

# Power Conscious Design with ProASIC

## Introduction

The last few years have catapulted designers into another realm of high-speed and complex products, where on-chip operation frequency is routinely over 100 MHz. The first hurdle in designing such systems is meeting timing requirements. Another important concern is mastering all parameters and sources of power consumption within a certain budget. This consumption is particularly tied to the switching activity and is data pattern dependent. Also, power improvements tend to reduce noise effects and help solve the third hurdle, signal integrity.

Power consumption, a persistent concern for digital designers, is becoming more of an issue as programmable logic providers offer devices with higher performance and gate counts. The more power the part utilizes, the hotter it operates and the slower the implemented application runs. Developers of battery operated designs that are used in portable products and systems employing interface cards struggle with this problem. In addition, a lack of tools and accurate models to estimate and verify the power consumption at each stage of the design cycle adds to the problem.

To significantly improve the chances of designing under power constraints, designers must consider and make use of the most power friendly FPGA architectures, power-conscious design techniques and practices, and a design methodology combined with a power estimation tool.

This application note uses concrete results measured on silicon to demonstrate that ProASIC is the most power-efficient FPGA on the market.

It is organized into six sections:

- The first section describes commonly used theoretical models to estimate overall static and dynamic power consumption.
- The second section evaluates the contribution of ProASIC power-friendly features to the reduction of both static and dynamic internal power consumption. It demonstrates that ProASIC technology offers the most appropriate feature set to implement designs under tight power constraints.
- The third section provides several RTL design techniques that allow efficient management of static and dynamic power. It covers the definition of clock domains and their

correlation, gating clocks, HDL coding to avoid or reduce glitches, and implementation selection for datapath basic blocks. This section also covers the effect of other RTL architectural decisions such as pipelining, state encoding, and buffering.

- The fourth section introduces the block-based methodology and the use of the power estimation tool integrated in ASICmaster.
- The fifth section presents experimental results obtained on real life designs classified in several categories that cover all the major application domains.
- The final section presents some conclusions.

## Static vs. Dynamic Power Models

The main distinguishing factor between static and dynamic power is that the dynamic power is frequency dependent, while static is not. Static power is defined as the product of the power supply voltage and static current, which itself has two components: leakage current and through current (Equation 1). Leakage currents have parasitic effects and are small in magnitude and therefore, can be ignored. Through currents occur in normal operation and result from transistors being continuously operated in their saturation region.

$$P_{static} = V \cdot I = \frac{V^2}{R} \quad (1)$$

Dynamic power has two components: the capacitive load power and the cell power (Equations 2 and 3). The latter is consumed internally by the cell primitives. This component accounts for the power that is primarily required to charge and discharge the internal cell capacitance. Capacitive load power represents the currents required to charge the external loads driven by each cell. The overall dynamic power for an entire chip is given by

$$P_{dynamic} = P_{dynamic\_loads} + P_{dynamic\_cells} \quad (2)$$

Where,

$$P_{dynamic\_loads} = V_{DD}^2 \cdot C_{node} \cdot f_{node} \quad (3)$$

$$P_{dynamic\_cells} = E_{dynamic\_cells} \cdot f_{cell} \quad (4)$$

The total power dissipation is the sum of the dynamic and the static components.

### Average Power Dissipation

When computed over a number of clock cycles, the equations listed above produce time-averaged power used to analyze the effect of power on battery life, junction temperature, etc. Temperature analysis also relies on the same analysis, i.e. steady-state temperature estimates. Average power consumption is used as a rough estimate. However, system power budgets are often based on the peak power.

### Peak Power Dissipation

Performing the same analysis on a cycle-by-cycle basis produces peak-power value, which is most useful in determining the power and the number of ground pins needed to minimize ground-bounce effects and to check noise limits.

## ProASIC Power-Friendly Features

The following subsections highlight the power-efficient features of the ProASIC flash-based technology, which help implement power-conscious design rules.

### The Flash Switch

To store electrical charges, the flash technology needs only one transistor with a floating gate, compared to a larger number of transistors required by SRAM-based technologies (Figure 1). This results in a smaller die size and reduces power requirements.

### The Logic Tile

The basic logic “tile” is very similar to a gate array gate (Figure 2 on page 3). It is a programmable 3-input, 1-output cell. Each of the inputs may be programmed for signal inversion, enabling easy netlist optimization. Unlike other

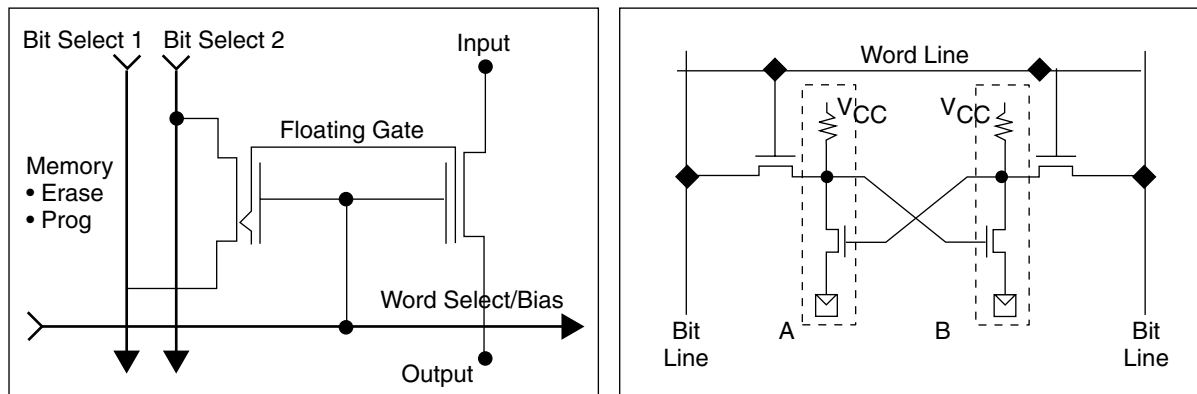
fixed architectures, the tile can be configured to operate as either a 3-input combinatorial cell or as a flip-flop. This eliminates the unnecessary burning of power for unused registers that occurs in SRAM-based technologies. Finally, an unused tile is completely isolated and does not contribute to power consumption.

### Embedded Memory Blocks

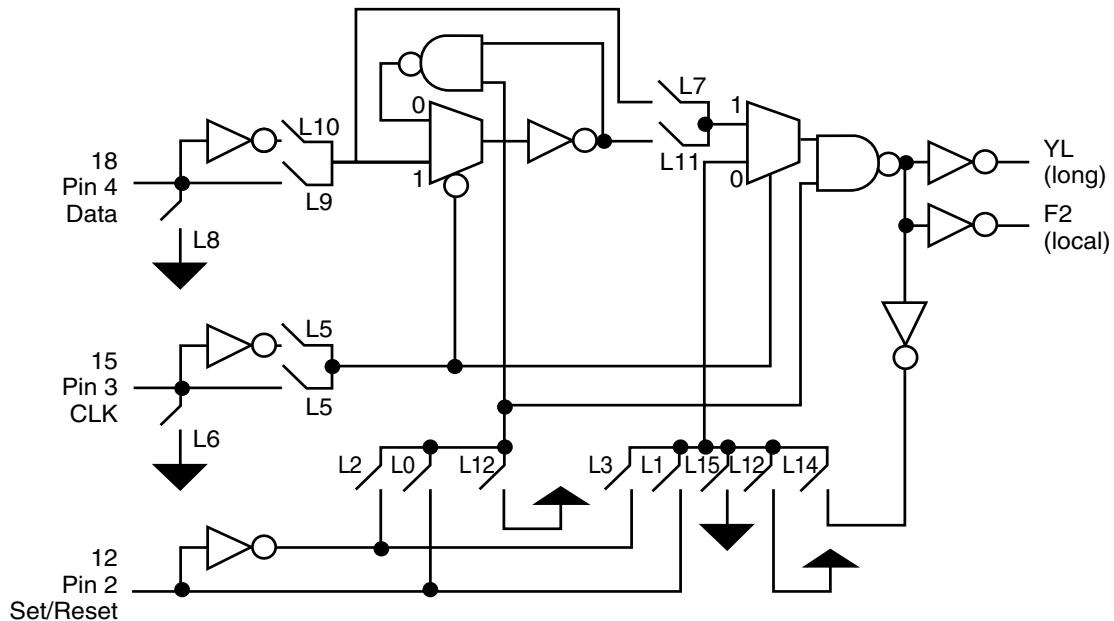
The configuration and the cascading of memory have a major impact on performance and power dissipation of portable applications. Without embedded memory, power is consumed at the chip's interface to external memory. Additionally, external memory has to be powered separately from the power source provided to the ProASIC part. In most networking applications such as Ethernet switches, where lower power, cost, and optimized bandwidth are critical, integrating as much embedded memory as possible onto the ProASIC part will save power for the entire system. Another important advantage of embedding RAM blocks is that it enables the conversion of pad-limited designs with high pin count packages to core-limited designs with lower pin count packages. The power-friendly cascading of basic memory blocks is discussed in [BZ99].

### Routing Resources

The routing architecture offers five levels of routing resources [BZ99]. The combination of these resources helps not only reduce power consumption, but also allows low power design techniques such as gating signals or clocks. For instance, the global routing networks may be mapped to external clock signals or to high fanout internal nets such as gated clocks. The high-speed very long lines have slightly higher capacitance than the discrete one-, two-, and four-tile long lines. However, if a signal routing requirement is long, the high-speed very long line offers an overall lower capacitance and better timing characteristics.



**Figure 1 • Flash Switch vs. SRAM Switch**



**Figure 2 • Architecture of the Logic Tile**

Additionally, all these routing resources are segmented, so the router is able to avoid using the unnecessarily long tracks, resulting in lower power consumption. The global routing network can be split if the internal or external clock distribution is limited to a part of the die. If not completely used, the global free portion is isolated.

### Input and Output Pads

The architecture offers separate I/O and logic core power rings. The core logic is driven by a 2.5V supply, while the I/Os are individually selectable as 3.3V or 2.5V [ProASIC99]. Moreover, the I/Os may be configured to operate with three different slew rates and support a low-power mode. Recommendations on how to configure low-power I/Os taking into account board considerations are introduced in [BZ99].

### Low Power Design Rules

The power driven methodology considers power dissipation at all levels. It is based on the use of tools and techniques at each of the design phases. As in the performance domain, early power specification and analysis helps with critical architecture decisions.

Power analysis tools that enable designers to make informed decisions at an early stage about the most power-efficient architecture and design technique are mandatory. However, these tools alone are not sufficient and should be combined with design rules that address unnecessary switching activity propagation.

As follows from the equations listed on [page 1](#), there are four factors that ultimately determine power consumption of a device: the magnitude of the supply voltage, the clock

frequency, the switching capacitive loads, and the switching activity in the circuit. Different optimization methods targeting each of these factors have been explored [Bernard96, IA96, Rabe96, Zafalon97, DS98]. Reduction of supply voltage, multiple voltage supplies, reduction of “capacitive” loads through gate sizing, and minimization of switching activity by exploiting the correlation between signals are just a few. On the other hand, the four factors strongly interact in ways that may cancel out the power optimization benefits obtained by adjusting only one of them. Additionally, many studies have shown that only optimizations applied sufficiently early in the design cycle, when a design's architecture is not yet fixed, have the potential to reduce power. In the ASIC world, gate size tuning at the logic level produces reductions averaging 10 percent. This is not possible when targeting an FPGA. However, optimizations at behavior and architectural levels can potentially slash power consumption by close to a factor of 10. Thus, to make intelligent decisions in power optimization, designers have to simultaneously consider all four factors affecting power dissipation, and apply the power conscious analysis and design rules early in the design cycle.

### RTL Power-Conscious Architectural Decisions

The main RTL architectural decisions are relative to selection of basic arithmetic blocks, state machine encoding, clocking schemes as well as buffering and pipelining. The following sections analyze their effect on power dissipation.

## Arithmetic/Data Path Elements Selection

Careful selection of appropriate arithmetic blocks is a source of large power savings. In this section, several adder and multiplier architectures are studied with regard to area, speed, and power dissipation. These architectures are provided by DesignWare, the Synopsys macro generator. This tool automatically generates the appropriate architecture for arithmetic blocks based on user timing constraints and mapping efforts.

### Adders

The selected architectures are the Forward Carry Look Ahead (CLF), the Brent and Kung (BK), the Carry Look-Ahead (CLA), the CSM, and the Ripple (RPL) adders. [Figure 3 on page 5](#) shows that the CLF is the fastest architecture compared to CLA, CSM, and RPL for a variety of bit widths. A closer look shows that BK architecture leads to the best speed/area trade-off [BK82]. For a comparative power study, experimental measurement on real silicon, illustrated for a 32-bit adder with speed oriented mapping, shows that the BK is the most power-friendly architecture on ProASIC ([Figure 4 on page 6](#)). This is because both the number of logic levels and the number of internal nets, in the BK architecture, are the smallest among all the architectures.

These results are easily explained when analyzing the fanout distribution of the internal nets and the number of logic level curves presented in [Figure 5 on page 6](#) and [Figure 6 on page 7](#). On one hand, the number of internal nets (i.e. nets with fanout ranging from 6 to 38) in the BK architecture is the smallest. On the other hand, the BK architecture has the lowest number of logic levels. The combination of these two factors implies that the switching activity and its propagation through the logic are the smallest in the BK when compared to the other architectures. An identical comparative power study was performed on the same adder architectures with various bit-widths but with area oriented mapping. The results show that on ProASIC, the BK architecture is an optimal implementation of adders since it provides a speed close to the one delivered by a CLF architecture for minimal area and power consumption. The same results show that CLF and RPL architectures have almost the same power dissipation. This emphasizes the effect of both the number of logic levels and the net fanout on the switching activity propagation and thus on the dynamic power consumption.

---

### Adders Selection Rule

*The fact that BK architecture is leading to the best power budget does not necessarily mean that all the adders must have this architecture. However, a reasonable selection rule consists of replacing all the adders in the critical path or critical range and forcing DesignCompiler/FPGACompiler to infer the Brent and Kung architecture.*

---

### Multipliers

For multipliers, the study considered first CSA, Wallace and Non Booth Encoding Wallace (NBW) architectures. Experimental power measurements have been done on 16-bit multipliers. The results presented in [Figure 7 on page 7](#) show that the Wallace architecture is significantly more power-friendly than the CSA multiplier. However, the NBW architecture is by far the most power-friendly of all the architectures.

First we explain the difference between the CSA and Wallace power consumption. This difference occurs because the Wallace tree is more equilibrated and the switching activity propagation is uniform. Additionally, the number of logic levels in the Wallace tree is significantly less than in its CSA counterpart. Another important advantage is related to the fanout distribution of the Wallace architecture.

The number of high fanout nets in the CSA architecture is larger than in the Wallace (see [Figure 8 on page 8](#)). Consequently, the switching propagation is more limited in the Wallace multipliers.

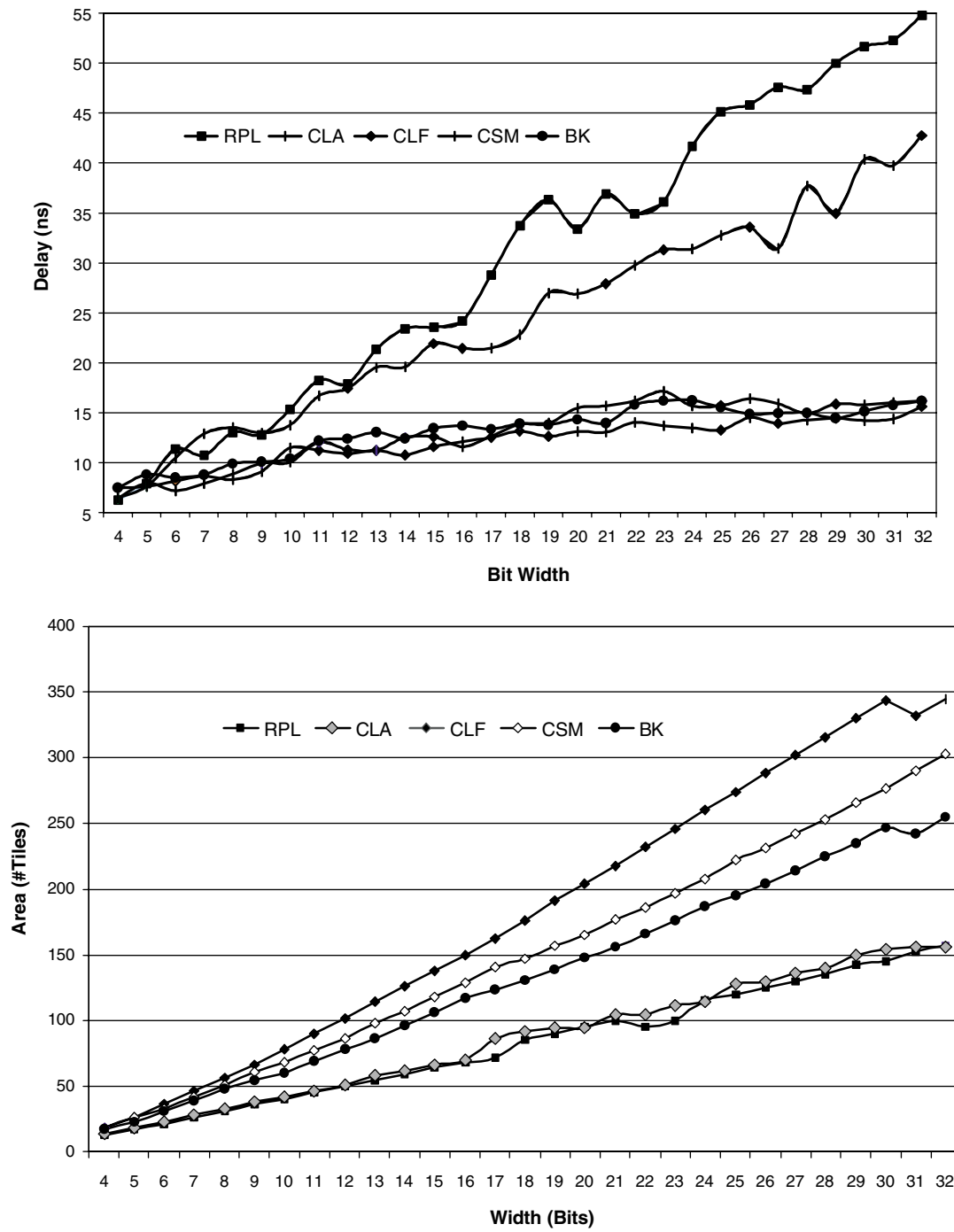
Second, we explain the huge power difference between Wallace and NBW: a closer look at the fanout distribution difference does not explain the amount of the difference. To better understand the source, the effect of the fanout on the place-and-route performance was studied. [Figure 9 on page 8](#) shows the delay variation for various post-layout wire lengths. It also translates the congestion and the delay hit inside the Wallace architecture. This better explains the difference in power dissipation.

---

### Multipliers Selection Rule

*The NBW architecture leads to the least power consumption. A rule of thumb consists of forcing DesignCompiler or FPGA Compiler to infer the NBW architecture particularly for the multipliers that are part of the critical path or in the paths that are close to critical, i.e. in a reasonable critical range. Another recommendation is to seriously consider pipeline multipliers with one or two stages, even if they meet the timing requirements with a non-pipelined configuration.*

---



**Figure 3 • Postlayout Performance and Area for Various Adder Architectures**

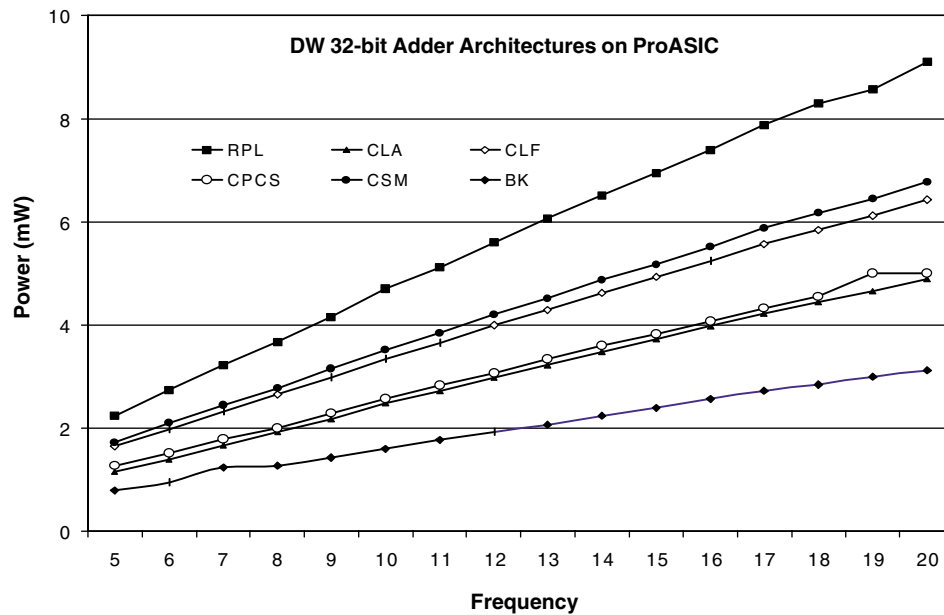


Figure 4 • ProASIC Silicon Power Characterization for Various 32-bit Adders

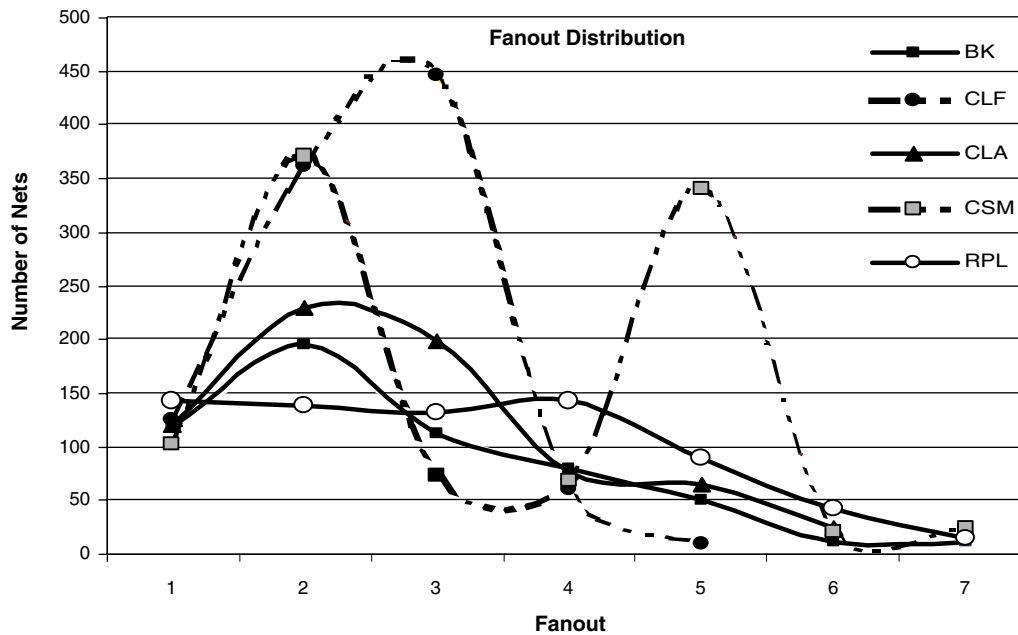
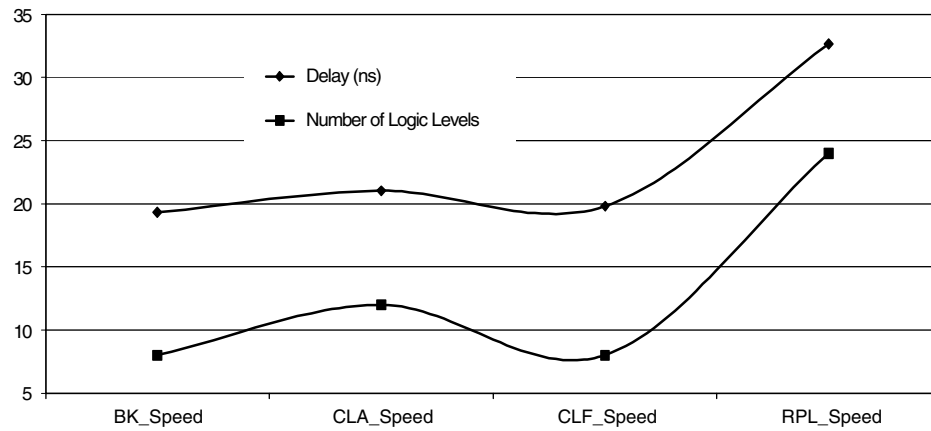
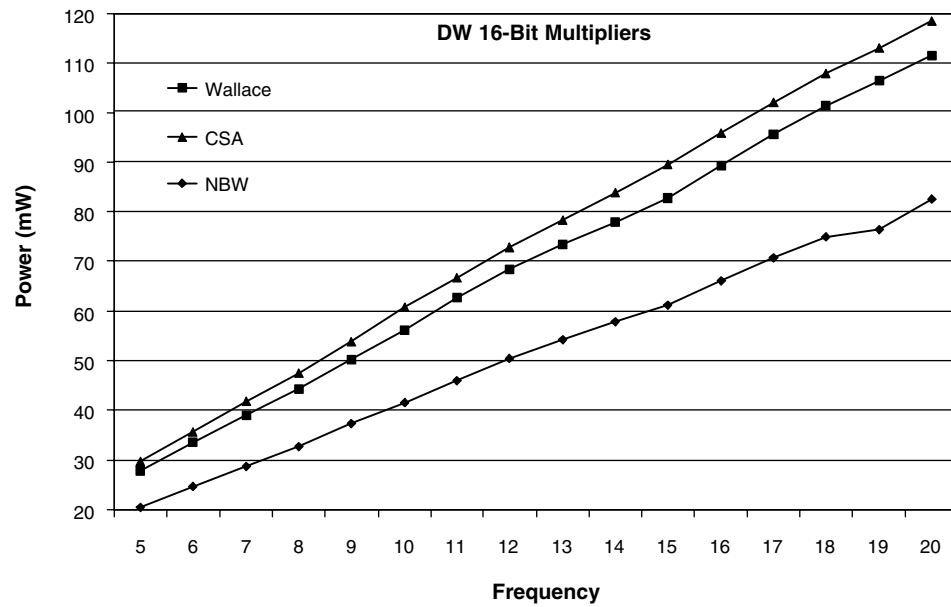


Figure 5 • Fanout Distribution for Various Adders' Internal Nets



**Figure 6 •** Number of Logic Levels for various 32-bits Adder Architectures



**Figure 7 •** ProASIC Power Characterization for various 16-bit Multipliers

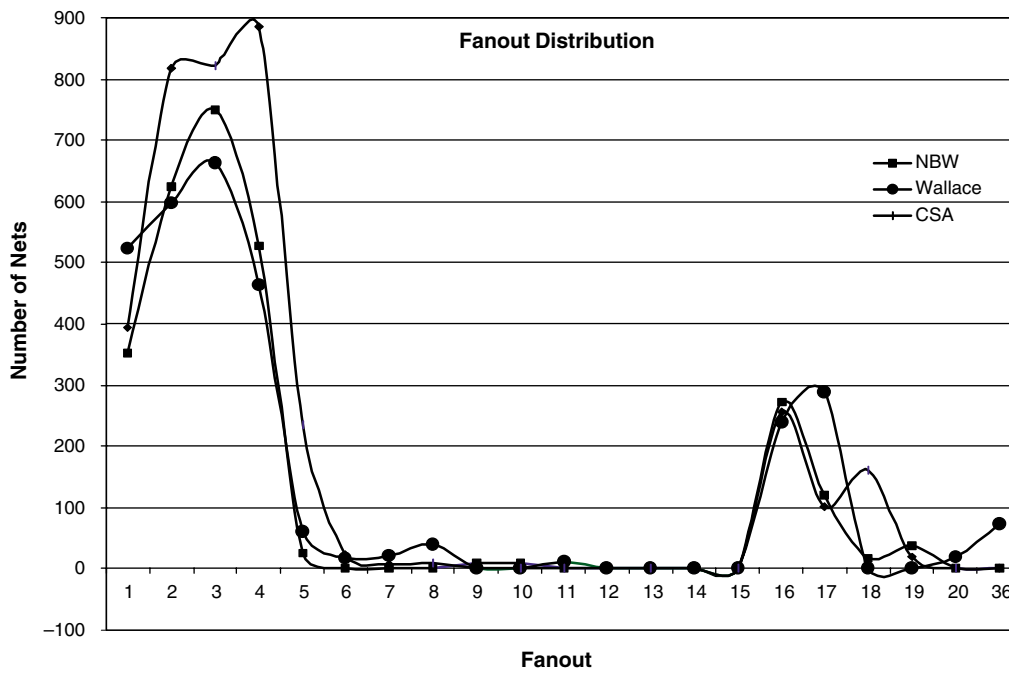


Figure 8 • Fanout Distribution for 32-Bit DesignWare Multipliers Mapped on ProASIC

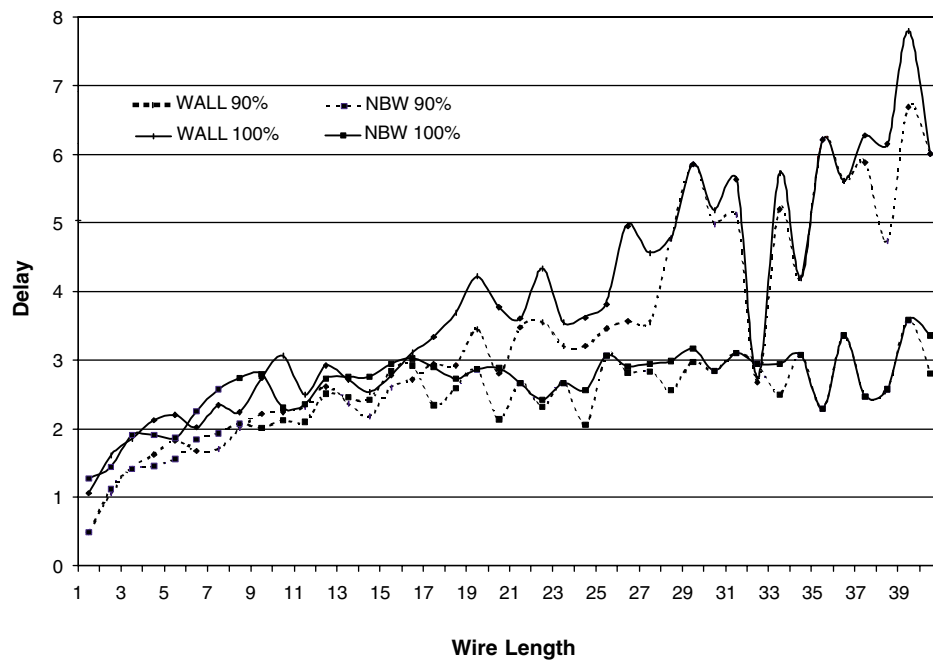


Figure 9 • Delay Variation for Various Wire Lengths in Wallace and NBW Architectures



### Finite State Machine (FSM) and Counter Encoding

Several studies compare the impact of the encoding options on performance and area results when targeting FPGAs [Belhadj94]. When considering lower dynamic power as an optimization criterion, the number of possible state registers and their transitions is a credible metric to use when comparing encoding options. To make this measure more accurate, it must be combined with the impact of the state register transitions on the output and next state logic. When targeting FPGAs, both the number of state registers, i.e. clock loads, and the number of state code bits changing per clock are considered.

#### Counter Encoding Impact on Power

**Table 1** compares one-hot, Gray, binary and LFSR state assignments for a counter with 8-states. Results show that one hot and linear-feedback shift-register (LFSR) and other shift-register-based state encoding exhibits large clock loads due to the number of flip-flops or a high average number of flip-flops toggling at each clock cycle. The comparison also shows that the Gray technique reduces both the average number of logic transitions per clock and the overall number of transitions for a cycle of the state machine. With more focus on common return-to-zero-state transitions, more power reduction can be achieved. Probabilistic studies determining the most frequent paths in the state machine also help to save more power [Bde94]. The experimental power measures on silicon confirm the conclusions based on the criteria introduced earlier. **Figure 10 on page 10** presents the power dissipation for 200 instances of 8-bit counters.

#### FSM Encoding Effect on Power

The main difference between counters and FSMs is that predicates on transitions between FSM states are not always “true,” which complicates next state and output functions. The power consumed by the combinatorial next state and output logic is important and can counterbalance savings implied by reduced clock load and transitions of the state register itself.

In this context, the study focused more on the output logic. Unlike the case of counters, the minimal number of registers also implies a more complex decoding of the output logic. In turn, the one hot encoding implied output logic is a simple OR of the product terms associated with the active states for each of the outputs of the FSM. The power measures on ProASIC silicon validate this point, as the selected state machine has 170 states and a large number of outputs. Even if the clock load is higher for the one hot configuration, the switching activity of the next state and output logic is substantially smaller than in the case of a Gray or binary sequential codes (**Figure 11 on page 10**).

Future studies will look at the power dissipated by the next state logic with a focus not only on the state assignment, but also on the structure of the state graph. An earlier study [Belhadj94] revealed that the number of states, the number of paths and their lengths, and the number and the complexity of the fork situations, have a huge impact on timing and area.

**Table 1 • State Codes and Number of Transitions and Clock Loads per Clock**

State	One Hot	Gray	Binary	LFSR
S0	00000001	000	000	111
S1	00000010	001	001	110
S2	00000100	011	010	100
S3	00001000	010	011	000
S4	00010000	110	100	001
S5	00100000	111	101	010
S6	01000000	101	110	101
S7	10000000	100	111	011
Total Number of Transitions	18	8	14	13
Maximum Transitions Per Clock Cycle	2	1	3	3
Clock Load	8	3	3	3

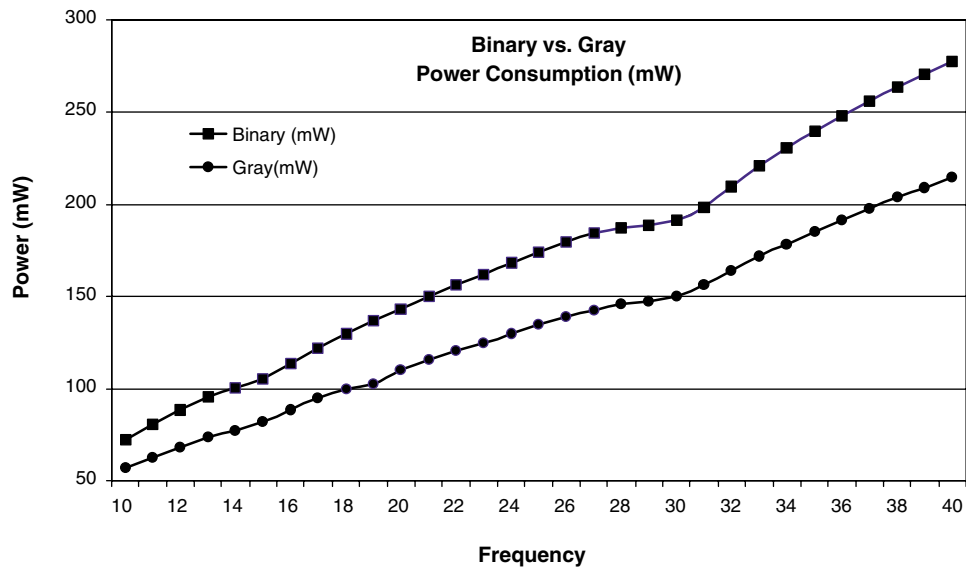


Figure 10 • Comparative Power Consumption for 200 instances of 8-bit Counters

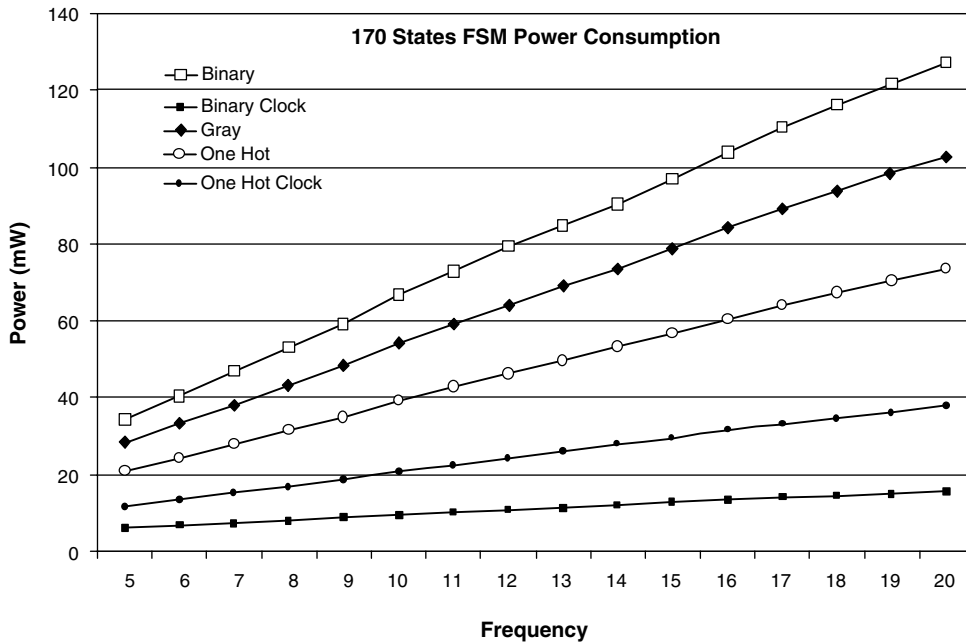


Figure 11 • Power Measure on ProASIC of 170 States Controller

### State Assignment Selection Rule

*The selection of the state assignment depends on several parameters such as the complexity of the state machine, i.e. the number of states, the number of paths and their lengths, the number of fork situation and the complexity of the predicates on transitions between states. As a rule, if the number of active states for each output of the FSM is relatively reduced compared to the total number of states, then the one hot encoding is the best candidate. Remember that in the case of one hot encoding of a Moore machine, extracted output Boolean functions are simply an OR of all  $Q_i$ , the outputs of the active states' hot register. Also, the next state Boolean equations will excite at a maximum of two registers at each transition between states, thus switching activity propagation is very local.*

*If the number of active states is very large, the output logic will need a deeper logic compared to the depth of the output logic extracted for a sequential encoding. Gray encoding is selected in the case of counters only.*

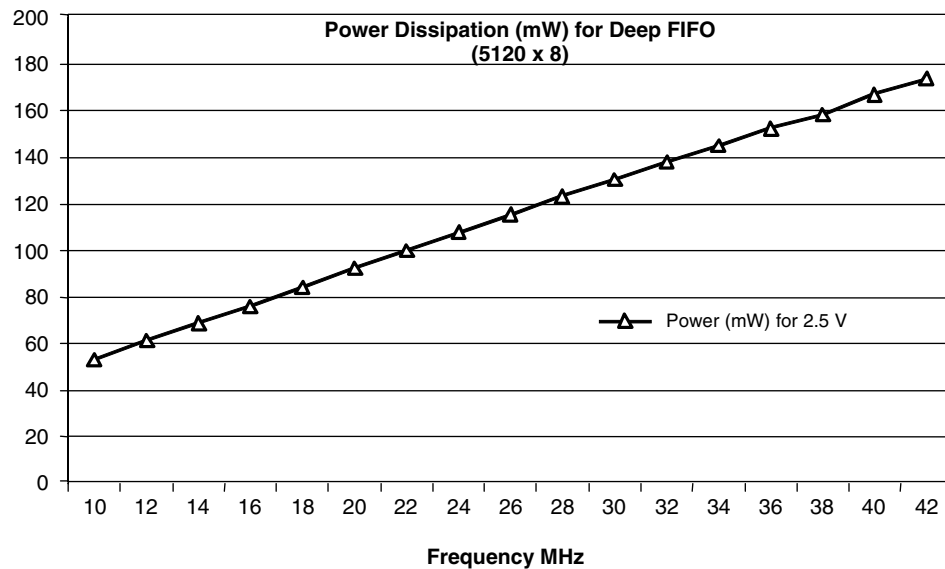
### Embedded Memory Blocks Power Characterization

Configuration and cascading of ProASIC embedded memory blocks have a major impact on the performance and power dissipation of portable applications. Without embedded

memory, power is consumed at the chip's interface to external memory. Additionally, external memory has to be powered separately from the power source provided to the ProASIC part. In most applications such as Ethernet switches, where lower power, cost, and optimized bandwidth are critical, integrating as much embedded memory as possible onto the ProASIC device will save power for the entire system. Another advantage of embedding RAM blocks is that it enables the conversion of pad-limited designs with high pin count packages to core-limited designs with lower pin count packages. The power-friendly cascading of basic memory blocks is discussed in [BZ99]. Figure 12 draws the power consumption for a deep Synchronous Read/Synchronous Write FIFO.

### Rule for Low Power Reduced RAM/FIFO Implementations

*The ProASIC embedded memory blocks are very low power blocks as the available embedded blocks were needed to start measuring the current with very sensitive measuring equipment. If designers prefer to customize these blocks and make up the address decoding themselves, rather than using MEMORYmaster, the recommendation is to use a Gray type of address counter.*



**Figure 12 • Power Dissipation of Deep FIFO Using ProASIC Embedded Memory Blocks**

### Pipelining Effect on Power

In addition to the speed-up that a pipeline stage may introduce, it is also supposed to stop the switching activity for a given data pattern and to reduce the fanout distribution. The side effect is related to the increase of the clock load and the parallel execution. Another important aspect to consider is related to the number of pipeline stages to introduce. As for timing optimization, the power consumption is reduced significantly with the first couple of stages and then becomes less significant. [Figure 13](#) shows experimental results obtained for Wallace architecture with various pipeline stages. As expected, the power consumption is reduced substantially.

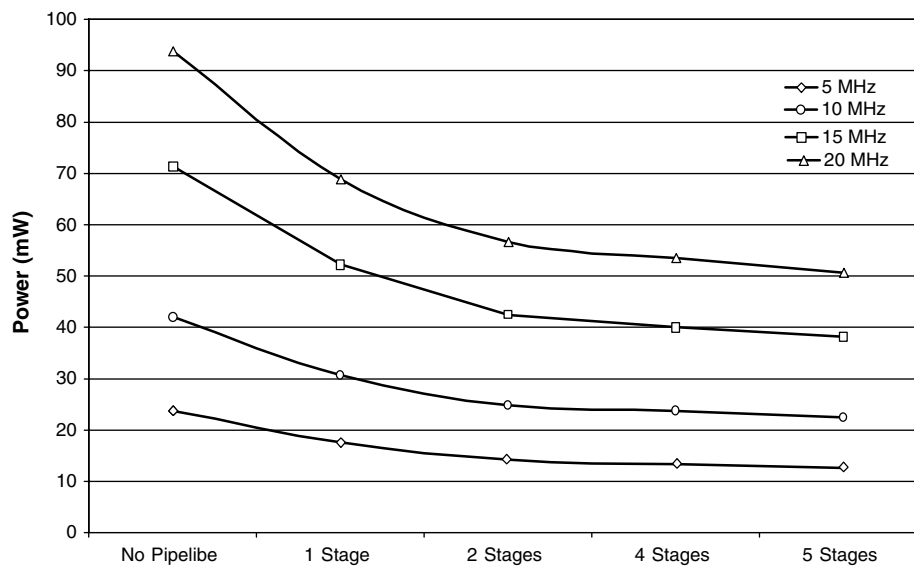
A slightly higher power dissipation for the 3-Stages multiplier configuration has been noticed. Deep investigation revealed marginal place-and-route effects as the experience included all the various configurations in one device, which apparently stressed the block-based place-and-route tool for the 3-stages block. Further

investigations are in progress to find other root causes of this slight increase.

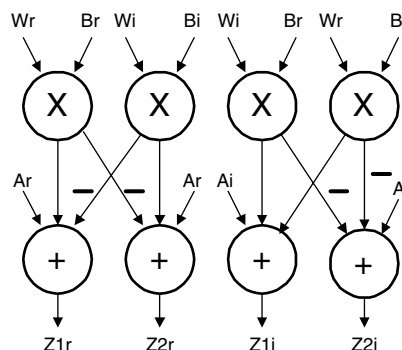
To complete the study of pipelining effect, a power characterization of ModuleCompiler designs is introduced. The design set considered during the experimentation included several ModuleCompiler blocks with various complexities. For the purpose of simple illustration, only two configurations (pipelined and non-pipelined) of a Fast Fourier Transform design are discussed.<sup>1</sup>

The FFT design consists of a set of multipliers followed by an array of adders that add to or subtract from the multiplier outputs an externally applied value as depicted in [Figure 14](#). For more details on the MCL description of this design see [BGLS2000].

1. Module Compiler has been selected because this tool has the ability to automatically pipeline a design with the appropriate number of pipeline stages based on the targeted timing constraints.



**Figure 13 • ProASIC Power Characterization for 16-Bit Pipelined Multipliers**



**Figure 14 • Fast Fourier Transform (FFT) Block Diagram**

Figure 15 draws the power dissipation for two configurations of the FFT design as well as their correspondent clock trees. Although the clock tree dissipation of the pipelined configuration is for very high frequencies, the power dissipated through the correspondent logic blocks is drastically reduced in comparison to the non-pipelined configuration.

To explain this variation, effects of fanout distribution and switching propagation have been investigated. Table 2 provides information on obtained postlayout results. The

column “Number of Logic Levels” shows the large difference between the depth of the most critical paths that partially explains the hit on power for the non-pipelined FFT.

On the other hand, the curves of high fanout distribution presented in Figure 16 on page 14 demonstrate the power-relaxation of the final architecture when introducing the pipeline stages.

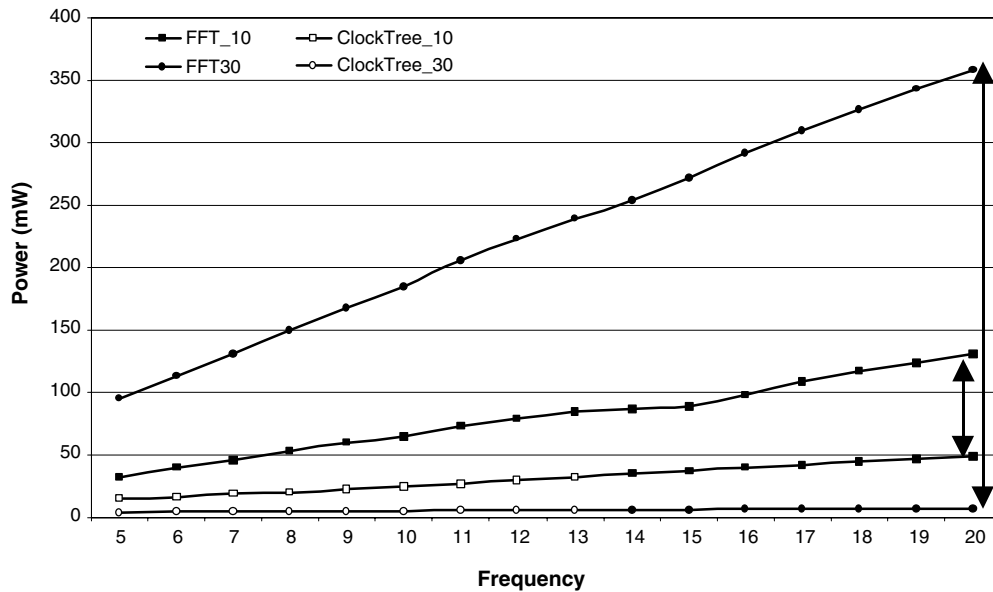


Figure 15 • ProASIC Power for Pipelined and Nonpipelined FFT Configurations

Table 2 • FFT Summary of Results

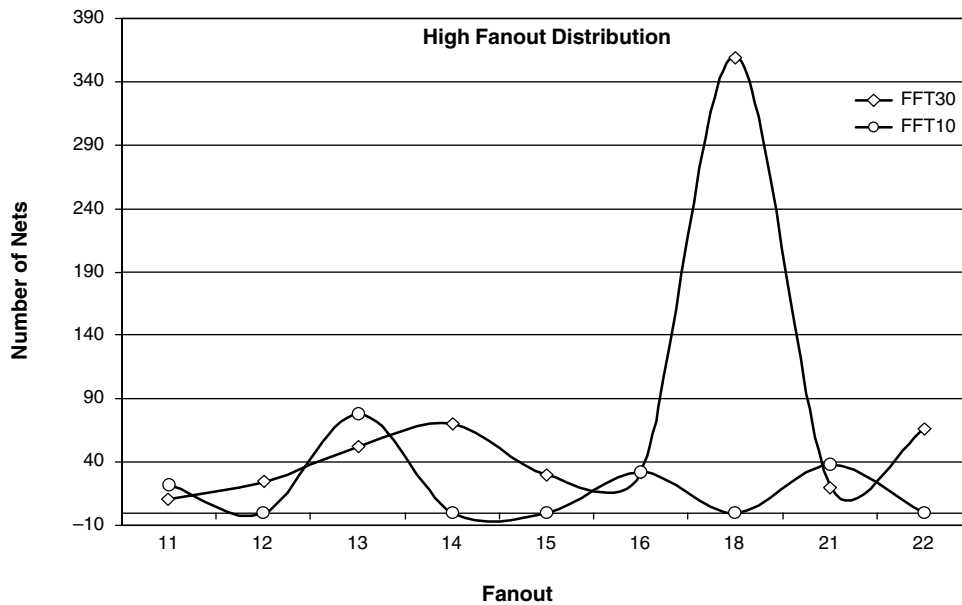
Design	Target Clock Speed	Input Pin Count	Output Pin Count	Number of ASIC Gates	Number of Logic Levels	Post-Layout Clock Period on ProASIC Netlist
FFT10	10	49	32	18039	7	10.70
FFT30	30	49	32	15032	24	31.10

### Rules for Pipelining

Introducing pipeline stages shows a real power reduction. The designer needs to determine the number of stages. A high number of registers may increase the power because of the higher utilization of the resources and clock load. The recommendation is to introduce one to two stages if the frequency of the design is low. If the frequency is above 50 MHz, 3, 4 or even 5 pipeline stages will significantly reduce the power consumption.

### Clocking Schemes

As clock frequency is the primary determinant of dynamic power for synchronous designs, ProASIC provides four different low skew global networks that enable designers to drive each group of flip-flops from one of the 40 external or internal clock “splines” (for the smallest ProASIC devices). This helps to avoid the use of a generic input as the flip-flop clock and tradeoff increased skew and input setup- and hold-time requirements.



**Figure 16 • High Fanout Distribution for Pipelined and Non-Pipelined FFT Configurations**

### Clocks' Scope Separation

It is a common design practice to drive different groups of registers with distinct clocks at different clock frequencies. Besides the setup- and hold-time requirements, the designer must master the skew between the rising and falling edges of these clocks in order to avoid metastability in the design. This problem is particularly tedious if the logic blocks interact with each other. A workaround is to have the clocks act as multiples of each other or to use the ProASIC clock spines.

### Clocks Gating

One clock-enable approach simply multiplexes the normal D-register input and its previous output. This eliminates possible glitches. However, a portion of the D-register still respond to falling or rising-clock' edges. Gating clocks is an alternative implementation of synchronous load enable registers and is considered an efficient way to prevent clock propagation to registers' clock pins whenever the load-enable signal is false. [Figure 17 on page 15](#) and [Figure 18 on page 15](#) introduce the general principle. Notice the power saving is due to the significant reduction of capacitance on the clock network and the internal power of the affected registers and the elimination of the N-Bits wide multiplexer and its connections.

### Gating Signals

Effective power implementation can be achieved using gating signals for particular parts of the design. Similar to the concept of gating clock, signal gating reduces the transitions in clock free signals. The most common example is the decoder enable. As part of an address decoding mechanism, signals used by other parts of the design may

toggle as a reflection of activity in these parts. Switching activity on one input of the decoder will induce a large number of toggling gates. Controlling this with an enable or select signal prevents the propagation of their switching activity, even if the logic is slightly more complex ([Figure 19 on page 15](#)).

### Rules for Clocks and Signals Gating

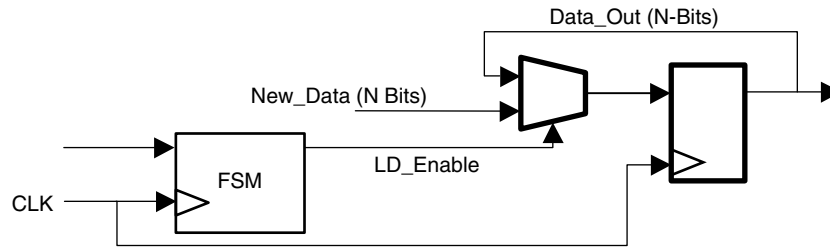
*If possible, gating clocks and signals saves power. It also complicates the testability and the clock and control signals' skew balancing [SNUGTutorial98]. The recommendation is to study the opportunity to reduce power and apply the gating accordingly. For clock gating, the saving opportunity is defined in terms of the number of affected registers (static factor) and the percentage of time the gated clocks are enabled (dynamic factor).*

### Code Motion for Data Path Re-ordering

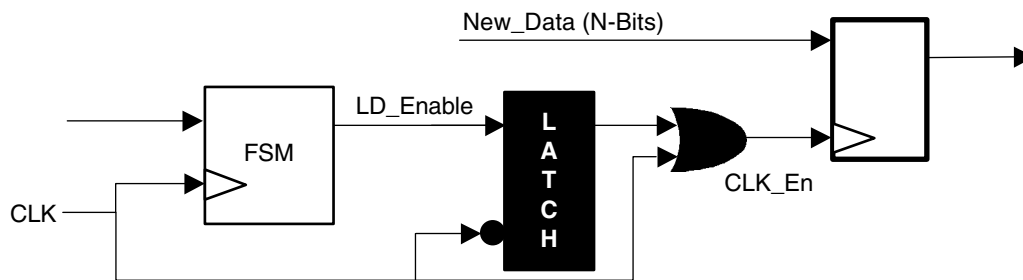
Several data path elements, such as decoders or comparison operators, as well as "glitchy" logic may significantly contribute to power dissipation. The glitches, caused by late arrival signals or skews, propagate through other data path elements and logic until they reach a register. This propagation burns more power as the transitions traverse the logic levels. To reduce this wasted dissipation, designers need to rewrite the HDL code and shorten the propagation paths as much as possible. [Figure 20 on page 15](#) illustrates two implementations of two "If ... Then ... Else" constructs where the "glitchy" and "stable" conditions are ordered differently.

The same re-organization is applicable for multiplexer-trees used for resource sharing. Balancing such a tree is recommended if the switching activity is uniform. However, when case one of the inputs of an equilibrated multiplexer-tree has a high “glitching potential,”

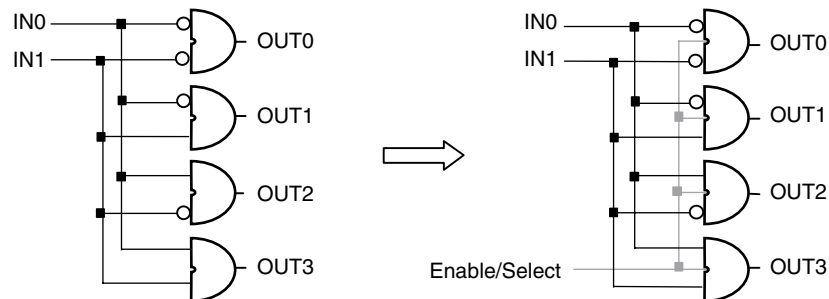
dis-equilibration of the tree must reduce the number of levels traversed by this signal. The same recommendations hold for CRC “Xor-trees” and chained arithmetic operators, particularly, if they are commutative.



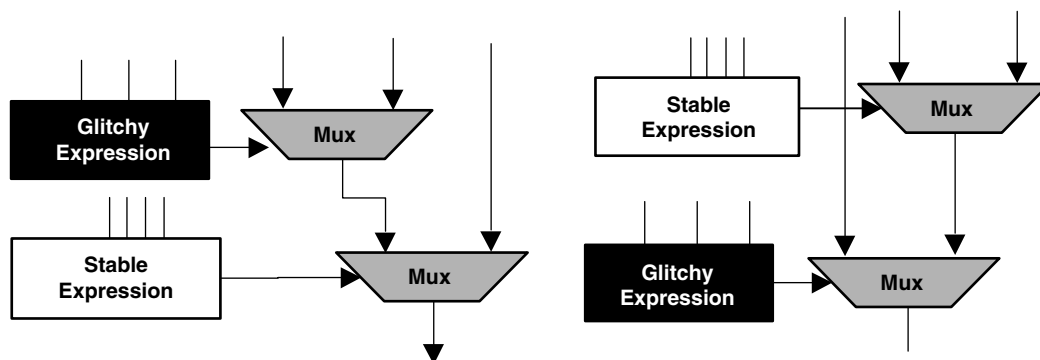
**Figure 17** • Clock-enable N Bits Wide Register Implementation.



**Figure 18** • Gated-Clock Implementation.



**Figure 19** • Decoder with Enable



**Figure 20** • HDL Code Motion or Datapath Re-ordering to Reduce Switching Propagation



## Block-Based Power-Driven Methodology

ProASIC's ASIC-like fine-grain library allows ASIC designers as well as FPGA designers to easily apply a hierarchy-based methodology. Figure 21 introduces the suggested approach and focuses on links between design phases. A more detailed presentation of the timing-only-oriented block methodology is introduced in [BABZ2000]. The Synopsys tools are presented here for illustration purposes only. Other tools such as Synplify from Synplcity and LeonardoSpectrum from Exemplar also support the ProASIC family of devices.

### Methodology Principles

The block-based design methodology can be roughly presented as follows:

1. Manipulation of the initial design hierarchy in order to better fit the optimization algorithm embedded in DesignCompiler, ModuleCompiler, and ASICmaster, the place-and-route tool.
2. For each block, synthesis is performed to get an estimation of the performance and to generate forward-timing constraints to the place-and-route tool.
3. The block is then placed and routed. In addition to the previous forward timing constraints, the user can define various floorplanning constraints.
4. After successful layout of the block the user can evaluate the timing and estimate the power, and then generate an SDF backannotated timing to update the top-level design.
5. If power budgets are not met, users can modify the synthesis script, select more power-friendly architectures, ask for re-timing or use pipelined configurations of some blocks.
6. Once all blocks are processed, the top-level design is compiled with accurate time and power budgets.
7. If the power budget is not met, users have the choice to optimize the design using the high-level decisions presented earlier such as more effective arithmetic resource selection, pipelining, wise state encoding, gating clocks or even HDL code re-investigation. The system architect or block integrator can also implement power control logic that switches on and off exclusively active blocks

Notice that in Figure 21, timing budgets have a higher priority over power. This can be changed if the design is not timing critical. The power-driven part of the flow is based on an estimation tool that is integrated into ASICmaster.

To help implement this design approach, ASICmaster integrates a Power Estimator that is briefly introduced in the next section.

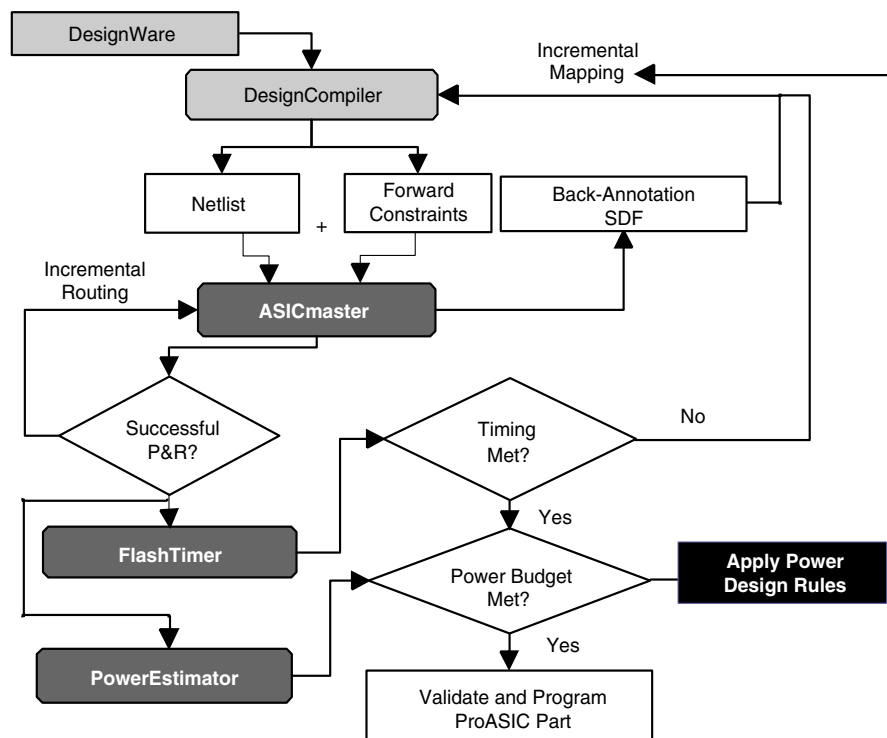


Figure 21 • Block Diagram Design Methodology



**ASICmaster Power Estimator Utility**

In this tool, design power is estimated in the same manner as CMOS gate arrays and includes both static and dynamic terms. The dynamic part is a function of both the number of tiles utilized and the frequency. The overall power dissipation estimator uses the following equation:

$$P = V_{dd} \cdot (I_{static} + I_{output} + I_{logic}) \quad (5)$$

where

$I_{static} = I_{static\_core} + I_{static\_io}$ , is the static current

$I_{output} = C_{typ} \cdot V \cdot f_{average} \cdot N$ , is the current due to output logic

$I_{logic} = 0.35 \cdot I_E \cdot G \cdot f \cdot F$ , is the current due to the internal logic

and where,

$C$  is the typical capacitance on a load

$V$  is the average voltage swing

$f_{average}$  is the average output switching frequency

$n$  is the number of active outputs

$I_E$  is the effective mA/gate/MHz of the parts

$G$  is the number of used gates (in thousands)

$f_m$  is the operating frequency in MHz for memories

$F_m$  is the fraction of memory devices active on each clock edge in %

The total power dissipation in Watt is:

$$(V \cdot (I_{ddq} + N \cdot C_{typ} \cdot V_{dd\_io} \cdot f_{avg} \cdot 0.001)) +$$

$$V_{dd} \cdot \left( \frac{0.35 \cdot I_E \cdot G \cdot f_c \cdot F_c}{100} \right) + \quad (6)$$

$$V_{dd} \cdot \left( \frac{0.35 \cdot I_E \cdot M \cdot 0.5 \cdot f_m \cdot F_m}{100} \right) \cdot 0.001$$

The user can set all these parameters according to his specific design and the tool will calculate the corresponding power dissipation. Figure 22 shows the menu of the tool.

ntw-belhardth1: "Tensilica" - Power Estimation

**Device Output**

Number of Active Outputs: 61

Average Pad Load (pF): 16

Average Frequency (MHz): 46

Average IO Voltage (V): 3.63

**Device Logic**

Gates (x 1000): 40

Operating Frequency (MHz): 50

Active Fraction (%): 15

**Device Memory**

Number of Memory Blocks: 10

Operating Frequency (MHz): 50

Active Fraction (%): 15

**Environment**

Air Flow (linear feet per minute): 10

Heatsink Reduction (%): 4

Ambient Temperature (C): 70

**Results**

Calculated Power (W): 1.689

Junction Temperature (C): 14.5

Close Calculate Help

Figure 22 • ProASIC Power Estimator Main Menu

### Methodology' Practical Advantages

If the architectural partitioning is done carefully, the manageable complexity of the created blocks favors incremental refinement and reduces the design time caused by iterations and late engineering changes. The block designer can then thoroughly investigate the solution space and select the most stable and efficient implementations. This investigation may include achieving certain objectives such as balancing timing performance, power dissipation, and testability.

At the integration level, integrators worry less about the blocks because they are validated and all of the complexity, performance, and power dissipation attributes are known. Integrators have an easier task when balancing competing design constraints. If the place-and-route tool supports certain capabilities, the timing and functional validations are straightforward. In the power arena, the system designer can implement an overall power control system that turns on and off clocking domains of exclusively active hierarchical blocks.

For the whole design team, the evidence of re-use advantages certainly creates the incentive to negotiate economical and technical barriers. Even if implementing such a methodology looks tedious at first, it is quite beneficial in the long run especially in terms of conserving resources and saving time.

### A Final Look

To meet design goals in terms of performance and power budgets, designers have to carefully select the target technology and think thoroughly at the architecture level. Experiences have demonstrated that curing is a tedious approach. To avoid iterations, a power-driven design approach has been proposed. Several RTL architectural decisions have been investigated with regard to power dissipation. Combined with wise functional partitioning and a power estimation tool, these rules ease the power consumption challenge and lead to a successful design validation.

---

*References:*

- [BASZ99] H. Belhadj, V. Aggarwal, N. Soria, B. Zahiri, "Power Conscious Design on A500K," International Workshop on Low Power Design, Moscow, September 1999.
- [Belhadj94] Hichem Belhadj, "State Assignment Selection for FSM implementation on FPGAs and CPLDs," IFIP Int'l Workshop on Logic Synthesis, December 1994, Grenoble, France.
- [BGLS2000] H. Belhadj, S. Goette, J. Lofgren, S. Sharif, "Mapping Module Compiler Designs into FPGAs," In SNUG'2000 Proceedings, San Jose, March, 2000.
- [BK82] R.T. Brent and H. Kung, "A Regular Layout for Parallel Adders," IEEE Trans. on computers, Vol. 39, pp. 260-264, March 1982.
- [BZ99] H. Belhadj and B. Zahiri, "Programmable ASIC Design Methodology Using Synopsys" SNUG Boston'99, Boston, October 1999.
- [Cha95] Chandrakasan et al., "Low Power Digital CMOS Design," K.A.P., 1995.
- [DS98] A. Dauman and B. Small, "Putting the Design Back in HDL Design," In Proceedings of PLD-Conference, January 1998.
- [Gailhard97] S. Gailhard et al, "Area/Time/Power Space Exploration in DSP High Level Synthesis," In Proceeding IP and Prototyping Workshop, December 1997.
- [Ghosh92] A. Ghosh et al., "Estimation of Average Switching Activity in Combinatorial and Sequential Circuits," In Proceedings of DAC'92, pp.: 249-299, 1992.
- [HC87] T. Han and D.A. Carlson, "Fast Area Efficient VLSI Adders," 8th Symposium on Computer Arithmetic, pp.: 49-56, May 1987.
- [Hwang99] E.O. Hwang, "Functional Partitioning for Low Power," PhD dissertation, University of California Riverside, June 1999.
- [IA96] M. Ikeda and K. Asada, "Bus Data Coding with Zero Suppression for Low Power Chip Interface," In Proceeding of Int'l Workshop in Logic Synthesis, 1996.
- [ProASIC99] ProASIC TM 500K Family Data Sheet, December 1999, ACTEL.
- [Rabe96] D. Rabe et al., "A New Approach to Gate Level Glitch Modeling," In Proceedings of IWLAS, Grenoble, December 1996.
- [SNUGTutorial98] "Low Power Design," Tutorial on Synopsys Power Tools, In Proceedings SNUG'98.
- [Tsui94] C. Tsui et al., "Technology Decomposition and Mapping Targeting Low Power Dissipation," In Proc. Design Automation Conference, San Diego, June 1994.
- [Tiwari93] V. Tiwari et al., "Technology Mapping for Low Power," In Proc. Design Automation Conference, June 1993.
- [Zafalon97] R. Zafalon et al, "Power Estimation and Synthesis: An Industrial Perspective," Invited Talk at PATMOS, September 1997.

Actel and the Actel logo are registered trademarks of Actel Corporation.  
All other trademarks are the property of their owners.



<http://www.actel.com>

Actel Europe Ltd.  
Daneshill House, Lutyens Close  
Basingstoke, Hampshire RG24 8AG  
United Kingdom  
Tel: +44-(0)125-630-5600  
Fax: +44-(0)125-635-5420

Actel Corporation  
955 East Arques Avenue  
Sunnyvale, California 94086  
USA  
Tel: (408) 739-1010  
Fax: (408) 739-1540

Actel Asia-Pacific  
EXOS Ebisu Bldg. 4F  
1-24-14 Ebisu Shibuya-ku  
Tokyo 150 Japan  
Tel: +81-(0)3-3445-7671  
Fax: +81-(0)3-3445-7668