Automatic RTL Code Generation for Low Power

Wei Wang, Li Li and Ken Choi

VLSI Design and Automation Laboratory

Department of Electrical and Computer Engineering

Illinois Institute of Technology

Chicago, U.S.A.

From literature review, we find 22 techniques for RTL power reduction as follows.

1. Glitch Reduction

a) Basic Approach: In the design-for-low-power techniques based on glitch reduction for register-transfer level circuits, we analyze the generation and propagation of glitches in both the control and data path parts of the circuit. Based on the analysis, we develop techniques that attempt to reduce glitching power consumption by minimizing generation and propagation of glitches in the RTL circuit. Our techniques include restructuring multiplexer networks (to enhance data correlations, eliminate glitchy control signals, and reduce glitches on data signals), clocking control signals, and inserting selective rising/falling delays. Our techniques are suited to control-flow intensive designs, where glitches generated at control signals have a significant impact on the circuit's power consumption, and multiplexers and registers often account for a major portion of the total power. Application of the proposed techniques to several examples shows significant power savings, with negligible area and delay overheads. [Rag96]

b) Examples and Results:

Techniques	Test Circuits/Benchmarks	Power Savings (%,x)
Restructuring multiplexer networks, clocking control signals, and inserting selective rising/falling delays [Rag96]	GCD	17.27 %
	Barcode	17.42%
	UAV	26.25%
	Vend	26.22%

2. Operand Isolation

a) Basic Approach: Designs which do not fully utilize their arithmetic data path components typically exhibit a significant overhead in power consumption. Whenever a module performs an operation whose result is not used in the downstream circuit, power is being consumed for an otherwise redundant computation. Operand isolation is a technique to minimize the power overhead incurred by redundant operations by selectively blocking the propagation of switching activity through the circuit. Reference [Mun00] discusses how redundant operations can be identified concurrently to normal circuit operation, and presents a model to estimate the power savings that can be obtained by isolation of selected modules at the register transfer (RT) level. Based on this model, an algorithm is presented to iteratively isolate modules while minimizing the cost incurred by RTL operand isolation. Experimental results with power reductions of up to 30% demonstrate the effectiveness of the approach. Figure 1 and 2 show the design without/with operand isolation.

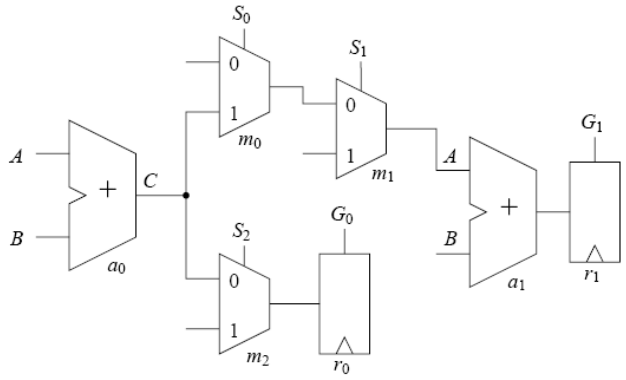

Figure 1: Design without operand isolation

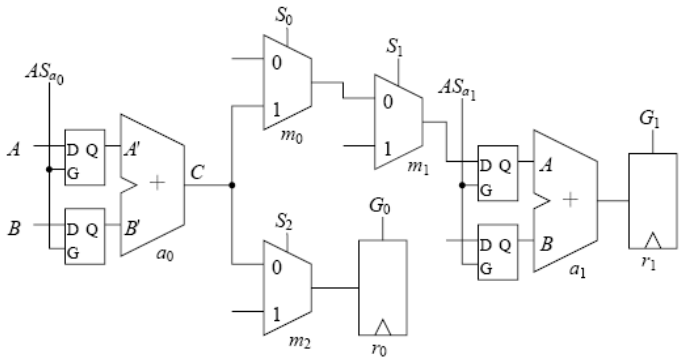

Figure 2: Design with operand isolation

b) Examples and Results: In the following, we shall summarize the results of applying our model to two industrial benchmark circuits. Both designs were data path blocks extracted from more complex designs:

Techniques	Test Circuits/Benchmarks	Power Savings (%,x)
non-isolated [Mun00]	industrial benchmark circuit 1	N/A
	industrial benchmark circuit 2	N/A
AND-isolated [Mun00]	industrial benchmark circuit 1	13.76%
	industrial benchmark circuit 2	31.95%
OR-isolated [Mun00]	industrial benchmark circuit 1	18.02%
	industrial benchmark circuit 2	32.01%
LAT-isolated [Mun00]	industrial benchmark circuit 1	12.04%
	industrial benchmark circuit 2	31.93%

3. Cache Design

a) Basic Approach: Caches consume a large portion of chip real estate. This naturally makes them the target for aggressive power reduction techniques by computer architects. With decreasing feature sizes, leakage power is becoming a major concern. Various power models and architecture/RT level techniques have been proposed to reduce the power consumption. In [Fla02], the author presented a method that put the cold cache lines into a state preserving, low-power drowsy mode. It shows that with simple architectural techniques, about 80%-90% of the cache lines can be maintained in a drowsy state without affecting performance by more than 1%. In [Zho03], the authors talk about applying the sleep mode to just the data cache to achieve low power.

b) Examples and Results:

Techniques	Test Circuits/Benchmarks	Power Savings (%,x)
Put the cold cache lines into a state preserving, low-power drowsy mode [Fla02]	32-KB 4-way set associative cache	50%-75%
Adaptive Mode Control [Zho03]	16kB, 32kB, and 64kB L1 instruction caches (I-Cache) and 64kB 4-way set associative data caches (D-Cache)	73% (I-Cache); 56% (D-Cache)

4. Precomputation Logic

a) Basic Approach: Precomputation logic [Ped05], relies on the idea of duplicating part of the logic with the purpose of precomputing the circuit output values one clock cycle before they are required, and then uses these values to reduce the total amount of switching in the circuit during the next clock cycle. In fact, knowing the output values one clock cycle in advance allows the original logic to be turned off during the next time frame, thus eliminating any charging and discharging of the internal capacitances. Obviously, the size of the logic that precalculates the output values must be kept under control since its contribution to the total power balance may offset the savings achieved by blocking the switching inside the original circuit. Several variants to the basic architecture can then be devised to address this issue. In particular, sometimes it may be convenient to resort to partial, rather than global, shutdown, i.e. to select for power management only a (possibly small) subset of the circuit inputs. A precomputation architecture realization of this same logic block placed between register sets R1 and R2 is depicted in Figure 3. The key elements of the precomputation architecture are two n-input, single-output predictor functions g1 and g2, which satisfy the following constraints:

g1 =1 **à** f = 1 g2 = 1 **à** f = 0

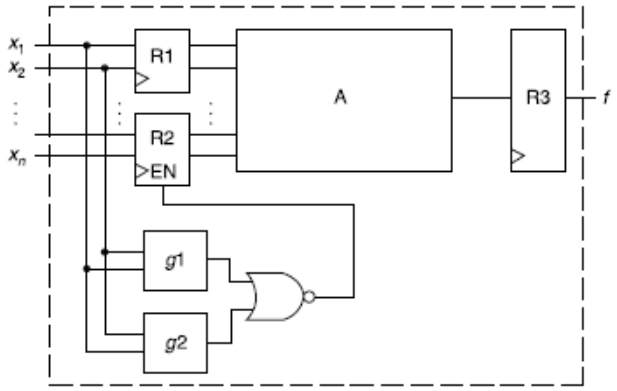

Figure 3: Precomputation logic realization of the pipeline stage

b) Examples and Results:

Techniques	Test Circuits/Benchmarks	Power Savings (%,x)
Selectively precomputing the output logic values of the circuit one clock cycle before they are required, and using the precomputed values to reduce internal switching activity in the succeeding clock cycle [Ali94]	Carryselect adders, comparators, and interconnections of adders and comparators	75%

a general precomputation architecture for sequential logic circuits [Moj98]	MCNC benchmark set; SIS package circuits	66%

5. Clock Gating

a) Basic Approach: Clock gating [Ped05] provides a way to selectively stop the clock, and thus force the original circuit to make no transition, whenever the computation that is to be carried out at the next clock cycle is redundant. In other words, the clock signal is disabled according to the idle conditions of the logic network. For reactive circuits, the number of clock cycles in which the design is idle in some wait states is usually large. Therefore, avoiding the power waste corresponding to such states may be significant.

The logic for the clock management is automatically synthesized from the Boolean function that represents the idle conditions of the circuit (cf. Figure 4). It may well be the case that considering all such conditions results in additional circuitry that is too large and too power consuming. It may then be necessary to synthesize a simplified function, which dissipates the minimum possible power and stops the clock with maximum efficiency. The use of gated clocks has the drawback that the logic implementing the clock-gating mechanism is functionally redundant, and this may create major difficulties in testing and verification.

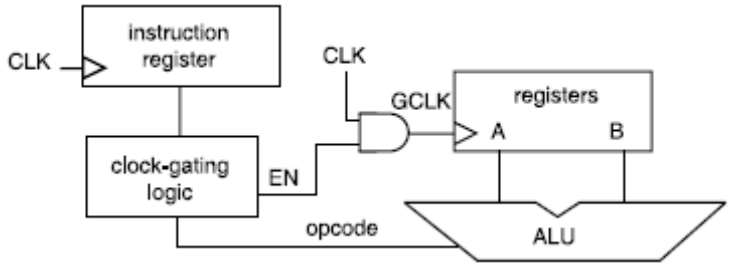

Figure 4: Clock gating logic for ALU in typical processor microarchitecture with negative-edge triggered flip-flops

Another difficulty with clock-gating is that one must stop hazards/glitches on the EN signal from corrupting the clock signal to the register sets. This can be accomplished by introducing a transparent negative latch between EN and the AND gate as shown in Figure 5.

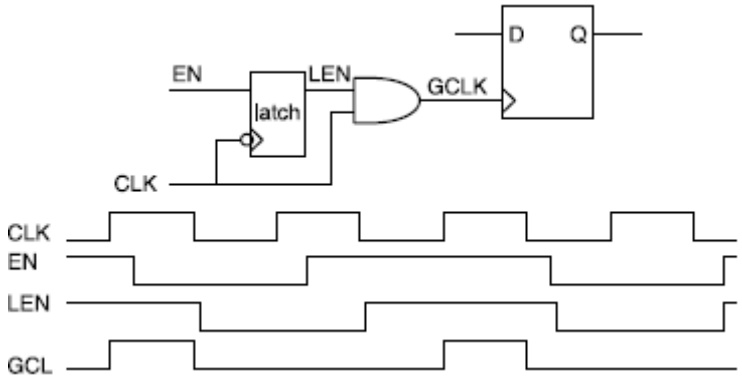

Figure 5: Clock is disabled when EN 0; furthermore, a hazard on EN will be stopped from reaching GCLK

b) Examples and Results:

Techniques	Test Circuits/Benchmarks	Power Savings (%,x)
Automatically synthesizes finite state machines with gated clocks [Ben94]	MCNC benchmarks	10%-30%
Adaptive Clock Gating Technique [Cha07]	Dhrystone and Whetstone	62.2% (dynamic) 70.9% (leakage)

6.Local Explicit Clock Enable - LECE

a) Basic Approach: The following schematic shows a local explicit clock enable. The q output is updated on a rising edge of clk, but only when the signal en is high.

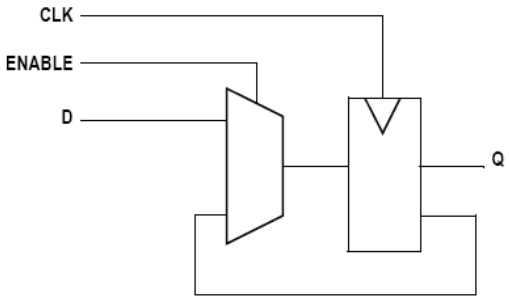

Figure 6: LECE

Normally logic synthesis will implement the indicated circuit with a 2:1 mux, or a muxed D flip-flop if there is one available in the target technology. If the enable is low for a significant percentage of the circuit operation, and if d and q are multi-bit buses, then a substantial amount of power dissipated by the clock driver is wasted. That is, driving the clock to this register when the enable is low does not change the circuit behavior.

Another way to implement this circuit is to gate the clock. Simply replacing the clock input to the flip-flop with an AND gate whose inputs are the clock and the enable is not safe. Consider the situation when the clock is high and transitions occur on the enable: edges will appear on the register clock input, possibly causing incorrect circuit behavior.

To avoid these edges, a latch must also be inserted so that when the clock is high, no activity on the enable will be transferred to the clock input. The following schematic illustrates the needed additional circuitry.

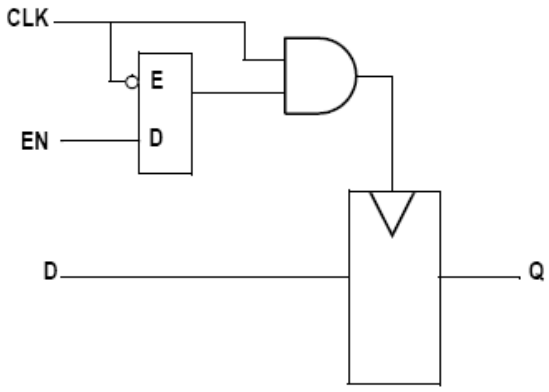

Figure 7: LECE with latch

Depending on the clock insertion technology, there are two possible ways to implement this change in your RTL code. In most cases, the physical clock insertion software will limit the choice of gates that can be used to implement the logic. For example, a special clock-and gate may be needed, or the tool may require insertion of a gate with the correct drive strength. In this case, the RTL design must be modified to instantiate the required gates. In some cases, it may be possible to simply write the required functionality in RTL and allow the synthesizer to select the needed gates.

7. Computational Kernels

a) Basic Approach: Sequential circuits may have an extremely large number of reachable states, but during normal operation these circuits tend to visit only a relatively small subset of the reachable states [Ped05]. A similar situation occurs at the primary outputs; while the circuit walks through the most probable states, only a few distinct patterns are generated at the combinational outputs of the circuit. Many researchers have proposed approaches for synthesizing a circuit that is fast and power-efficient under typical input stimuli, but continues to operate correctly even when uncommon input stimuli are applied to the circuit. Reference [Ben01] presents a power optimization technique by exploiting the concept of computational kernel of a sequential circuit, which is a highly simplified logic block that imitates the steady-state behavior of the original specification. This block is smaller, faster and less power consuming than the circuit from which it is extracted, and can replace the original network for a large fraction of the operation time. In [Ben99], the authors raise the level of abstraction at which the kernel-based optimization strategy can be exploited and show how RTL components, for which only a functional specification is available, can be optimized using the computational kernels. They present a technique for computational kernel extraction directly from the functional specification of a RTL module.

b) Examples and Results:

Techniques		**Test Circuits/Benchmarks**	**Power Savings (%,x)**
A highly simplified logic block that imitates the steady-state behavior of the original specification [Ben01]	Symbolic extraction	SIS, CUDD package	51%
	Simulation based extraction		34%
	Structural extraction		28%
Computational kernel extraction directly from the functional specification of a RTL module [Ben99]		SIS, CUDD package	51%

8. State Machine Decomposition

a) Basic Approach: Decomposition of finite-state machines for low power has been proposed in [Mon98]. The basic idea is to decompose the STG of a finite-state machine (FSM) into two STGs that jointly produce the equivalent input–output behavior as the original machine [Ped05]. Power is saved because, except for transitions between the two sub-FSMs, only one of the sub-FSMs needs to be clocked. The technique follows a standard decomposition structure. The states are partitioned by searching for a small subset of states with high probability of transitions among these states and a low probability of transitions to and from other states. This subset of states will then constitute a small sub-FSM that is active most of the time. When the small sub-FSM is active, the other larger sub-FSM can be disabled. Consequently, power is saved because most of the time only the smaller, more power efficient sub-FSM is clocked. In [Cho96], the combinational logic block is partitioned (for example to CL1 and CL2) and the active part is decided based on the encoding of the present state. The states selected for one of the sub-FSMs (i.e. M1) are all encoded in such a way that the enable signal is always on for CL1, while it is off for CL2. Conversely, for all states in the other sub-FSM (i.e. M2), the enable signal is always off for CL1, while it is on for CL2. Consequently, for all transitions within M1, only CL1 will be active and vice versa. Consider as an example dk27 FSM from the MCNC benchmark set, depicted in Figure 8.

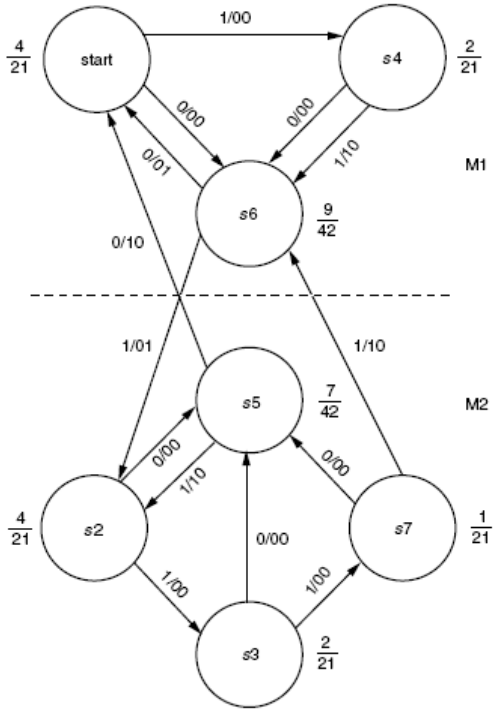

Figure 8: Example of an FSM (dk27) that may be decomposed into two sub-FSMs such that one sub-FSM can be shut off when the other is active and vice versa

b) Examples and Results:

Techniques	Test Circuits/Benchmarks	Power Savings (%,x)
Compute two sub-FSMs that together have the same functionality as the original FSM [Mon98]	MCNC91 and ISCAS89 benchmark set	80%

Decomposition of a finite state machine into submachines, synthesis of the coupled submachines to optimize the logic circuits [Cho96]	MCNC benchmark set	15%-59%

9. Guarded Evaluation

a) Basic Approach: Unlike precomputation and gated clocks, guarded evaluation does not require one to synthesize additional logic to implement the shutdown mechanism [Ped05]. Instead, it exploits existing signals in the original circuit. The approach is based on placing some guard logic, consisting of transparent latches with an enable signal, at the inputs of each block of the circuit that needs to be power-managed. When the block must execute some useful computation in a clock cycle, the enable signal makes the latches transparent. Otherwise, the latches retain their previous states, thus blocking any transition within the logic block. Guarded evaluation provides a systematic approach to identify where transparent latches must be placed within the circuit and by which signals they must be controlled. This technique, referred to as pure guarded evaluation, has the desirable property that when applied, no changes in the original combinational circuitry are needed. However, if some resynthesis and restructuring of the original logic is allowed, a larger number of logic shutdown opportunities may become available. Figure 9 shows an example of guard logic insertion.

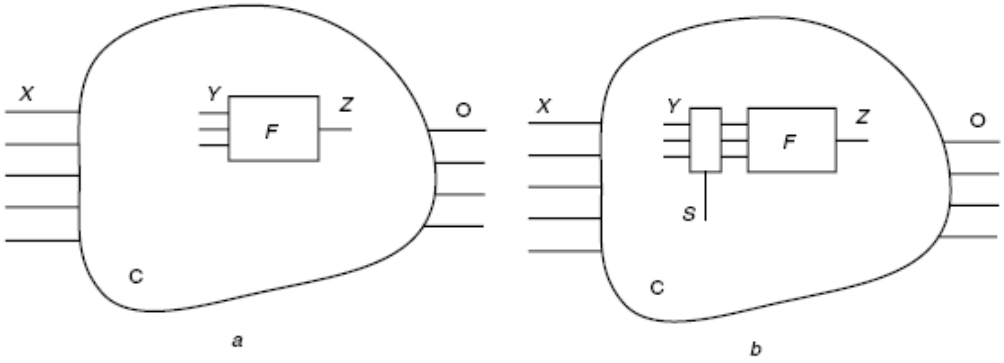

Figure 9: An example of guard logic insertion

b) Examples and Results:

Techniques	Test Circuits/Benchmarks	Power Savings (%,x)
Guarded evaluation [Tiw98]	MCNC and ISCAS suites	65%

10. Bus Encoding

a) Basic Approach: A lot of power is consumed in the on-chip and off-chip buses in a VLSI circuit. These buses, which connect various internal blocks of the circuit or connect the circuit to the external environment, have large capacitive loads and high transition counts. Power on these buses can be reduced by properly coding the data and/or address bus values so as to minimize the number of transitions that occur on the bus.

Musoll, et al. proposed the working zone method in [Mus97]. Their method takes advantage of the fact that data accesses tend to remain in a small set of working zones. In [Sta95] bus-invert method has

been introduced. The bus-invert selects between the original and the inverted pattern in a way that minimizes the switching activity on the bus. In [Ben97], Benini et al. proposed the T0 code, which exploits data sequentiality to reduce the switching activity on the address bus. Reference [Ram99] proposes a low-power coding framework for address and data buses. They describe the general architecture of a low power encoder (cf. Figure 10).

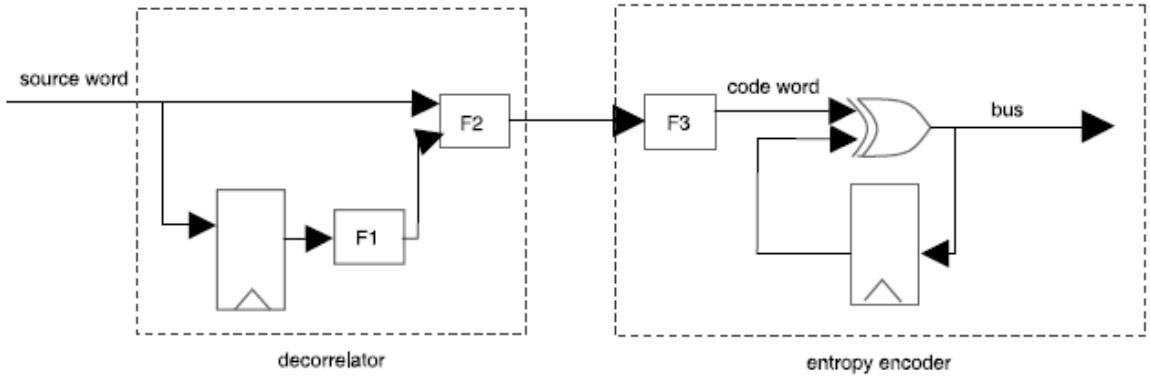

Figure 10: Block diagram of a generic low-power encoder

In [Agh01], Aghaghiri et al. presented offset–XOR–SM encoding, whereby the new code word is again transition signaled over the bus. In [Mam01], Mamidipaka et al. proposed an encoding technique based on the notion of self-organizing lists. In [Agh04], the authors introduced a class of irredundant low power techniques for encoding instruction or data source words before they are transmitted over buses. In [Lam04], the authors provide a modified bus-invert (MBI) technique which, besides reducing delay and power, also minimizes the crosstalk noise that results from inductive coupling between the bus lines.

b) Examples and Results:

Techniques	Test Circuits/Benchmarks	Power Savings (%,x)
Workang-Zone Encoding technique [Mus97]	DSPlGxx, DSP32C and DSP32xx (AT&T Microelectronics), DSP56lxx (Motorola), TMS320C2x1 TMS320C209, TMS320C5x and TMS320C54x (Texas Instruments), or Z894xx (Zilog).	47%
T0 code [Ben97]	32-bit address bus of the MIPS microprocessor	36%
A source-coding framework to describe encoding schemes to reduce transition activity [Ram99]	SIS circuits	36%

11. Data Bus Specific Clock/Data Bus Power Reduction

a) Basic Approach: To transfer a small number, we inherently need the small number of bits. But all bit lines on a data bus change their status and redundant power is consumed [Mur03]. To reduce the redundant power consumption, the author introduced a concept named active bits and proposed a power reduction scheme for data buses using the active bits. Suppressing switching activity of inactive bits, we

can reduce redundant power consumption. Experimental results illustrate 20% - 35% on average and up to 54.2% switching activity reduction.

b) Examples and Results:

Techniques	Test Circuits/Benchmarks	Power Savings (%,x)
Dynamic detection of active bits [Mur03]	SPEC2000 benchmarks; SimpleScalar	20%-35%

12. Convert Regfile to Latchfile

a) Basic Approach [Seq07]: When a two-dimensional array appears in RTL code, the synthesizer will infer a register or latch file. This is a random logic block containing a number of registers or latches, and logic to implement the decode and write lines. The result is typically not laid out in any regular fashion. The register file synthesizes to flipflop elements connected to the global clock. These flipflops clock every cycle, even though only one word's worth of registers could possibly toggle. Therefore, the synthesized register file consumes a high amount of clock power. On the other hand, a level-sensitive latch file is synthesized to latches that are only enabled when their word is selected. This uses a far smaller amount of clock power. This method examines each clocked register file and determines how much power could be saved by using a level sensitive latch array.

13. FSM Re-encoding

a) Basic Approach [Seq07]: For state machines, this method creates a state transition diagram and estimates power savings gained from reducing the number of state bit toggles for most common transitions. The following schematic (Figure 11) shows a general finite state machine. As the FSM operates, it transitions from one state to another to another. At each transition, a certain number of state bits toggle. If few or no state bits toggle, there is little activity in the two combinational logic blocks. However, if many state bits toggle, there may be considerable activity in the combinational logic.

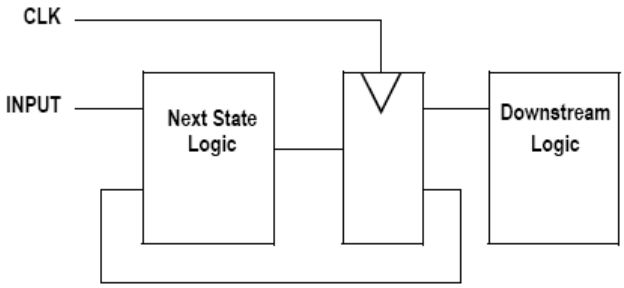

Figure 11: a general finite state machine

This method creates a state transition table showing the number of transitions from any states to any other, and showing the number of state bit toggles for each transition. It then examines the transition table to see if a different state encoding would reduce the total number of state bit toggles. For example, if 90 percent of the transitions were between states 0000 and 1111, choosing a new encoding where these two states differ by only one bit instead of four will greatly reduce the total toggle count. This method estimates the percent reduction that could be achieved in the state machine, and reduces the power of the two combinational logic blocks by the same percentage.

14. High Power Bus

a) Basic Approach [Seq07]: This method finds multi-bit buses with significant power dissipated in wiring and estimates savings gained from re-floorplanning. The following floorplan (Figure 12) view shows a long bus. A signal is routed from block A to block C across block B, and it is not connected to block B. If the signal is very active, this may dissipate more power than needed.

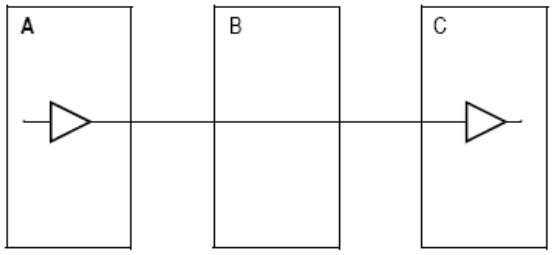

Figure 12: A long bus floorplan

No logical change is required; the only change required is at the floorplan level. The following diagram (Figure 13) shows a floorplan that is better from the standpoint of the power dissipated by the indicated bus.

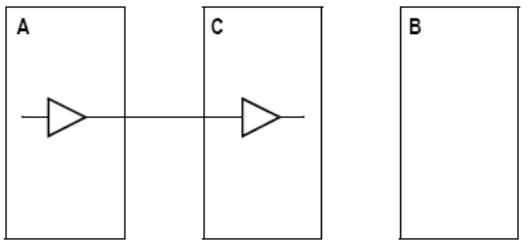

Figure 13: Revised floorplan

This method estimates the power reduction by assuming the wiring capacitance in the intermodule net is reduced to zero. The total pin capacitance does not change, and the length of the intra-module part (from the driving logic to the output of A, and from the input of C to the receiving logic) is also assumed to remain the same.

15. Low Frequency Register

a) Basic Approach [Seq07]: This method finds register banks that rarely toggle, and estimates power savings gained from adding a clock enable to the register. The following schematic and timing diagram (Figure 14) shows a low frequency register. In this circuit, the data input is active during one phase of operation only, and does not change for a long period of time.

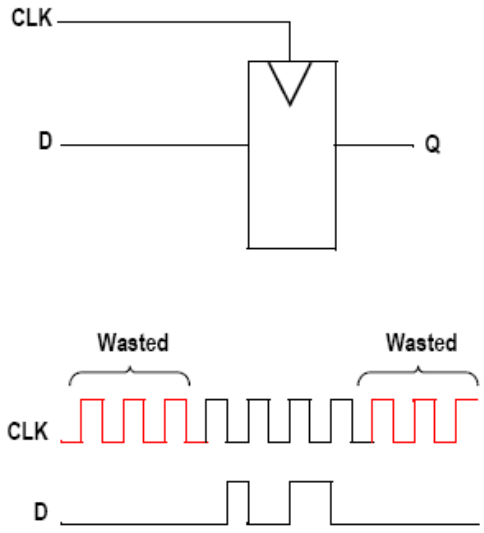

Figure 14: A low frequency register

In this circuit, power is wasted by the clock driver as well as by circuitry inside the register because most of the clock toggles are not needed. That is, driving the clock to this register when the data is not changing does not change the circuit behavior. For this design change, we must locate or create an enable signal. By examining the circuit architecture and the testbenches, it should be possible to locate or create a signal that will indicate whether the data might be changing on a global basis. For example, if the register is in a datapath and the datapath knows when it receives valid data, then a signal can be created that is high only when valid data is in the datapath. Once we have located an enable signal, we can gate the clock to this register.

16. Pad Static Power

a) Basic Approach [Seq07]: This method finds finds pads with pullups or pulldowns. If the pullup or pulldown is not required by the off-chip system, the static power drawn can be eliminated. The following schematic (Figure 15) shows an output pad with a pullup. When the data input is high, the resistor does not consume any static power. However, when the data input is low, the resistor does consume static power. In some cases, the external system requires the pullup for correct operation. In other cases, the pullup may not be required, but may have simply been placed there by default.

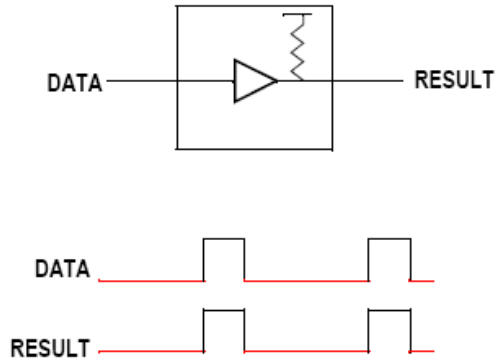

Figure 15: An output pad with a pullup

If the pullup is not required by the system, choose a pad that has the same drive strength but no pullup. This reduces the state-dependent part of the static power to zero. This method reduces the number of separate pad static power reports by grouping together all the pads for a single bus. It is determined that pads are on the same bus if they are connected to an input or output net which ends in an indexed name

like address.

17. Split Memory Words

a) Basic Approach [Seq07]: This method finds wide memories and determines if power could be saved by splitting it into two smaller memories with an output mux. The following schematic (Figure 16) shows a memory with a large number of words split into two smaller memories, each with half the number of words.

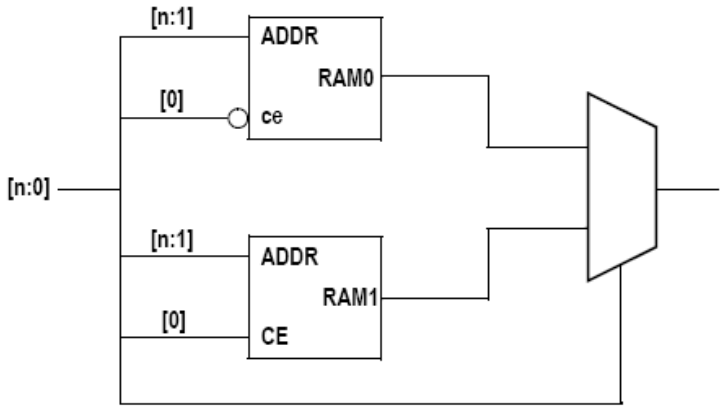

Figure 16: A memory with a large number of words split into two smaller memories

For most memory architectures, the power of a read or write to the half-size memory is much less than the power of a read or write to the full size memories. Even though the same number of reads and writes are occurring in the new architecture, the power is reduced because each read or write is on only one of the smaller memories. To implement this change, replace the single memory with two memories, each having half the number of words. Select a bit of the address, usually the MSB or LSB, to operate as a .bank select.. This bit should be used as the chip enable to one memory, and inverted to be used as the chip enable to the other memory. This bit should also be used as the select input to a new multiplexor on the memory outputs. This mux selects the appropriate memory bank.

18. Tristate Float/Fight

a) Basic Approach [Seq07]: This method finds RTL or gate-level tristate logic and estimates power wasted by tristate nets floating or fighting drivers. It estimates the power due to tristate bus contention and floating. If finds significant power waste, we should change the design to remove the waste.

To remove power waste due to a floating bus, add a weak pullup or pulldown to the net. If the technology library does not provide this type of cell, add another tristate driver to the net. The task of this additional driver is to drive the net to a known value when none of the other drivers on the net are active. Therefore, the enable of the new driver should be the NOR of the other enables. That is, the OR of the other enables is high when any one is active, so the NOR is high when none of the other enables is active.

To remove power waste due to bus contention requires more understanding of the design. The situation occurs when the expressions for two enables are overlapping (when there exists a condition when two enables are on). The logic for the enables should be changed so that they are mutually exclusive.

19. Memory Power Linter

a) Basic Approach [Seq07]: This method finds the memories in the design and monitors the data input to the memories to see if changes in the data input ports were wasted because the memory was not selected for a write access. It also estimates the amount of power lost due to the wasted activity on the input data bus.

If the input data bus to a memory changes one or more times and that memory is not selected for a write access before the input data line toggles again, the toggles on the input data bus of the memory can be considered as extraneous toggles, which cause power to be wasted. The following schematic (Figure 17) shows data changing at the DATA input to the memory. However, the write enable (WE) to the memory is not valid during the data change and will consequently is not written into the memory at ADDR. Therefore, this extraneous transition of the data causes power to be wasted.

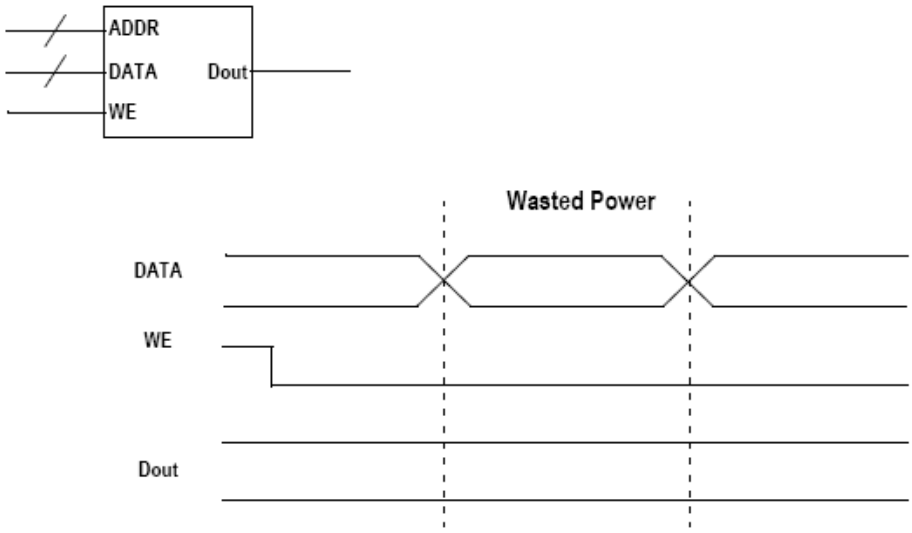

Figure 17: Data changing at the DATA input to the memory

Unused toggles on the memory input data bus can be desirable or undesirable. Depending on your design, there could be several ways to implement more efficient circuitry to reduce extraneous activity on the input data bus for the memory. This is essentially an analysis tool that points out the potential areas of concern. The analysis is very much design and simulation dependent. This method will monitor the input data bus of the memory only if the bits of the input data bus and the write enable are monitored by your simulator. Inferred nets are not monitored for extraneous activity/wasted power.

20. MUX Power Linter

a) Basic Approach [Seq07]: This method finds the multiplexors in the design and monitors the inputs to see if certain input toggles were unnecessary because the input was not selected. It also estimates the amount of power lost due to the extraneous activity on the inputs.

If the input net to a multiplexor toggles one or more times and that input is not selected by the multiplexor before it toggles again, the toggles on that input of the multiplexor can be considered wasted. This means that there is activity that consumes power on that pin of the multiplexor, but that activity is meaningless. The following schematic (Figure 18) shows a multiplexor which initially selects the B input. The power consumed by the transitioning that appears on A during the B select period is considered wasted power as it will not be relayed to any other part of the design.

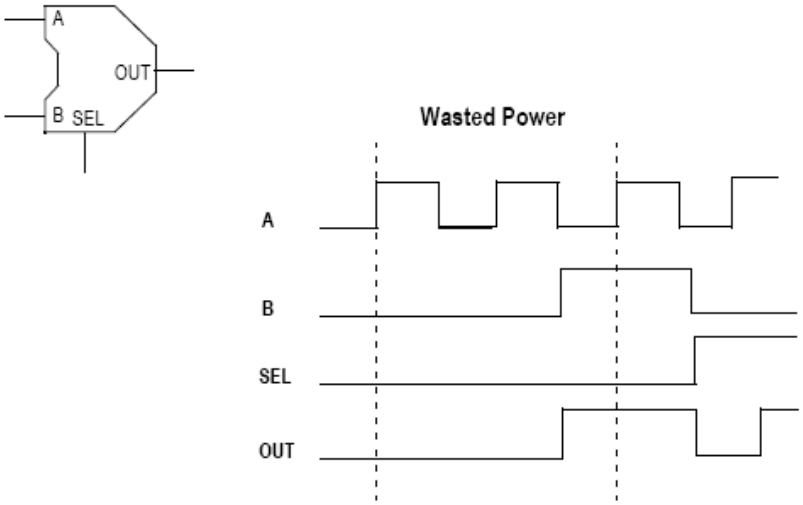

Figure 18: A multiplexor which initially selects the B input

The unused toggles on the multiplexor can be desirable or undesirable. Depending on your design, there could be many ways of implementing more efficient circuitry upstream from the multiplexor which would reduce wasted activity at the multiplexor. This is essentially an analysis tool that points out the potential areas of concern. The analysis is very much design and simulation dependent. This method will monitor an input of a multiplexor only if the associated net is monitored by your simulator. Inferred nets are not monitored for extraneous activity/wasted power.

21. Register Power Linter

a) Basic Approach [Seq07]: This method finds registers in the design and monitors the inputs to see if certain toggles were unnecessary because the clock did not cycle. It also estimates the amount of power lost due to the extraneous activity on the inputs.

If the input net to a register or a flip-flop toggles two or more times before the clock of the register completes one cycle, the toggles on the inputs could all be wasted. This means that there is activity that consumes power on the input of the register, but that activity is meaningless. The following schematic (Figure 19) shows a register that is clocked infrequently (CLK) compared to the data changing at its D input. The transitioning occurring on the D input between the clocking of the register is considered extraneous in that it causes power to be wasted.

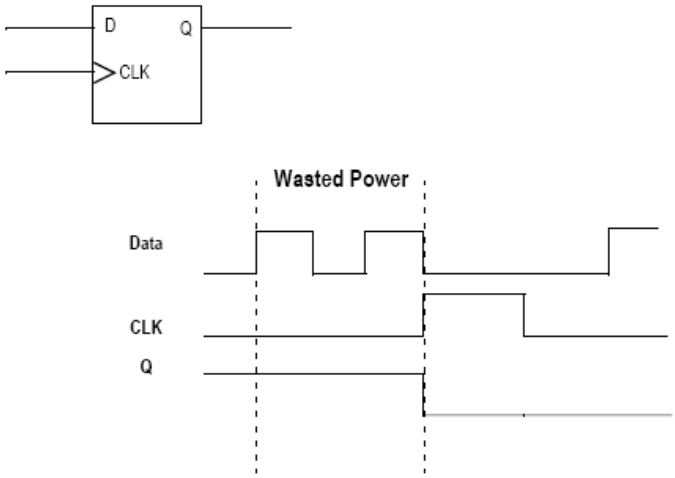

Figure 19: A register that is clocked infrequently

The unused toggles on the register input can be desirable or undesirable. If it is undesirable, there may be many ways of implementing a more efficient circuitry upstream from the register, which would reduce wasted activity of the register. This is essentially an analysis tool that points out the potential areas of concern. This analysis is very much design dependent and simulation dependent. This method will monitor the input of the register only if the associated net is monitored by your simulator. Inferred nets are not monitored for extraneous activity/wasted power.

22. Clock Enable Condition Linter

a) Basic Approach [Seq07]: This method detects clock gating situations where the data input to the register is driven by a feedback mux. The goal is to determine situations where the mux select line, which will act as the clock gate enable signal, is not optimally designed. A simple example is where the clock is enabled but that data input is not changing.

The following schematic (20) shows a mux-flop feedback loop topology, which is a potential candidate for clock gating. The power benefit of clock gating is only as good as the clock enable condition (ENABLE). If the input D does not cycle when the mux is enabled, the corresponding clock toggles are extraneous, and therefore contribute to wasted power. If this topology is replaced with a gated clock with the same enable condition, the wasted power would remain the same.

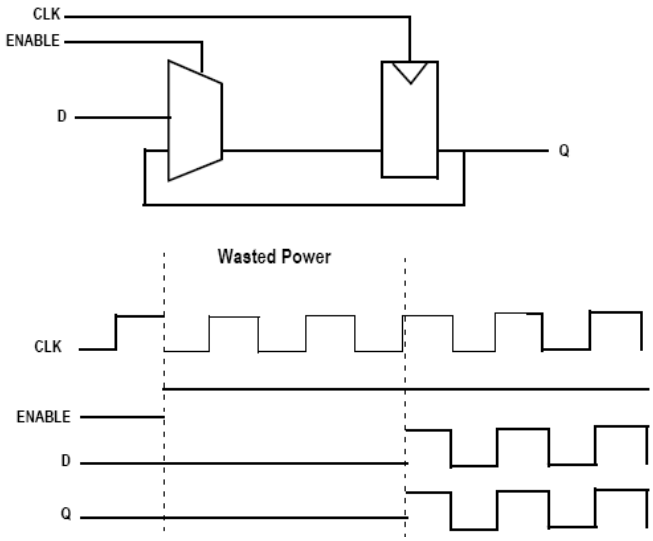

Figure 20: A mux-flop feedback loop topology

The unused clock toggles on the register clock pin can be desirable or undesirable. If they are undesirable, there may be ways of implementing more efficient circuitry that would reduce wasted activity of the clock. For example, you could replace the topology in the previous schematic with a gated clock such that the gated clock enable signal is in sync with the data activity of the register.

References

[Agh01] Aghaghiri, Y., Fallah, F., Pedram, M., Irredundant address bus encoding for low power. International Symposium on Low Power Electronics and Design, 2001. 6-7 Aug. 2001 Page(s):182 - 187 .

[Agh04] Aghaghiri, Y., Fallah, F., Pedram, M., Transition reduction in memory buses using sector-based encoding techniques. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Volume 23, Issue 8, Aug. 2004 Page(s):1164 – 1174.

[Ali94] Alidina, M., Monteiro, J., Devadas, S., Ghosh, A., Papaefthymiou, MAlidina, M., Monteiro, J., Devadas, S., Ghosh, A. and Papaefthymiou, M, Precomputation-based sequential logic optimization for low power. Very Large Scale Integration (VLSI) Systems, IEEE Transactions, Volume 2, Issue 4, Page(s):426 – 436, Dec. 1994.

[Alp99] C. J. Alpert, A. Devgan, and S. T. Quay, "Is wire tapering worthwhile?" Proc. of IEEE Intl. Conf. on Computer-Aided Design, pp. 430–435, Nov. 1999.

[Ben01] Benini, L., De Micheli, G., Lioy, A., Macii, E., Odasso, G. and Poncino, M., Synthesis of power-managed sequential components based on computational kernel extraction, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Volume 20, Issue 9, Sept. 2001 Page(s):1118 - 1131

[Ben94] Benini, L., Siegel, P., and De Micheli, G.: Automatic synthesis of gated clocks for power reduction in sequential circuits, IEEE Des. Test Comp., 1994, 11, (4), pp. 32–40.

[Ben97] Benini, L., De Micheli, G., Macii, E., Sciuto, D. and Silvano, C., Asymptotic zero-transition activity encoding for address busses in low-power microprocessor-based systems, Seventh Great Lakes Symposium on VLSI, 1997. Proceedings. 13-15 March 1997 Page(s):77 – 82.

[Ben99] Benini, L., De Micheli, G., Macii, E., Odasso, G. and Poncino, M., Kernel-based power optimization of RTL components: exact and approximate extraction algorithms, 36th Design Automation Conference, 1999. Proceedings. 21-25 June 1999 Page(s):247 – 252

[Cha07] Xiaotao Chang, Mingming Zhang, Ge Zhang, Zhimin Zhang and Jim Wang, Adaptive Clock Gating Techniquefor Low Power IP Core in SoC Design. IEEE International Symposium on Circuits and Systems, 2007. ISCAS 2007. Page(s):2120 - 2123, 27-30 May 2007.

[Che04] Tai-Chen Chen, Song-Ra Pan, and Yao-Wen Chang, "Timing Modeling and Optimization Under the Transmission Line Model," IEEE TVLSI Systems, vol. 12, no. 1, pp. 28–41, Jan. 2004.

[Cho04] Seung Hoon Choi, Bipul C. Paul and Kaushik Roy, "Novel Sizing Algorithm for Yield Improvement under Process Variation in Nanometer Technology," DAC 2004, San Diego, CA, USA, June 7-11, 2004.

[Cho96] Chow, S-H., Ho, Y-C., and Hwang, T., Low power realization of finite state machines a decomposition approach', ACM Trans. Des. Autom. Electron. Syst., 1, (3), pp. 315–340, 1996.

[Chu01] C. C. N. Chu, and D. F. Wong, “Closed Form Solution to Simultaneous Buffer Insertion/Sizing and Wire Sizing,” ACM Trans. on Design Automation of Electronic Systems, vol. 6, no. 3, pp. 343–371, July 2001.

[Chu97] C. Chung-Ping, and D. F. Wong, “Optimal Wire-Sizing Function with Fringing Capacitance Consideration,” Proc. of ACM DAC, Nov. 1997, pp. 604–607.

[Con02] J. Cong, and D. Z. Pan, “Wire Width Planning for Interconnect Performance Optimization,” IEEE TCAD, vol. 21, no. 3, pp. 319–329, 2002.

[Con94] J. Cong, and K. Cheng-Kok, “Simultaneous Driver and Wire Sizing for Performance and Power Optimization,” Proc. of ICCAD, Nov. 1994, pp. 206–212.

[Fla02] Flautner, K., Nam Sung Kim, Martin, S., Blaauw, D., Mudge, T., Drowsy caches: simple techniques for reducing leakage power, Computer Architecture, 2002. Proceedings, 29th Annual International Symposium, Page(s):148 – 157, May 2002.

[Han06] Narender Hanchate and Nagarajan Ranganathan, “A Linear Time Algorithm for Wire Sizing with Simultaneous Optimization of Interconnect Delay and Crosstalk Noise,” Proceedings of the 19th International Conference on VLSI Design (VLSID’06), 2006.

[He98] Jiang-An He, and H. Kobayashi, “Simultaneous Wire Sizing and Wire Spacing in Post-Layout Performance Optimization,” Proc. of ASPDAC, 1998, pp. 373–378.

[Hui00] J. I. Hui-Ru, C. Yao-Wen, and J. Jing-Yang, “Crosstalk-Driven Interconnect Optimization by Simultaneous Gate and Wire Sizing,” IEEE of TCAD, vol. 19, no. 9, pp. 999–1010, Sept. 2000.

[Jyu94] Horng-Fei Jyu and Sharad Malik, ”Statistical Delay Modeling in Logic Design and Synthesis,” 31st ACM/IEEE Design Automation Conference, 1994.

[Lam04] Lampropoulos, M., Al-Hashimi, B.M., Rosinger, P., Minimization of crosstalk noise, delay and power using a modified bus invert technique. Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings. Volume 2, 16-20 Feb. 2004 Page(s):1372 - 1373 Vol.2.

[Mam01] Mamidipaka, M., Hirschberg, D., Dutt, N., Low power address encoding using self-organizing lists. International Symposium on Low Power Electronics and Design, 2001. 6-7 Aug. 2001 Page(s):188 – 193.

[Moj98] Monteiro, J., Devadas, S. and Ghosh, A., Sequential logic optimization for low power using input-disabling precomputation architectures, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions, Volume 17, Issue 3, Page(s):279 – 284, March 1998.

[Mon98] Monteiro, J.C., Oliveria, A.L., Finite state machine decomposition for low power, Design Automation Conference, 1998. Proceedings, Page(s):758 - 763, 15-19 Jun 1998.

[Mun00] Munch, M., Wurth, B., Mehra, R., Sproch, J. and Wehn, N., Automating RT-Level Operand Isolation to Minimize Power Consumption in Datapaths, Design Automation and Test in Europe Conference and Exhibition 2000 Proceedings. Page(s):624 – 631, March 2000.

[Mur03] Muroyama, M., Hyodo, A., Okuma, T., Yasuura, H., A power reduction scheme for data buses by dynamic detection of active bits. Euromicro Symposium on Digital System Design, 2003. Proceedings. 1-6 Sept. 2003 Page(s):408 – 415.

[Mus97] Musoll, E., Lang, T. and Cortadella, J., Exploiting the locality of memory references to reduce the address bus energy, International Symposium on Low Power Electronics and Design, 1997. Proceedings. 18-20 Aug 199,7Page(s):202 – 207.

[Oka03] Kenichi Okada, Kento Yamaoka, and Hidetoshi Onodera, "Statistical Modeling of Gate-delay Variation with Consideration of Intra-gate Variability," Proceedings of the 2003 International Symposium on Circuits and Systems, Vol. 5, pp.V-513 - V-516, May 2003.

[Ped05] M. Pedram and A. Abdollahi, Low-power RT-level synthesis techniques: a tutorial, Computers and Digital Techniques, IEE Proceedings, Volume 152, Issue 3, Page(s):333 – 343, 6 May 2005.

[Rag96] Raghunathan, A., Dey, S., Jha, N.K., Glitch Analysis and Reduction in Register Transfer Level Power Optimization, Design Automation Conference Proceedings 1996, 33rd, 3-7 Page(s):331 – 336, June 1996.

[Ram99] Ramprasad, S., Shanbhag, N.R. and Hajj, I.N., A coding framework for low-power address and data busses. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume 7, Issue 2, June 1999 Page(s):212 – 221.

[Rao05] Rahul Rao, Kanak Agarwal, Anirudh Devgan, Kevin Nowka, Dennis Sylvester and Richard Brow, "Parametric Yield Analysis and Constrained-Based Supply Voltage Optimization," Proceedings of the Sixth International Symposium on Quality Electronic Design, 2005.

[Sap96] S. S. Sapatnekar, "Wire Sizing as a Convex Optimization Problem: Exploring the Area-Delay Tradeoff," IEEE TCAD, vol. 15, no. 8, pp. 1001–1011, Aug. 1996.

[Seq07] Sequence Design, Inc, "PowerTheater User Guide", 2007.

[She00] Rupesh S. Shelar, Sacheendra Nath and Jagmohan S. Nanaware, "Parameterized Reusable Component Library Methodology," Proceedings of the 26th Euromicro Conference. Volume 1, pp.410–416, Sept. 2000.

[Sri08] Ashish Srivastava, Kaviraj Chopra, Saumil Shah, Dennis Sylvester and David Blaauw, "A Novel Approach to Perform Gate-Level Yield Analysis and Optimization Considering Correlated Variations in Power and Performance," IEEE Transactions on computer-aided design and systems, vol. 27, No. 2, Feb. 2008.

[Sta95] Stan, M.R. and Burleson, W.P., Bus-invert coding for low-power I/O, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume 3, Issue 1, March 1995 Page(s):49 – 58.

[Tiw98] Tiwari, V., Malik, S. and Ashar, P., Guarded evaluation: pushing power management to logic synthesis/design, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Volume 17, Issue 10, Page(s):1051 - 1060 , Oct. 1998.

[Wes05] Neil H.E. Weste and David Harris, "CMOS VLSI Design – A Circuits and Systems Perspective", Pearson Education, Inc., 2005.

[Zho03] Huiyang Zhou, Mark C. Toburen, Eric Rotenberg, Thomas M. Conte, Adaptive Mode Control: A Static-Power-Efficient Cache Design. ACM Transactions on Embedded Computing Systems (TECS). Volume 2, Issue 3, Pages: 347 – 372, Aug. 2003.