most of the time will be disabling all the state registers in the larger sub-FSM. The overhead associated with the FSM decomposition makes this method not very effective for typical FSMs with a small number of states. However, for large machines, impressive gains up to 80% power reduction are possible

1.1 Estimation of Power

Entropy is a measure of the randomness carried by a set of discrete events observed over time. In the studies of the information theory, a method to quantify the information content C_i of an event E_i in this manner is to take logarithmic of the event probability

$$C_i = \log_2 (1/P_i)$$

Since $0 \leq P_i \leq 1$, the logarithmic term is non negative and we have $C_i > 0$.

The average information contents of the system is the weighted sum of the information content of C_i by its occurrence probability. This is also called the entropy of the system.

$$H(X) = \sum_{i=1}^{m-1} p_i \log_2 \frac{1}{p_i}$$

1.2 Problems Faced

The major shortcoming of this solution, however, is that lowering the supply voltage affects circuit speed. As a consequence, both design and technological solutions must be applied in order to compensate the decrease in circuit performance introduced by reduced voltage. In other words, speed optimization is applied first, followed by supply voltage scaling, which brings the design back to its original timing, but with a lower power requirement. A similar problem, i.e., performance decrease, is encountered when power optimization is obtained through frequency scaling. Techniques that rely on reductions of the clock frequency to lower power consumption are thus usable under the constraint that some performance slack does exist. Although this may seldom occur for designs considered in their entirety, it happens quite often that some specific units in a larger architecture do not require peak performance for some clock/machine cycles. Selective frequency scaling (as well as voltage scaling) on such units may thus be applied, at no penalty in the overall system speed. Optimization approaches that have a lower impact on performance, yet allowing significant power savings, are those targeting the minimization of the *switched capacitance* (i.e., the product of the capacitive load with the switching activity). Static solutions (i.e., applicable at design time) handle switched capacitance minimization through area optimization (that corresponds to a decrease in the capacitive load) and switching activity reduction via exploitation of different kinds of signal correlations (temporal, spatial, spatial-temporal). Dynamic techniques, on the other hand, aim at eliminating power wastes that may be originated by the application of certain system workloads (i.e., the data being processed).

2 Heuristic Algorithm Approach

- (1) Crossing Transition should be minimum to reduce power consumption.[6]
- (2) First bit of state code is Control bit distinguish between Sub machines.
- (3) To distinguish between states within each submachine inner bits are needed.

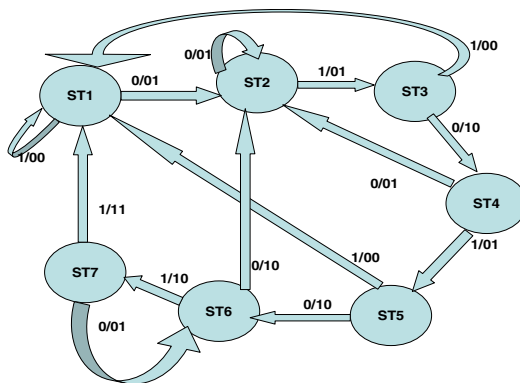


Fig. 1. Original FSM

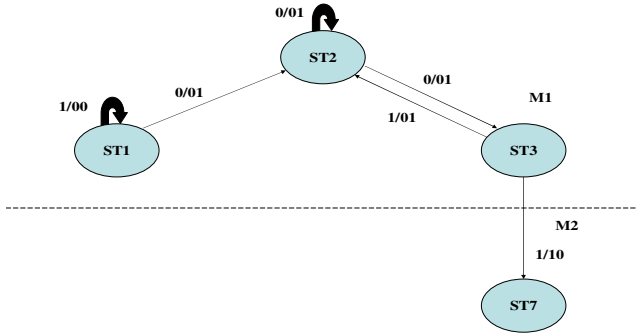


Fig. 2. Upper half FSM

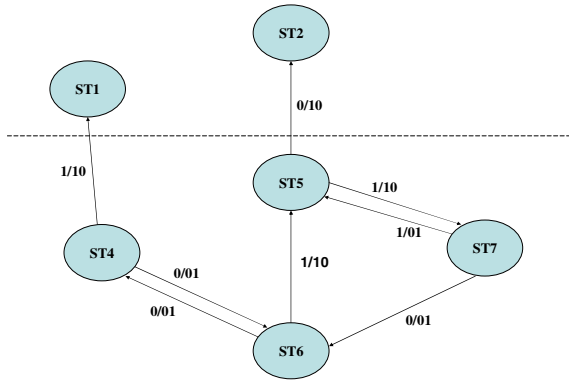


Fig. 3. Lower half FSM

2.1 Simulation of FSM

Following are the simulations over ModelSim for FSM

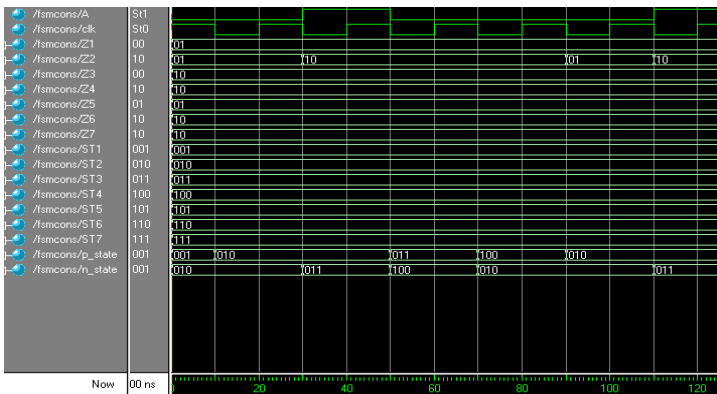


Fig. 4. State Transition in FSM for an input sequence

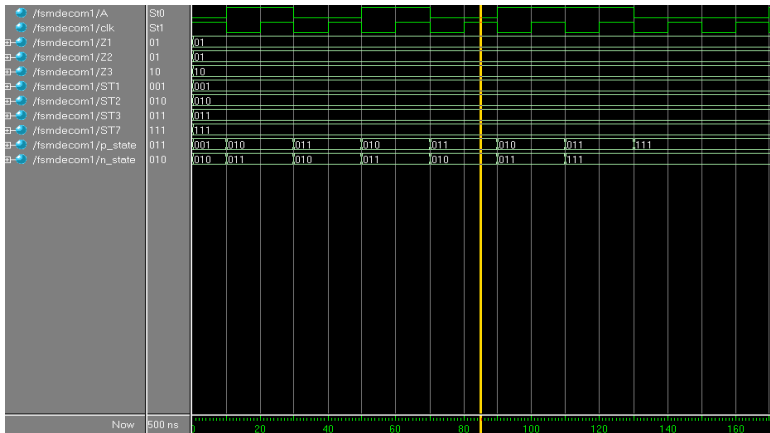


Fig. 5. State transition in Upper half FSM

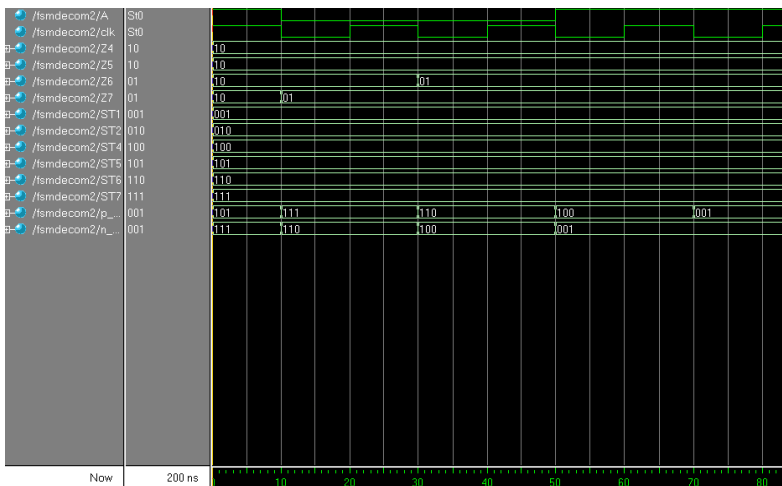


Fig. 6. State transition in Lower half subFSM

2.2 Observations

Power consumption in original FSM is calculated as 5.39mW.
 Power consumption in decomposed sub FSM is 3.47mW & 3.64mW.
 On taking the average, 3.55mW.
 Percentage in power saving[3] = $(P_{\text{originalFSM}} - P_{\text{subFSM}})$

$$\begin{aligned}
 & \frac{P_{\text{originalFSM}}}{5.39} \\
 & = 34.13\%
 \end{aligned}$$

3 Architecture of FSM

There is one control signal, control₁, which is the output of the first flip-flop; one 1x2 decoder which will generate the enable signals e1 and e2 for submachine M1 and M2; four AND gates A, B, C, D in front of M1 and four AND gates E,F,G,H in front of M2 which will block the state and primary input signals from propagating through M1 and M2, respectively; three multiplexers mux1, mux2, mux3 which will determine whether the next state registers will be loaded from M1 or M2; and two multiplexers mux4, mux5 which will determine the correct output signals[3]. Suppose submachine M1 is active and is in state s6. Since the first bit of the state code (the control signal control₁) of s6, which is the input to the 1x2 decoder, is 0, the output signal from the decoder to Com1, e1, is 1 which will turn on all the AND gates A, B, C, D in front of submachine M1. Thus, all signals fed to the circuit corresponding to submachine M1 will propagate through. However, the output signal from the decoder to Com2, e2, is 0 which will turn off all the AND gates D, E, F, G in front of submachine M2. Thus, all signals fed to the circuit corresponding to submachine M2 will be blocked. Similarly, with the control signal control₁ being 0, signals for the next state flip-flops and output will come from Com1 through the, F, G, H in front of M2 which will block the state and 2x1 multiplexers. Now, if the input is 0, submachine M1 will transit to state START. Since the first bit of the state code of START is also 0, submachine M1 will stay active in the next clock cycle. On the other hand, if the input is 1, submachine M1 will transit to state s2. Since the first bit of the state code (the control signal control₁) of s2 is 1, the output signal of the decoder e2 will be 1 which will allow inputs to propagate through the AND gates E, F, G, H and Com2, and thus turn on the submachine M2. In the meantime, the value

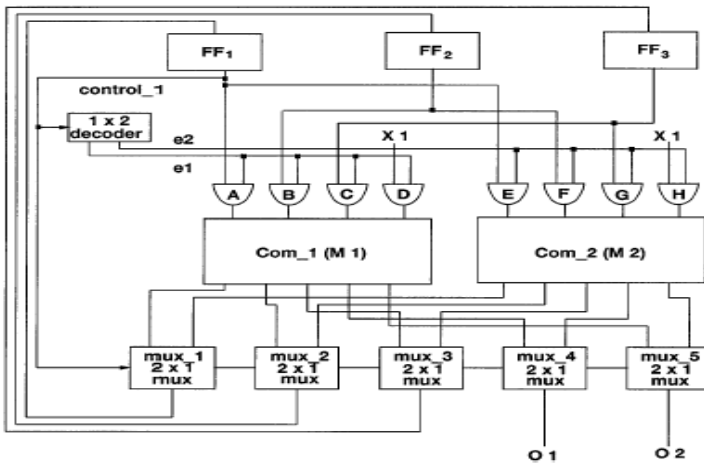


Fig. 7. Block diagram of low power architecture

of e1 is 0 which will inhibit the propagation of inputs through the AND gates A, B, C, D, and thus set all the inputs to submachine M1 to 0s in the next clock cycle, rendering submachine M1 inactive. If in subsequent clock cycles state transitions are confined to within submachine M2 , the value of e1 will remain 0 and submachine M1 will remain inactive.

3.1 Simulation at Architectural Level

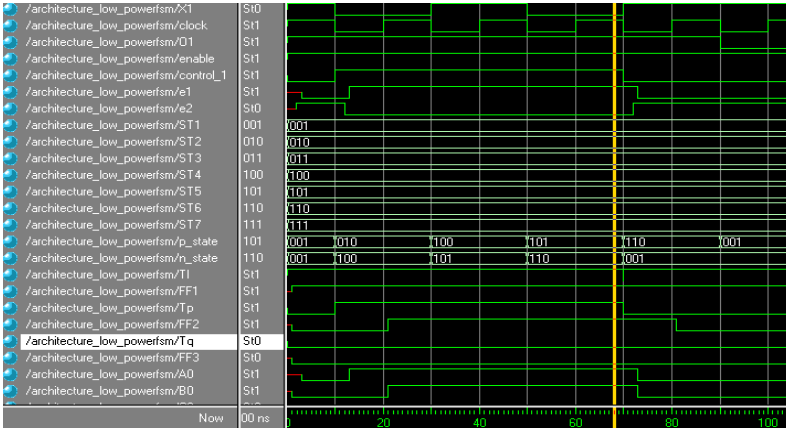


Fig. 8.

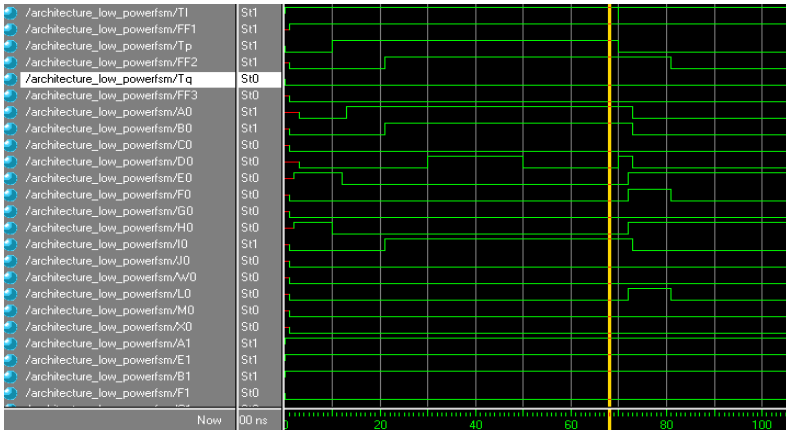


Fig. 9.

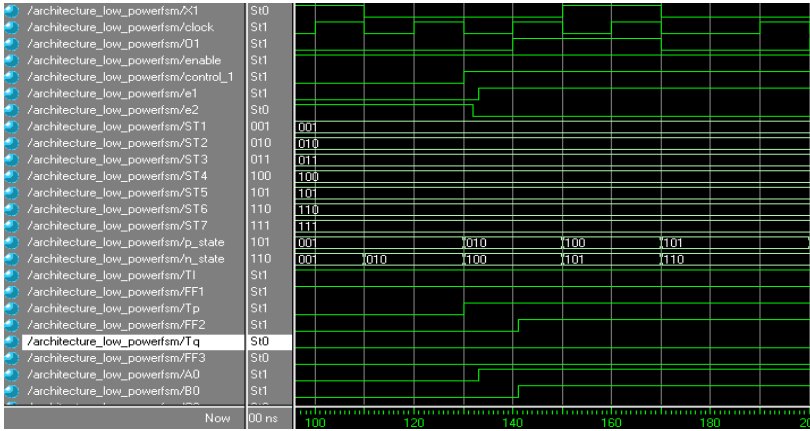


Fig. 10.

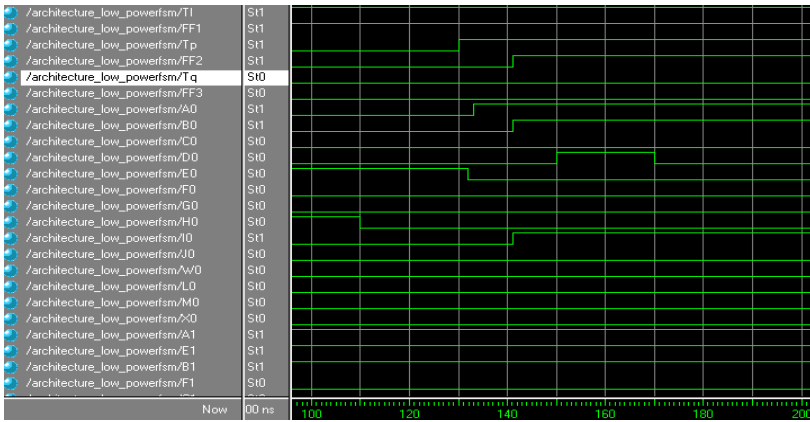


Fig. 11.

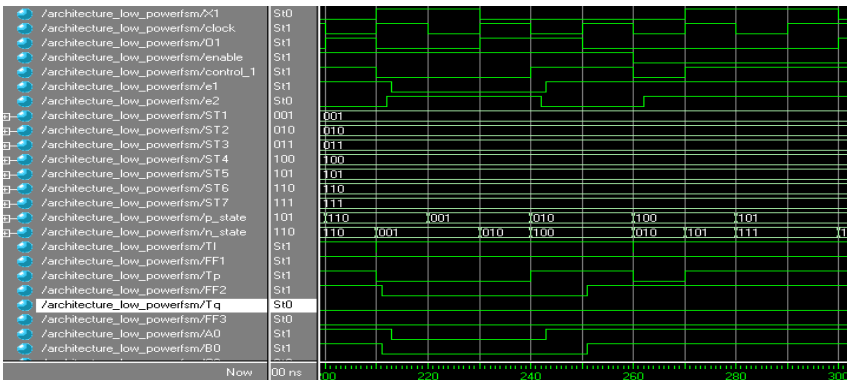


Fig. 12.

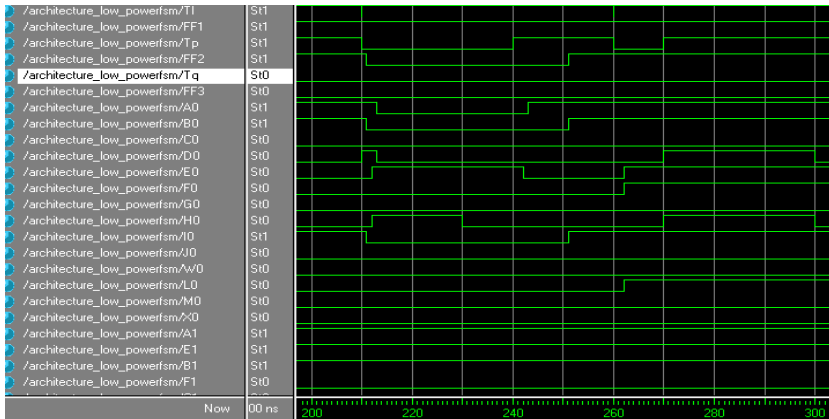


Fig. 13.

Above are the simulated output waveform at an architecture level .

4 Conclusion

Methodology for the decomposition of finite state machines targeted towards low power dissipation is presented. Heuristic algorithm approach gives good decomposition techniques to obtain a state machine at, for the majority of the large examples tested, exhibits a much smaller power dissipation than the original. The results show that savings of up to 33% are possible in some of the examples. Despite the good quality of the results obtained, there are several rections for future work that may improve these results and extend e **range** of applicability of the technique. The single most important direction is probably the extension of is work to the case where the state transition graph cannot be explicitly described. In our experiments, we verified that the bottleneck sides in the extraction of the STG from the sequential circuit description. Because the decomposition algorithm works with an explicit description of the STG, it is not possible to apply the method to machines where the STG cannot even be extracted. However, it may be possible to apply the decomposition idea even in this case, if a set of transitions with high enough probability exist to justify the method. The basic idea is that, in many cases, this set of transitions can be identified using Monte Carlo methods even without doing a complete traversal of the STG. With this set of transitions available, it is possible to perform a partition of the STG, considering only the states involved in these transitions and using only an implicit representation of the STG. Once the STG is partitioned, the rest of the method is directly applicable, making it feasible to use it in cases where the machine is too large to permit the extraction of the STG. Another interesting direction for future research is on the automatic selection of the size of the **small** machine. Although, in some cases, significant gains can be obtained with a variety of sizes for this machine, in other cases the result depends strongly on the

adequate selection of the value of this parameter. It may be possible to modify the cost function described above to automatically include a term that depends on the number of states, thereby removing the need for the user to specify the value of this parameter or to perform a search for its right value. Finally, it is clear that the results obtained in this paper can be improved if a more efficient implementation of the decomposition strategy is selected. In particular, it should be possible to reduce the overhead incurred by the addition of the extra outputs to each sub-FSM, by encoding these outputs in a different way that take into account the encoding of each of the sub-machines.

References

- [1] Alidina, M., Monteiro, J., Devadas, S., Ghosh, A., Papaefthymiou, M.: Precomputation-based sequential logic optimization for low power. *IEEE Trans. VLSI Syst.* 2, 426–436 (1994)
- [2] Benini, L., De Micheli, G., Lioy, A., Macii, E., Odasso, G., Poncino, M.: Computational kernels and their application to sequential power optimization. In: *Proc. 35th Design Automation Conf.*, pp. 764–769 (June 1998)
- [3] Benini, L., Siegel, P., De Micheli, G.: Automatic synthesis of lowpower gated-clock finite-state machines. *IEEE Trans. Computer-Aided Design* 15, 630–643 (1996)
- [4] Benini, L., Vuillod, P., Coelho, C., De Micheli, G.: Synthesis of lowpower partially-clocked systems from high-level specifications. Presented at the 9th Int. Symp. System Synthesis (November 1996)
- [5] Chow, S.-H., Ho, Y.-C., Hwang, T.: Low power realization of finite state machines—A decomposition approach. *ACM Trans. Design Automat. Electron. Syst.* 1(3), 315–340 (1996)
- [6] Devadas, S., Newton, A.: Decomposition and factorization of sequential finite state machines. *IEEE Trans. Computer-Aided Design* 8, 1206–1217 (1989)
- [7] Hachtel, G., Hermida, M., Pardo, A., Poncino, M., Somenzi, F.: Reencoding sequential circuits to reduce power dissipation. In: *Proc. Int. Conf. Computer-Aided Design*, pp. 70–73 (November 1994)
- [8] Kernighan, B., Lin, S.: An efficient heuristic procedure for partitioning graphs. *Bell Syst. Tech. J.*, 291–307 (February 1970)
- [9] Monteiro, J., Devadas, S., Ghosh, A.: Retiming sequential circuits for low power. In: *Proc. Int. Conf. Computer-Aided Design*, pp. 398–402 (November 1993)
- [10] Monteiro, J., Oliveira, A.: Finite state machine decomposition for low power. In: *Proc. ACM/IEEE Design Automation Conf.*, pp. 758–763 (June 1998)