Overview of the Power Minimization Techniques Employed in the IBM PowerPC 4xx Embedded Controllers

Anthony Correale, Jr.

PowerPC Integrated Solutions
IBM Microelectronics, RTP, NC

Introduction

The need for power efficient embedded controllers became obvious when the explosion in portable communications and consumer segments began. While the IBM PowerPC™ 4xx [1] embedded controller family addresses a wide variety of market segments, its low power consumption makes it particularly attractive for the consumer electronics and portable communications market segments. While low power consumption is most often thought of as being important to extend battery life, there are other advantages of using a power-efficient embedded controller:

The component can operate in a much harsher thermal environment because its low power results in a broader operating ambient.

The need for thermal enhancements such as heat sinks and forced air cooling is eliminated, thereby resulting in lower component and system costs.

In applications where component placement is critical due to thermal constraints, using the PowerPC 403GA gives the system designer additional flexibility.

In applications where thermal limitations are not of concern, the low power dissipation of the PowerPC 403GA affords enhanced reliability.

Additionally, a power-efficient embedded controller macro allows for a more robust 'Core-plus' strategy and implementation.

This paper discusses the many power conscious approaches used in the implementation of the PowerPC 403GA.

The PowerPC 403GA™ is the first member of the IBM 4xx embedded controller family. It is a 32-bit RISC embedded controller which contains a 2K-Byte Instruction Cache and a 1K-Byte Data Cache both of which are 2-way set associative. The chip also incorporates a two-level interrupt controller, a 4 channel DMA controller, a serial port unit, a Bus Interface Unit and a JTAG port which is also used for real-time debug. Currently, applications for the 403GA include office automation (eg., printers, copiers, fax machines), consumer electronics (eg., video games, set-top boxes, and PDAs), telecommunications, and networking.

Table 1 highlights the characteristics of the silicon process employed and the physical attributes of the PowerPC 403GA

Effective power management begins with the realization that capacitance and node switching must be minimized. Equation 1 shows that AC power dissipation is influenced by three elements: power supply voltage, capacitance, and switching frequency.

1. $\text{Power} = (0.5*C_l*V^2_{dd}*f) + (I_{sc} + I_{leakage})*V_{dd}$

Once the power supply voltage has been set, the remaining terms that influence power dissipation are the capacitance, C_l, and the switching frequency, f. The other terms in total power equation are the short circuit power, $I_{sc}*V_{dd,}$ expended when both n-channel and p-channel devices are on concurrently during a switching event, and leakage power, $I_{leakage}*V_{dd}$. Short circuit power can be minimized by managing transition rates, and leakage power is dictated by the technology employed (provided that circuit design is done properly).

Parameter	Value
Technology	IBM™ CMOS5L [2] 0.5um, N-Well CMOS, 3 Levels of metal
Chip Size	6.05 X 6.05 mm
Transistors	Approximately 600,000
Arrays	1-Kbyte, 2-way set associative Data Cache 2-Kbyte, 2-way set associative Instruction-Cache 32 X 32 5-port General Purpose Register File
Peripheral Units	Interrupt Controller 4 channel DMA Controller Serial Port Bus Interface Unit JTAG Port
PowerSupply	3.3 volt nominal
Power Dissipation @25MHz	200 mW typical @55C, 3.3v, 10pF load 660 mW maximum @ 85C, 3.3v, 50pF load 30 mW Wait State
Performance	54.5K Dhrystones @ 25 MHz 72K Dhrystones @ 33 MHz
Package	160 pin QPFP
I/O	127 signal, CMOS/TTL levels, 5V compliant

Table 1. Process and chip physical attributes

Capacitance minimization can be accomplished by minimizing the wiring distance between the driving and receiving circuit(s), using a routing medium with the least capacitance per unit length, selecting the proper driving and receiving circuit drive strength (power/performance code) and eliminating needless stray parasitics. With IBM's advanced 0.5 micron CMOS5L technology, metal layer capacitance associated with the area component of the metal line has been minimized through the use of thick oxides and inter-layer planarization. Lateral (line- to-line) capacitance becomes the

dominant term in the wiring capacitance equation. Therefore, while the area component is minimized, the importance of the lateral capacitance term increases as wiring pitches decrease. Hence, wiring capacitance while reduced, still plays a significant role in the overall net capacitance. For nets which have short routing distances, the dominant capacitance is associated with the gate oxide, the diffusion junction of the driving circuit and the parasitic field polysilicon. For a given frequency of operation, minimization of any of the above will result in lower power dissipation.

Power management techniques employed on the PowerPC 403GA embedded controller include:

1. A rich power/performance circuit library
2. Post processing algorithms for minimization of undesirable parasitics due to place and route
3. Compiled 1xN Macros
4. Local clock splitting & generation
5. Dynamic on-chip clock activation
6. Operand transition isolation
7. Gated and clocked receivers and common I/O
8. Extensive use of NFET transfer gates

Circuit Library

To minimize capacitance, a very robust circuit library was created that emulated a hand-crafted custom realization, but afforded the use of automated place and route. To this end, a broad spectrum of power/performance optimized circuits was realized. The granularity between the power/performance codes for each library element was defined to provide offerings in the capacitive loading ranges most frequently used. Based on our design experience from developing other IBM embedded controllers, it was expected that the average capacitive loading per net would need to be approximately 0.2pF to achieve the desired power/performance for this embedded controller. Although there would exist nets which had significantly higher capacitance, those nets are typically limited to a relatively narrow class of circuits and logical function. With this in mind, circuit drive strengths in increments of 25fF were realized.

The lowest power circuit, A-power, has a drive strength of 25fF, whereas the highest power code supports 2500fF. Table 2 defines the correspondence between the power/performance code and allowed maximum output capacitance. As seen by the table, the highest concentration of drive strengths is in the range of 25fF to 250fF. Here, the capacitance granularity is 25fF. Drive strengths in the range from 250fF to 500fF have a granularity of 50fF; the range from 500fF to 1pF have a granularity of 100fF; and finally, the range from 1pF to 2.5pF have a granularity of 500fF.

Power/ Perf Code	Maximum Cap.(fF)	Power/ Perf Code	Maximum Cap.(fF)
A	25	P	600
B	50	Q	700
C	75	R	800
D	100	S	900
E	125	T	1000
F	150	U	1500
G	175	V	2000
H	200	W	2500
I	225	X	
J	250	Y	
K	300	Z	
L	350		
M	400		
N	450		
O	500		

Table 2. Assignment of Power/Performance Codes to Maximum Cap

To keep the number of elements in the circuit library manageable, not all circuit elements were created with all power codes. Rather, the power codes for each element were created as a function of the circuit complexity. The broadest offering of power/performance codes were associated with the simplest function circuit elements (inverters and buffers), and the fewest power/performance codes were associated with the most complex functions (RAMs, GPR, Multiplier). Figure 1 illustrates the association of power/performance breadth to function complexity.

In a typical microprocessor design, single level logic primitives are used most frequently. Therefore, we created the broadest spectrum of power/performance circuits for single level logic primitives.

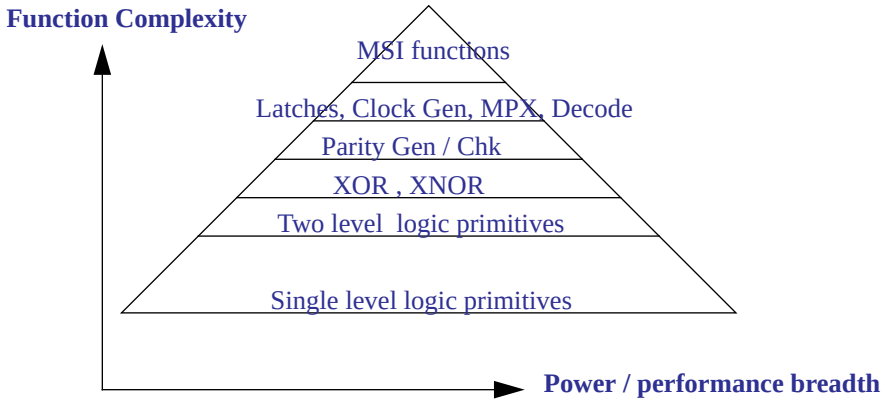

Figure 1. Function Complexity versus Power/Performance Breadth

The next most frequently used functions are two level logic realizations, and the range of their power/performance circuits offered is smaller than the single level logic primitives. Synthesis can create these functions by using components from the single level logic primitives, but not with the same overall area, wiring efficiency and power/performance. Therefore, this class of circuits was designed, albeit with fewer power/performance options than the single level logic class. The most complex functions are used with less frequency and as such, required fewer power/performance options. Hence, as function complexity increases, the power/performance breadth decreases.

The library of circuits was comprised of the following functions;

- NOR 1 - 4
- OR 1 - 4
- NAND 2 - 4
- AND 2 - 4
- AOI !(ab+c, ab+cd, ab+cd+e, ab+cd+ef, ab+cd+ef+g, ab+cd+ef+gh)
- AO (ab+c, ab+cd, ab+cd+e, ab+cd+ef, ab+cd+ef+g, ab+cd+ef+gh)
- XOR 2
- XNOR 2
- 4-bit Full Adder
- Selectors (1-2, 1-3, 1-4, 1-5, 1-6, 1-8)
- Decoders (1-2, 2-4)
- Polarity Hold L1/L2 LSSD latches
- Clock Splitters
- Off Chip Driver / Off Chip Receiver

The number of inputs for the primitive logic functions, such as NOR, NAND, AND and OR was limited to four. Both inverting and non-inverting functions were created to take advantage of the power/performance and area of the library element. The circuits do not support both polarities concurrently.

All the circuits were designed to satisfy a 1ns output transition time, as measured from the 10 to 90% points, at rated maximum capacitance when the applied input transition rate was between 0.1ns and 1ns. The fast transition time requirements were imposed to minimize the power associated with the time that both the pull-up and pull-down devices were active. This transition time is also consistent with the 0.5 micron technology being employed.

The higher complexity random logic circuits were synthesized. This eliminated the guesswork of how best to realize the higher input count circuits for optimal power, performance and area. The number of power/performance codes presented a logistics problem for the synthesis engine and, as such, a three-phase synthesis approach was employed. The first stage presented the synthesis engine with a subset of the library wherein the power/performance increments were relatively large, with a maximum of 5 options for each function. The second stage used a post-synthesis optimization wherein the entire library was represented. After place and route, an additional in-place optimization was employed, wherein circuits were replaced with different drive strengths to better match the needed minimal power/performance requirements.

Since the library was very granular in power/performance, accurate prediction of capacitance loading was imperative. The chip was implemented using hierarchical logic, synthesis, and place and route partitions. To account for the different type of domains and requirements, each partition could use a different wire load model for synthesis. A domain wherein the organization was composed of primarily random logic could be treated differently than a data flow stack. Post route wire capacitance was fed back to the synthesis engine for in-place optimization. The circuits were physically designed so that many power codes of a given function would occupy the same physical space with only wiring pin target differences. The synthesis engine was informed of eligible replacements for the in-place optimization. If capacitance violations remained after these optimizations, then engineering evaluations were made as to how to proceed. On occasion, manual manipulation of the layout was needed to effect the desired results.

In addition to the synthesis engine's limited ability to choose from the broad spectrum of power/performance circuits available in the library, the inability to directly specify the power consumption of the circuit and request a power minimization synthesis left much to be desired. This was especially frustrating because the typical metrics used in the minimization process are performance and area. Since many like function circuits of differing power/performance codes occupied the same area, the tool would choose the higher power circuit. To this end, we heuristically adjusted the area metric to steer the tool to a more reasonable choice. This is an area where much improvement is needed for power optimized synthesis.

Compiled Entities

All circuits were designed with the same physical form factor (pitch). This allowed combinatorial logic to be placed and routed with more complex functions. Registers were compiled from the base latches and clock splitters. The compiled register allowed variations in bit width, polarity and power code on a per-bit basis, and variations in clock gating and scan direction and definition. Similarly, selectors were compiled from the base select circuits and decoders. Power code definition was specifiable on a per-bit basis. The power code of the decoder, like the clock splitter, was a function of the bit width. Defaults could be overridden by the designer to effect improved performance, but there were minimum defaults that could not be overridden. Five power/performance codes were used for the clock splitters and four for the decoders. Great care had to be exercised in the development of the clock splitters to ensure that under the broad range of register bit widths, clock overlap was minimized without sacrificing performance.

The compiled entities were instantiated in the schematic with a "don't touch" property, thereby disallowing the synthesis engine from modifying them. To this end, all compiled entities were initially realized as having minimum power code components unless the designer required otherwise. The synthesis engine was deceived to think that the compiled entities had the highest power code to minimize the addition of unnecessary re-powering buffers by the synthesis engine. After synthesis and place and route, the actual power codes were revealed, and timing analysis determined which bits of the compiled entities required change. The register and selector bits were then changed to the appropriate power code. Since the register and selector bits were all interchangeable from a power viewpoint, wiring modifications weren't necessary. This practice ensured that the minimum power code on a per-bit basis was employed.

The registers also permitted polarity changes without area increases, but did result in wiring perturbation. This was especially helpful when the synthesis engine needed higher drive strength than the highest power code register bit could support. In this case, high power buffers which were added to the register output would be replaced by the same power inverters, and the phase of the latch could be changed. While this did require some routing modifications, it saved power and enhanced performance. Area could be saved if the region were compressed and rerouted.

Parasitic Capacitance Minimization

IBM's CMOS5L technology requires that each device gate be connected to a diffusion at completion of M1 processing to avoid

potential threshold voltage shifts and reduced hot-electron reliability due to subsequent process steps. This "floating gate" requirement can be most easily satisfied by employing a floating gate contact structure on all gate inputs. While this approach has the benefit of ensuring all gates are protected without elaborate checking algorithms, it is the least power efficient approach. For example, each floating gate contact structure has a 2fF capacitance, and while this may seem insignificant it represents 40% of the overall input capacitance of the A-power code inverter. Furthermore, in the design of a low power library, even the field polysilicon connections need to be minimized.

Protecting each gate requires positioning the floating gate contact in a given wiring channel and connecting the field poly to it. Since many of the devices are extremely small (less than 3um), there would be considerable field poly between the gate and the outermost wiring track in the cell.

We chose to implement an algorithmic approach to find all floating gates and correct them automatically.

The approach was approximately 99% effective. In less than 1% of the cases, manual intervention was required. Since the registers and selectors were compiled, all necessary intra-macro connections requiring floating-gate protection were integrated in the base cells during design. In addition, the extensive use of NFET transfer gates for the selector and latch inputs helped minimize the number of floating gate contacts. Transfer gates were used for the obvious reasons of power, performance and area.[3] The floating gate protection benefit, while secondary, was quite welcome.

In this regard the use of algorithmic application of floating gates not only minimizes the capacitance of the floating gate contact structure, but it also reduces the field polysilicon interconnect.

Field polysilicon connects were also minimized by algorithmically creating the needed polysilicon straps between the wiring target location and the actual gate. In many libraries, the polysilicon extends from each gate into the wiring tracks to provide ample wiring targets. This adds field polysilicon and hence, input capacitance to all circuits employing it. Pseudo-polysilicon paths or flight lines were incorporated into each circuit from the wiring field pin to the respective gate. This allowed for contour routing to avoid blockages and minimize area. The appropriate flight line was then algorithmically converted to field polysilicon based on global wiring target personalization. This approach saved at least 50% of all field polysilicon. Figure 2 illustrates this concept. A similar approach was employed on diffusions associated with transfer gate source inputs and output circuit source and drains.

Local Clock Splitting and Generation

Figure 3 shows that the on-chip clock distribution network took the form of a pitchfork with the source being located at the handle. No on-chip receiver or redrive buffer was employed other than the distributed local clock splitters located along the tines of the pitchfork. While this results in increased module pin capacitance (22pF) the power saved in receiving and repowering the clock tree is estimated at 10mW. The tines were all routed in M3 to minimize capacitance. To handle the current density requirements, the widths of the lines were increased but the spacing to adjacent lines was also increased to reduce coupling and perimeter capacitance.

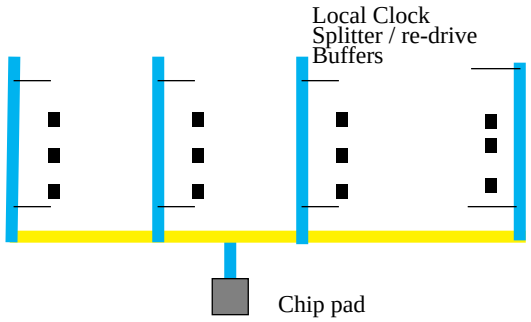

Figure 3: Clock Distribution Network

Dynamic On-chip Clock Activation

Dynamic on-chip clock activation was used extensively to minimize power. Each compiled register had an integrated clock overhead cell which served as a 2-phase clock splitter, re-drive buffer and power management control. The power management was in the form of a pair of control signals which, when deactivated, disabled the clock splitters and the redrive buffers from switching. This scheme allowed independent control of the C1 (L1 latch) and the C2 (L2 latch) clocks. This independent control allowed the designer the ultimate flexibility in managing power and performance. For cases where performance was of critical importance, the C1 clock gate would be enabled. This was necessary only for critical half-cycle paths. Very rarely would the C2 clock gate be bypassed. This clock gating scheme was employed throughout the chip design.

The gating of clocks not only affords power reduction in the clock splitter, but also in the latches themselves. This is evident in the case of L2 latches which are disabled for a given cycle and the content of the L1 latch is a "don't care". Loading the L1 latch on this cycle not only causes needless power dissipation in the clock splitters, but also in the L1 latches. Furthermore, L1 latches with outputs cause needless power dissipation in the circuits they drive when unnecessarily perturbed.
If a C2 clock were allowed to free-run, then for each cycle when the latch content was a don't care, all the downstream logic would

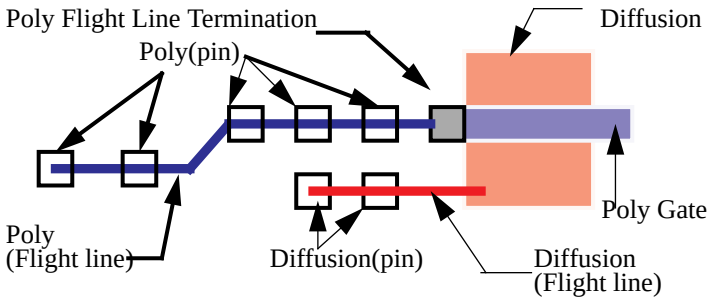

Figure 2: Poly & Diffusion Flight line Illustration

be unneccessarily perturbed. Since an LSSD [4] design has the bulk of its logic between L2 and L1 latches, this is an area where power management is concentrated.

Operand Isolation

Operand isolation is another technique used in the 403GA for power conservation. Operand isolation, as the name implies, isolates sections of circuitry from "seeing" changes on their inputs unless they are expected to respond to them. Figure 4 illustrates operand isolation for a simple ALU. In an ALU, all the execution units (logical units, arithmetic units, shifters and rotators) share (a) common operand bus(es). If operand isolation is not used, all execution units execute concurrently and a single result is selected and propagated. Meanwhile, power has been needlessly expended in those non-selected units. By employing operand isolation, operands are steered only those execution units which are going to be active in a given cycle. This may not be critical in the primitive logical units, but it saves power in the more robust functions. For the adders, it is important not to forget to control the carry-in signal since the entire state of the adders can be affected by a change in carry-in. The control signals for the operand isolation can be the same as those used to select the result in the most simple case. Power distribution area can also be minimized, as only selective functions will be activated.

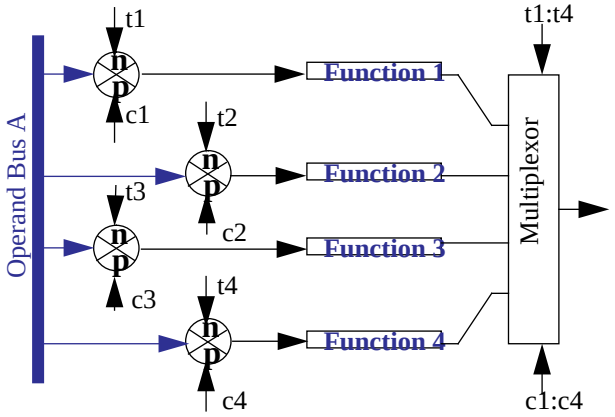

Figure 4: Operand Isolation

Gated and Clocked Receivers and Common I/O

Additional power savings were achieved by both clocking and gating the 403GA's synchronous inputs and common I/O receivers.[5] In the case of common I/O, the receivers are only active when the system clock is low and when the gate enable is active. In the most simplistic realization, the receiver gate enable is the inverse of the driver enable. This disables the receiver when the driver is active and eliminates any needless power due to the receiver and any downstream circuit switching.

Extensive use of NFET transfer gates

NFET transfer gates were employed extensively in the realization of latches, selectors, exclusive-ORs, receivers, and adders. The use of NFET-only transfer gates in the selectors and latches allowed for the use of single phase clocks and controls. This saved both power and area. In the case of latches, the transfer gates were used for clock and scan activation, and transfer between L1 and L2.[6] The selectors employed a PFET pull-up whose gate was controlled by the output of the first inverter of the selector to provide up-level voltage adjustment. In fact, a PFET stack was employed wherein one of the series PFET's gate was grounded. This was employed to more tightly control the feedback current over broad process ranges. The channel length of both devices was significantly larger than minimum to minimize delta L effects, and since two devices were employed, the channel width variances could also be minimized. Trade-offs were made for how much additional internodal capacitance was added versus control of feedback current.

The use of transfer gates certainly presented some challenges in input capacitance representation. For the purposes of timing analyses only the input capacitance when the gate was activated was presented. A second capacitance value was generated for the case when the gate was deactivated, but this value was used to modify the net capacitance files manually when knowledge of the gate activation was known. This is especially important when multiple selector data inputs are being driven from a single source and only a partial set of selects are active. The difference in input capacitance between an active and inactive gate is approximately one order of magnitude. Hence, by applying knowledge of the gate activation, the capacitance file could be reduced for the net and a lower power circuit could be used to drive it. As mentioned earlier, the use of transfer gates had the added benefit of easing the floating gate incorporation.

Results

The PowerPC 403GA has a typical power dissipation of 200mW at 25 mHz. This was determined by running a "typical" instruction mix comprised of the following: 33% branch, 25% load, 17% compare, 17% arithmetic and 8% store operations. The code assumes a 90% Instruction Cache hit rate. The worst case power assumes maximum I/O switching activity and heavy use of the Caches. The wait state power is 30 mW with the system clock running.

The average internal net capacitance is approximately 90 fF. The distribution of power-performance codes is heavily biased to the low end of the offering, with the average being D-power. Figure 5 is a histogram of all random logic gate drive strengths employed on the PowerPC 403GA.

Summary

In summary, techniques have been described wherein a robust granular power/performance cell library has been developed and used in conjunction with algorithmic parasitic capacitance minimization and other power management techniques to produce a power efficient embedded controller.

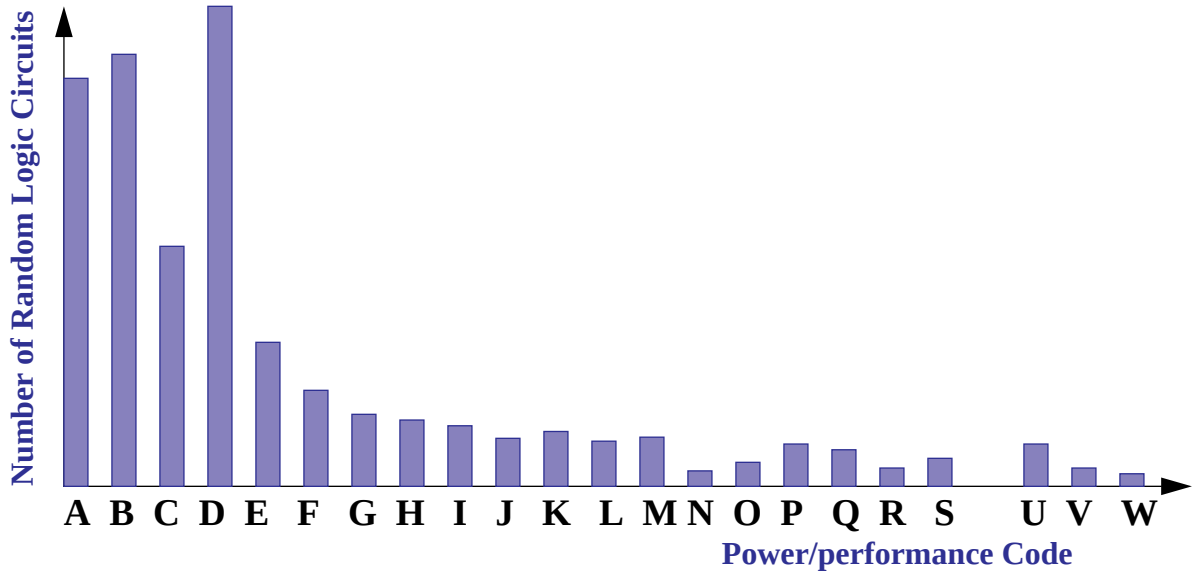

Figure 5: Histogram of Random Logic Circuit Power/Performance Codes

Acknowledgments

I would like to thank the entire PowerPC403 development team, based in Research Triangle Park, N.C., for all their efforts in developing a first pass successful design and their conscious efforts to manage power. I also thank John McKeeman, David Artz, Charlie Hoffman, Dave Weitzel, Donna Correale, Chuck Freeman and Steve Harris for their contributions in the preparation of this manuscript. Finally, many thanks to Alan Botula for his CMOS5L technology knowledge.

References

1. IBM PowerPC 403GA 32-Bit RISC Embedded Controller data sheet Document No.MPR403DSU-01 09.02.94
2. CMOS5L Design Manual, IBM Engineering Specification 02G1508
3. A.P.Chandrakasan,S.Sheng and R.W.Brodersen, "Low-Power CMOS Design," in IEEE Journal of Solid State Circuits, Vol. 27, NO. 4, April 1992, pg.475
4. E.B.Eichelberger and T.W.Williams, "A logic design structure for LSI testability," in Proc. 14th Design Automation Conf., June 1977, pp. 462-468
5. Anthony Correale, Jr., "Power Reduction Technique for Chips Using Common I/O'", in IBM Technical Disclosure Bulletin, Vol. 37, No. 01, January 1994
6. Anthony Correale, Jr., "Design considerations of a static LSSD polarity hold latch pair", in IBM Journal of Research and Development, Vol.28, No. 4, July 1984, pp.370-377