A 160-MHz, 32-b, 0.5-W CMOS RISC Microprocessor

James Montanaro
Richard T. Witek
Krishna Anne
Andrew J. Black
Elizabeth M. Cooper
Daniel W. Dobberpuhl
Paul M. Donahue
Jim Eno
Gregory W. Hoeppner
David Kruckemyer
Thomas H. Lee
Peter C. M. Lin
Liam Madden
Daniel Murray
Mark H. Pearce
Sribalan Santhanam
Kathryn J. Snyder
Ray Stephany
Stephen C. Thierauf

This paper describes a 160 MHz 500 mW StrongARM microprocessor designed for low-power, low-cost applications. The chip implements the ARM V4 instruction set[1] and is bus compatible with earlier implementations. The pin interface runs at 3.3 V but the internal power supplies can vary from 1.5 to 2.2 V, providing various options to balance performance and power dissipation. At 160 MHz internal clock speed with a nominal Vdd of 1.65 V, it delivers 185 Dhrystone 2.1 MIPS while dissipating less than 450 mW. The range of operating points runs from 100 MHz at 1.65 V dissipating less than 300 mW to 200 MHz at 2.0 V for less than 900 mW. An on-chip PLL provides the internal clock based on a 3.68 MHz clock input. The chip contains 2.5 million transistors, 90% of which are in the two 16 kB caches. It is fabricated in a 0.35-μm three-metal CMOS process with 0.35 V thresholds and 0.25 μm effective channel lengths. The chip measures 7.8 mm × 6.4 mm and is packaged in a 144-pin plastic thin quad flat pack (TQFP) package.

© 1996 IEEE. Reprinted, with permission, from *IEEE Journal of Solid-State Circuits*, volume 31, number 11, November 1996, pages 1703–1714.

Introduction

As personal digital assistants (PDA's) move into the next generation, there is an obvious need for additional processing power to enable new applications and improve existing ones. While enhanced functionality such as improved handwriting recognition, voice recognition, and speech synthesis are desirable, the size and weight limitations of PDA's require that microprocessors deliver this performance without consuming additional power. The microprocessor described in this paper—the Digital Equipment Corporation SA-110, the first microprocessor in the StrongARM family—directly addresses this need by delivering 185 Dhrystone 2.1 MIPS while dissipating less than 450 mW. This represents a significantly higher performance than is currently available at this power level.

CMOS Process Technology

The chip is fabricated in a 0.35 μm three-metal CMOS process with 0.35 V thresholds and 0.25 μm effective channel lengths. Process characteristics are shown in Table 1. The process is the result of several generations of development efforts directed toward high-performance microprocessors. It is identical to the one used in Digital Equipment Corporation's current generation of Alpha chips[2] except for the removal of the fourth layer of metal and the addition of a final nitride passivation required for plastic packaging.

The factors which drive process development for low-power design are similar to those which drive the process for pure high-performance although the motivation sometimes differs. For example, while both types of designs benefit from maximizing Idsat of the transistors at the lowest acceptable Vdd, the motivation for a pure high-performance design is reducing power distribution and thermal problems rather than extending battery life. Similar arguments apply to minimizing transistor leakage and on-chip variation of transistor parameters. This convergence of goals has been essential to our ability to develop one process to satisfy the requirements of both low-power and high-performance families.

Table 1
Process Features

Feature size	0.35 µm
Channel length	0.25 µm
Gate oxide	6.0 nm
Vtn/Vtp	0.35 V/-0.35 V
Power supply	2.0 V (nominal)
Substrate	P-epi with n-well
Salicide	Cobalt-disilicide in diffusions and gates
Metal 1	0.7 µm AlCu, 1.225 µm pitch (contacted)
Metal 2	0.7 µm AlCu, 1.225 µm pitch (contacted)
Metal 3	1.4 µm AlCu, 2.8 µm pitch (contacted)
RAM cell	6 transistor, 25.5 µm^2

Power Dissipation Tradeoffs

RISC microprocessors operating at 160 MHz are fairly common using current CMOS process technology. The novel aspect of this design is the ability to achieve this operating frequency at power levels which are low enough for handheld applications. Several design tradeoffs were made to achieve the desired power dissipation. In order to illustrate their effect on the design, it is interesting to imagine applying these tradeoffs to an earlier design whose power dissipation occupies the opposite end of the power spectrum, the first reported Alpha microprocessor.[3] This Alpha chip was fabricated in a 0.75-µm CMOS process and operated at 200 MHz dissipating 26 W at 3.45 V. The impact of these tradeoffs is summarized in Table 2.

The first decision is to reduce the internal power supply to 1.5 V. This change cuts the power by a factor of 5.3. While this has the desired effect, it has implications for the cycle time which are considered in the section Circuit Implementation.

The next step is to reduce the functionality. As compared to the early Alpha chip, the most obvious sections missing in this design are the floating point unit and the branch history table. Floating point is not required in the target applications and the low branch latency of this design eliminates the need for the branch history table. Less obvious, but very important, is reduced control complexity. This is a simple machine and we have worked hard to keep it so. We estimated that the reduced functionality would cut power by a factor of three.

Table 2
Power Dissipation Tradeoffs

Start with Alpha 21064: 200 MHz @ 3.45 V. Power dissipation = 26W			
Vdd reduction:	Power reduction =	5.3x	⩾ 4.9 W
Reduce functions:	Power reduction =	3x	⩾ 1.6 W
Scale process:	Power reduction =	2x	⩾ 0.8 W
Reduce clock load:	Power reduction =	1.3x	⩾ 0.6 W
Reduce clock rate:	Power reduction =	1.25x	⩾ 0.5 W

Process scaling reduces node capacitances and therefore chip power. Note that although the area components of the capacitance will decrease as the square of the scale factor, the total capacitance change with scaling will be less dramatic primarily due to the effect of periphery capacitance. We estimate that scaling from 0.75 µm of the early Alpha chip to our current 0.35 µm process results in a power reduction of about a factor of two, a linear reduction with scale factor. Once again, coupled with this positive effect of process scaling are a host of other issues. Some of those issues are considered in the section Power Down Modes.

Next, consider the clock power. The clock power of the Alpha chips is fairly large and while that clocking strategy works well for Alpha machines, it is not appropriate for a low-power chip. Our clocking strategy and our latch circuits are described in some detail later. One major change from the Alpha design was to reject the pair of transparent latches per cycle used on the Alpha design. Instead, on this design, we switched to a single edge-triggered latch per cycle to reduce clock load and latch delay. Our estimate is that the changes in the clocking reduced the clock power by a factor of two. Since the clock power was about 65% of the total power on the first Alpha chip, this results in a reduction of about 1.3.

Finally, the reduction in clock frequency from 200 MHz to 160 MHz drops the power by 1.25.

Clearly, this analysis is not rigorous, but it suggests that it is reasonable to build a 160 MHz processor chip that dissipates around half a watt. A similar analysis was performed at the beginning of the project to select the power supply voltage and operating frequency and to determine whether significant changes in design method would be required to meet the performance and power goals. It is interesting to note that with the exception of the clocking changes, the design methods and philosophy used on this design were very similar to that used on the Alpha chips.

Instruction Set

The microprocessor implements the ARM V4[1] instruction set. The architecture defines thirty 32-b general purpose registers and a program counter (PC). Registers are specified by a 4-b field where registers 0 to 14 are general purpose registers (GPR) and register 15 is the PC. The current processor status register contains a current mode field which selects either an unprivileged user mode or one of six privileged modes. The current mode selects which set of GPR's is visible.

In addition to basic RISC features of fixed length instructions and simple load/store architecture, the architecture implemented includes several features to improve code density. These include conditional execution of all instructions, load and store multiple instructions, auto-increment and auto-decrement for loads and stores, and a shift of one operand in every ALU operation. The architecture supports loads and stores of 8-, 16-, and 32-b data values. In addition to the standard 32-b computations, there is a 32-b × 32-b multiply accumulate with a 64-b product and accumulator.

Chip Microarchitecture

As shown in Figure 1, the chip is functionally partitioned into the following major sections: the instruction unit (IBOX), integer execution unit (EBOX), integer multiplier (MUL), memory management unit for data (DMMU), memory management unit for instructions (IMMU), write buffer (WB), bus interface unit (BIU), phase locked loop (PLL), and caches for data (Dcache) and instructions (Icache). To minimize pin power and support the high-speed internal core, one half of the chip area is devoted to the two 16 K caches. The pad ring occupies one-third of the chip area and the processor core fills the remaining one-sixth of the chip area.

The processor is a single issue design with a classic five-stage pipeline—Fetch, Issue, Execute, Buffer, and Register File Write (Figure 2). All arithmetic logic unit (ALU) results can be forwarded to the ALU input and there is a one-cycle bubble for dependent loads.

For example, the pipeline diagram in Figure 2 shows a SUBTRACT followed by a dependent LOAD. Note that at the end of cycle 3, we bypass the result from the SUBTRACT back into the ALU to compute the load address in cycle 4 without stalling the pipe.

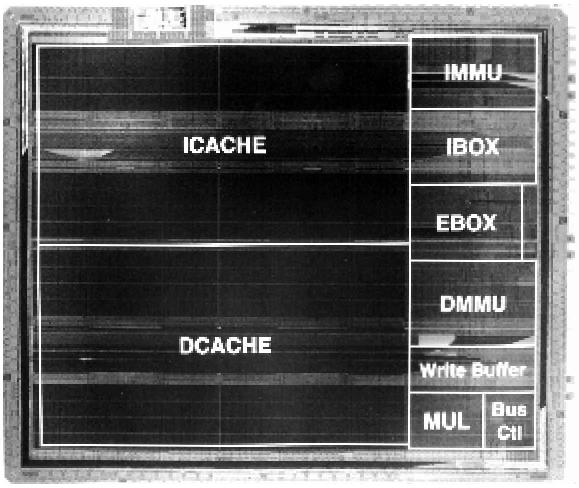
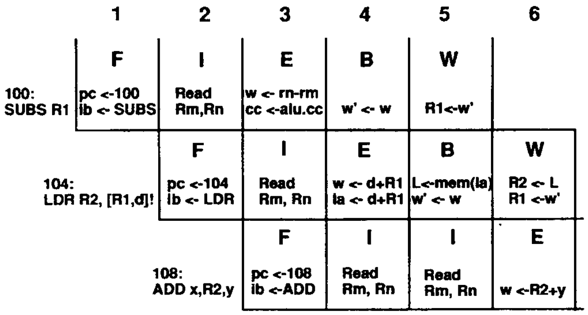


Figure 1
Chip Photo with Overlay



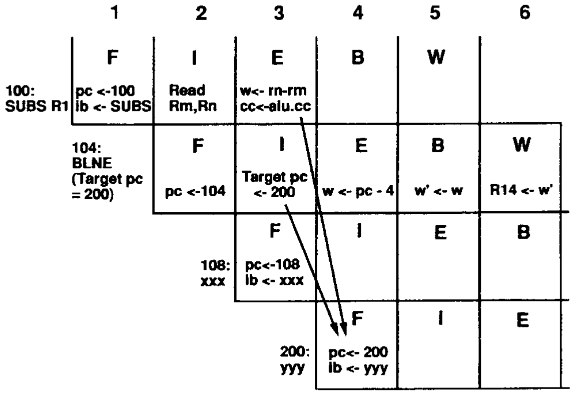
	1	2	3	4	5	6
	F	I	E	B	W	
100: SUBS R1	pc <-100 ib <- SUBS	Read Rm,Rn	w <- rn-rm cc <-alu.cc	w' <- w	R1<-w'	
		F	I	E	B	W
104: LDR R2, [R1,d]!		pc <-104 ib <- LDR	Read Rm, Rn	w <- d+R1 la <- d+R1	L<-mem(la) w' <- w	R2 <- L R1 <-w'
			F	I	I	E
108: ADD x,R2,y			pc <-108 ib <-ADD	Read Rm, Rn	Read Rm, Rn	w <-R2+y

Figure 2
Basic Pipeline Diagram

The third instruction is an ADD which uses the result of the previous LOAD. The ADD is held in the Issue stage for one additional cycle until the LOAD data is available at the end of cycle 5.

The IBOX can resolve conditional branches in the Issue stage even when the condition codes are being updated in the current Execute cycle. By providing this optimized path, the IBOX incurs only a one-cycle penalty for branches taken, so the chip does not require branch prediction hardware. For example, in the pair of instructions shown in Figure 3, the BRANCH and LINK instruction at the (program counter) PC of 104 depends on the condition codes which are being generated by the SUBTRACT in the previous instruction. The condition codes from the Execute stage of the SUBTRACT are available at the end of cycle 3, in time to swing the PC multiplexer in the IBOX to point at the branch target PC during the next Fetch cycle.

The optimization of the branch path represents a power versus performance tradeoff in which perfor-



	1	2	3	4	5	6
	F	I	E	B	W	
100: SUBS R1	pc <-100 ib <- SUBS	Read Rm,Rn	w<- rn-rm cc<-alu.cc			
104: BLNE (Target pc = 200)		F pc <-104	I Target pc <- 200	E w <- pc - 4	B w' <- w	W R14 <- w'
			F	I	E	B
108: xxx			pc<-108 ib <- xxx			
				F	I	E
200: yyy				pc<- 200 ib <- yyy		

Figure 3
Pipeline Diagram of a Branch

mance won. In our effort to hold the one cycle branch penalty, we included a dedicated adder in the IBOX to calculate the branch target address and consumed additional power in the EBOX adder to meet the critical speed path to control the PC multiplexer. Due to critical path constraints, the adder in the IBOX must run every cycle, even if the instruction is not a branch.

In the early stage of the design, one of our concerns was whether the decision to pursue this optimized branch path would increase our cycle time. As the design turned out, our best efforts in this ALU path and in the cache access path resulted in nearly identical delays for these two longest critical speed paths.

Data for integer operations comes from a 31-entry register file with three read and two write ports. Sixteen of the registers are visible at any time with 15 additional shadow registers specified by the architecture to minimize the overhead associated with initiating exceptions. The EBOX contains an ALU with a full 32-b bidirectional shifter on one of the input operands. It includes bypassing circuitry to forward the data from the Dcache or the ALU output to any of the read ports. Figure 4 shows the circuit blocks involved in the branch path. The path starts at a latch in the bypassers and, in a single cycle, includes a 0-to 32-b shift, a 32-b ALU operation, and a condition code computation to swing the PC multiplexer for the next cycle. The registers to hold the condition codes were implemented in the EBOX so that this path could be locally optimized. Analysis of code traces indicated that most ALU operations included a shift of zero, so for this case, the shifter is disabled and bypassed to reduce power.

The EBOX also contains a 32-b multiply/accumulate unit. The multiplier consists of a 12- by 32-b carry-save multiplier array which is used for one to three cycles depending on the value of multiplicand and a 32-b final adder to reduce the carry-save result. For multiply accumulate operations, the accumulate value is inserted into the array so that an additional cycle is not required for the Multiplies with Accumulate. Multiply Long instructions require one additional cycle. This results in a MULTIPLY or MULTIPLY/ACCUMULATE in two to four cycles and MUL LONG or MUL LONG/ACCUMULATE in three to five cycles.

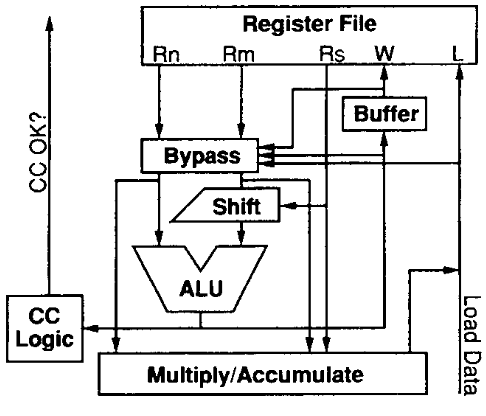


Figure 4
EBOX Block Diagram

The Wallace tree implementation was chosen to minimize the delay through the array. This implementation required careful floor planning and custom layout to keep the wiring under control. The decision to perform 12 b of multiply per cycle was based on wiring tradeoffs made during the physical planning phase of the design rather than critical path concerns. When the multiplier is not in use, all clocks to the section stop and the input operands do not toggle.

The chip features separate 16 kByte, 32-way set associative virtual caches for instructions and data. Each cache is implemented as 16 fully associative blocks. Each cache is accessed in a single cycle for both reads and writes, providing a two-cycle latency for return data to the register file. One eighth of each cache is enabled for a cache access.

The Dcache is writeback with no write allocation. The block size is 32 bytes with dirty bits provided for each half block to minimize the data which needs to be castout in the event of a dirty victim. The physical address is stored with the data to avoid address translation during castouts.

Given the size of the caches and the low power target for the chip, it was important that we have fine granularity of bank selection. In addition, we required associativity of at least four-way for cache efficiency and it was important to performance that we maintain a single cycle access. We considered several solutions to this problem, including traditional four-way set associative caches, and decided that the simplest approach which satisfied all the requirements was to implement the caches as smaller, bank-addressed, fully associative caches. This resulted in 32-way associativity but this level of associativity was a side effect of the implementation used, not the result of a goal to get associativity significantly above four-way.

The chip includes separate memory management units (MMU) for instructions and data. Each MMU contains a 32-entry fully associative translation lookaside buffer (TLB) with entries which can map either 4 kB, 64 kB, or 1 MB pages. TLB fills are implemented in hardware. In addition to the standard memory management protection mechanisms, the ARM architecture defines an orthogonal memory protection scheme to allow the operating system easy access to large sections of memory without manipulating the page tables. This functionality requires a set of additional checks which must be performed after the TLB lookup. The resulting critical path was sufficiently long that we self-timed the RAM access in the TLB to allow us to perform the lookup and complex protection checks in a single cycle.

A write buffer with eight 16-byte entries handles stores and castouts from the Dcache. The write buffer includes a single-entry merge latch to pack up sequential stores to the same entry.

During normal operations, an external load request takes priority over stores on the pin bus. However, in the event of a load which hits in the write buffer, the chip executes a series of priority stores which raises the priority of the Write Buffer on the external bus above that of any loads. External stores occur and the write buffer empties until the store which was pending at the load address completes. At this point, top priority reverts back to loads.

Power Down Modes

There are two power down modes supported by the chip—Idle and Sleep.

Idle mode is intended for short periods of inactivity and is appropriate for situations in which rapid resumption of processing is required. In Idle mode, the on-chip PLL continues to run but the internal clock grid and the bus clock stop toggling. This eliminates most activity in the chip and the power dissipation drops from 450 mW to 20 mW. Return from Idle to normal mode is accomplished with essentially no delay by simply restarting the bus clock.

Sleep mode is designed for extended periods of inactivity which require the lowest power consumption. The current in Sleep mode is 50 μA which is achieved by turning off the internal power to the chip. The 3.3 V I/O circuitry remains powered and the chip is well behaved on the bus, maintaining specified levels if required by the drive enable inputs. Return from Sleep to normal operation takes approximately 140 μs.

As was noted earlier, a low voltage process is key to the design of a microprocessor which will run at 160 MHz while dissipating less than 450 mW. However, the same low device thresholds which allow the reduction of Vdd also result in significant device leakage. While this leakage is not large enough to cause a problem for normal operation, it does pose problems for standby current, especially if the process skews toward short channel devices. Our initial analysis indicated that the chip would dissipate over 100 mW in Idle mode with the clocks stopped. To reduce this leakage, we lengthened devices in the cache arrays, the pad drivers, and certain other areas. This brought the leakage power to within the required value of 20 mW in the fastest process corner. As a backup, we relaxed our design rules to allow the remaining gate regions, which are drawn with a standard 0.35 μm gate length, to be biased up algorithmically without violating design rules in case it was necessary to meet the leakage requirements.

The requirement for standby power in Sleep is more than two orders of magnitude lower than the Idle power. To meet the power limit in Sleep, we considered a variety of options including integrated power supply switches and substrate biasing schemes before choosing the simple approach of turning off the internal supply. This approach is reasonable for this generation of parts since they have a dedicated low voltage supply. As more parts of the system shift to the low voltage supply, this may no longer be acceptable. The conflicting requirements of high performance at low voltage and low standby current promise to create interesting challenges in future designs.

The power switch to turn off the internal power supply during Sleep is implemented off-chip as part of the power supply circuit for the low voltage supply. No state is stored internally during Sleep since in typical PDA systems, the Sleep state corresponds to the user turning the system off. Therefore the time associated with reloading the cache upon return from Sleep is acceptable.

The requirements in Idle and Sleep complicated the design of the bus interface circuits. This section includes the level-shifting interface between the internal low voltage (1.5 to 2.2 V) signals and the 3.3 V external pin bus. The bus interface circuits must drive and receive signals which are higher voltage than those nominally supported by the 0.35-μm process without using circuits which would cause us to exceed the current limit specified by the Idle spec. In addition, during Sleep the pads must be able to sustain the value on the output pins despite the loss of internal Vdd (Vddi) from the low voltage supply which is powered off by the system. The circuitry used to implement this function is shown in Figure 5.

Since Vddi will be driven to zero by the system during Sleep, it is used not only as a power supply but also as a logic signal. All circuitry which must be active in Sleep is driven from the external, 3.3 V supply (Vddx) which has been dropped through diode-connected PMOS devices to reduce the stress on the oxide of these devices. Before signaling the chip to enter Sleep, the system asserts the nRESET pin (active low) which drives all enabled outputs to a specified state—disabled for control signals and zero for addresses and data. It then asserts nPWRSLP (active low) which is ANDed with the appropriate output enable control to turn on small leaker devices which will hold the output pin in the appropriate state during Sleep. In the circuit shown in Figure 5, the output is an address. Therefore, the address bus enable (ABE) pin is the control pin on the lower NMOS leaker and a

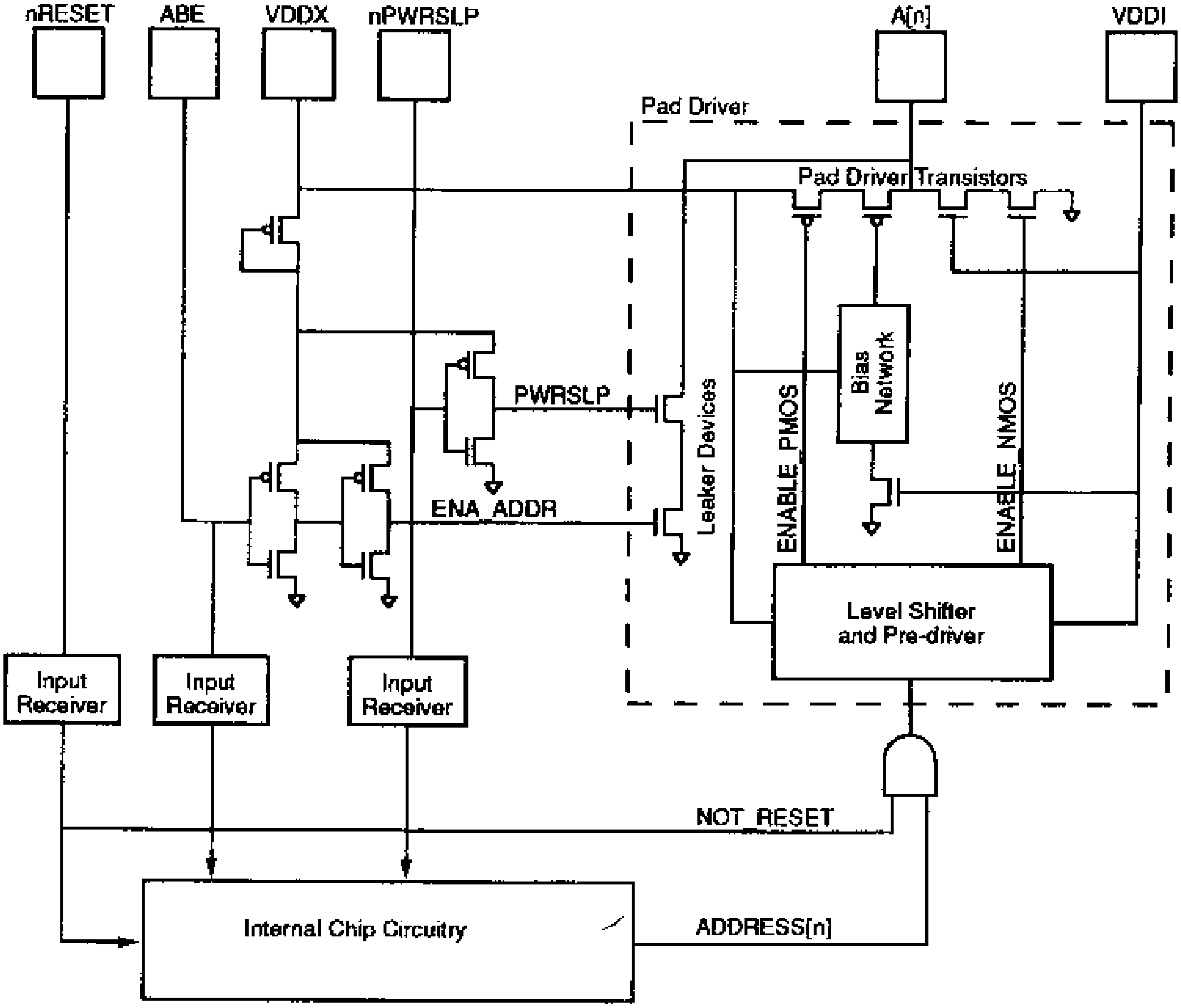


Figure 5
Pad Circuitry

buffered version of nPWRSLP controls the top device. Finally, the Vddi pins are actively driven to zero by the system. This action disables the output stage of the pad driver circuit by turning off the transistors closest to the pad—the NMOS directly and the PMOS via the bias network whose output goes to Vddx when its path to Vss is cut off. Note that for any input whose value is required during Sleep (ABE and nPWRSLP in the example described), a separate parallel input receiver must be implemented since the normal input receiver requires Vddi.

Circuit Implementation

The circuit implementation is pseudostatic and allows the internal clock to be stopped indefinitely in either state. Use of circuits which might limit low voltage operation was strictly controlled and the design was simulated to ensure operation significantly below the nominal 1.5 V level of the low voltage supply. The values of the internal supply and operating frequency were optimized to achieve maximum performance for less than half a watt.

The vast majority of the design is purely static, composed of either complementary CMOS gates or static differential logic. In certain situations, wide NOR functions were required and these were implemented in a pseudostatic fashion using either static weak feedback circuits or self-timed circuits to latch the output data and return the dynamic node to its precharged state.

The register file (RF) uses the self-timed approach to return the bit lines to the precharged state after an access (Figure 6). In this circuit, an extra self-timing column of bit cells with a dynamic bit line was implemented to mimic the timing of the data bit lines.

Figure 6 shows one cell from a column of register file data bit cells and one cell from the extra self-timing column (only one read port is shown). The bit cells in this extra column are all tied off so that the SELF_BITLINE signal will always discharge when the READ_WORDLINE goes high. When the SELF_BITLINE falls, it will set an RS latch causing the SELF_ENABLE signal to fall. This will disable the READ_WORDLINE and cause the bit lines to be precharged high when the read access is complete. Since the DATA_BITLINE's are received by low sensitive RS latches, the output data will be held when the bit line is precharged high. The self-timing RS latch is cleared when CLOCK_L goes low. This causes the SELF_ENABLE signal to go high, enabling the read port for the access in the next clock cycle. A separate SELF_BITLINE signal is implemented for each of the three register file ports so that the clocks for the three ports can be enabled independently.

The transistor leakage associated with the low threshold voltages is problematic for these pseudo-static circuits. If a weak feedback circuit is used in a NOR structure which is precharged high, excessive leakage in the parallel NMOS pulldowns would require that the feedback be fairly strong, which in turn would reduce the speed of the circuit. In the limit of very wide NOR's, it may not be possible to size a PMOS leaker so that it can supply the leakage of all the off NMOS pulldowns without making the leaker too large to be overpowered by a single active pulldown. In the case of a self-timed approach, a similar problem exists but it usually is manifested as a vanishingly small timing margin for the self-timed circuit to fire before the data on the dynamic node decays away. In either case, we addressed this issue by requiring the length of pulldowns on dynamic nodes to be slightly larger than minimum. Transistor leakage current is a strong function of channel length so a 12% increase in device length results in a leakage reduction in the worst case of about a factor of 20. The resulting leakage makes implementation of either weak feedback or a self-timed approach very reasonable.

The operating frequency at 1.5 V can be roughly derived by starting with the frequency of the Alpha

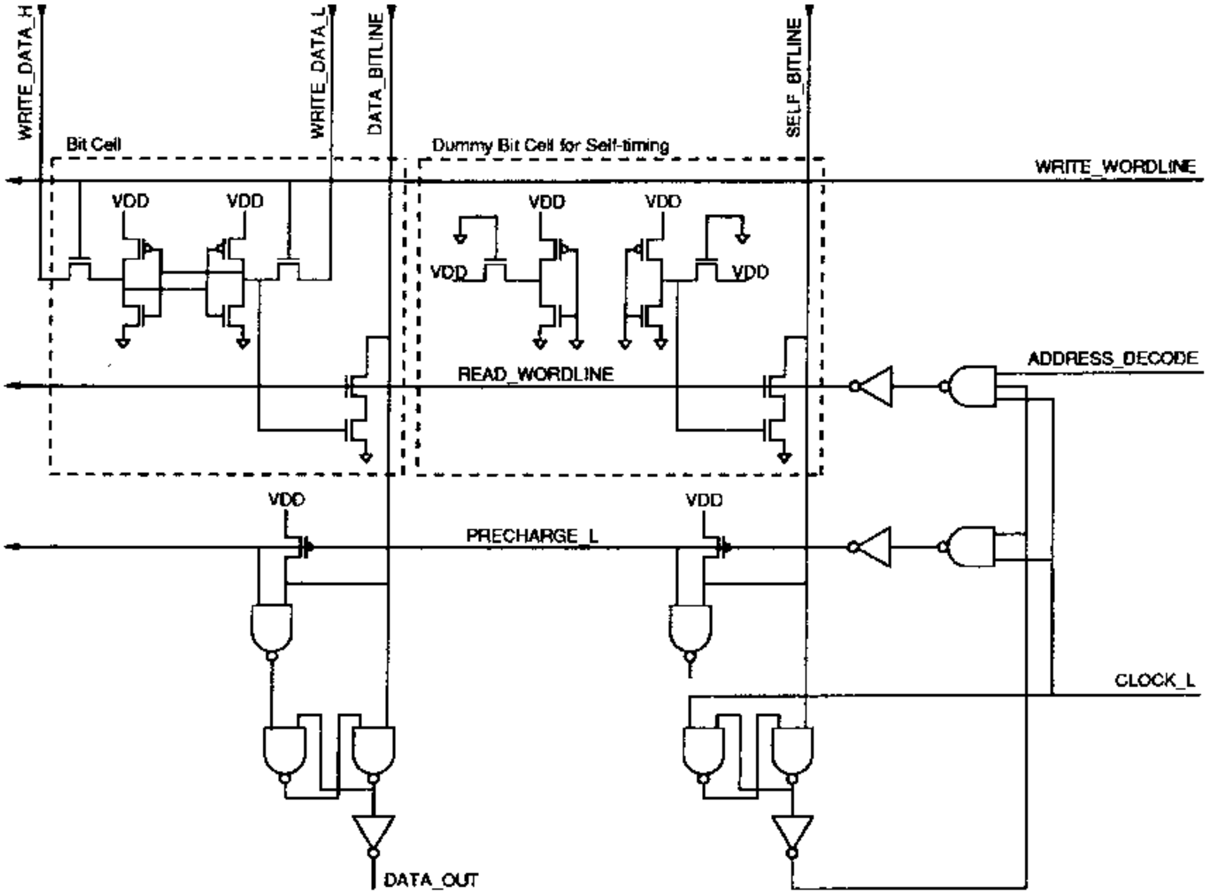


Figure 6
Self-timed RF Precharge

processor in the same process technology[2] and scaling for the use of a longer tick model and then Vdd. Since the long tick design requires the chip to perform a full SHIFT and a full ADD in a single cycle, this approximately doubles the cycle time required. The effect of Vdd scaling is roughly linear for this range of Vdd. Combining these effects results in an operating frequency at 1.5 V given by

$$433 \text{ MHz} * 0.5 * (1.5 \text{ V}/2.0 \text{ V}) = 162 \text{ MHz}.$$

This pair of voltage and frequency values agrees well with the power estimate outlined in the section Power Dissipation Tradeoffs. Note that for power supply voltages much lower than 1.5 V, the operating frequency decreases with voltage in a manner which is significantly stronger than linear. This fact sets a practical lower limit on the power supply voltage in most applications.

Power estimates made early in the design are prone to errors in either direction. In the case of this design, the power dissipated at 1.5 V was lower than the 450 mW target, so we shifted the nominal internal Vdd to 1.65 V to increase the yield in the 160 MHz bin.

Clock Generation

An on-chip PLL[4] generates the internal clock at one of 16 frequencies ranging from 88 to 287 MHz based on a fixed 3.68 MHz input clock. Due to internal resource constraints and our limited experience with low-power analog circuits, we contracted with Centre Suisse d'Electronique et de Microtechnique (CSEM) from Neuchâtel, Switzerland, to design the PLL and engaged Professor T. Lee from Stanford as a consultant on the project. Our initial feasibility work resulted in several design tradeoffs.

First, while there was a system requirement that the chip return quickly from the Idle state to normal operation, there was no such constraint on returning from the Sleep state. Based on this determination and our 20 mW power budget in Idle, we concluded that if we could keep the PLL power below 2 mW, we could leave the PLL running in Idle and remove the requirements on the PLL lock time. Thus, the need for a very low power PLL is dictated by the power budget in Idle, not in normal operation.

Next, we had specified a large frequency multiplication factor to allow the use of a common and cheap low frequency crystal clock source for consumer products. Early investigations indicated that this would make tight phase locking difficult. However, when we looked at target systems, we found no pressing need for phase locking. Consequently, we removed phase locking as a design criteria and concentrated our efforts and design tradeoffs on minimizing phase compression.

Finally, while the PLL was designed to handle the noise expected on the chip power supplies, we discovered toward the end of the design that the PLL was under its area budget and there was additional space available in the vicinity. We took advantage of this opportunity to provide cleaner power to the PLL by RC filtering our internal supply and we dedicated 1 nF of on-chip decoupling cap to this purpose.

CSEM performed the circuit and layout design and we placed the completed block into the microprocessor. Since we anticipated that the characterization of the PLL integrated in the chip would present some difficulties, we reserved one of the six die sites on our first pass reticle set for a test chip which contained several variants of the full PLL and interesting sub-blocks. These circuits allowed access to a variety of nodes in the PLL without compromising the design of the PLL instantiated in the chip. The results of the PLL characterization are reported in Reference 4.

Clock Distribution

The chip operates from two clocks as shown in Figure 7. An internal clock, called DCLK, is usually generated by the PLL. The second clock is a bus clock, known as MCLK which operates up to 66 MHz. MCLK can be supplied by an external asynchronous source or by the chip based on a division of the PLL clock signal.

There are five clock regimes in the chip. The first two regimes are sourced by MCLK and consist of the pad ring which receives MCLK directly and the bus interface unit (BIU) and part of the write buffer which receive MCLK through conditional clock buffers. The last three regimes are sourced by the internal DCLK clock tree and contain the Dcache, the Icache, and the

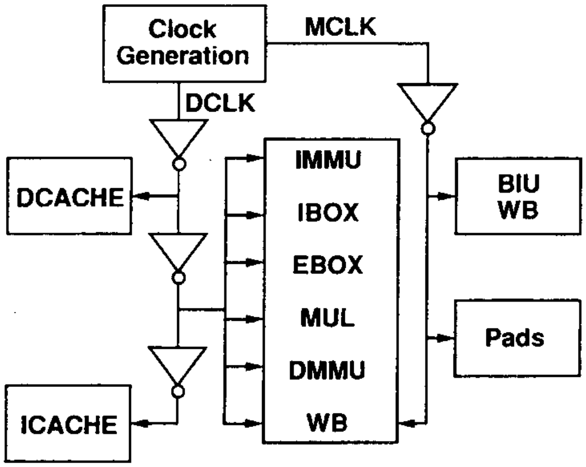


Figure 7
Clock Regimes

core. In this case, the core includes the IBOX, EBOX, MUL, IMMU, DMMU, and part of the write buffer.

Both MCLK and DCLK are distributed by buffered H-trees to conditional clock buffers in the various sections of the chip. The buffers in the H-tree allow the use of smaller lines for distribution and result in lower clock power. Although the three internal clock regimes are all sourced by the same H-tree, the topology of the chip did not allow corresponding sections of the H-tree to be routed in the same metal. This resulted in an increase in the expected skew between the caches and the core. In addition, we discovered that we could squeeze a bit more performance from the chip if we intentionally offset the clock in the caches relative to the clock in the core. Consequently, we used the clock buffers in the H-tree to tune the clock so that the Dcache receives a clock which is one gate delay earlier than the core and the Icache receives a clock which is one gate delay later than the core.

Figure 8 shows the physical routing of the internal clock tree. The buffer stages are not shown but they exist in the center of the chip and in four symmetric locations—two in the center of the I and D caches and two in locations at the cache/core interface. The final leg of the H-tree is tied to conditional clock buffers in the caches and the core. The problems associated with clock skew within the caches are reduced by the fact that only a single bank in each cache is enabled. This limits the physical distance over which tightly controlled clocks need to be delivered in the cache regions.

The clocks in the core present a more interesting problem. The final leg of the clock tree in the core stretches the full height of the chip and tight control of skew along this node is required for speed and functionality. It is implemented as a vertical, metal 2 line driven from four nominally equidistant points. The clock buffers are standard cells of varying drive strength built directly under this M2 line to minimize local variation in delay.

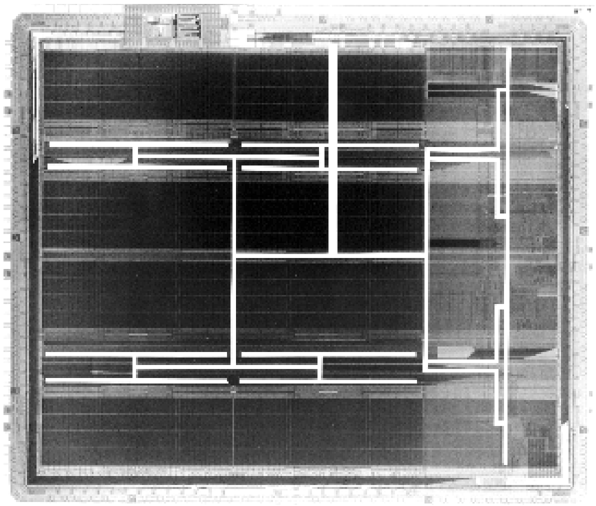


Figure 8
Physical Routing of Clock Tree

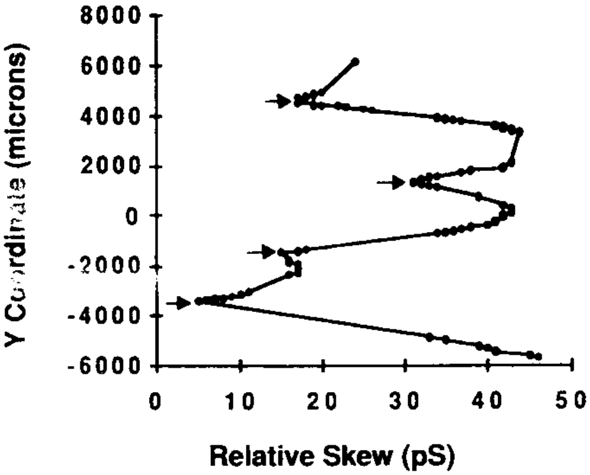


Figure 9
Clock Arrival Time in the Core

Circuit simulations of the H-tree were performed using SPICE to determine the skew between clock regions and within each of the clock regions. The nodes in the grid were extracted from layout and contained more than 30,000 R and C elements. Figure 9 shows the relative clock arrival time versus the Y coordinate for each conditional clock buffer on the vertical leg of the clock tree in the core. The four arrows on the graph indicate the points from which the final leg is driven. The data points are the relative arrival times of the clock input to the conditional clock buffers sourced by the clock tree. The total simulated skew is 41 pS assuming maximum metal resistance.

Clock Switching

One additional complication related to the internal clock tree is that it is not always driven by the clock from the PLL, known as CCLK. During cache fills, the clock source for the internal sections of the chip switches over to MCLK so that the whole chip is running synchronous to the bus (Figure 10). This simplifies fills and it reduces power since the bus clock is significantly slower than CCLK. Note that since this machine has a blocking cache, not much happens while waiting for a cache fill. Therefore, running on the slower bus clock during fills has essentially no performance impact.

Since MCLK and CCLK might be asynchronous, switching the driver of DCLK quickly between the two clock sources is difficult. Careful attention must be paid to the synchronization of the Mux control signals to prevent glitch pulses on the clock during the transition between the clock sources.

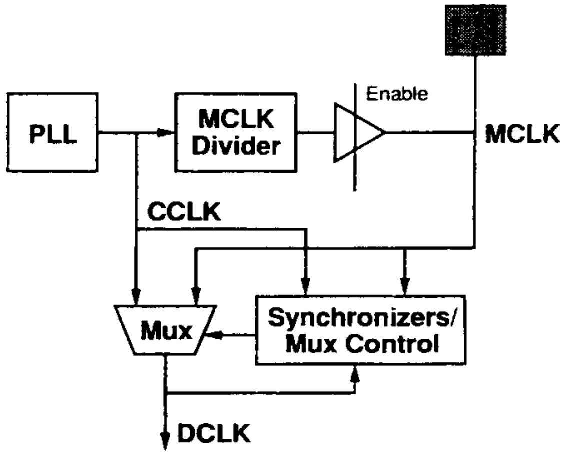


Figure 10
Clock Switching Circuit

Clock switching is only used during fills. Stores which miss in the cache and castouts are written to memory through the write buffer without switching the internal clock over to MCLK. The write buffer receives both DCLK and MCLK and passes the data for external stores across the DCLK/MCLK interface with proper attention to synchronization issues between the two clock regimes. One interesting characteristic of clock switching is that it gives the system designer another option to save power in situations for which the full performance of the chip is not required. By disabling clock switching on the fly, you can configure the chip to run off the bus clock. There is no limit on asymmetry or maximum pulse width of the bus clock, so the chip can be operated at very low frequencies if desired.

Conditional Clock Buffers

Conditional clock buffers are simple NAND/invert structures with an integral latch on the condition input. The buffers must be matched to their load to minimize skew. Since adding dummy clock loads is contrary to the low-power design philosophy, we created scaled clock buffers which would produce matched clocks for a wide range of loads and only needed to add dummy clock loads for a small number of very lightly loaded clock nodes. The task of matching the clock buffers to the load was greatly simplified by the fact the clock load presented by our standard latches is largely data-independent.

While the use of conditional clock buffers is central to the design method used on the chip, it should be noted that the critical paths to generate the condition input to these buffers represent some of the most difficult design problems in the chip. In this case, we decided that the power saving associated with the conditional clocking was worth the additional design effort and possible performance reduction.

Latch Circuits

The standard latches used in the design are differential edge-triggered latches (Figure 11). The circuit structure is a precharged differential sense amp followed by a pair of cross-coupled NAND gates. The sense amp need not be particularly well balanced because the inputs to the latch are full CMOS levels. The NMOS shorting device between nodes L3 and L4 provides a dc path to ground for leakage currents on nodes L1 and L2 in case the inputs to the latch switch after the latch evaluates. At normal operating frequencies, this device is not particularly important but it is required for the latch to be static. Note that since the dc current flowing is due only to device leakage, the magnitude of the current is insignificant to the power in normal operation.

Testability

The chip supports IEEE 1149.1 boundary scan for continuity testing. In addition, it has two hardware features to aid in manufacturing testing. The first is a bypass to allow CCLK to be driven from a pin synchronous to MCLK. This allows the tester to control the timing between CCLK and MCLK to make the asynchronous sections appear to be deterministic. The second test feature provides a linear feedback shift register (LFSR) that can be loaded with instruction data from the Icache. Loading the LFSR can be conditioned based on the value of address bit 2 and the Icache hit signal. The LFSR is loaded after the Fetch stage to allow the instruction following a branch to be read from the Icache and loaded into the LFSR. This feature allows any random pattern to be loaded into the

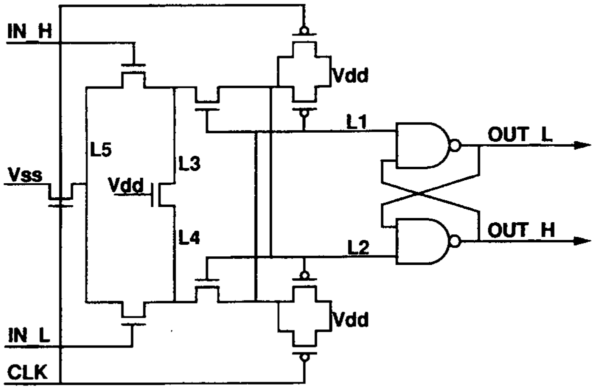


Figure 11
Latch Circuit

Icache and then read out by alternating branch instructions with data patterns words.

Power Dissipation Results

Measured Results

Power dissipation data was collected on an evaluation board running Dhrystone 2.1 with the bus clock running at one-third of the PLL clock frequency. Dhrystone fits entirely in the internal caches so, after the first pass through the loop, pin activity is limited. This is the highest power case because cache misses cause the internal clocks to run at the bus speed and result in a lower total power. For both sets of measurements, external Vdd is fixed at 3.3 V. For an internal Vdd of 1.5 V, the total power is 2.1 mW/MHz. If the internal supply is set to 2.0 V, the total power is 3.3 mW/MHz. Note that the ratio of the power at 1.5 and 2.0 V does not track Vdd^2 because it contains a component of external power and the external Vdd is fixed.

Simulated Power Dissipation by Section

An analysis of node transitions based on simulation was performed to estimate the power dissipation associated with the various major sections of the chip (Table 3). Toggle information was collected based on 160,000 cycles of Dhrystone and combined with extracted node capacitances to estimate power dissipation by node and this data was further grouped by section. The clock power listed in Table 3 is due only to the global clock circuits.

A few points are worth noting.

- First, the power is dominated by the caches as you might expect given their size. This is despite our efforts to reduce their power through bank selection and other means. The Icache burns more power than the Dcache because it runs every cycle.
- Next, the PLL power is insignificant in normal operation. As was noted earlier, its low power characteristics are only important in Idle.
- Finally, since reduction in clock power was one of our explicit goals, it is interesting to consider the total clock power. If you extract the local clock power from the nonclock sections and sum it, you get a total clock power, including the global clock trees, the local clock buffers and the local clock loads. This power is 25% of the total chip power, significantly less than the 65% consumed by the clocks in the Alpha microprocessor used in our initial feasibility studies.

Table 3
Simulated Power Dissipation by Section

ICACHE	27%
IBOX	18%
DCACHE	16%
CLOCK	10%
IMMU	9%
EBOX	8%
DMMU	8%
Write buffer	2%
Bus interface unit	2%
PLL	<1%

Conditional clocking was an integral part of the design method, so it is difficult to determine the power saving associated with it. However, the power associated with driving the conditional clocks is 15% of the chip power and if the conditions on all the conditional clock buffers were always true, this power would quadruple. This does not account for the additional power savings that has been achieved by blocking spurious data transitions.

CAD Tools

The CAD tools used on this chip were largely the same as those used on our Alpha designs.[5] This is not surprising since the performance target of the chip roughly parallels that of the Alpha family as noted in the section Circuit Implementation. The most significant departure was in the area of static timing verification and race analysis where the adoption of edge-triggered latching required significant modifications to the tools used in the Alpha designs.

Project Organization

One of the challenging aspects of this project was geographical. The detailed design was performed at four sites across a nine hour time zone range. The initial feasibility work and architectural definition was done at Digital Semiconductor's design center in Austin with on-site participation by personnel from Advanced RISC Machines Limited (ARM). The implementation was more widely distributed with the caches, MMU's, write buffer, and bus interface unit at Digital Semiconductor's design center in Palo Alto, the instruction unit, execution unit, and clocks in Austin, the pad driver and ESD protection circuits at Digital Semiconductor's main facility in Hudson, MA, and the PLL at the CSEM design center in Neuchâtel, Switzerland. In addition, we consulted with Hudson for CAD and process issues, with ARM in Cambridge, England, for all manner of architectural issues and implementation tradeoffs associated with ARM designs and with T. Lee from Stanford on the PLL. The implementation phase of the project took less than nine months with about 20 design engineers.

Conclusion

The microprocessor described uses traditional high performance custom circuit design, an intentionally simple architectural design, and advanced CMOS process technology to produce a 160 MHz microprocessor which dissipates less than 450 mW. The internal supplies can vary from 1.5 to 2.2 V while the pin interface runs at 3.3 V. The chip implements the ARM V4 instruction set and delivers 185 Dhrystone 2.1 MIPS at 160 MHz. The chip contains 2.5 million transistors and is fabricated in a 0.35-μm three-metal CMOS process. It measures 7.8 mm × 6.4 mm and is packaged in a 144-pin plastic thin quad flat pack (TQFP) package.

Acknowledgments

The authors would like to acknowledge the contributions of the following people:

F. Aires, M. Bazor, G. Cheney, K. Chui, M. Culbert, T. Daum, K. Fielding, J. Gee, J. Grodstein, L. Hall, J. Hancock, H. Horovitz, C. Houghton, L. Howarth, D. Jaggar, G. Joe, R. Kaye, J. Kapp, I. Kim, Y. Lou, S. Lum, D. Noorlag, L. O'Donnell, K. Patton, J. Reinschmidt, S. Roberts, A. Silveria, P. Skerry, D. Souyadalay, E. Supnet, L. Tran, D. Zoehrer, and the PLL design team at CSEM.

The support which they received on many aspects of the design from the people at Advanced RISC Machines, Ltd. was very important and keenly appreciated.

References

1. *ARM Architecture Reference* (Cambridge, England: Advanced RISC Machines, Ltd., 1995).

2. P. Gronowski et al., "A 433 MHz 64b Quad-Issue RISC Microprocessor," *ISSCC Digest of Technical Papers* (February, 1996): 222–223.

3. D. Dobberpuhl et al., "A 200 MHz 64b Dual-Issue CMOS Microprocessor," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 11 (1992).

4. V. von Kaenel et al., "A 320 MHz, 1.5 mW CMOS PLL for Microprocessor Clock Generation," *ISSCC Digest of Technical Papers* (February, 1996): 132–133.

5. T. Fox, "The Design of High-Performance Microprocessors at Digital," *31st ACM/IEEE Design Automation Conference*, San Diego, Calif. (June 1994): 586–591.

Biographies

James Montanaro
James Montanaro received the B.S.E.E. and M.S.E.E. degree from the Massachusetts Institute of Technology, Cambridge, MA, in 1980. He joined Digital Equipment Corporation in 1982 and worked as a circuit designer on several RISC microprocessor chips including the first two Alpha designs. In 1992, he joined Apple Computer as a circuit designer on the PowerPC 603 chip. In 1993, he returned to Digital, working in the Austin Research and Design Center on the design of the first StrongARM microprocessor chip.

Richard T. Witek
Rich Witek received a B.S. in computer science from Aurora College, Aurora, IL, in 1976. He is the lead architect on the StrongARM microprocessors at Digital's Austin design center. He was co-architect of the Digital Alpha architecture and lead architect on the first Alpha microprocessor. Rich was one of the lead designers on the MicroVAX II microprocessor, the first single chip VAX. At Digital, Rich also worked on Phase 2 and Phase 3 DECnet architecture and implementation along with other PDP11 and VAX software projects. Rich was part of the Apple PowerPC architecture team at Somerset in Austin. His current professional interests include processor architecture and implementations. Rich has numerous patents and technical publications on microprocessors and caches.

Krishna Anne
Krishna Anne received the B.E. degree in electronics engineering in 1991 from Andhra University, Vizag, India, and the M.S.E.E. degree from the University of Texas at Arlington in 1993. After a brief stay at Tensleep Design, Inc., Austin, TX, in 1994, he joined Austin Research and Design Center of Digital Equipment Corporation as a design engineer responsible for the full-custom design and development of high-performance low-power processors. He worked on the design and implementation of the multiplier on the StrongARM project and is currently working on another low-power chip.

Andrew J. Black
Andy Black received a B.S.E.E. from Pennsylvania State University and an M.S.E.E. from the University of Southern California. He joined Digital in 1992 after working for International Solar Electric Technology. He was a senior hardware engineer in Digital's Palo Alto Design Center, where he led the bus interface unit design for the StrongARM SA-110 microprocessor chip. During his work on the Alpha 21164 CPU, he was a member of the design team for the memory management unit and contributed to the chip's clock design. He is currently with Silicon Graphics Inc. as a member of the technical staff in the MIPS Technology Division where he is working on high-performance consumer-oriented products. Andy is a member of I.E.E.E., Tau Beta Pi, and Eta Kappa Nu.

Elizabeth M. Cooper
Elizabeth Cooper received the B.S. degrees (summa cum laude) in electrical engineering and computer science from Washington University in St. Louis in 1985. She received the M.S. degree in computer science from Stanford University in 1995. She joined Digital Equipment Corporation in 1985. Her previous responsibilities include design contributions to several CMOS VAX and Alpha CPUs. She was responsible for the design of the memory management unit on the SA-110 StrongARM chip. She is currently employed at Silicon Graphics MIPS Technology Division.

Daniel W. Dobberpuhl
Daniel Dobberpuhl received the B.S.E.E. degree from the University of Illinois in 1967. He joined Digital Equipment Corporation in 1976 and has been responsible for five generations of microprocessor designs including the initial Alpha CPUs. Most recently he has been the Technical Director of the Low Power Microprocessor Group with Digital's Palo Alto Design Center. He is the co-author of *The Design and Analysis of VLSI Circuits* (Addison-Wesley, 1985).

Paul M. Donahue
Paul Donahue received the B.S. degree in computer science from Cornell University, Ithaca, NY, in 1994. Upon graduation he joined Digital Semiconductor's Palo Alto Design Center and worked on the SA-110. He is currently working on the microarchitecture and verification of a StrongARM variant.

Jim Eno
Jim Eno received the B.S.E.E degree from North Carolina State University, Raleigh, in 1989. He is employed as a senior engineer at Digital Equipment Corporation's Austin Research and Design Center in Austin, TX, working most recently on the microarchitecture of the SA-110 StrongARM microprocessor. Before his employment with Digital, he was with the Somerset Design Center in Austin, working on the microarchitecture and design of the PowerPC 603 microprocessor. Previous to this, Jim was involved in ASIC design support and tool development at Compaq Computer Corporation. His research interests include low-power microprocessor design and the propagation of acoustic waves in various materials, enhanced by interaction with selected organic compounds.

Gregory W. Hoeppner
Gregory Hoeppner graduated with distinction from Purdue University, West Lafayette, IN, in 1979. In 1980 he worked at General Telephone and Electronics Research Laboratory, Waltham, MA, performing basic properties research on GaAs. From 1981 to 1992 he held a number of positions with Digital Equipment Corporation, Hudson, MA, including CMOS process development, device characterization and modeling, circuit design, chip implementation, and finally co-led the 21064 Alpha chip implementation team. In 1992 he joined IBM's Advanced Workstation Division before returning to Digital Equipment Corporation in 1993 to co-found their Austin Research and Design Center, Austin, TX. Here he contributed to the microarchitecture, implementation and verification of Digital's first StrongARM processor.

David Kruckemyer
David Kruckemyer received the B.S. degree in computer engineering from the University of Illinois at Urbana-Champaign in 1993 and received the M.S. degree from Stanford University in 1995. After graduation, he joined Digital Equipment Corporation's Palo Alto Design Center to work on the implementation of the Instruction Memory Management Unit for the SA-110, the first StrongARM microprocessor. He is currently involved in the microarchitecture and implementation of a next-generation StrongARM variant.

Thomas H. Lee
Thomas Lee received the S.B., S.M., and Sc.D. degrees in electrical engineering, all from the Massachusetts Institute of Technology, Cambridge, MA, in 1983, 1985, and 1990, respectively. He joined Analog Devices in Wilmington, MA, in 1990 where he was primarily engaged in the design of high-speed clock recovery devices. In 1992, he joined Rambus, Inc. in Mountain View, CA, where he developed high-speed analog circuitry for 500 megabyte/s DRAMs. Since 1994, he has been an Assistant Professor of Electrical Engineering at Stanford University where his research interests are in low-power, high-speed analog circuits and systems, with a focus on gigahertz-speed wireless integrated circuits built in conventional silicon technologies, particularly CMOS. He has twice received the "Outstanding Paper" award at the International Solid-State Circuits Conference.

Peter C. M. Lin
Peter Lin was born in Taichung, Taiwan, on March 17, 1960. He received the B.S.E.E. degree from Feng Chia University, Taichung, Taiwan, in 1982 and the M.E. and E.E. degrees from University of Utah, Salt Lake City, in 1987 and 1989, respectively. From 1990 to 1993 he designed 2M VRAM and 8M WRAM for Samsung Semiconductor, San Jose, CA. From 1994 to 1995 he worked for Digital Equipment Corporation, Palo Alto, CA, where he contributed to the design of low power Alpha and StrongARM microprocessors. He is currently working for C-Cube Microsystems, Milpitas, CA. He holds one patent in output buffer design.

Liam Madden
Liam Madden received the B.E. degree from University College, Dublin, Ireland, in 1979 and the M.E. degree from Cornell University, Ithaca, NY, in 1990. Over the past 15 years he has designed CMOS CISC and RISC microprocessors, including the 21064 Alpha processor. He led the design team in Palo Alto which delivered the caches, write buffer, memory management, and bus interface units for the SA-110 StrongARM microprocessor. He is currently employed at Silicon Graphics, Mountain View, CA, where he is Director of Circuit Design and Technology.

Daniel Murray
Daniel Murray received the B.S. degree in electrical engineering in 1994 from the University of California, Berkeley. In 1994, he joined Digital Semiconductor's low power microprocessor group in Palo Alto, CA. He contributed as a circuit designer on the first StrongARM CPU and is currently involved in the implementation of another high-performance, low-power microprocessor.

Mark H. Pearce
Mark Pearce was born in Geneva, Switzerland, on June 12, 1969. He received the B.S.E.E. degree from University of Pennsylvania, Philadelphia, in 1992, and the M.S.E.E. degree from Stanford University, Stanford, CA, in 1994. In 1994 he joined Digital Equipment Corporation, at their Palo Alto Design Center, working initially on a low power Alpha processor prototype. He designed the write buffer on SA-110, the StrongARM processor. He is currently working on another high-performance, low-power processor.

Sribalan Santhanam
Sribalan Santhanam received the M.S.E. degree in computer science and engineering from the University of Michigan, Ann Arbor, in 1989. He joined Digital Equipment Corporation, in Hudson, MA, where he worked on the design of the floating-point unit of the 21064 CPU and subsequently on the design of the cache control unit of the Alpha 21164 CPU. He then moved to Digital's Palo Alto Design Center where he was responsible for the design of the caches for the SA-110 StrongARM microprocessor. He is currently a principal hardware engineer working on the implementation of a follow-on StrongARM microprocessor.

Kathryn J. Snyder
Kathryn Snyder (formerly Hoover) received the B.S. and M.S. degrees from the University of Michigan, Ann Arbor, in 1990 and 1992, respectively. She is a circuit designer with Digital Equipment Corporation working on low-power microprocessor designs in Austin, TX. She designed a variety of custom circuits for the SA-110 StrongARM microprocessor. Prior to employment with Digital, she worked for IBM in Austin, doing custom array design for PowerPC microprocessors.

Ray Stephany
Ray Stephany received the B.S.E.E. from Rensellaer Polytechnic Institute, Troy, NY, and an M.B.A. from Worcester Polytechnic Institute, Worcester, MA. He joined Digital's Austin Research and Design Center in July, 1993. Since that time, he has been one of the project leads on the StrongARM line of microprocessors. He has contributed to the development of low power circuit design techniques, CAD tools, verification, and overall methodology. He is currently leading the implementation of a next-generation StrongARM CPU and looking at SOI as a potential lower power process for future generations of microprocessors.

Stephen C. Thierauf
Stephen Thierauf is a consulting hardware engineer at Digital Equipment Corporation's Digital Semiconductor Group, located in Hudson, MA, and is responsible for I/O circuit design, on- and off-chip signal integrity, and I/O modeling for Alpha microprocessors, PCI peripherals, and other ULSI/VLSI devices. His previous work includes system level signal integrity analysis, micropackaging analysis and micropackaging design for numerous high-performance microprocessors and peripherals.