The Implementation of the Itanium 2 Microprocessor

Samuel D. Naffziger, Glenn Colon-Bonet, Timothy Fischer, Reid Riedlinger, Thomas J. Sullivan, and Tom Grutkowski

***Abstract*—This 64-b microprocessor is the second-generation design of the new Itanium architecture, termed explicitly parallel instruction computing (EPIC). The design seeks to extract maximum performance from EPIC by optimizing the memory system and execution resources for a combination of high bandwidth and low latency. This is achieved by tightly coupling microarchitecture choices to innovative circuit designs and the capabilities of the transistors and wires in the .18-μm bulk Al metal process [16]. The key features of this design are: a short eight-stage pipeline, 11 sustainable issue ports (six Integer, four floating point, half-cycle access level-1 caches, 64-GB/s level-2 cache and 3-MB level-3 cache, all integrated on a 421 mm^2 die. The chip operates at over 1 GHz and is built on significant advances in CMOS circuits and methodologies. After providing an overview of the processor microarchitecture and design, this paper describes a few of these key enabling circuits and design techniques.**

***Index Terms*—Clock design, computer architecture, design methodology, dynamic logic, IA-64, integrated circuit design, microprocessors.**

I. Introduction

THE DESIGN of the Itanium 2 processor sets out to achieve industry leading performance on a broad range of 64-b technical and commercial applications. The team's approach to achieving this goal involved close cooperation and synergy between control design engineers (architects) and circuit design engineers. All of the significant microarchitectural features arose out of careful analysis of the capabilities of the .18-μm bulk technology transistors and interconnect in conjunction with innovative approaches to CMOS circuit design. As the performance benefits of the many possibilities were explored, several approaches were adopted.

1) Maximize parallelism available to compilers for a two bundle (six instructions per cycle) explicitly parallel instruction computing (EPIC) machine by employing dense wiring techniques (e.g., 20 64-b busses in the IEU datapath) and dataflow domino circuits in execution units and caches.
2) Reduce memory system inefficiencies with lowered latencies and increased bandwidth for the cache system based on innovative SRAM designs and self-timed circuits.
3) Reduce branch impacts beyond the predication that the Itanium architecture enables by exploiting the high-speed SRAM technology [1] developed for the data cache coupled with advanced prediction algorithms.

Manuscript received March 15, 2002; revised June 14, 2002.

S. D. Naffziger, G. Colon-Bonet, T. Fischer, R. Riedlinger, and T. J. Sullivan are with Hewlett Packard Company, Fort Collins, CO 80525 USA (e-mail: sdn@fc.hp.com).

T. Grutkowski is with the Intel Corporation, Santa Clara, CA 80525 USA.

Digital Object Identifier 10.1109/JSSC.2002.803943

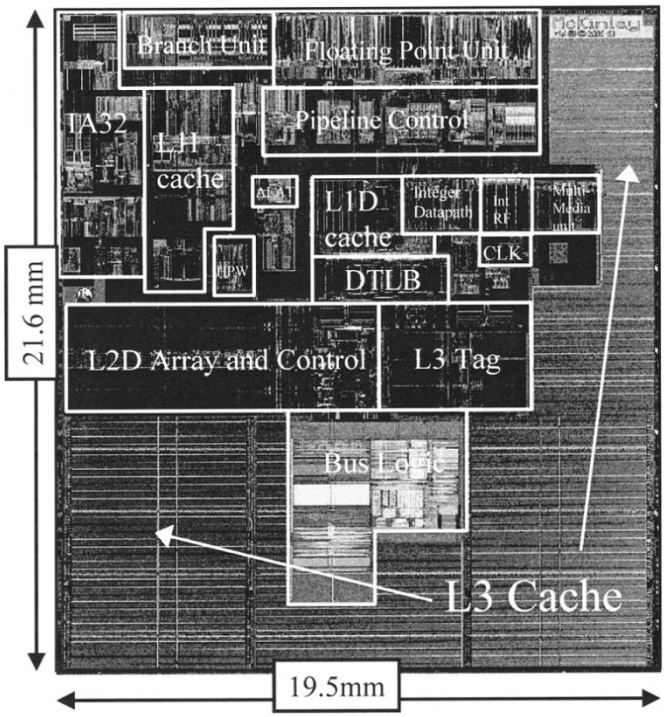


Fig. 1. Itanium 2 overview.

4) Achieve the best balance between high frequency and instruction execution efficiency (IPC) with advanced circuit techniques that greatly reduce latch overhead and clock sensitivity, many of which are discussed in this paper.

Since the details on several of the above features are explored in other articles in this journal, this paper will provide more depth on the key enabling circuits used in all areas of the chip and the methodologies surrounding them, which produced a design that not only breaks new ground in density and performance, but is robust enough for the requirements of high-volume manufacturing. But first, an overview of the processor is in order.

II. Overview

From the die photo (Fig. 1), it can be seen that about half the processor area is consumed by the 3-MB level-3 cache. This cache is optimized for density while still providing a latency of 12 cycles of load-to-use, about half that of a similar sized off-chip cache [2]. It is implemented with 135 separate "subarrays" that enable high density and the ability to conform to the irregular shape of the processor core with flexible subarray placement. Memory and I/O are accessed through the 128-b, 400-MHz front side bus (FSB), which at 6.4 GB/s

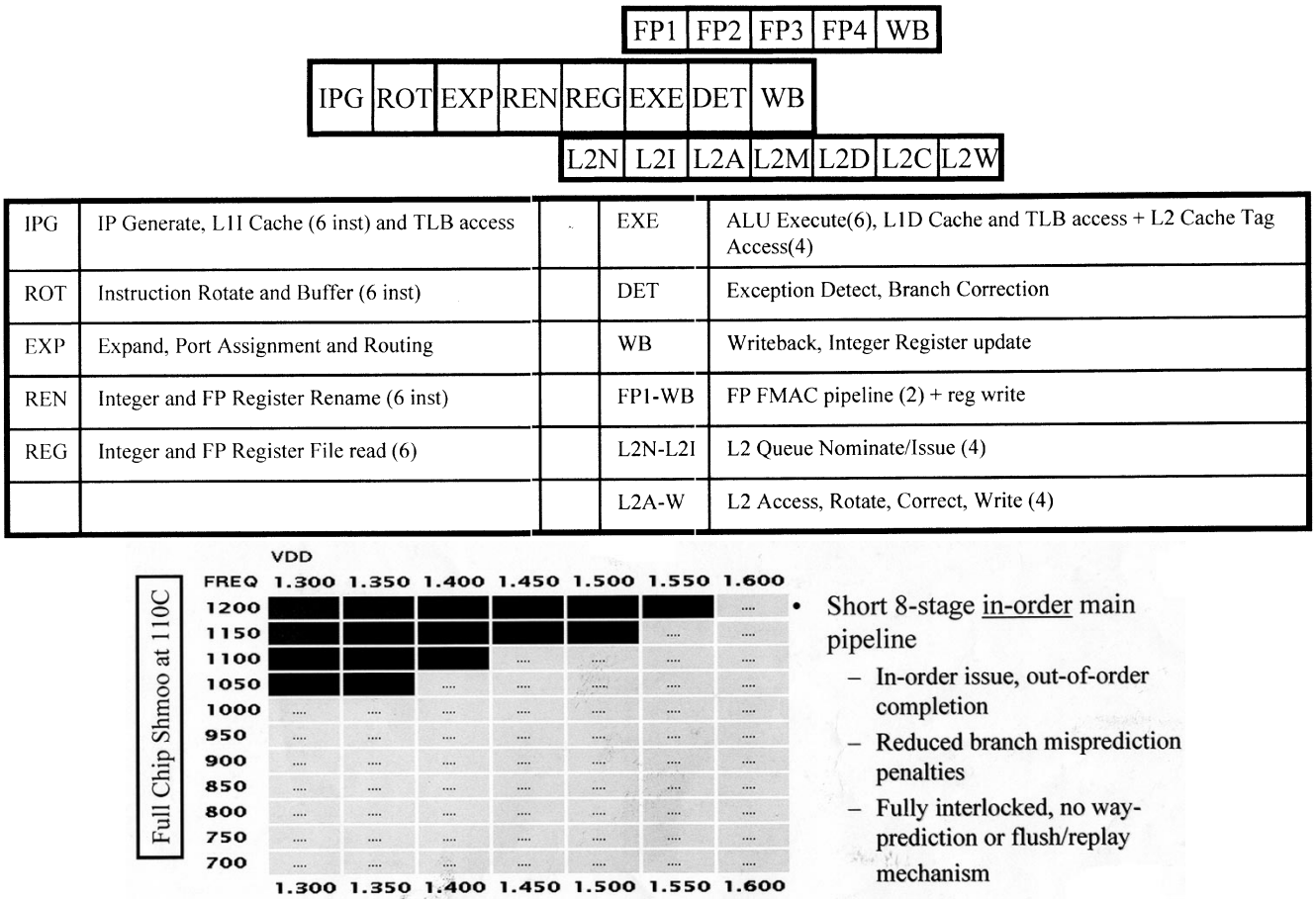


Fig. 2. Itanium 2 pipeline and chip shmoo.

provides about three times the bandwidth of the first-generation Itanium processor. This capability provides headroom for future Itanium-2 based designs, since bandwidth is a key performance enabler for high-end technical and commercial applications. The I/O pads are distributed across four stripes embedded in the L3 cache arrays which reduced the area consumed for I/O distribution to the C4 bumps. The processor core itself provides tremendous integer execution resources with the six 1-cycle integer and six 2-cycle multimedia units. These accomplish full symmetric bypassing with each other and the L1D cache in combination with a 20 ported, 128×65-b register file [3]. The integer units and register file are 4 mm $\times$ 1.9 mm, including hardware support for IA-32 code. The dual 82-b floating-point units have a four-cycle multiplier accumulator (MAC) latency and are fully bypassed with each other. In combination with the 14 ported, 128×82-b register file and other miscellaneous floating point (FP) support, these units are 9 mm $\times$ 2.2 mm. The prevalidated four-port 16-KB L1D cache is tightly coupled to the integer units to achieve the half-cycle load [1]. As a result, the less latency-sensitive FPU directly interfaces to the L2D cache [4] with four 82-b load ports (six-cycle latency) and two 82-b store ports.

The front-end instruction fetch pipe stages are decoupled from the backend stages via an eight-bundle queue. The backend pipeline stages use a scoreboard to resolve register dependencies; this pipeline is a fully stalled design and does not rely on a replay or flush mechanism (Fig. 2). The same prevalidated, single-ended cache technology that enables the half-cycle latency L1D cache is leveraged to improve instruction fetch and branch resteer. Each set of six instructions (two bundles) stored in the 16-KB L1I cache is accompanied by branch target address and branch prediction information. The data is read out of the cache in the first half-cycle. In the next half-cycle, the prediction information is examined, and if the branch to which the stored target corresponds is predicted taken, then this address steers the instruction pointer mux for the next instruction fetch. Thus, correctly predicted taken branches incur no pipeline bubbles, which provides up to 33% more instruction fetch bandwidth with an assumption of one predicted taken branch every six bundles (i.e., every third cycle or 18 instructions). A 12K entry branch history cache and a 16K entry 2-b pattern history table back up the 1K set of histories. The result is 95.2% prediction accuracy for TPC-C (the optimization target for the branch unit design).

The challenge of fitting the large caches, numerous execution resources and datapath busses (Fig. 3) within the die size limit for this process put extreme pressure on the use of all six layers of interconnect. A careful balance was maintained between the allocation of metal to the power grid for supply integrity and to signals for improved density and performance (Fig. 4). In some sections of the chip, we routed 16 signal wires per bit slice before interspersing supply wires. Noise and inductance issues were managed either by active noise cancellation [3] or noise-resistant circuits and hand-generated layouts. The clock

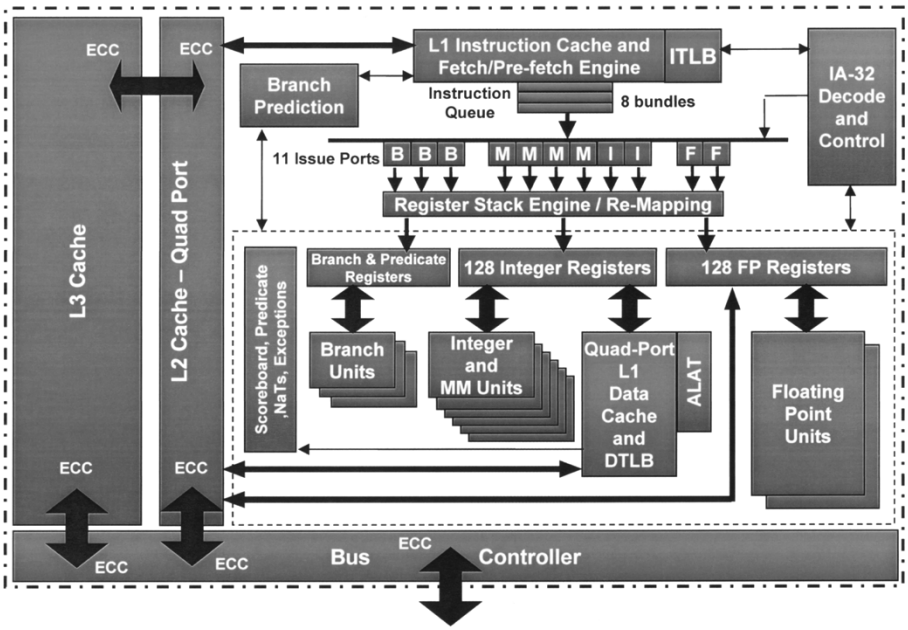


Fig. 3. Itanium 2 block diagram.

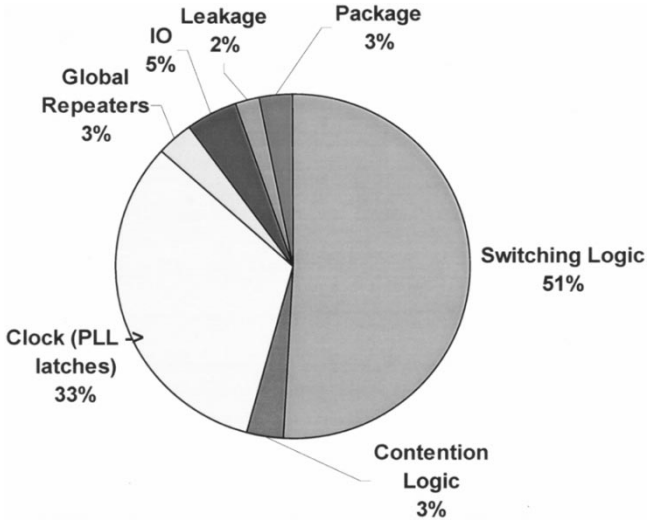


Fig. 4. Processor statistics: power breakdown from 130 W. Over 18 200 global nets connecting up units at the top level. ~700 nF explicit bypass cap. Power grid uniform in M5/6—consumes 22% of M5 and 62% of M6. Bus I/Os distributed on four stripes in the L3 cache. 25M logic transisors, 221M total.

system consumed about 1/3 of the total power (Fig. 4), which is quite good for a high performance processor. This reduction can largely be attributed to the balanced H-tree versus full grid distribution and state versus phase based design for fewer latch elements.

Referring to Table I, where the Itanium 2 processor is compared to other enterprise class processors in 0.18-μm technology, it can be seen that while the Itanium 2 processor provides the shortest pipeline, the densest cache system and highest execution and memory bandwidth, it also maintains a competitive frequency and core area (where core is defined as all the execution resources including the lowest level cache). This is a direct result of the many advanced circuit design techniques and methodologies discussed in this paper.

TABLE I
ITANIUM 2 VERSUS OTHER 64-B 18-μm PROCESSORS

	Itanium 2	USIII	EV7	Power 4 (2 cores)
Frequency (GHz)	1.0	1.05	1.0-1.2	1.3
Pipe stages (mpb)	**8 (6)**	14 (8)	9 (~11)	12 (~11)
Sustainable Int BW	**6 / cycle**	3 / cycle	4/cycle	3/cycle
FP units	2/cycle	2/cycle	2/cycle	2/cycle
On chip cache	**3.3MB 4 arrays**	96KB 2 arrays	1.8MB 3 arrays	1.7MB 2.5 per core
D Cache read BW	**64GB/s**	16.8	19.2	41.6 / core
Die size (mm^2)	**421**	244	397	400 est.
Core size (no IO, only lowest level caches)	142	206	115	100 est.
Power (Watts)	**130**	75	125	125

[1]Pulse latch excludes scan slave latch. First value excludes pulse generator, second value includes pulse generator.

III. CLOCKING AND LATCHING

A. Clock Distribution and Management

The Itanium-2 central processing unit (CPU) has a three-level clock distribution system, as described in [5], consisting of: a PLL with primary driver; a second-level clock buffer (SLCB) driving a route called SLCBO; and a local, third-level buffer called a *gater*. The first and second level clocks are 50% duty cycle and distributed as balanced H-trees. The gater-level distribution consists of local routes to latches and circuits with carefully controlled load and resistance–capacitance (*RC*) delays.

Gaters are high-gain buffers, variants of which generate 24 types of true and complement phase and pulse clocks from SLCBO. Standard gater waveforms are shown in Fig. 5. CK is 180° out of phase with respect to SLCBO, with pulsed clock (PCK) and pulsed not clock (PNCK) at the rising edge of the corresponding phase clocks. Other gater waveforms include delayed precharge clocks (PRECK) for domino logic, wide

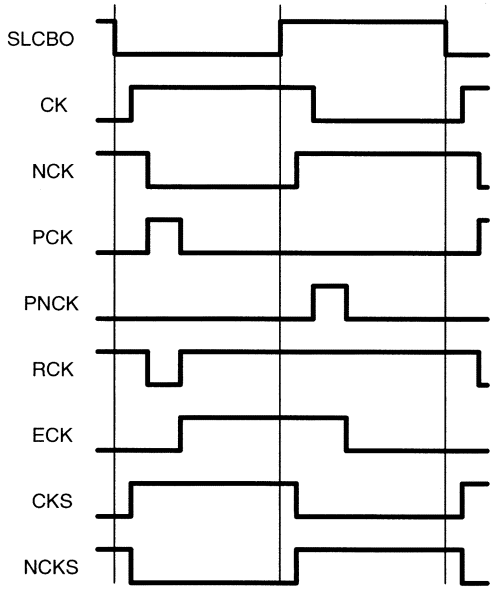


Fig. 5. Gater clock waveforms.

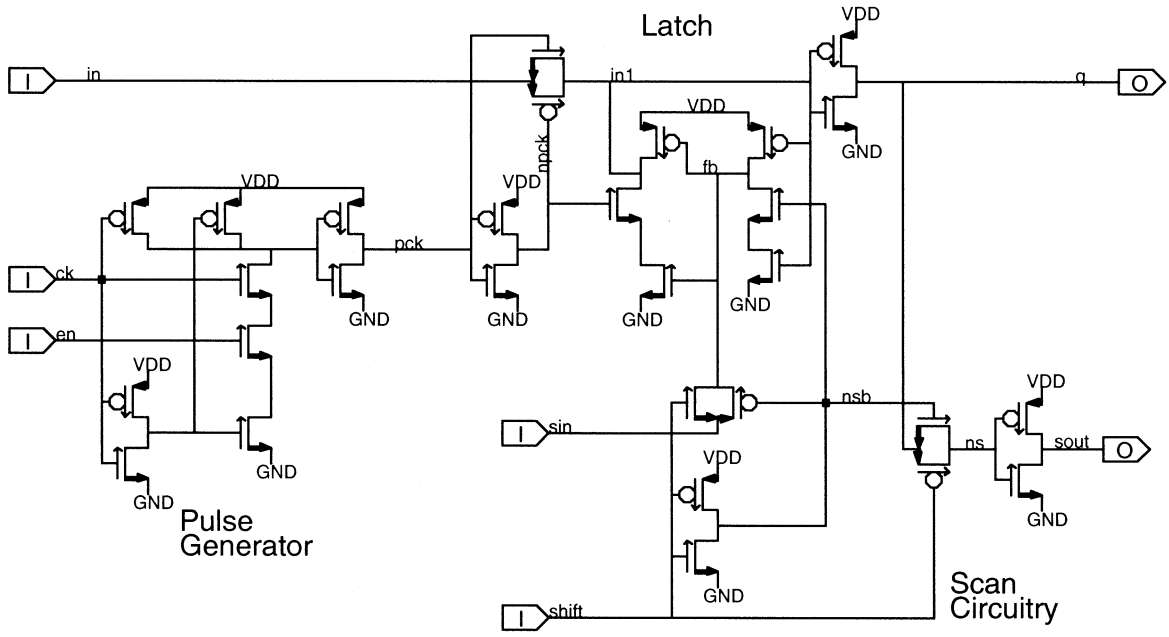


Fig. 6. Itanium 2 pulse latch schematic.

pulses (WPCK) for enhanced cycle stealing and RCK/ECK for dynamic latching (see *dynamic clocking/latching*). Gaters consist of a predriver, which generates a specific clock waveform and an output driver tuned to drive fixed-increment loads.

Gater-level clock nets are typically less than 1000 μm in length and on the Itanium 2 processor, were designed and routed by individual block owners. Tight control and analysis of gater clocks was critical to managing clock skew. Most importantly, no buffering of clocks by nonstandard gaters was allowed, thereby controlling clock delay mismatch and PVT shift, which increase skew. Gater loads are managed to $+/-15\%$ of the nominal gater load spec and gater-to-latch *RC* delay held to less than 20 pS. Maximum allowable coupling to gater-level clocks is 10%. Clock timing specs were checked by parasitic *RC* extraction and SPICE simulation of over 18 000 gater-level routes.

Initially, we anticipated that pulse waveforms would be easily distorted and skewed by typical phase-clock routing techniques. However, following the strict *RC* delay and coupling guidelines described above, we found that simply adding extra gater drive allowed pulse clocks to be routed over the same regions and load ranges as phase clocks without undue pulse distortion.

Clock skew due to data-dependent gate load variation was also less problematic than expected. While CMOS gate capacitance varies by up to 11 times in this 0.18-μm process under various data-dependent bias conditions, typical gater clocks have much lower gate load variation. Field effect transistor (FET) bias conditions in common latching, gating, and dynamic circuits limit gate cap variation to a practical range of about five times. In addition, Itanium 2 pulse latches—the most common pulse clock receiver—have clock loads varying by only a frac-

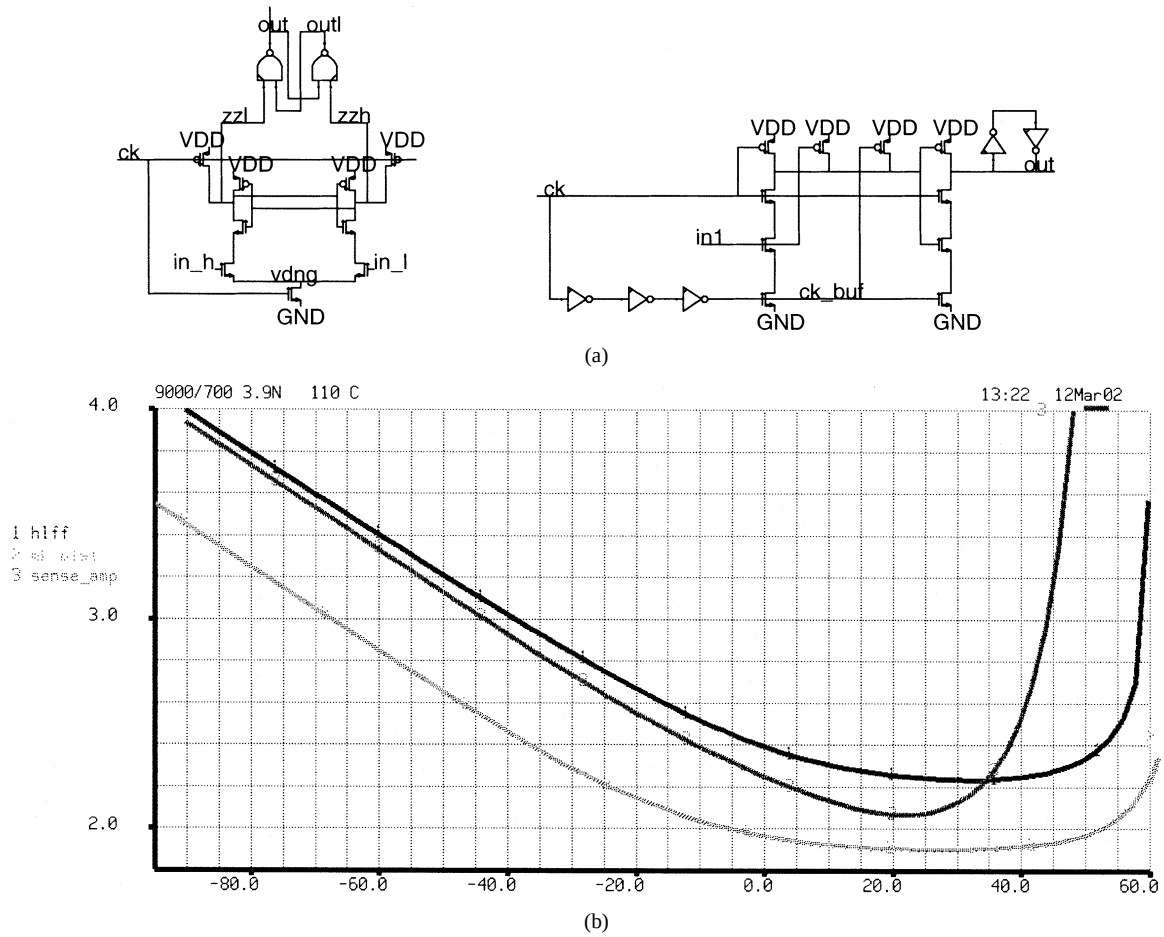


Fig. 7. Transparency window comparison of edge-triggered latches. (a) Two contemporary edge-triggered latches: sense amp latch (top) [8] and HLFF (bottom) [7]. (b) Transparency window simulation results for McKinley pulse latch, HLFF latch, and sense amp latch, 0.18-μm process, 1.2 V. Plot of D-Q delay (FO4 inverter equivalents) versus input-to-clock arrival time.

tion of worst case. Thus, data-dependent clock-load variation was a small skew component, allowing simpler skew and timing analyses than described in [6].

The second-level route consists of 33 separate SLCB zones, each containing a balanced H-tree feeding gater groups. SLCB load mismatch between zones and imbalances in the individual H-tree routes are principal sources of clock skew. The routing of each SLCB zone was performed with a custom clock routing/analysis tool called EZROUTE. Second level routing to each gater farm is balanced, with an extracted RLC delay mismatch simulated at less than 10 pS in SPICE.

B. Static Latching Methodology

Goals for the Itanium 2 static latching methodology are similar to other high-speed designs [17]: *state-based* (area, timing overhead), *high data-transparency* (skew tolerance, cycle stealing), *low latch delay* (cycle time), and *fully scanable latch elements* (testability). To meet these goals, pulse latch (Fig. 6) consists of a transparent pass-gate latch (Tlat) opened by a nominal 125-pS wide pulse. Tlats are clocked with either a local pulse generator driving only one Tlat (as shown), or by a pulse gater driving many Tlats. Pulse clocks provide a relatively wide transparency window, which allows cycle stealing and skew tolerance while avoiding the long hold times of phase-transparent clocking. 95% of all static latching on the Itanium 2 processor uses pulse clocking.

As shown in Fig. 6, Tlats have pass-gate inputs and can be driven by any logic gate provided that drive strength requirements are satisfied. The library provides latches with both pass-gate latches and inverter-isolated inputs for design flexibility. The scan slave latch incorporated into the Tlat saves area by omitting static feedback. Over 85% of all static latches incorporate scan functionality.

Fig. 7(a) shows two other edge-triggered latch designs found in contemporary CPUs: the “hybrid latch flip-flop” (HLFF) [7] and an EV6-style sense amp latch [8]. The latch circuits were sized for wide transparency windows and output loads similar to the Itanium 2 Tlat (in a 0.18-μm process), then simulated in SPICE. Simulation results are plotted in Fig. 7(b) and latch details are listed in Table II. The Itanium 2 pulse latch, with a 125-pS pulse width, has the widest transparency window (58 pS) and the lowest delay (1.9 FO4 inverter equivalents) of the group. Though its internal pulse width is also 125 pS, the HLFF pulse latch transparency window is approximately 31% smaller and its minimum D-Q delay is 15% slower, than the pulse latch. In addition, the HLFF has a weakly held output node; in many applications it requires another buffer stage, increasing its D-Q delay. The EV6-style sense amp latch has the smallest transparency window at 26 pS. The sense amp

TABLE II
ITANIUM 2 VERSUS OTHER 64-B 18-μm PROCESSORS EDGE-TRIGGERED LATCH CHARACTERISTICS IN A 0.18-μm PROCESS

Latch Type	Transparency Window (pS)	Min. D-Q Delay (in FO4 inv delays)	Total Fet Width (um)[1]	Est'd Layout Area (um^2)[1]
Itanium 2 Pulse Latch	58	1.9	20.9 / 24.4	39.9 / 55.4
HLFF Latch	40	2.2	32.8	61.9
Sense Amp Latch	26	2.1	38.5	58.4

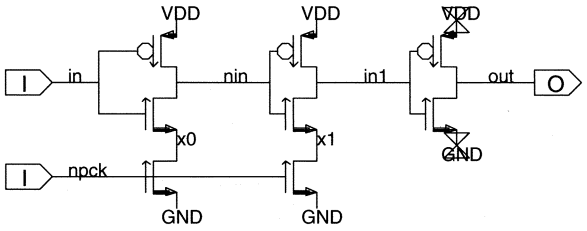


Fig. 8. Clocked deracer schematic.

latch delay and transparency window are proportional for this circuit, making it impractical to tune for greater transparency.

C. Hold-Time Analysis

Itanium 2 min-time verification had three parts. First, designer guidelines specified how many typical gate delays were required between the various source/sink clocking/latching combinations. Next, a fast gate-counting tool provided early feedback to designers on how well they met these hold-time guidelines. Third, static timing analysis was run on layout netlists with extracted parasitics.

By tuning clock routes to a fixed delay spec, we were able to use "ideal" clock waveforms during static timing analysis and thereby decouple clock design from path design. This greatly improved designer productivity and reduced the design thrashes common during min and max time path fixes. Likewise, latch setup/hold times were checked to a spec and specified for static timing, further decreasing design dependencies and increasing accuracy.

On the Itanium 2 processor, *deracer* circuits were used to fix min-time problems. As shown in Fig. 8, *clocked deracers* use the receiving latch pulse clock to briefly block incoming paths while the receiving latch is open. The clocked deracer adapts its delay to receiver pulsewidth distortion by remaining opaque longer and provides a significant advantage over fixed delays: their impact to max-time paths is only 60% of the hold-time benefit they provide.

IV. DOMINO METHODOLOGY

Domino circuits play a critical role in the design of the processor. The high speed of domino logic combined with the ability to implement complex logic functions in a single gate makes it an important tool in achieving the processor goals of low pipeline latencies and high frequency operation. Domino design is inherently phase-based, with latching required in each phase in order to preserve pipeline state. This has the disadvantage of making it potentially more sensitive to clock skew and latch insertion delays compared with cycle-based design. Incorporating testability into domino circuits also presents significant challenges, but is required in order to achieve program test coverage goals.

In designs such as the HP PA-8000 [9], [10], these limitations are partially addressed through the use of delayed clocking to provide time borrowing, combined with a self-timed latching structure [11]. Although this methodology is skew tolerant and incorporates test features, it has a high overhead and is not well suited to general use. More recent designs use opportunistic time borrowing (OTB) [12] to eliminate the need for the explicit latch; however, OTB does not provide for scan controllability or simple transitions into static logic. These shortcomings are addressed by adding a simple "bolt-on" circuit, called a dynamic latch converter (DLC). Through the addition of the DLC, a conventional domino gate becomes a latched domino cell. The DLC (discussed in detail below) meets the goals of eliminating insertion delay, increasing skew tolerance and adding testability.

A. Domino Pipeline

An example domino pipeline is shown in Fig. 9. The pipeline begins with an entry latch, which is used to convert static logic signals to the monotonic signals needed by the domino domain. The entry latch is essentially a fully held domino gate with a pulsed evaluate clock, allowing low setup and hold times. Although arbitrary logic functions may be implemented in the entry latch, evaluation must complete within the clock pulse width, which limits the complexity of functions. This latch is shown in Fig. 14 and is described in detail later in this paper.

A typical domino phase consists of up to seven domino logic gates, the last of which is a DLC. The outputs of a DLC pass data transparently during the evaluate phase and are held throughout the precharge phase. This allows DLC outputs to be used as inputs to the domino gates of the next phase or as inputs to static logic. Alternate latching structures, such as self-timed zero/one catching latches, are also supported.

B. Domino Clocking

Two clocking schemes are used for domino logic on the Itanium 2 processor. The first uses a pair of overlapping phase clocks, designated CK and NCK (Fig. 5). These clocks allow for several gates worth of time borrowing between phases and are quite easy for the designer to implement; however, potential

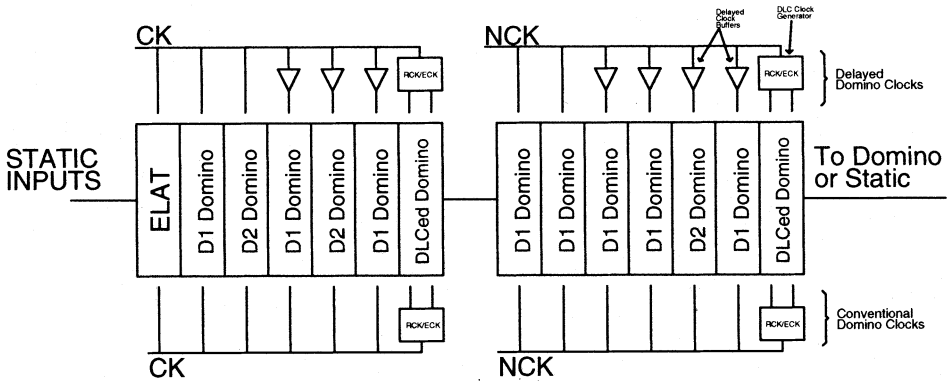


Fig. 9. Representative domino pipeline.

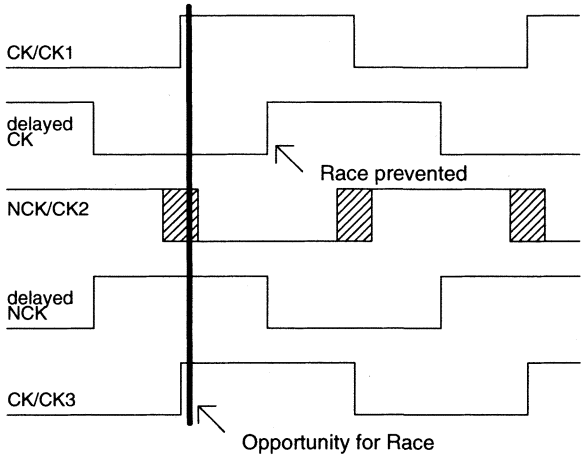


Fig. 10. Clocks for custom domino logic.

races require significant analysis, as the overlap time is magnified by clock skew, exacerbating the race problem. Races are resolved by inserting delay buffers in the logic path.

For more regular domino data paths, an alternative clock scheme is used to reduce the overhead of race management, while still providing significant time borrowing. Pictured in Fig. 10, it borrows the idea of delayed clocks from OTB [12], [13]. It is evident from Fig. 11 that data early in CK1 can race through the end of CK2 and into CK3. By clocking at least one CK1 gate with a delayed clock, the data is prevented from racing through CK2 into CK3. While race analysis isn't eliminated, the increased skew tolerance results in nearly correct by construction designs, which need minimal repairs.

A second advantage of the delayed clock system is reduced global clock load. As less than 30% of the domino gates are driven by the undelayed clock, the majority of the clock load is buffered from the clock gaters, reducing the number of gaters required and the load presented to the global clock routes. The generation of delayed clocks is made simple by using a library of load-matched noninverting buffers.

C. Domino Logic

Domino logic [Fig. 11(a)] is widely used for high-speed or large fan-in logic [14]. Typically, an evaluate N channel FET (NFET) is added between the logic network and ground to eliminate excess current during precharge and permit the use of a smaller precharge P channel FET (PFET). The presence of an evaluate NFET results in a D1 gate, while lack of the NFET results in a D2 style gate, which can be faster and use smaller logic FETs. Both include a weak keeper PFET to maintain high levels on the storage node, which may droop due to the leakage current of the NFETs, noise events on the inputs, or charge sharing inside the logic network. The D2 gates reduce clock and signal loads, saving power; however, care must be taken to appropriately resize the precharger to overcome the increased short circuit current and to meet precharge timing. It is imperative that the inputs of the NFET network discharge quickly to minimize the short circuit current and to reduce precharge time. This is addressed by requiring each D2 gate to be preceded by a D1 gate and by slightly delaying the precharge clock edge on the D2 gates. Due to increased engineering effort, the use of D2 style gates is limited to certain critical paths.

Increased leakage currents and greater noise sensitivity are prevalent in submicron processes and force a modification of wide fan-in gates to reduce the number of parallel pull-down paths. Fig. 12(a) shows an eight-input domino multiplexer. The eight parallel paths make the gate more susceptible to NFET leakage and multiple noise events. This issue is alleviated by splitting the pull-down network and replacing the output inverter with a NAND gate [Fig. 12(b)]. The pull-down networks can be smaller and faster, as the storage node capacitance is reduced. Finally, the P:N ratio of each domino gate's output is restricted to a 5:1 ratio, which maintains a reasonable trip point, thus improving noise immunity.

D. Dynamic Latch Converter

A dynamic latch is created by adding the DLC "bolt-on" to an existing domino gate. The "bolt-on" includes a clock generator [Fig. 11(b)] and the additional NFET holders and scan logic shown in Fig. 11(c). The resulting circuit performs the same function as the original gate, while providing latch and scan functionality, without unnecessary insertion delay. Unlike typical domino gates, the precharge and evaluate transistors of the DLC are clocked by separate clocks, designated RCK and ECK respectively [Figs. 5 and 11(d)]. The RCK pulse, at the beginning of the evaluate phase, precharges the gate. The rising transition of ECK causes the DLC to enter the evaluate phase, which extends into the next phase to allow time borrowing across clock phases. The latch's output becomes valid while the ECK clock is high and holds its value throughout the following clock phase, allowing it to drive either dynamic or static logic.

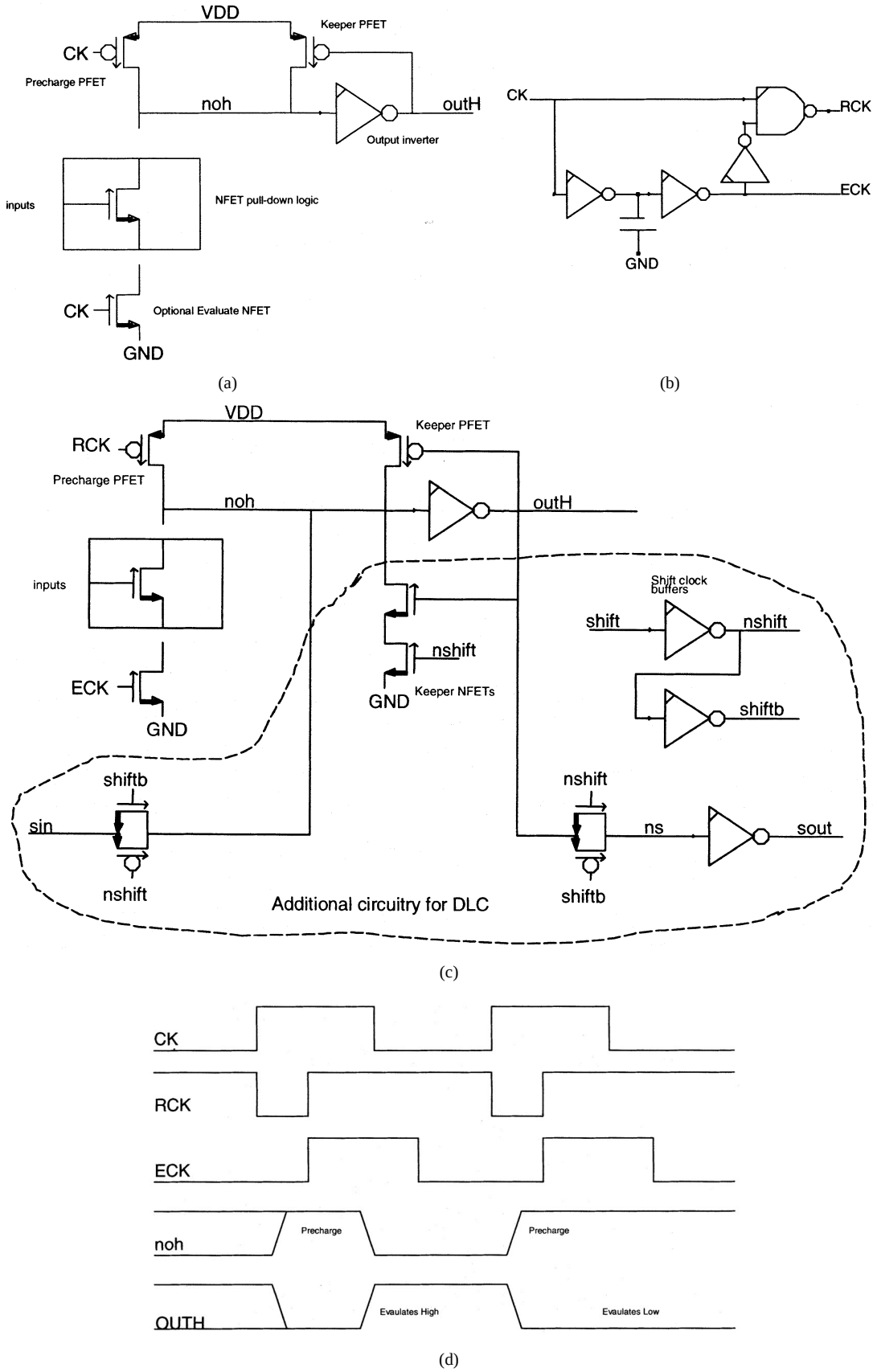


Fig. 11. (a) Typical domino gate. (b) DLC clock generator. (c) Domino gate with DLC logic. (d) Clock and data waveforms for a DLC.

The DLC also allows scan on all dynamic phase boundaries, greatly increasing test coverage and debug capability. Different DLC scan circuits are used for library and custom designs. The library's scan [15] enjoys the advantages of a single shift clock

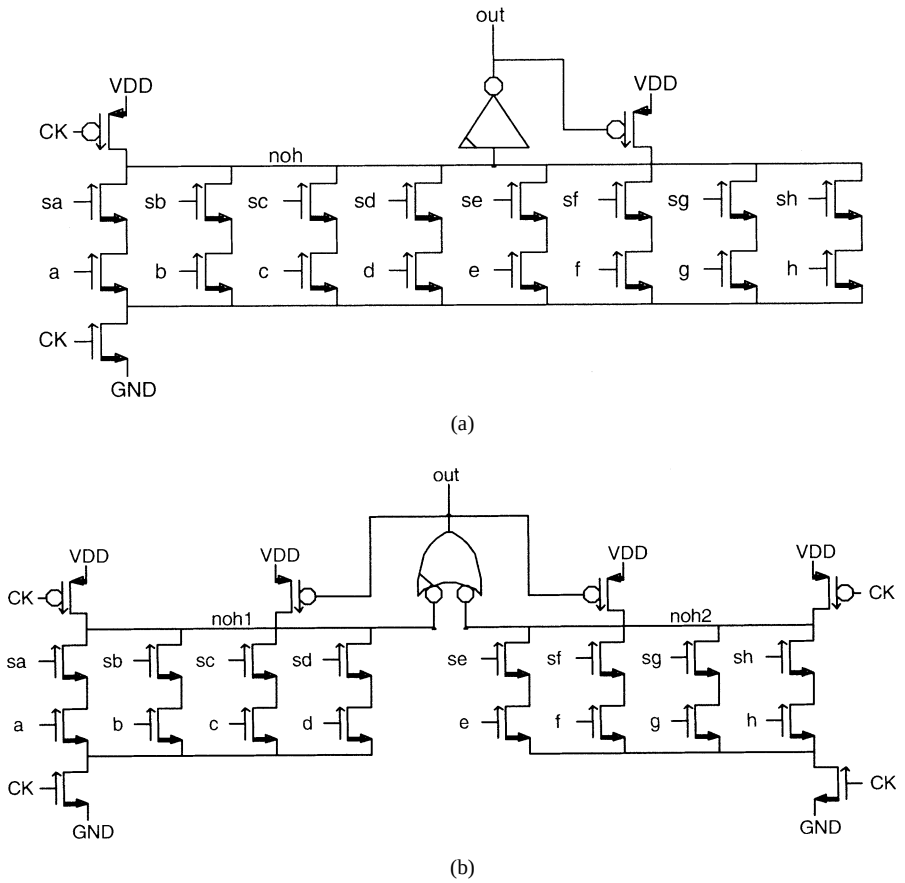


Fig. 12. (a) Eight-input domino multiplexer.(b) Eight-input multiplexer with split storage node.

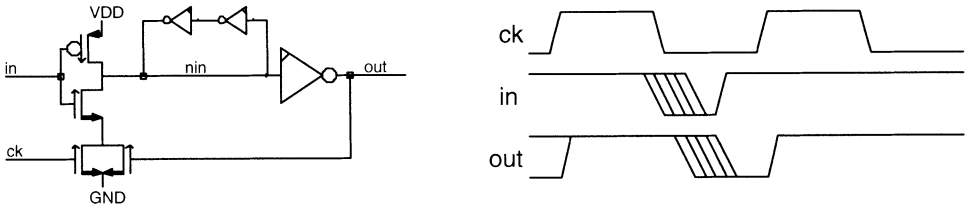


Fig. 13. Zero-catching latch and associated waveforms.

and no individual circuit tweaking. The custom dynamic scan uses a pair of traditional nonoverlapping clocks and requires a limited amount of circuit sizing; however, it is more easily incorporated into the wide range of latch sizes found in full custom circuits. It also exploits the nature of dual-rail domino logic by using the logical high and low storage nodes as the scan latch's master and slave.

E. Hybrid Latching Circuits

In addition to the standard DLC, alternate latching structures are sometimes needed. These structures include the zero/one catcher and the zero catching DLC. The zero-catching latch, shown in Fig. 13, allows for self-timed capture of a monotonic active low input. The circuit is fully transparent during the precharge phase, passing either a logic one or zero. During the hold phase, the output is latched, but a falling input during this time period will be passed to the output. This circuit is used when the timing relationship of a monotonic input relative to the clock is uncertain such as long routes with significant *RC* delay. The circuit has further advantages of low insertion delay, reduced noise sensitivity, and testability. The circuit can also be used to interface from the domino domain into the static domain. A one-catching latch has similar characteristics, but is used to latch monotonic active high signals. The zero-catching DLC is a combination of a conventional domino gate and a zero-catching latch. Its outputs are similar to those of a DLC, but it does not require a pulsed precharge clock, making the structure useful for signals requiring a longer precharge time.

V. Tricky Circuit Methodology

In order to solve the numerous timing and area challenges presented by the processor implementation, numerous critical circuits known as "tricky circuits" are incorporated. These include pulsed-dynamic circuits, self-timed gates and contention based logic. Proliferation of tricky circuits is strictly controlled and each instance has electrical verification performed "in context" of its usage. Many of these involve the interface between static and dynamic logic.

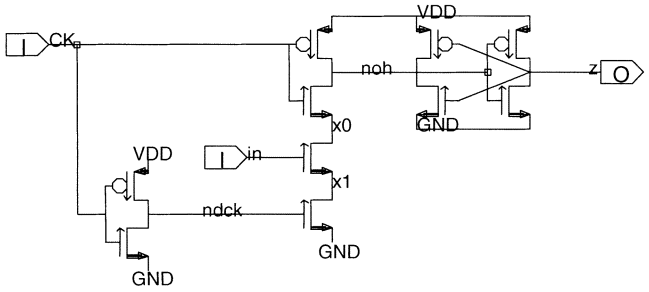


Fig. 14. Elat circuit.

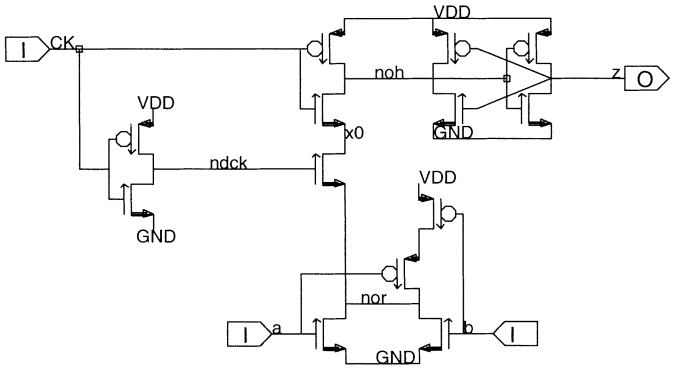


Fig. 15. OR Elat circuit.

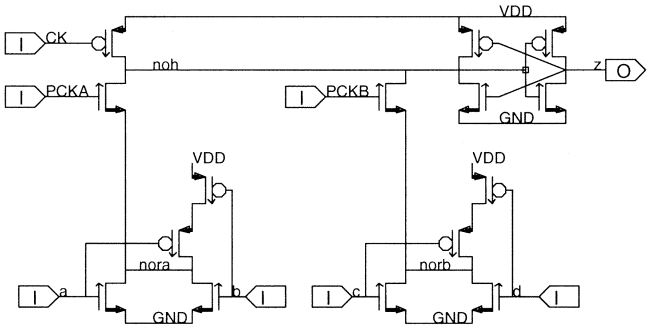


Fig. 16. OR Elat circuit.

A. Static-Dynamic Interface

In the Itanium 2 processor, a large effort went into to the design of a seamless interaction between static and dynamic circuit families. In past CPUs, this interface has been a frequent source of failures during the silicon debug portion of the design. Our circuit methodology provides a number of standard circuit topologies that provide a low-overhead transition from static to dynamic logic families. The skew penalty must be paid at this interface, so low setup and hold-time requirements are a necessity. The goal is to give designers a large amount of flexibility in this transition while not introducing problems, such as output glitches or hold-time issues, that might only be caught during silicon debug.

The entry latch (Elat) captures a static input and converts it into a single rail monotonic dynamic signal. The Elat structure, shown in Fig. 14, has a number of important attributes. It has a very short clock to Q, short setup time, and the hold-time requirements are not excessive. A single-phase clock is used to capture the static signal on the rising edge of CK. Controlling the falling edge of the ndck signal minimizes the hold-time requirements. The period where both CK and NDCK are high is defined as the evaluate pulse; its width is tightly controlled to minimize hold-time problems.

The Elat structure has a negative setup time for the 0 to 1 transition and has a longer hold-time requirement on the low level for the input. The NFET pull-down stack is designed to robustly discharge the storage node and to overcome the PFET holder during the evaluate pulse width.Static logic functions can be easily incorporated into the front end of the Elat. This embedded logic function helps ease the timing requirements of many critical paths. In addition, the front end can be implemented with a static CMOS gate, making the circuit more resistant to noise on the inputs. This is particularly useful, as receiving a static signal into an NFET only structure is a recipe for a noise-corrupted event. This complementary logic function on the front end has the added advantage of negating charge-sharing events that may occur in the NFET stack. Fig. 15 illustrates a two input NOR gate front end used to create an OR-Elat.

An alternative design for the static-to-dynamic transition is widely used in the integer register file; this structure is called pulsed dynamic and is illustrated in Fig. 16. The key difference between this device and the standard Elat used on this processor is that the pulse clock is generated external to the circuit. The disadvantage of having two clocks routed to the circuit is overcome by the increased flexibility that this design offers. The ability to have multiple NFET pull-down stacks connected to differing pulse clocks, with separate enable functionality, is widely exploited in the integer register file. The NFET stack in pulsed dynamic is inherently one FET shorter than the standard Elat. This enables a smaller input loading and more complex front-end logic functions. The combination of smaller input loading with the external pulse clock generator enables a much higher gain out of the pulsed dynamic circuit as compared to the standard Elat.

As with the standard Elat design, the NFET stack must be robustly designed to discharge the storage node and to overcome the PFET keeper with the pulsed dynamic design. To help mitigate charge sharing, traditional CMOS logic front-ends are incorporated into a number of designs.

There are a large number of register file structures and random access memory (RAM) arrays inthe Itanium 2 processor; as with any large CPU, die area was at a premium. This led to the design of a very small dynamic decoding structure called an *annihilation gate*. These structures are similar to the standard Elat design in that they are latching and have a monotonic output; however, the annihilation gate enables a NOR logic function. A typical circuit configuration is shown in Fig. 17.

The annihilation gate (or "flying NOR" or "cancellation gate") is an elegant way to perform high fan-in dynamic AND computations much more quickly and compactly than traditional static or dynamic logic. In addition, the latching structure enables a very low hold-time requirement on the inputs. By design, it has very fast clock to Q and its monotonic output enables it to be used as a static to dynamic interface circuit. The annihilation gate was used extensively in decoders where the low input load

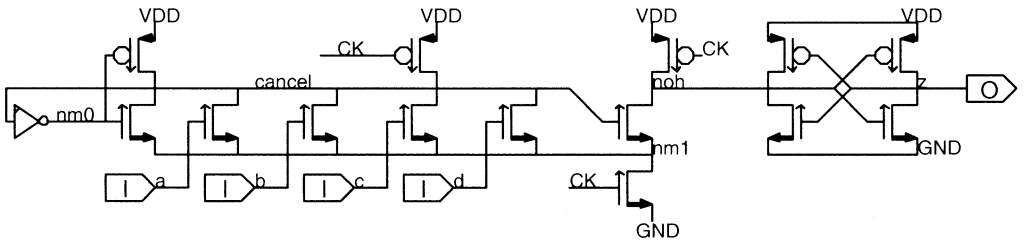


Fig. 17. Pulsed dynamic logic gate.

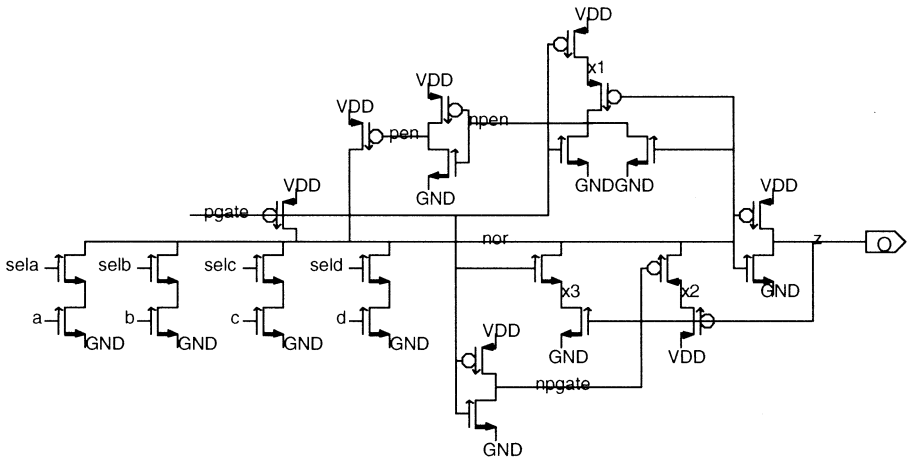


Fig. 18. Annihilation gate circuit.

and high gain structure are desired. A typical decoder is 6-b wide. A traditional dynamic decoder requires a seven NFET stack to accomplish the required AND function, or two levels of logic. Since the annihilation gate inputs are connected to a single NFET pull-down, NFET sizes are significantly decreased over an equivalent dynamic design.

The challenge when designing an annihilation gate is to ensure that the droop on the evaluation node during the cancellation does not cause a glitch on the output. The delay from the clock to the output transition is optimized for these gates. With an NFET-only receiving structure, the engineers have to be careful to ensure that the inputs are adequately shielded to protect against noise events. When using the annihilation gate as a decoder, there is the added concern that the cancellation node of the entries that are not evaluating will couple back to the input signals. This coupling can degrade the input signal value and cause an erroneous evaluation.

B. Pseudodynamic Logic

The desire for the small size and wide fan-in functions provided by dynamic logic, combined with the constraints of timing and circuit interfaces, led to another breakthrough in circuit design that was highly leveraged across the Itanium 2 processor. This structure is called pseudodynamic and is shown in Fig. 18. Pseudodynamic devices are similar to the traditional pseudo-nMOS designs, with the exception of a feedback network. Pseudodynamic implementations garner most of the benefit of dynamic logic while maintaining the ease of use and design associated with static gates. Although these gates are higher power than traditional static designs due to the inherent drive fight, the low input capacitance, small area required by the NFET-only evaluate structure and high fan-in that these devices achieve more than make up for their costs. In addition, significant power reductions can be realized by clocking the PFET pull-up (pgate input).

A pseudodynamic gate achieves a shorter delay than a traditional pseudo-nMOS structure, as the circuit optimizes the PFET pull-up depending on the current state of the shared node (NOR). When the shared node is at a high voltage the strong pull-up structure is disabled. This shortens the delay for a falling edge on the shared node (NOR). When the shared node is at a low voltage, both PFETs are enabled. This speeds up the rising edge of the shared node. The impact of this design yields a 15%–20% delay improvement over traditional pseudo-nMOS.

There are a number of design considerations that must be accounted for when implementing pseudodynamic circuitry. As the shared node is subject to high unidirectional current loads, the artwork for the gate must be robustly designed to meet all the self-heating and electromigration requirements of the process. Close attention must be paid to the ratio of the receiving inverter trip point relative to the minimum value ($V_{\rm ol}$) that is driven on the shared node when both PFETs are enabled and one of the NFET pull-downs is active. In addition, the feedback delay and response to input glitches are carefully checked. Coupling noise and current–resistance (*IR*) drop to $V_{\rm ol}$ and power supply issues are accounted for in the design of all the pseudodynamic.

VI. Conclusion

The Itanium 2 processor microarchitecture breaks new ground in integration, instruction parallelism, latency, and efficiency, all of which are founded on a rich set of carefully

designed and managed circuit families and methodologies. Clocking and latching circuits were carefully controlled and analyzed to reduce skew and eliminate hold-time problems. The static and domino methodologies address key issues of skew tolerance, insertion delay, clocking complexity and testability. They also provide low overhead solutions for the domino-static interface and are fully compatible with either cell-based or custom design. Elats, DLCs, pulsed dynamic, annihilation, and pseudodynamic gates are in the class of "tricky circuits" and were carefully controlled and analyzed to ensure that each instantiation is a robust design. These methodologies and circuits, essential to the Itanium 2 processor's success, have been shown to be robust through post-silicon analysis.

This combination of aggressive circuits and microarchitecture enables McKinley to achieve industry leadership 64-b performance of 810 SPECint2000 and 1356 SPECfp2000 at 1 GHz.

ACKNOWLEDGMENT

The authors recognize the extraordinary efforts of a committed design team from Intel and Hewlett Packard in making the Itanium 2 processor a success.

REFERENCES

[1] D. Bradley, P. Mahoney, and B. Stackhouse, "The 16kB single-cycle read access cache on a next-generation 64b Itanium microprocessor," in *ISSCC Dig. Tech. Papers*, Feb. 2002, pp. 110–111.

[2] D. Weiss, J. Wuu, and V. Chin, "The on-chip 3MB subarray based 3rd level cache on an Itanium microprocessor," in *ISSCC Dig. Tech. Papers*, Feb. 2002, pp. 112–113.

[3] E. Fetzer and J. Orton, "A fully-bypassed 6-issue integer datapath and register file on an Itanium microprocessor," in *ISSCC Dig. Tech. Papers*, Feb. 2002, pp. 420–421.

[4] R. Reidlinger and T. Grutkowski, "The high-bandwidth 256KB 2nd level cache on an Itanium microprocessor," in *ISSCC Dig. Tech. Papers*, Feb. 2002, pp. 418–419.

[5] F. Anderson, S. Wells, and E. Berta, "The core clock system on the next generation Itanium microprocessor," in *ISSCC Dig. Tech. Papers*, Feb. 2002, pp. 146–147.

[6] D. Bailey and B. Benschneider, "Clocking design and analysis for a 600-MHz Alpha microprocessor," *IEEE J. Solid-State Circuits*, vol. 33, pp. 1627–1633, Nov. 1998.

[7] H. Partovi *et al.*, "Flow-through latch and edge-triggered flip-flop hybrid elements," in *ISSCC Dig. Tech. Papers*, Feb. 1996, pp. 138–139.

[8] B. Gieseke *et al.*, "A 600-MHz superscalar RISC microprocessor with out-of-order execution," in *ISSCC Dig. Tech. Papers*, Feb. 1997, pp. 176–177.

[9] C. Heikes and G. Colon-Bonet, "A dual floating-point coprocessor with FMAC architecture," in *ISSCC Dig. Tech. Papers*, Feb. 1995, pp. 354–355.

[10] P. Dorweiler, F. Moore, D. Josephson, and G. Colon-Bonet, "Design methodologies and circuit design trade-offs for the HP PA8000 processor," *Hewlett Packard J.*, Aug. 1997.

[11] J. Yetter, "Universal pipeline latch for mousetrap logic circuits," U.S patent 5 392 423, Feb. 1995.

[12] S. Rusu and G. Signer, "The first IA-64 microprocessor," *IEEE J. Solid-State Circuits*, vol. 35, pp. 1539–1544, Nov. 2000.

[13] D. Harris and M. Horowitz, "Skew-tolerant domino circuits," *IEEE J. Solid-State Circuits*, vol. 32, pp. 1702–1711, Nov. 1997.

[14] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*. Reading, MA: Addison-Wesley, 1993.

[15] D. Josephson, S. Poehlman, V. Govan, and C. Mumford, "Test methodology for the McKinley processor," in *Int. Test Conf.*, Oct. 2001, pp. 578–585.

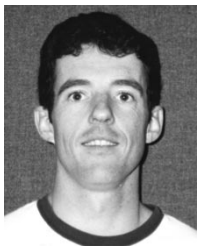
[16] S. Yang *et al.*, "A high performance 180 nm generation logic technology," in *IEDM Dig. Tech. Papers*, 1998, pp. 197–200.

[17] V. Stojansvic and V. G. Oklobdjia, "Comparative analysis of MS latches and flip flops for high performance and low-power VLSI systems," *IEEE J. Solid-State Circuits*, vol. 34, pp. 536–548, Apr. 1999.

[18] G. Lauterbach *et al.*, "UltraSPARC-III: A 3rd Generation 64b SPARC Microprocessor," in *Proc. ISSCC 2000*, 2001, pp. 410–411.

[19] A. Jain *et al.*, "A 1.2 GHz alpha microprocessor with 44.8 GB/s chip pin bandwidth," Proc. ISSCC 2001 and Microprocessor Report, vol. 10, no. 14, pp. 240–241, 2001.

[20] C. Anderson *et al.*, "Physical design of a fourth-generation power GHz microprocessor," Proc. ISSCC 2001 and Microprocessor Report, vol. 13, no. 13, pp. 232–233, 2001.



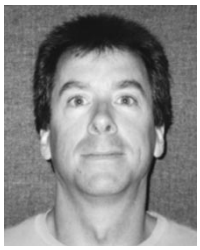
Samuel D. Naffziger received the B.S. degree in electrical engineering from the California Institute of Technology, Pasadena, in 1988. and the M.S.E.E. degree from Stanford University, Stanford, CA, in 1993.

He joined Hewlett Packard Company, Fort Collins, CO, in 1988, where he has since worked on all aspects microprocessor design and implementation. He holds 27 U.S. and foreign patents on processor circuits and architecture and has authored several papers in the field. He is currently the circuit design lead for a next-generation Itanium processor.

Glenn T. Colon-Bonet received the B.S. degree in computer and electrical engineering in 1987 and the M.S. degree in electrical engineering in 1988, both from from Purdue University, West Lafayette, IN.

In 1989, he joined the Colorado Integrated Circuits Division of Hewlett Packard Company, Fort Collins, where he performed circuit design and algorithm development of floating-point division and square-root hardware for the PA-RISC processor family. In 1994, he joined the Engineering Systems Laboratory as a Technical Contributor to participate in the development of the floating-point instruction-set architecture for the Itanium processor family. He was Lead Architect for the Itanium 2 floating-point unit and also participated in the development of domino circuit and test methodologies. In 1998, he became Project Manager for the Itanium 2 floating-point unit. He is currently a Project Manager in the Microprocessor Technology Laboratory, managing the integer/multimedia units, as well as being a Test Manager for a future Itanium processor family product. He currently holds 12 U.S. patents.

Mr. Colon-Bonet organized the 15th IEEE Symposium on Computer Arithmetic as Finance/Local Arrangements Chair.

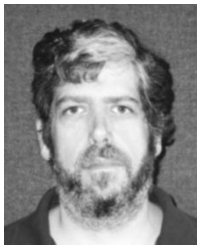


Timothy Fischer received the M.S. degree in computer engineering from the University of Cincinnati, Cincinnati, OH, in 1989.

He was withDigital Equipment Corporation, Hudson, MA, from 1989–1998, designing Alpha and VAX microprocessors, including the EV5 floating-point and EV6 instruction-issue units. In 1998, he joined Hewlett Packard Company, Fort Collins, working on L2 cache design and global circuit methodology on the Itanium 2 microprocessor. He is now involved in circuit methodology and clock design for a next-generation Itanium microprocessor.

Reid Riedlinger received the M.S. degree in electrical engineering from Montana State University, Bozeman, in 1993.

He then joined Hewlett Packard Corporation, Fort Collins, CO, where he worked on a number of PA-RISC implementations on cache and TLB designs, including the L2 cache design and global circuit methodology, and recently completed working on the post-silicon verification of the Itanium 2 microprocessor. He is now involved in circuit methodology and the L1 cache design for a next-generation Itanium microprocessor.



Thomas J. Sullivan received the B.S. degrees in computer science and electrical engineering and the M.S. and D.Sc. degrees in electrical engineering from Washington University, St. Louis, MO, in 1983, 1984, and 1993, respectively.

After working with the McDonnell Douglas Microelectronics Center, St. Louis, MO, in 1985, he was a Research Associate at Washington University from 1986 to 1993. His research included custom VLSI designs for digital signal processing applications and massively parallel architectures. In 1994, he joined Hewlett Packard Company, Fort Collins, CO, where he has been involved in the physical design of both PA-RISC and iA64 processors. Most recently, he has focused on development of floating-point designs.

Tom Grutkowski received the B.S. degree from The Cooper Union for the Advancement of Science and Art, New York, in 1987, and the Master's degree from Georgia Institue of Technology, Atlanta, in 1992.

He has been with Intel Corporation, Santa Clara, CA, since 1994, contributing to the Pentium II, Itanium, and Itanium II design efforts. He has been awarded patents for both arithmetic and cache designs. In addition to the microprocessor design effort, his interests extend to post-silicon debugging and characterization. He is currently working on the next-generation IPF processor.