A 533-MHz BiCMOS Superscalar RISC Microprocessor

Cliff A. Maier, *Member, IEEE*, James A. Markevitch, *Member, IEEE*, Cheryl Senter Brashears, Tim Sippel, *Member, IEEE*, Earl T. Cohen, Jim Blomgren, James G. Ballard, Jay Pattin, Viki Moldenhauer, Jeffrey A. Thomas, *Member, IEEE*, and George Taylor

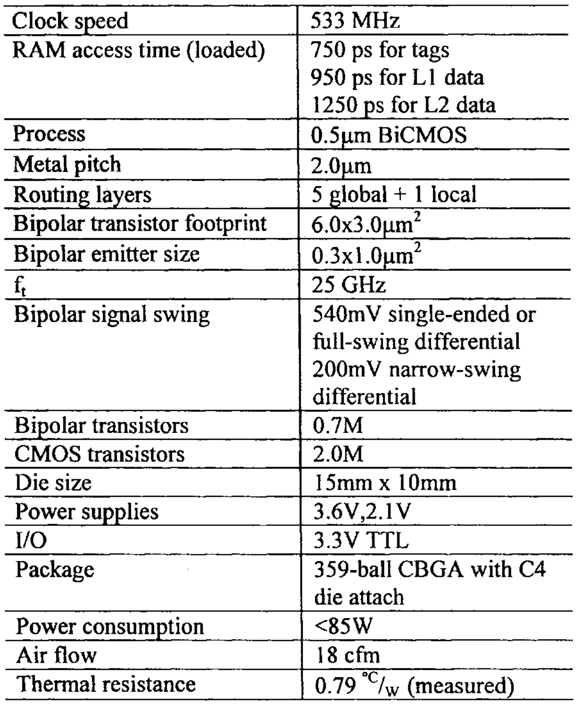
***Abstract*—This 533-MHz BiCMOS very large scale integration (VLSI) implementation of the PowerPC architecture contains three pipelines and a large on-chip secondary cache to achieve a peak performance of 1600 MIPS. The 15 mm × 10 mm die contains 2.7M transistors (2M CMOS and 0.7M bipolar) and dissipates less than 85 W. The die is fabricated in a six-level metal, 0.5-μm BiCMOS process and requires 3.6 and 2.1 V power supplies.**

***Index Terms*—BiCMOS integrated circuits, bipolar digital integrated circuits, emitter coupled logic, microprocessors.**

I. Introduction

A THREE-WAY superscalar microprocessor chip implementing the PowerPC Architecture [1], [2] has been designed to operate at 533 MHz by taking advantage of an advanced BiCMOS process and through innovative design techniques. The processor is designed to be compatible with existing air-cooled desk-side systems and is fully compatible with the PowerPC 60× bus standard [3], supporting bus speeds up to 100 MHz. The chip contains 2.7M transistors on a 15 mm × 10 mm die fabricated with a 0.5-μm BiCMOS technology providing five global and one local interconnect layer (Table I). The estimated SPECint_base95 and SPECfp_base95 for the chip, assuming a 1-MB off-chip cache and a 66-MHz bus, are 12 and 10, respectively. The processor's high speed was achieved by devising a microarchitecture designed with attention to the capabilities of the process.

TABLE I
Process and Chip Features



Clock speed	533 MHz
RAM access time (loaded)	750 ps for tags 950 ps for L1 data 1250 ps for L2 data
Process	0.5μm BiCMOS
Metal pitch	2.0μm
Routing layers	5 global + 1 local
Bipolar transistor footprint	6.0x3.0μm^2
Bipolar emitter size	0.3x1.0μm^2
f_t	25 GHz
Bipolar signal swing	540mV single-ended or full-swing differential 200mV narrow-swing differential
Bipolar transistors	0.7M
CMOS transistors	2.0M
Die size	15mm x 10mm
Power supplies	3.6V,2.1V
I/O	3.3V TTL
Package	359-ball CBGA with C4 die attach
Power consumption	<85W
Air flow	18 cfm
Thermal resistance	0.79 $^{\circ C}/_{W}$ (measured)

The three-way superscalar microarchitecture issues one integer or floating-point, one load/store, and one branch instruction in parallel per cycle. Two-to-three way superscalar is an architectural "sweet spot": enough hardware parallelism to be effective without requiring complex, out-of-order execution.

All logic circuits are implemented in three-level emitter coupled logic (ECL) with input ORing, allowing extremely complex gates. Only RAM structures were implemented with CMOS circuits. Each gate may contain an arbitrary number of emitter followers, having inverted or noninverted inputs, and having outputs on any combination of the three voltage levels. Since several circuit core and emitter follower power levels are available, the cell library is extremely large. Advanced design tools allow most cells to be automatically generated, providing layout as well as timing information for use in the design process.

All single-ported RAM circuits are implemented with 6T CMOS cells and bipolar word line drivers and sense amps, with additional access devices for multiported RAM structures.

Manuscript received April 17, 1997; revised July 31, 1997.

C. A. Maier and C. Senter Brashears were with Exponential Technology, San Jose, CA 95131 USA. They are now with AMD, Sunnyvale, CA 94088 USA.

J. A. Markevitch was with Exponential Technology, San Jose, CA 95131 USA. He is now with Evergreen Technology, Palo Alto, CA 94301 USA.

T. Sippel was with Exponential Technology, San Jose, CA 95131 USA. He is now with Alchemy Designs, Sunnyvale, CA 94087 USA.

E. T. Cohen was with Exponential Technology, San Jose, CA 95131 USA. He is now with BiT Microsystems, Fremont, CA 94538 USA.

J. Blomgren was with Exponential Technology, San Jose, CA 95131 USA. He is now with EVSX, Austin, TX 78735 USA.

J. G. Ballard was with Exponential Technology, San Jose, CA 95131 USA. He is now with Sun Microsystems, Mountain View, CA 94043 USA.

J. Pattin was with Exponential Technology, San Jose, CA 95131 USA. He is now with Epigram, Palo Alto, CA 94301 USA.

V. Moldenhauer was with Exponential Technology, San Jose, CA 95131 USA. She is now with Philips Electronics, Sunnyvale, CA 94088 USA.

J. A. Thomas was with Exponential Technology, San Jose, CA 95131 USA. He is now with DEC, Palo Alto, CA 94301 USA.

G. Taylor was with Exponential Technology, San Jose, CA 95131 USA. He is now a consultant in Los Altos, CA 94024 USA.

Publisher Item Identifier S 0018-9200(97)08028-1.

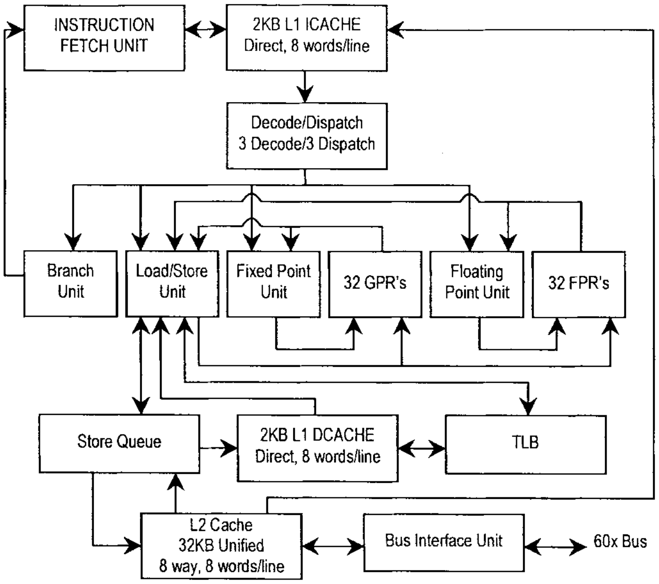

INSTRUCTION FETCH UNIT
2KB L1 ICACHE Direct, 8 words/line
Decode/Dispatch 3 Decode/3 Dispatch
Branch Unit
Load/Store Unit
Fixed Point Unit
32 GPR's
Floating Point Unit
32 FPR's
Store Queue
2KB L1 DCACHE Direct, 8 words/line
TLB
L2 Cache 32KB Unified 8 way, 8 words/line
Bus Interface Unit
60x Bus

Fig. 1. Datapath block diagram.

The chip incorporates Harvard 2-kB level 1 (L1) caches with 0.8 ns access as well as a unified 32-kB level 2 (L2) cache with 1.3 ns access. The secondary cache was optimized for power rather than speed. All RAM's were designed to fit optimally into the microarchitecture. For example, many of the RAM's are synchronous and incorporate address and data latches within the custom block to allow an extra half-cycle of hold time.

The chip requires a standard air-cooled enclosure. Plenum air flow is specified at 18 cubic ft/min. Thermal resistance was modeled to be 0.76°C/W, and measured at 0.79°C/W.

II. Microarchitecture

The BiCMOS process permits high-speed (but relatively high-power compared to CMOS) logic circuits as well as CMOS RAM elements. In order to support a high internal clock frequency while minimizing power requirements and transistor count, it was necessary to adopt a microarchitecture that took best advantage of the process technology.

The processor is three-way superscalar, capable of issuing floating point or integer, load/store, and branch instructions in parallel. The processor contains several functional units, including the integer unit, floating point unit (FPU), branch prediction unit, instruction cache, data cache, translation look-aside buffer, and L2 cache (Fig. 1).

The integer pipeline is shallow for a microprocessor operating at this clock rate, with only six stages: fetch (F), decode (D), address calculation (A), cache access (C), miss detection (M), and write (W). Innovative features of this microarchitecture are a "moving X stage" and aggressive grouping rules. The moving X stage allows the integer arithmetic logic unit (ALU) to adjust its position in the pipeline for either minimal branch latency or for a zero-cycle load/use penalty. When the ALU operates early in the pipe, quick branch resolution and minimal address generation interlocks are possible. Moving the ALU later in the pipe allows dual issue of load/use pairs.

F 0x100: bge .+0x3c 0x13c: add r3.r3.r2	
D Decode and issue as a pair	**F** 0x140: lwz r4,4(r3) 0x144: cmpwi r4,0 0x148: bne .+0x18
Ax branch resolve ALU add	**D** Decode and issue as a triple
C	**A** address generation
M	**C** cache access
W write r3	**Mx** cache tag check ALU compare
	W write r4 branch resolve

Fig. 2. Pipeline diagram showing grouping and moving X stage.

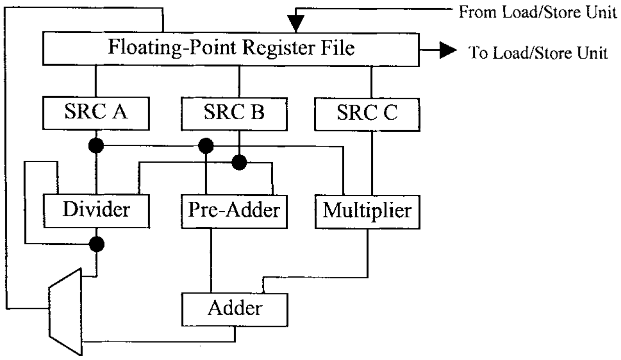


Fig. 3. Simplified FPU block diagram.

The grouping rules allow a load/use pair to issue as a group (due to the moving X stage) and also let a branch issue as the first of a pair of instructions.

Fig. 2 shows a pipeline diagram illustrating the moving X stage and grouping. For the instructions issued in the first cycle, the ALU can calculate the result of the *add* in the address calculation stage as well as resolve the branch. In the next group of three instructions, the *compare* depends on the results of the *load*. As a result, the X stage "slides out" to the miss detection stage, and the results of the *compare* are available in time for the write stage. The ALU will "slide back" whenever idle cycles allow it to.

A. Floating Point Unit

The FPU contains an adder, multiplier, and divider, as well as its own register file (Fig. 3). The FPU is capable of performing a normal multiply–add instruction without stalling except in the case of double-precision arguments, which require two passes through the multiplier. A double-precision floating point addition can be accomplished without stalling.

The floating point pipeline is six stages long (not including fetch) and is loosely coupled to the integer pipeline. Fig. 4 shows the allowed alignments between the integer and floating point pipelines. Loosely coupling the pipelines prevents one

	F0	F1	F2	F3	F4	F5
D	■					
A		■	■	■	■	
C		■	■	■	■	
M		■	■	■	■	
W				■	■	■
					■	■
						■

Fig. 4. Allowed pipeline alignments.

	F0	F1	F2	F3	F4	F5
D	■					
A		■				
C			■			
M				■		
W					■	
						■

Fig. 5. Highest performance pipeline alignment.

pipeline's stalls from having to be connected to the other pipeline, reducing critical paths. In addition, if one pipeline stalls, the other pipeline can proceed, increasing performance slightly.

Fig. 5 represents the highest performance pipeline alignment in which neither pipeline stalls. This alignment is not always possible due to the need to coordinate the pipelines to handle exceptions and cache RAM updates. There are several restrictions to the allowed pipeline alignments that require the pipelines to pass through the M/F4 alignment.

The divider uses a radix-4 Sweeny, Robinson, and Tocher (SRT) algorithm and may be operated in either an interlocked or noninterlocked fashion, depending on whether an exception is possible. The divider produces two bits of result per cycle and was designed to minimize circuit complexity. When operated synchronously, the divide will stall in the F4 stage until completion (in some rare cases involving denormalized numbers requiring in excess of one hundred cycles). When processed in a noninterlocked fashion, the divide is placed in a special "F5" stage, and once complete will remain "valid" until an opportunity to write its results to the FP register file is found (in an idle cycle when the FP register file is not being written by subsequent FP operate instructions).

B. Load/Store Unit and Caches

The 2-kB instruction and data caches are direct-mapped to support the high clock rate, while the 32-kB L2 cache is eight-way set associative. The primary caches are write-through while the L2 cache is copyback. All L1 cache entries are guaranteed to be in the L2 cache. A 256-entry branch prediction table is coupled to the instruction cache. The branch prediction mechanism uses a 2-b history field. To provide sufficient storage capacity for high performance, the L2 cache, both tags and data, is contained on-chip. Most cache operations are managed by the L2 cache control circuitry, keeping the L1 caches simple and fast. In addition, the two-level on-chip scheme allows snoop operations for coherency to be handled without interrupting the processor pipeline and without the need for either dual-ported or duplicated cache tags. The average miss latency to the L2 cache for L1 data and instruction cache misses is 3.5 and 4.5 cycles, respectively.

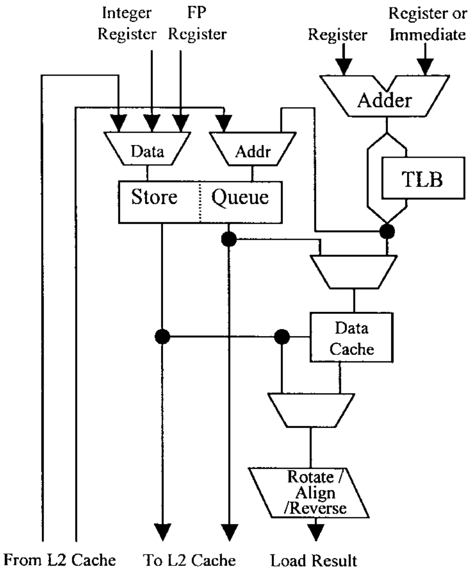


Fig. 6. Simplified load/store unit block diagram.

The L2 cache data RAM is arranged as two interleaved banks, each of which is divided into four 64-b subbanks. Each access requires two cycles, but sequential accesses to alternating banks allow one operation to be started on every cycle. The L2 cache implements the modified, exclusive, shared, invalid (MESI) cache coherency protocol. Due to the parallelism implemented in the L2 cache, two data cache misses that hit in the L2, two instruction cache misses that hit in the L2, one store or cache management operation, one cache line writeback, one snoop tag check, one snoop push, and three missed L2 accesses can be processed in parallel.

The load/store unit (LSU) executes load, store, and cache operations in conjunction with the L2 cache and bus interface unit. The LSU contains an effective address adder, the translation lookaside buffer (TLB), the data cache, a store queue, and a rotator to handle misaligned data and byte-reversal (Fig. 6).

The LSU reads the results of load instructions from the data cache and bypasses them to any execution unit that may need them as operands for other instructions, including fixed-point instructions issued during the same cycle as the load. As a result, the load-use penalty for ALU operations is effectively zero cycles.

Store instructions ordinarily can be issued without stalling the pipeline, as a four double-word store queue holds the data until it can be written into the L2 cache and possibly the data cache. The store queue implements a store-around protocol.

C. Branch Predict and Instruction Fetch Units

The instruction fetch unit contains the instruction cache, the instruction TLB, a six-word instruction buffer and a branch prediction RAM. The instruction buffer contains a four-entry decode buffer and a two-entry fetch buffer.

The fetch unit contains a copy of the program counter that is updated based on whether the upcoming instruction is a branch and whether the branch prediction RAM predicts that the branch will be taken.

The branch unit predicts whether branches are taken and computes the branch target addresses. All branch predictions are tracked in a 256-entry combined 2-b history and target buffer. Mispredicted branches are handled by flushing the pipeline and sending the correct branch target address back to the fetch unit where the fetch program counter is updated. The branch misprediction penalty depends on where in the pipeline the branch is resolved and can range from three to five cycles. The condition register resides in the branch unit where all condition register logical instructions are executed.

D. Integer Execution Unit

The integer execution unit consists of a single pipe stage that executes instructions in one of five subunits: an adder unit, a shifter/rotator, a leading-zero counter, a divider, and a multiplier. Only one subunit can execute an instruction in any given cycle.

The multiplier takes three to five cycles for a 32 b × 32 b multiply and uses two internal registers to hold intermediate results. Divides use a combination of the adder and the shifter/rotator.

III. Global Implementation

In order to support the processor's high clock speed, power, and voltage reference distribution requirements, it was necessary to pay careful attention to floorplanning and global distribution strategies.

A. Clocking

The chip uses a bipolar phase-locked loop (PLL) design with a capture range of from 250 to 650 MHz. This PLL can multiply the input bus clock by any integer from 2 to 32, allowing use in a variety of currently available and future systems. The jitter specification for the PLL is ±30 ps (simulated by varying the supply voltage by 1 V in a sinusoidal fashion). As the PLL is fully differential, common-mode noise is rejected and substrate injection problems, already minimized due to the insulated substrate, are reduced.

There are 16 500 clocked elements on the chip (each flip-flop counting as two elements, while latches count as one). Clock distribution is accomplished using a modified H tree with one large driver supplying a differential clock to 450 tributary drivers. The ends of the clock tree are shorted to reduce skew, and there is one H-tree for each polarity of the differential clock. Each tributary driver supplies one row of up to 72 clocked elements. Tuning the speed of the tributary drivers based on their load minimizes clock skew. Clock skew between any two clocked elements on chip was SPICE simulated to be less than 125 ps, 75 ps of which is local skew between the tributary drivers and the clocked elements. Timing analysis tools calculate the total clock delay to every logic gate in the design, taking into account both global and local skew.

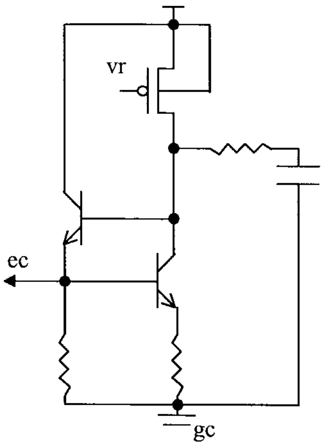


Fig. 7. Reference circuit current mirror.

B. Floorplanning

The floorplan is arranged so as to minimize the length of large buses. Instructions flow from the L1 instruction cache through the instruction fetch circuitry where control signals are sent to the other logical units. The control circuitry is centralized to minimize control critical paths. The FPU receives control signals from the nearby pipeline control circuitry. The L1 data cache is located central to the FPU and integer ALU and register file. The CMOS RAM circuits are generally isolated from ECL nets in order to reduce the effect of crosstalk caused by the larger CMOS swing.

Most block-to-block communication is accomplished through abutment, and in many cases nets feed through blocks without connecting to circuitry. Two global routing channels, one vertical and one horizontal, are used to route large buses without interfering with local routing channels. Due to the length of these channels, care had to be taken to avoid crosstalk between wires of a bus. These channels were prerouted and connected by abutment to the other blocks.

The chip is packaged using area-bonding, with power and signal bumps located in an array over the entire area of the die. Due to the requirement to place power and ground bumps in a regular array in order to simplify power distribution, signal bumps are disallowed over ECL logic. Placing signal bumps over ECL logic circuits would disrupt the M5–M6 power distribution grid pattern. Signal bumps are allowed over CMOS circuits, which have reduced power requirements, only if a metal shielding layer is included. The chip is fabricated using a six-metal layer process, the upper five layers of which may be used for global routing. M1 is reserved for local routing.

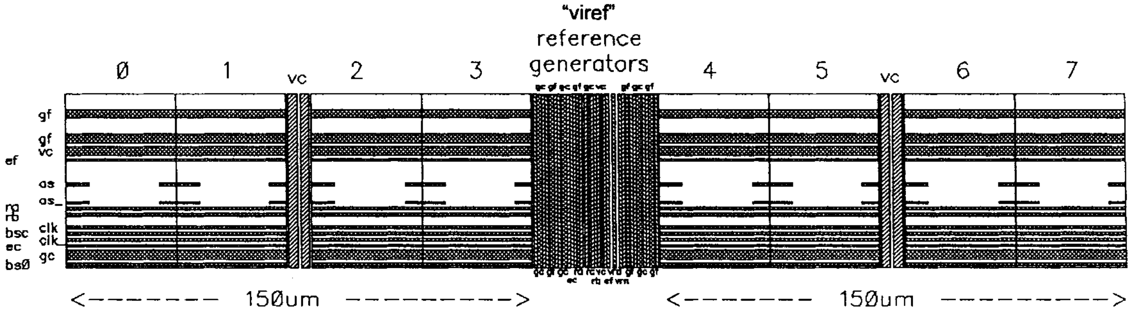


Fig. 8. Power and reference distribution.

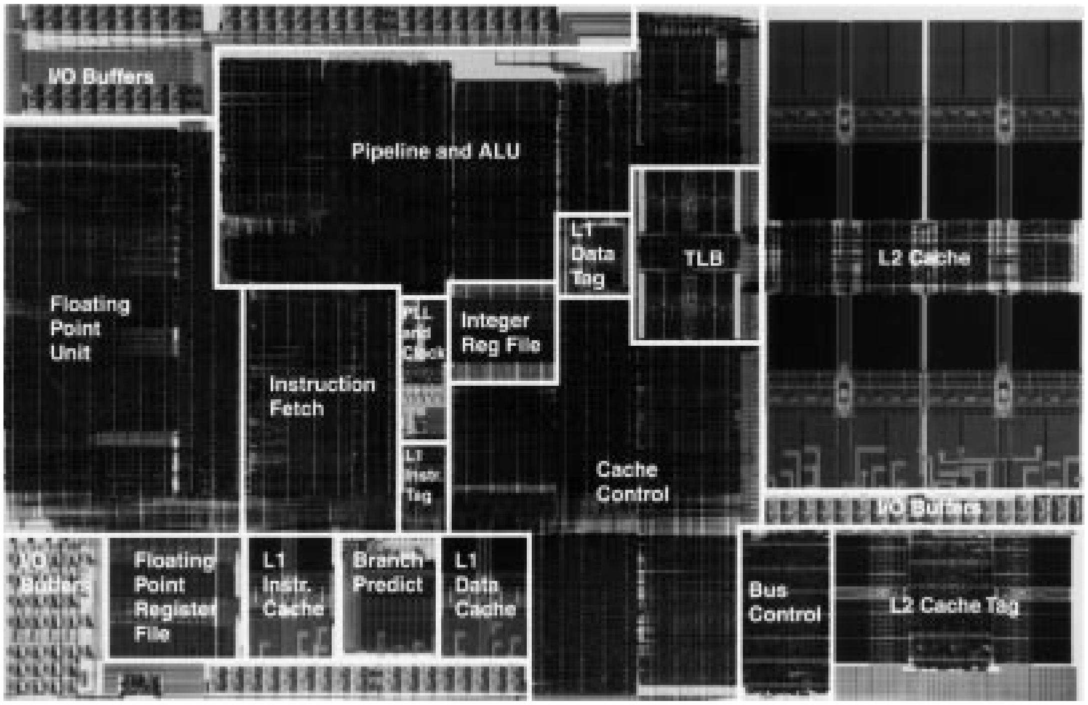


Fig. 9. Chip microphotograph with overlay.

The sixth metal layer is reserved for routing of global signals such as VDD and VSS, to distribute the differential clock from the PLL to the ECL logic areas, and to route the system clock input to the PLL.

The current and voltage references do not require low skew distribution since they are static signals, but they are required to track with G_{C} (the ECL core ground). This tracking is accomplished with current mirror circuits (Fig. 7). In the figure, a global reference, V_{R} is used to generate E_c, the voltage reference for the ECL current sources. The reference and power connections, as well as clock connections, are made by abutment within the cell rows.

The global signals are distributed vertically through the logic rows using global straps (Fig. 8). The distance between the logic cells and the global bumps is minimized through the use of wide metal-6 straps which run orthogonally to the metal-2/metal-4 straps. The "viref" strap has cells placed in it to generate the voltage references for each of the logic voltage levels, as well as the current references. IR drop is simulated to be <5 mV between the reference generator and any logic cell. The total IR drop on the power rails is simulated to be <25 mV.

IV. Implementation Examples

A. Circuits

The chip (Fig. 9) is implemented in bipolar-based BiCMOS. Bipolar transistors are dielectrically isolated with trenches and bonded wafer oxide. While a bipolar transistor is larger than a minimum-size CMOS device, bipolar gates provide two area advantages. First, logic functions implemented by a single gate can be complex without sacrificing speed; the layout of more complicated functions tends to be metal rather than device-limited. Second, drive strength of bipolar devices can be tuned over a wide range by adjusting resistors controlling the current, eliminating the levels of buffering often found in CMOS designs. All logic circuits were implemented in ECL while all RAM arrays were implemented in CMOS.

The cell library uses up to three-level series gating allowing functions such as three-input XOR's or eight-to-one MUXes

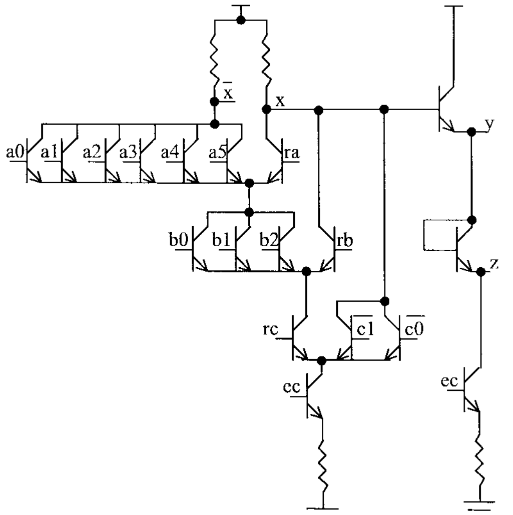


Fig. 10. OA6311 gate with Y and Z followers.

TABLE II
DELAYS (PS) FOR TWO-INPUT OR WITH FOLLOWER

Core / Follower	Load (mm of wire / number of base loads)							
(μA)	0.0/0	0.25/1	0.5/2	1.0/4	2.0/8	4.0/16	6.0/24	1.5/32
200/200	40/40	70/67	94/89	136/128	217/200	392/357	614/552	265/205
200/400	43/43	68/66	89/85	126/118	194/180	338/310	513/467	232/183
200/800	52/52	71/69	88/85	119/113	177/166	300/277	450/412	206/167
400/800	43/43	58/57	71/68	95/90	142/133	250/230	391/356	165/134
400/1600	53/53	64/63	75/72	95/91	137/129	235/218	364/333	156/129

TABLE III
RISE/FALL DELAYS FOR TYPICAL GATES

Gate type	From Input	Core / Follower (μA)	Unloaded gate delay (ps)
2-input OR	-	200/200	40/40
2-to-1 MUX	Data	200/200	44/36
2-to-1 MUX	Select	200/200	61/69
8-to-1 MUX	Data	200/200	92/62
8-to-1 MUX	Select	200/200	120/105
8-to-1 MUX	Data	400/400	63/46
8-to-1 MUX	Select	400/400	83/70

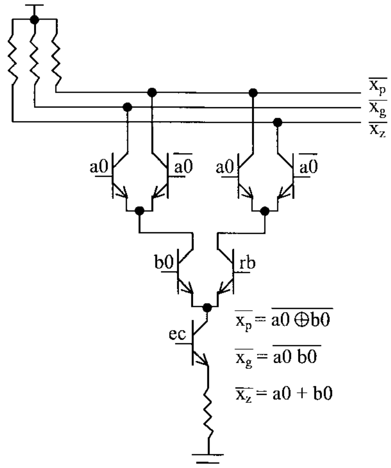

$\overline{x_p}$
$\overline{x_g}$
$\overline{x_z}$
a0
$\overline{a0}$
a0
$\overline{a0}$
b0
rb
ec
$\overline{x_p} = \overline{a0 \oplus b0}$
$\overline{x_g} = \overline{a0\ b0}$
$\overline{x_z} = a0 + b0$

Fig. 11. Generate/propagate/zero gate.

to be implemented in a single gate delay. Multiple output functions such as one of eight decoders and generate-propagate are possible. A typical example of a bipolar ECL gate is a OA6311 (Fig. 10). In addition to the inputs a0–a5, b0–b2, and $\overline{c0} - \overline{c1}$, the gate also requires reference voltages, r_a, r_b, and r_c, and a current source control reference, e_c. The a0–a5 inputs are OR'd together, as are the b0–b2 inputs. The $\overline{c0}$ and $\overline{c1}$ input devices are wired to the x signal, forming a NOR or an AND with inverted inputs, thus allowing AND gates with more than three inputs. The power overhead required for the voltage references is approximately 12%, and the additional area required for the reference generator circuitry based on current mirrors is approximately 13%. Both true and complement outputs are available for single-output functions.

The fundamental operation of the gate is simple, relying on current steering through emitter-coupled devices. A group of two or more emitter-coupled devices acts as a current switch, steering the current generated by a current source at the bottom of the tree through whichever side of the current switch has a device which is on (due to a "high" base voltage). By setting the base voltages on the input devices appropriately, the current through one of the load resistors at the top of the tree may be turned on, pulling down the appropriate output.

Followers added to gates improve drive strength and shift output levels down to be used as lower-level inputs. The OA6311 gate shown in Fig. 10 has both a y (one diode drop) and a z (two diode drop) follower on its x output. Followers use a separate current source, hence the core current (controlling the unloaded propagation delay and the speed of the x and $\overline{x}$ outputs) can be set independently from the follower current (controlling the speed of the y and z-level outputs). Core currents for cells range from 70–800 μA, while follower currents range from 70–1600 μA. Table II provides examples of speed/power tradeoffs. Table III shows that the speed of more complex gates is very close to the speed of a simple OR gate, and that slower speeds can be compensated for by increasing cell core current, which has small impact on cell area.

The loading on one output of a follower can effect the timing of the other outputs of the follower, an effect called "reflected delay." In order to avoid this, an inverted output is often chosen due to its relative isolation from the noninverted outputs.

The x-level outputs are not appropriate for heavily loaded nets and are typically reserved for light loads or, when differential narrow swing outputs are used, for datapath critical paths.

Fig. 11 is a schematic for a generate-propagate-zero (GPZ) gate, a typical example of a multiple-output ECL circuit. The GPZ gate is possible because for any combination of inputs it is necessary to pull down through one (and only one) load resistor. The gate takes one differential and one single-ended input and produces generate, propagate, and zero-detect outputs, each of which is inverted polarity.

The advantages of bipolar logic are best illustrated by the gate-depth of the logic. Even with the shallow, six-stage pipeline, the worst-case flip-flop to flip-flop paths in the chip have no more than 15 gate delays while most paths contain

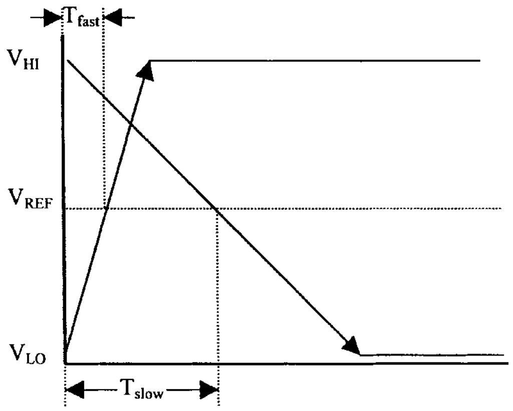


Fig. 12. Effect of rise/fall-time disparity on coupling.

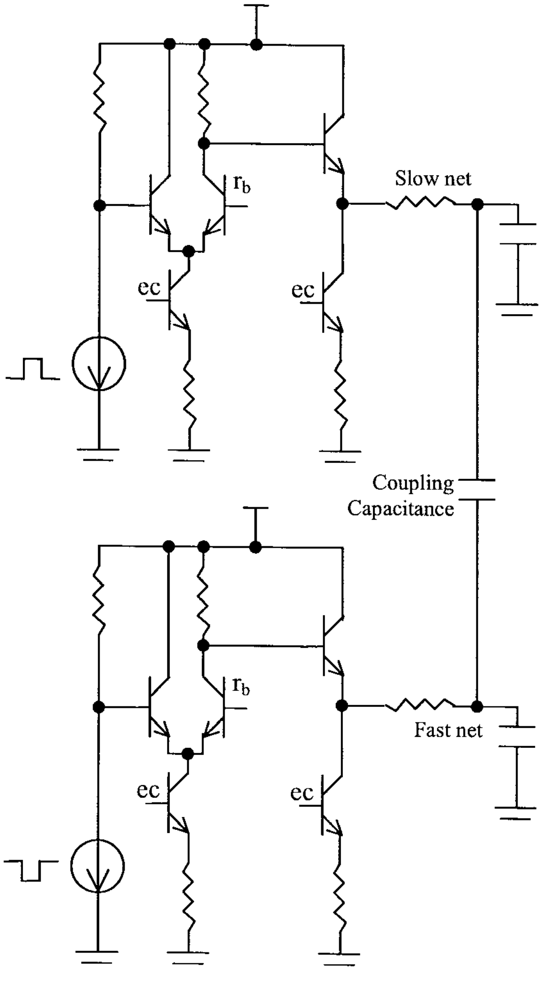


Fig. 13. Circuit used for coupling analysis in SPICE.

fewer than ten gate delays. There are 36k bipolar gates in the design. Due to the three-level series-gated logic and the need to operate at low temperature, a 3.6-V supply is required to avoid device saturation. To save power, an additional 2.1-V supply is provided for cells with a y-level follower but no z-level follower.

Differential outputs are allowed, and reduced-swing "x-level" differential outputs are commonly used for speed-critical paths where capacitive loading is light. Normal single-ended voltage swing is 540 mV, while reduced differential swing is 200 mV. Wider single-ended and differential swing levels are available for situations such as high fan-in that reduce the effective swing. Differential signals are not routed together, and narrow swing differential signals and full-swing signals are generally not routed near each other. Outputs may be wire-OR'ed subject to IR drop considerations, allowing complicated sum-of-products expressions to be implemented in a single gate delay in many cases.

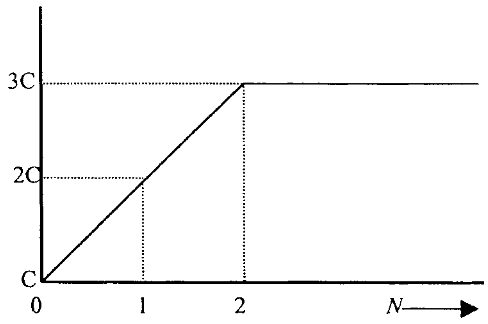


Fig. 14. Effective capacitance as a function of N.

A worst-case capacitive coupling factor of three times the static capacitance between neighboring wires is assumed if the signals on the wires have the same signal swing. A factor of three rather than two [4] was used to compensate for disparate rise and fall times. In order to simplify the worst-case analysis, this factor of three was applied, regardless of the relative direction of the swings, to both nets (even the net undergoing the fast transition, which in actuality "sees" an effective capacitance of at most twice the static capacitance). This was based on a first-order analysis that shows that the effective coupling capacitance between two nets is three times the static capacitance for nets undergoing an odd mode transition if the rise time (or fall time) is at least twice as fast as the fall time (or rise time).

In Fig. 12, the rise and fall times, T_{fast} and T_{slow} are indicated at 0 to 50% transition times, since the effective switching time is determined by a transition to the reference voltage. Assume $T_{\mathrm{slow}} = N\,T_{\mathrm{fast}}$ and that the total capacitance seen by the wire with the falling signal is the average of the capacitance seen while the net it is coupled to is undergoing a positive transition (C_1) and the normal static capacitance ($C_2 = C$) when the rising signal has reached V_{HI}. The voltage swing is $\Delta V = V_{\mathrm{HI}} - V_{\mathrm{LO}}$. C is the static capacitance between the two wires. As both narrow-swing differential and full-swing single-ended signals may be routed in close proximity, the case of full-swing signals coupling to narrower swing signals was analyzed. Assume that the two signals have swings ΔV and $\Delta V/A$. The effective capacitance seen by the wire with swing $\Delta V/A$ and transition time T_{slow} while the wire it couples to is transitioning is

$$C_1 = \left(\frac{\Delta V}{\dfrac{\Delta V}{A \cdot N}} + 1 \right) C = (A \cdot N + 1) C.$$

If N is greater than two then the effective capacitance "seen" by the signal making the slower transition is the average

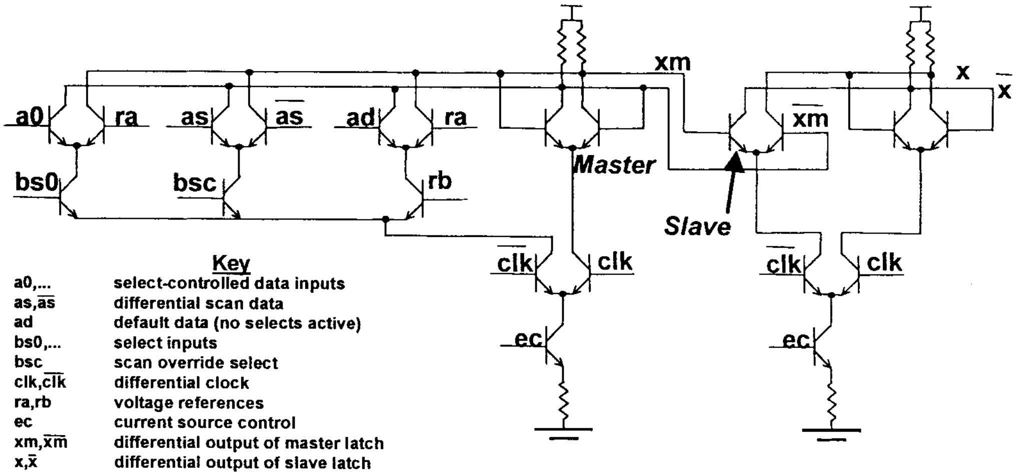

a0
ra
as
$\overline{as}$
ad
ra
bs0
bsc
rb
xm
$\overline{xm}$
x
$\overline{x}$
Master
Slave
$\overline{clk}$
clk
$\overline{clk}$
clk
ec
ec
Key
a0,... select-controlled data inputs
as,$\overline{as}$ differential scan data
ad default data (no selects active)
bs0,... select inputs
bsc scan override select
clk,$\overline{clk}$ differential clock
ra,rb voltage references
ec current source control
xm,$\overline{xm}$ differential output of master latch
x,$\overline{x}$ differential output of slave latch

Fig. 15. Flip-flop with one-hot MUX and scan override.

of C_1, which applies while the fast signal is transitioning, and the static capacitance which applies from the time the fast signal has stopped transitioning until the slow signal has reached its reference point

$$C_{\text{average}} = \frac{2t_{\text{fast}}C_1 + C_2(Nt_{\text{fast}} - 2t_{\text{fast}})}{Nt_{\text{fast}}} = (2A+1)C.$$

From this analysis it is apparent that the effective average capacitance for the case where $N > 2$ and $A = 1$ is simply $3C$. This factor was confirmed by simulating coupled nets in SPICE (Fig. 13) and laboratory experiments with fabricated circuits.

For $N = 1$ it is expected that the effective capacitance would be equal to twice the static capacitance, while for $N = 0$, corresponding to the coupled-to signal being static, the effective capacitance is equal to the static capacitance. Using this simple analysis, it is a simple matter to generate the coupling coefficient for any positive value of N (Fig. 14). For $1 < N < 2$, the fast signal has completed only $1/N$ of its entire transition by the time the slow signal has reached its reference. As a result, the effective capacitance is simply C_1. This is likewise the case for $0 < N < 1$, in which case "t_{slow}" will transition past its reference to N times its full swing by the time "t_{fast}" reaches its reference.

While the effective capacitance is shown analytically to depend on N, we simplified our delay calculations by using $3C$ for all coupling cases. While capacitive coupling could also induce spurious transitions, SPICE analysis showed this noise to be less than 50 mV. We chose this level of approximation to simplify the calculations and allow universal application. More detailed analyzes of similar phenomena were performed in [5].

B. Testability

In order to facilitate testability, all flip-flops are scannable. A solution requiring low cycle time and area overhead is used. Flip-flops take advantage of three-level series gating and incorporate a one-hot (unencoded selects, no more than one of which is active) MUX function for their data inputs. An additional scan select signal chooses a data input that connects all the flip-flops in a scan chain (Fig. 15). The MUX can also be used for a data hold function. Differential scan connections avoid the requirement that all flip-flops use the top-level reference voltage.

The flip-flop is based on a standard ECL master-slave latch. The master and slave latches consist of emitter-coupled current switches with bases tied to the outputs. The master latch current switch in Fig. 15, for example, has its right transistor tied to $\overline{xm}$ and its left transistor tied to xm. If xm is asserted, the left transistor is turned on, pulling down $\overline{xm}$, and the right transistor is shut off, preventing a voltage drop between V_c and xm.

For flip-flops with normal-swing outputs, scan connections use full-swing differential signals. Scan select must take precedence over normal selects because the state of normal select signals may be arbitrary during scanning operations. A special scan select driver cell generates a logic one 400 mV higher than the norm (and a normal logic zero), allowing scan selects to override normal selects. As this technique drives the devices into saturation, it is allowed only during low-speed scan operations. The scan mechanism requires three additional devices per flip-flop, plus one scan-select driver per flip-flop row, and adds fewer than 11 ps of delay to the flip-flop data input MUX.

Basic JTAG support (IEEE 1149.1) [6] as well as parallel and serial-load internal scan capability are supported. The testing circuitry interfaces with the PLL to allow speed-testing by causing the on-chip clock to be triggered twice, effectively running one cycle at the internal processor clock speed. Unlike other published schemes [7], [8], this scheme makes use of the on-chip PLL so that the testing clock can be much slower than the on-chip clock.

V. Design Tools/Methodology

The logic blocks are constructed using a standard cell methodology. The placement of datapath cells is accomplished through the designer's inclusion of register transfer language (RTL) directives. Due to the quantity of standard cells used in the design (36 100 cells with 7580 unique designs) an automated cell generation methodology is used. Based on parameters encoded in the name of a cell, the SPICE deck, Verilog model, timing view, and layout can generally be created automatically using programs designed in-house. Approximately 75% of the cells in the library had layout automatically generated. The layout for commonly used control cells and most datapath cells is hand optimized to minimize area. In cases where a layout cannot be generated automatically, layout parasitics can often be extracted from similar cells by the automated cell generation tools to improve accuracy until final cell layout is available.

The RAM structures, PLL, bandgap, and I/O cells were all custom-designed to optimize area and performance.

In order to minimize power, most blocks were run through a power-optimizer that adjusted cell core and follower currents for all paths in the block to meet the targeted cycle time while minimizing power. The power optimizer uses a greedy algorithm to reduce the total power required for a block and takes advantage of the automated cell creation tools to generate timing views for candidate cells on-the-fly.

Crosstalk was evaluated by extracting net-to-net capacitances for all signals on the same metal layer and screening for coupled nets likely to experience odd mode switching (as predicted by the static timing analyzer). The coupling capacitances to neighboring wires were multiplied by a crosstalk factor that is a function of the relative voltage swing amplitudes of the coupled wires for wires which an interval timing tool (incorporated into the static timing analyzer) predicted could undergo simultaneous transitions. The static timing analyzer was capable of handling distributed RC interconnect networks by calling SPICE when heuristics determined that a lumped capacitive model would provide insufficient accuracy. Because ECL circuits provide very high stage gain, it was sufficient for the timing analyzer to consider only a single driver and its interconnect and active loads; the characteristics of the input signal could, to first order, be ignored.

C and structural Verilog were used to create the RTL netlist for the chip. No synthesis was used in the design. For verification purposes, those logical blocks written in Verilog code were mechanically translated to C, and all C code was compiled into two-state logic models. One model is designed for verification of the entire chip, while another focuses on the floating point unit. Billions of vectors have been run on the models, which run on a network of low-cost personal computers, against an architectural simulator. A Quickturn netlist is also created from the Verilog code used for the physical design and run on Quickturn hardware. Throughout the verification random patterns were used, while directed testing was used to investigate problems and to increase coverage of more complicated blocks of logic.

Compatibility was achieved by creating an architectural simulator and verifying it against a production PowerPC chip. Once the correctness of the simulator was shown, it became the standard against which compiled RTL models were compared.

VI. Summary

A 533-MHz superscalar processor compatible with the PowerPC Architecture has been described. The high clock speed was accomplished through the use of BiCMOS process technology and a carefully tuned microarchitecture. All custom RAM's worked at speed, and the MacOS operating system and applications were run on first silicon after one metal spin.

The chip runs in a standard air-cooled enclosure and conforms to standard bus and system standards.

Acknowledgment

The authors wish to express their appreciation for the tremendous efforts of the other members of the Exponential team, particularly the members of the Logic, Circuits, Physical Design, CAD, Design Verification, and Manufacturing groups.

References

[1] *IBM, The PowerPC Architecture*, 2nd ed. San Francisco, CA: Morgan Kaufmann, Dec. 13, 1994.

[2] E. T. Cohen *et al.*, "A 533MHz BiCMOS Superscalar Microprocessor," in *1997 Dig. Tech. Papers, IEEE Solid State Circuits Conf.*, 1997, pp. 164–165.

[3] *PowerPC 60x Microprocessor Interface Specification, Version 3.0*, Motorola Inc., Mar. 28, 1994.

[4] T. Sakurai, "Closed-form expressions for interconnection delay, coupling, and crosstalk in VLSI's," *IEEE Trans. Electronic Devices*, vol. 40, pp. 118–124, Jan. 1993.

[5] F. Dartu and L. Pileggi, "Calculating worst-case gate delays due to dominant capacitance coupling," in *Proc. 1997 Design Automation Conf.*, June 1997, pp. 46–51.

[6] IEEE Standard 1149.1-1990, "IEEE standard test access port and boundary scan architecture," May 21, 1990.

[7] H. Chang and J. A. Abraham, "Delay test techniques for boundary scan based architectures," in *Proc., Custom Integrated Circuits Conf.*, 1992, pp. 13.2.1–13.2.4.

[8] C. A. Maier *et al.*, "Embedded at-speed testing schemes with low overhead for high speed digital systems on multi-chip modules," in *IEEE Innovative Systems in Silicon Conf.*, Austin, TX, Oct. 8–10, 1996, pp. 210–216.

Cliff A. Maier (S'92–M'97) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Rensselaer Polytechnic Institute (RPI), Troy, NY, in 1992, 1994, and 1996, respectively.

While at RPI, he did research on high-speed microprocessors and cache memories, at-speed testing, advanced packaging, and GaAs bipolar circuits. He joined Exponential Technology in 1996 where he worked on bipolar logic and circuit design for the x704's floating point unit. He has worked at Sun Microsystems as a Designer of high-performance SPARC microprocessors. He is currently at AMD, Sunnyvale, CA, where he is working on microprocessor designs. He has authored and co-authored numerous papers on testing, bipolar circuits, packaging, and high-speed microprocessor design.

James A. Markevitch (A'86–M'97) left the Massachusetts Institute of Technology, Cambridge, in 1980.

He joined Fortune Systems in 1981 to port the UNIX operating system to desktop computers. In 1982, he founded Evergreen Technology, a consulting firm specializing in hardware and software architecture. From 1982 to 1987, he ported the UNIX operating system for clients, designed custom hardware systems, and developed and licensed products for CAD and optical disk systems. From 1988 to 1993, he worked with Lawrence Livermore National Laboratory, where he served as the Software Architect for a series of suborbital and deep-space missions. He began working with Exponential Technology in 1994, developing a set of CAD tools for a variety of functions, including automated timing analysis, as well as performing design and analysis of BiCMOS circuits. He continues consulting with Evergreen Technology, Palo Alto, CA.

Cheryl Senter Brashears received the B.S. degree in electrical engineering and computer science from Duke University, Durham, NC, in 1989.

From 1990 to 1993, she worked at SMOS Systems R&D Division, San Jose, CA, doing architectural and logic design of RISC-based microprocessors. She joined Exponential in 1993 to work on logic design, physical design, and design methodology. In 1996, she became the Physical Design Manager. She is currently employed at AMD, Sunnyvale, CA. She holds three patents in the area of floating point design and out-of-order load/store unit design.

Tim Sippel (S'79–M'81) received the B.S.E.E. degree from the University of California at Davis in 1981 and the M.S.E.E. degree from the University of California at Berkeley in 1982.

From 1982 until 1985, he was a Logic Designer at AT&T Bell Laboratories. From 1985 until 1990, he supported the GDT physical design tools at Mentor Graphics. Since 1990, he has worked as a Consultant, specializing in the setup and support of physical design flows for full-custom VLSI designs. This has included the support of several microprocessor design projects. He is currently on the consulting staff at Alchemy Designs, Sunnyvale, CA.

Earl T. Cohen received the S.B. degree from the Massachusetts Institute of Technology, Cambridge, in 1978 and the Ph.D. degree in E.E.C.S. from the University of California, Berkeley, in 1983.

He has worked in a wide variety of electronic design areas including error correction, supercomputers, mainframe computers, and microprocessor design. He joined Exponential in 1993 and was responsible for the micro-architecture and logic design of the integer ALU, the instruction fetch unit, the load/store unit, and the level-two cache; in addition, he was responsible for Exponential's C-based design language and their C-to-Verilog translator. He is currently the Chief Technology Officer at BiT Microsystems, Fremont, CA. He holds 13 patents, ten of which are from Exponential.

Jim Blomgren received the B.S. degree in EECS from the University of California, Berkeley, in 1982.

He has spent the past 15 years working on the design and production of 370, SPARC, PowerPC, and x86 microprocessors. He co-founded Exponential, San Jose, CA, in 1993 and was responsible for the micro-architecture of the x704 and the design of its pipeline. He holds 20 patents, half of which are from Exponential. He is now with EVSX, Austin, TX.

James G. Ballard received the B.S. degree in computer engineering from Case Western Reserve University, Cleveland, OH, in 1985 and the M.S. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1987.

He has worked on the design of mainframe systems as well as custom high-speed microprocessors. He joined Exponential in 1994 as a Physical Design Engineer and was promoted to Manager of Logic Design in 1996. He is currently working on high-performance microprocessor designs at Sun Microsystems, Mountain View, CA.

Jay Pattin has 15 years experience in operating systems, computer architecture, and processor design.

Viki Moldenhauer received the B.S. degree in electrical engineering and applied mathematics from Carnegie-Mellon University, Pittsburgh, PA.

She worked at Lawrence Livermore National Laboratory from 1982 to 1988, at the National Magnetic Fusion Energy Computer Center, and the S-1 Project. In 1988, she joined Key Computer Laboratories, where she designed a floating point multiply chipset. In 1995, she joined Exponential Technology, where she was responsible for various aspects of physical design and verification on the x704. She is now with Philips Electronics, Sunnyvale, CA.

Jeffrey A. Thomas (S'76–M'77) received the B.S. degree in electrical engineering from the University of Wisconsin-Madison in 1977.

He has worked on projects ranging from mainframe system design to microprocessor design. He joined Exponential, San Jose, CA, in 1994 as Vice President of Engineering. He holds 12 patents in the area of cache design. He is now with DEC, Palo Alto, CA.

George Taylor received the B.S.E.E. degree from Duke University, Durham, NC, in 1978 and the Ph.D. degree in computer science from the University of California, Berkeley, in 1991.

He was a co-founder of Exponential, San Jose, CA, in 1993 and became its Chief Technical Officer in 1994, where his responsibilities have included microarchitecture, arithmetic unit design, and CAD tools for cell generation and power/delay optimization. Prior to Exponential, he was the Design Manager for the MIPS R6000 microprocessor and Director of Experimental Architecture at Sun Microsystems Laboratories. His publications include papers on multilevel cache design and hardware for division and square root.