Volume 08 Issue 01 Published, February 18, 2004 ISSN 1535-864X

Intel® Technology Journal

Intel® Pentium® 4 Processor on 90nm Technology

LVS Technology for the Intel® Pentium® 4 Processor on 90nm Technology

A compiled version of all papers from this issue of the Intel Technology Journal can be found at:
http://developer.intel.com/technology/itj/index.htm

LVS Technology for the Intel® Pentium® 4 Processor on 90nm Technology

Dan J. Deleganes, Desktop Platforms Group, Intel Corporation
Micah Barany, Desktop Platforms Group, Intel Corporation
Daniel Chow, Technology and Manufacturing Group, Intel Corporation
Tom D. Fletcher, Desktop Platforms Group, Intel Corporation
George L. Geannopoulos, Technology and Manufacturing Group, Intel Corporation
Kurt Kreitzer, Desktop Platforms Group, Intel Corporation
Anant P. Singh, Desktop Platforms Group, Intel Corporation
Sapumal B. Wijeratne, Technology and Manufacturing Group, Intel Corporation

Index words: X86 integer core, adder, sense-amplifier, adder, rotator, microprocessor, Low-Voltage Swing, LVS

ABSTRACT

To meet the demands of low-latency integer operations, the Intel® Pentium® 4 processor architecture implements fast integer operations using a 2x frequency core clock. The frequency advances enabled by Intel's new 90nm technology when paired with a 2x frequency multiplier require novel circuit topologies if latency is to be optimized. The discussed solution uses unprecedented levels of small signal random logic to implement a double frequency X86 integer core. This circuit technology, termed "Low-Voltage Swing" (LVS) enables the Pentium 4 processor [1] to take full advantage of Intel's new 90nm technology [2].

INTRODUCTION

Microprocessor performance can be defined as the product of latency and parallelism. Since parallelism has been well exploited in previous microprocessor generations, the integer performance in the Intel Pentium 4 processor architecture is achieved using ultra-low-latency integer operands. The reduced latency when then paired with Hyper-Threading Technology (parallelism) empowers a one-generation-ahead design. Like the preceding Pentium 4 processor designs, the newest member of the family on Intel's 90nm technology enables ultra low-latency integer ops by running the integer core at twice the core frequency of the microprocessor. At today's clock rates, this operating frequency is in and of itself notable. For example, a 3.4 GHz processor would have the integer logic functioning at 6.8 GHz. Such a frequency target is on the low end of 90nm technology capabilities–that is, at the beginning of process life. End-of-life process technology frequency expectations are far higher. In this paper, we describe the implementation of the newest Pentium 4 processor integer logic core using Low-Voltage Swing (i.e., differential small signal) logic. This circuit topology, referred to most frequently as "LVS," is designed explicitly to take advantage of the frequency headroom enabled by Intel's new 90nm technology. In this paper we explain the overall circuit topology, and take you on a walk-through of three core blocks: the Alignment Mux, Adder, and Rotator. A section describing the tools/methodologies for pre-silicon verification necessary for high-volume manufacturing (HVM) is outlined, which includes small signal path tracing, merging dynamic and static timing, and matched layout. Finally, you will see up-to-date post-silicon data demonstrating the integer core running at higher frequencies than any other published X86 integer cores.

® Intel and Pentium are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

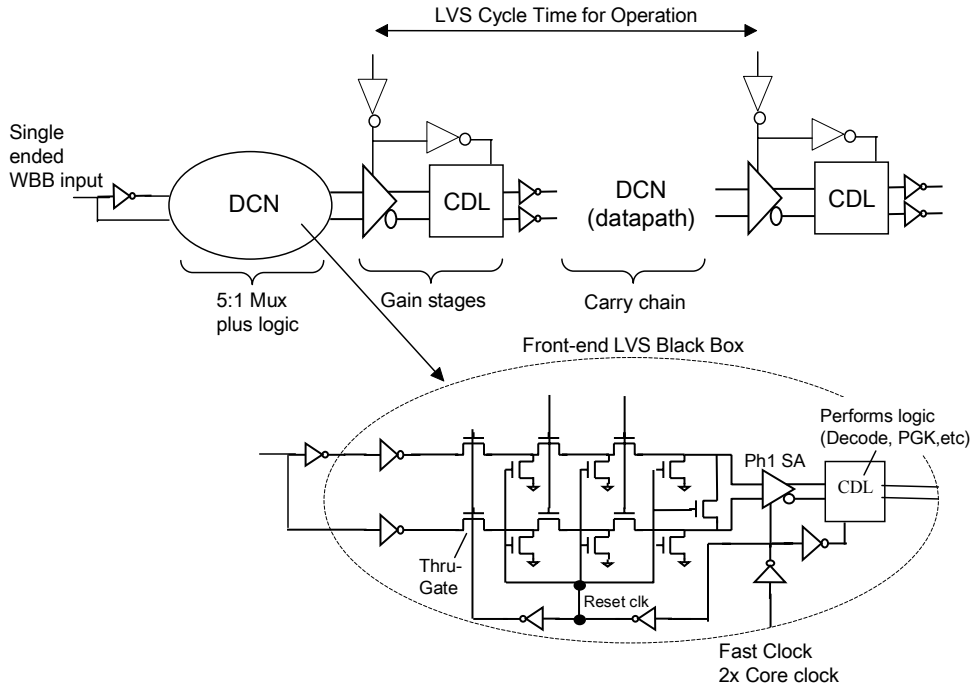

Figure 1: LVS circuit block diagram

LOW-VOLTAGE SWING LOGIC AT INTEL

In 1997 Intel researchers began investigating ways to continue designing the Intel Pentium 4 architecture's 2x frequency integer core on process technologies many years in the future. Looking several generations ahead, they were concerned that the self-resetting domino topologies used so effectively in the original Pentium 4 design would need to be replaced with even faster circuit topologies, if the integer core was to keep pace with the capabilities of future manufacturing technologies. These researchers, led by our co-author Tom Fletcher, determined that large Diffusion Connected Networks (DCN) with multiple inputs and outputs could be used to implement significant logic functions in a single stage. Although such structures are excruciatingly slow at creating standard CMOS voltage levels, it was recognized that by using differential (true and complement) functions, the resulting "small signal" voltages could be differentially sensed and amplified into a "large signal." This circuitry operated faster than even our fastest domino circuits. The delays through two stages of sense and gain were costly, but since the diffusion connected network was capable of doing six to eight stages of logic in a single stage, the overall time to implement a complex logic function was a net performance win over other topologies. Furthermore, it was determined that such networks could readily take advantage of straightforward pass-gate algorithms, such as carry skip addition, to minimize the number of series devices. The resulting differential Low-Voltage Swing (LVS) topology used fewer transistors to implement a given logic function, which lead to an area advantage over traditional static or domino circuits. The topology also promised low-power opportunities at equal frequencies due to reduced voltage transitions. The performance of speed, area, and power wins led to the technology being selected for the next-generation design. During technology development, the LVS circuit topology that delivered the best performance operated like domino logic, with evaluate and reset phases. The higher linear region currents of N-transistors make them the device of choice for DCN pre-conditioned to ground being selected. A P-type sense amplifier senses the differential output of these pass-gate DCNs.

The potential gains of this new topology promised to be significant. However, the design complexity was identified as a major concern. The sheer amount of small

signal logic that would be needed to implement an entire X86 logic core was unprecedented. Consider this: the transistor count of this execution core exceeds that of the entire Pentium Pro design! There were no tools for timing analysis, noise analysis, or logic verification. To minimize the differential and common-mode noise, new layout checks were needed to ensure that custom devices in random logic met analog layout requirements. Pulsed clocks required careful crafting. Clearly, the challenges of implementing an entire Pentium 4 integer core using small signal circuits to implement logic functions would be an extreme challenge. The work began!

LOW-VOLTAGE SWING CIRCUIT ARCHITECTURE

Figure 1 shows the basic topology of the LVS circuits used. An LVS circuit, called the Front End (FE), implements an LVS multiplexer to select among the Write Back Buses and the Source Buses. For simplicity, the DCN diagram shows only two N-pass-gates connected to each node, when in reality, each node would have multiple inputs. The Complementary Domino Logic (CDL) gain stage restores the sense-amplifier output's ratioed voltage levels. The CDL in most cases also implements logic; for example, in the Adder this would be a Propagate, Generate, Kill (PGK) function generator. Also shown is the thru-gate, which acts as a min-delay blocker by gating static data entering the DCN. The thru-gate is controlled by a clock that turns on one inversion after the de-assert of the reset clock. By ANDing the thru-gate clock with a logic signal, one series device can be removed, further improving speed. In Figure 1, the carry chain is collected into a second LVS blackbox that produces the result of a 16-bit add.

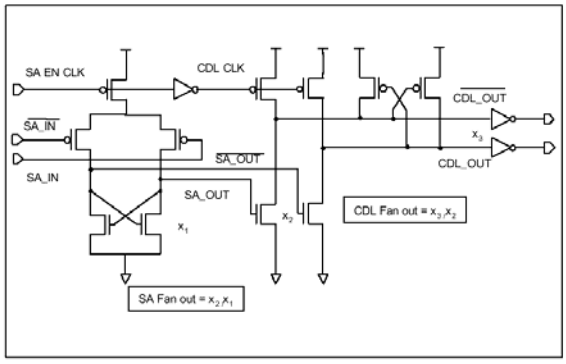

Figure 2: Sense-amplifier and CDL inverter followed by a CMOS inverter

Figure 2 illustrates the ratioed P-type sense-amplifier driving a simple CDL inverter. The reset devices are removed to simplify the diagram. The outputs of the LVS carry-chain DCN connect to "SA IN" and its complementary pin. The timing relationships between the LVS DCN, the sense-amplifier, and the CDL are shown in Figure 3. The sense-amplifier and the CDL are in phase and are controlled by the clocks named "SA EN" clock and "CDL CLK," respectively. The rising edge of "SA EN" clock initiates the reset of the sense-amplifier outputs, and it is immediately followed by the precharge of the CDL outputs. During this time the LVS DCN (carry-chain) is in evaluation and generates a differential voltage at the inputs of the 17 receiving sense-amplifiers of the 16-bit adder. The falling edge of the "SA EN" clock triggers evaluation of these sense-amplifiers. This event is depicted in Figure 3 with a vertical line that intersects the 50% transition point of the falling "SA EN" clock. In this example, it can be seen that at the sense-amplifier trigger point, the input differential is approximately 344 mV with 49 mV of common mode. The lower plot in Figure 3 illustrates the sense-amplifier outputs resolving this input differential. The non-zero offset level on the sense-amplifier '0 output can be seen to induce a glitch on the non-switching terminal of the CDL (middle plot) that is mitigated by the cross-coupled P-keepers on the CDL. The magnitude of this glitch is a decreasing function of the sense-amplifier input differential. Below a certain minimum input differential voltage, the CDL glitch could potentially induce a domino false-discharge failure mechanism on the CDL output, resulting in a speedpath or logic failure.

It can be seen in the bottom plot of Figure 3 that the DCN reset clock resets the inputs of the sense-amplifier shortly after the sense-amplifier trigger point. In fact the DCN reset is initiated only one inversion after the sense-amplifier. If the inputs to the sense-amplifier are reset before the sense-amplifier resolves, then a functional failure will occur. The part will then not operate at any frequency. This race is known as the "sense versus reset" race, and it is the only functional race in this LVS design.

A typical LVS circuit is allocated only about two inversions to develop differential!

Full details on the adder circuitry are detailed in the Adder Circuit section below.

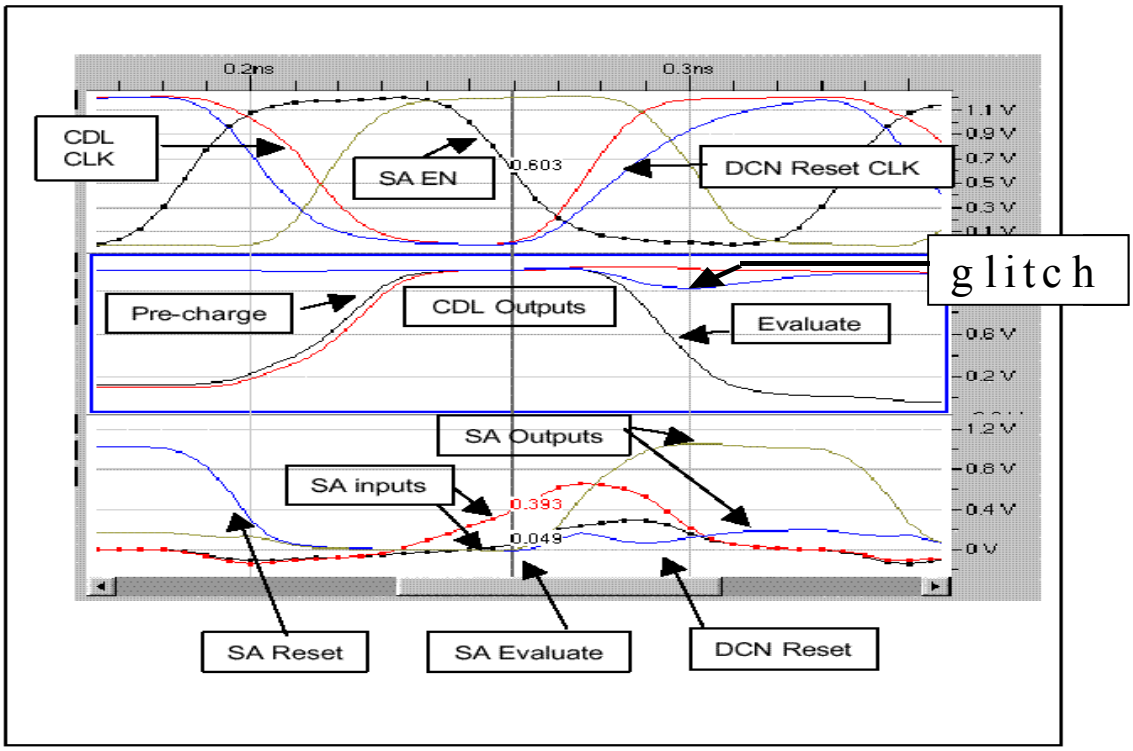

Figure 3: LVS waveforms

CLOCKING

The core logic of the CPU is running at the specified processor frequency. For a 3.5 GHz core clock frequency, the Main Core Clock (MCLK) period is about 285 ps. The pulsed Fast Clock (FCLK) doubler circuit doubles the MCLK frequency, in this case to 7 GHz. One FCLK phase is allocated for LVS DCN signal development, and the other FCLK phase is allocated for DCN precharge. The pulsed clocks used on the previous 2x Intel Pentium 4 integer cores were ideal for clocking LVS circuits because they would only stop in the reset phase. This meant that when the clock was stopped there would be zero source drain leakage for the pass-gates because all nodes would be reset to ground. The pulsed FCLK is generated by combining two tunable NAND chopper delay circuits into a pseudo wired-or. One chopper is sourced from the MCLK, the other, one inversion later. In order to provide symmetric FCLK pulses for both phases of the MCLK, the low phase of the MCLK is one inversion delay longer than the high phase to account for the additional inversion to the second chopper. This non-fifty percent duty cycle of the MCLK allows the core circuitry to do more work in the low phase of the MCLK for the non-LVS MCLK circuitry, but limits the bandwidth of the global clock distribution more than if it was a true fifty percent duty cycle clock. Figure 4 shows the clocking edge relationships.

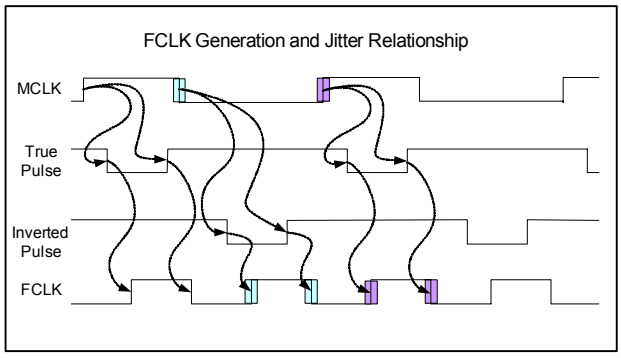

Figure 4: Fast Clock (FCLK) timing edge relationships

Clock skew and jitter posed significant challenges to LVS design, especially as these would degrade an FCLK phase speedpath four times as much as an MCLK cycle path. Exact control of the MCLK high and low phases have a direct impact on the allowable time for the low phase of FCLK; if either phase of MCLK gets smaller, the FCLK low phase will also get smaller by the same amount. A key advantage of pulsed clocking is that the FCLK high phase is not affected by MCLK jitter and skew, since both FCLK edges are generated from the same MCLK edge.

With cycle-to-cycle jitter, just the low phase of the FCLK will be impacted. The design of the LVS blocks took advantage of these effects and accounted for them directly during timing analysis.

LOW-VOLTAGE SWING USAGE WITHIN THE INTEGER CORE

Figure 5 provides an architectural block diagram, showing the critical integer core components. LVS circuitry enabled the Intel Pentium 4's low latency, used in the critical L0 load pipeline's alignment mux, adders, logic unit, rotator, and the address generation unit. Details of LVS used for the adder, rotator, and alignment mux circuits are given below.

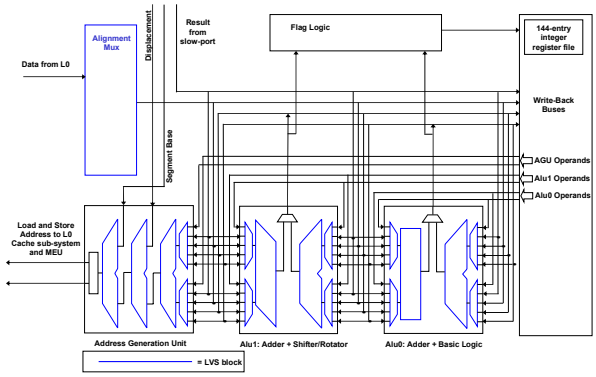

Figure 5: Integer core architectural diagram showing LVS usage

Adder Circuit

The LVS carry chain for a 16-bit adder is shown in Figure 6. It is built upon 16 cascaded LVS PGK cells, named "0-F," with carry-skip pass-gates "s0-s9" placed between them such that the span of any carry propagation path is limited to no more than 6 series devices. The LVS cells that make up the LVS carry-chain are shown in Figure 7. The LVS XOR gates in Figure 7 hook up to each polarity of the carry [n] nodes along the carry-chain to produce the sum [n+1] result. The typical critical path begins with the turning ON of the "s0" skip pass-gate that allows the "Cin" to charge up the precharged-low carry-chain and develop differential at the inputs of 17 PMOS sense-amplifiers that sense the 16 bit sum and the carry out. This 16 bit adder spans half the datapath height. It's length represents the total interconnect distance that has to be traversed for a carry-chain to propagate from "Cin" to "Carry<15>." A bit slice of the LVS adder, including adder PGK controls and clocking, is shown in Figure 8. The LVS front-end evaluates in phase 2 of the FCLK and presents source data to the first sense-amplifier "P-SA-1" that is triggered on the fall of the "ckxf1pb6_b" clock. The next stage "CDL-1" is an inverting level-restoring stage that has integrated PGK logic. This circuit is a complex CDL gate that begins evaluation on the rise of the "ckxf1p7c" clock. Exactly one FCLK phase later the fall of the "ckxf1p7c" clock triggers the 17 sense-amplifiers commonly titled "P-SA-2" (see Figure 8) that capture the sum and carry results of the 16 bit LVS adder.

Typically, the critical path goes through the "gp [n]" group-propagate signals. The wide NOR gates that generate "gp [n]" group-propagate functions are allocated only 16 ps. A conventional design of a fast ratioed-NOR, similar to the one illustrated in Figure 9, could not be used to implement wide, single-stage, precharged-low NOR functions required by the LVS adder. For certain input combinations, the gates' pull-up and pull-down networks are on simultaneously, and for these cases the ratioed-NOR gate's output can produce a steady-state noise source that is typically in the ~200-400 mV (Vcc = 1.2 V, 1262) range. A novel p-interruptible ratioed-NOR gate (RP-NOR), illustrated in Figure 10, was designed in place of a ratioed-NOR (RNOR) gate. RP-NOR gates can implement fast NOR2-NOR5 gates while limiting the contention-induced noise to a small and narrow glitch. The signals "pc," "pa," "pb" are the inverse of "pcn," "pan," and "pbn," respectively. All inputs are precharged high. During precharge, the top P-device is on, enabling it to precharge the internal node n1 thus facilitating a fast rising transition. If "pdn" falls, then the P-stack is enabled and the NOR will begin to rise. This transition is allowed to continue only if "pan," "pbn," and "pcn" also fall, because this is the only condition for which the top P-device will remain enabled. For all other combinations, the P-device is disabled within one gate delay, limiting the contention to a narrow pulse that is approximately one gate delay wide. Figure 11 shows the comparison of the RP-NOR contention noise pulse to the DC contention waveform produced by a RNOR within the context of the LVS adder's evaluation window. A RNOR produces a contention waveform that exists for the entire duration of the LVS evaluate window, allowing the off skip device driven by it to leak opposite charge onto the carry-chain, degrading or even destroying the DCN signal development. The new RP-NOR, however, produces its contention glitch only at the onset of LVS evaluation leaving the carry-chain a significant amount of time to recover and develop positive differential unimpeded by any further contention-induced differential noise.

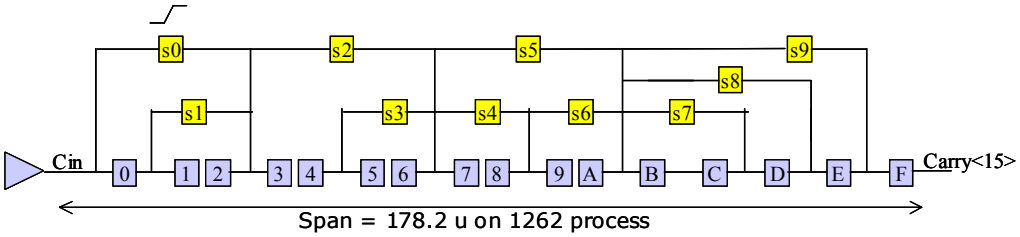

Figure 6: 16-bit adder LVS carry-chain structure

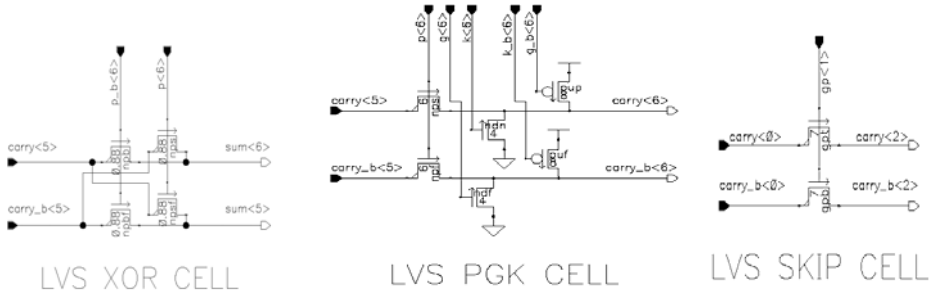

Figure 7: LVS PGK and XOR cells used in the LVS carry-chain

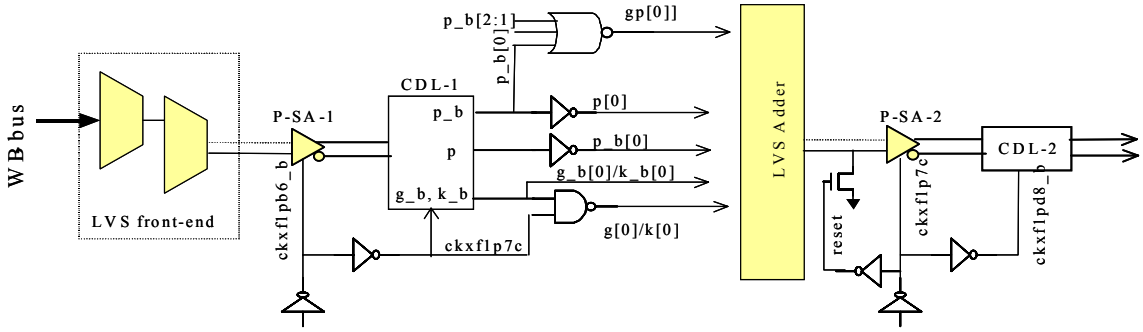

Figure 8: Logic for adder bitslice

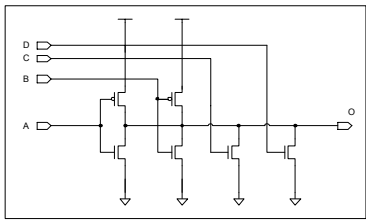

Figure 9: A 4-input ratioed NOR (R-NOR) gate

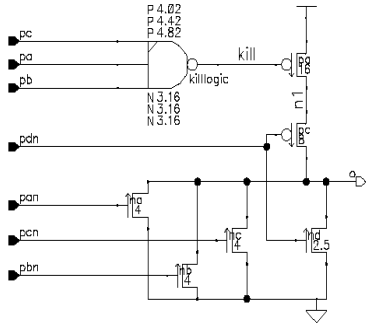

Figure 10: A 4-input ratioed P NOR (RP-NOR) gate

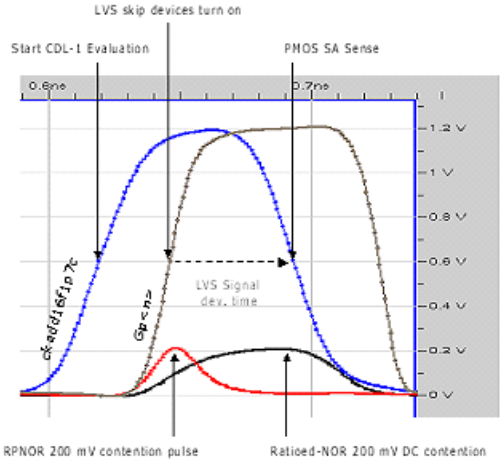

Figure 11: Contention behavior of ratioed-NOR gate (RNOR) versus new RP-NOR gate

Alignment Mux Circuit

LVS makes possible the implementation of the Alignment Mux function in an MCLK phase (144ps), reducing the critical "load pipeline" latency in the integer core. It provides a 2x speed improvement over the traditional multi-stage domino design. The Alignment Mux datapath function consists of 128 individual 32:1 dual rail muxes distributed across the datapath width of the entire L0 cache. Muxing is performed with a single-stage DCN pair followed by the large distributed mux node connected to the sense-amplifiers. This RC requires designing the Alignment Mux at MCLK frequencies, whereas all other LVS blocks operate at FCLK. Source inputs to the DCN are full-swing dual rail signals from the L0 cache. The clock gated control logic generates the full-swing DCN selects, which are replicated across the entire mux height to reduce loading and RC. Figure 12 shows the circuit topology of the Alignment Mux.

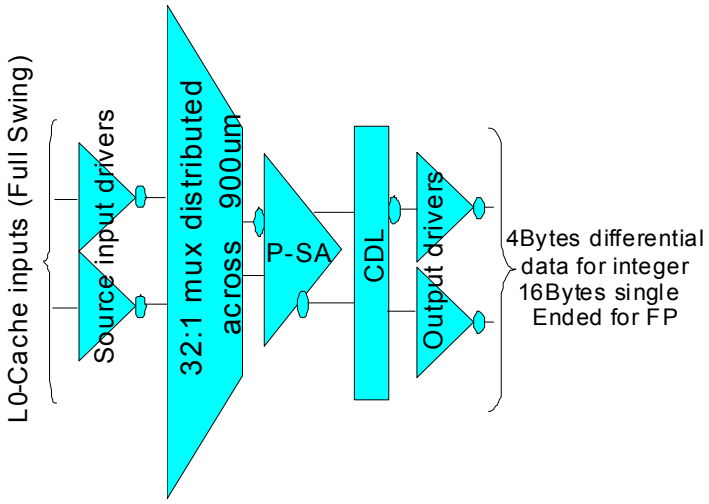

Figure 12: Alignment Mux circuit block diagram

LVS was the ideal technology for the Alignment Mux, with its muxing and distributed RC. LVS technology proved crucial in reducing the L0 cache latency and has enabled aggressive frequency headroom for subsequent Pentium 4 steppings.

LVS Rotator and Shifter

The LVS rotator/shifter performs these operations (ops): Rotate Left (ROL), Rotate Right (ROR), Shift Left (SHL), Shift Right (SHR), Shift Arithmetic Right (SAR), Byte Swap, and High-Low Swap. The only 8-bit operations supported are "8L," performed on bits [7:0] of the operand. "8H" rotates and shifts are done in the Intel Pentium 4 processor slow port datapath and are longer latency operations. For 8-bit and 16-bit rotates and shifts, the remaining bits of the operand are passed through unchanged to the result. For SHL and SHR ops, the value of '0 is padded in from the least significant or most significant position, respectively. For SAR ops, the value of the most-significant bit is padded in from the most significant position.

Rotate and shift operations are done by first rotating the operand according to the shift count, and then selecting either the rotated value or the "kill value" to produce the final result. The kill value is always '0 for SHL and SHR ops, and it is the size-appropriate, most-significant bit for SAR ops.

As in reference [3], the block algorithm takes advantage of symmetry to streamline the rotation portion of the datapath.

Right rotates with a shift count of *r* are done by rotating the operand to the left by *r*# + 1 places. For right ops, the shift count is complemented prior to entering the shift count decode logic, and the extra "+1" place is taken care of in the datapath.

Rotator circuits are traditionally a series of muxes wired up with long interconnects to steer the operand over the length and breadth of the datapath. This makes the rotator particularly suited to LVS technology. The datapath muxes are implemented with a wide DCN that ends up at 32 sense-amplifiers for the "prop value" and an additional 32 sense-amplifiers for the "kill value." A final muxing stage selects between the outputs of these two sets of sense-amplifiers. The selects for this muxing stage are bitwise. The complex shift logic is implemented using LVS circuits. The decode of the shift and rotate count is done in the Front-End by embedding logic into the CDLs and the subsequent static logic stage. LVS technology has enabled us to implement Fast Rotate and Shift ops in Pentium 4 processors, which provides a significant and measurable performance gain.

TOOLS AND METHODOLOGY

A large challenge to enabling the design of LVS circuitry was to provide the innovative tools and methodologies that enabled us to successfully apply this technology to a HVM environment. We created LVSTNT (LVS Timing and Noise Tool), a custom dynamic simulator and interface, that calculates required and valid times for groups of data and pass-gate signals interfacing with the LVS circuitry. These results were merged with our project standard static timing tools and flows. A custom LVS Layout Rule Checking (LRC) tool was developed to eliminate non-common mode noise and to ensure layout matching that can tolerate process variation.

Dynamic Simulation and Timing Issues

LVS timing performance depends upon the relative arrival of dozens of signals, followed by the generation of a small differential signal, which requires a uniquely complicated timing analysis. In contrast, traditional static timing analysis simply assumes that a single signal generates a timing path. And unlike a typical dynamic simulation, which simply verifies that a circuit operates at a target frequency, our LVSTNT dynamic simulator provides a key advantage by calculating the worst-required times for the input signals. By having the required times, we know how much timing margin a given input has, enabling us to make valuable area/power/delay trade-offs. To avoid exponential growth in the number of timing paths when calculating the required input timings of combinations of multiple signals, we made creative, specific assumptions to maintain a linear number of simulations. And even after pruning the number of timing paths using patented algorithms [4], the rotator alone required simulations on more than 60,000 paths to characterize the circuit. To address the associated huge runtime and database size, LVSTNT partitions the LVS circuitry into the minimum database needed to dynamically simulate each unique path. We can quickly, interactively, simulate a single path of interest. As even a single path requires 3-12 simulations to find the passing conditions and input required times, understanding so many simulation results proves daunting, so LVSTNT automatically merges the worst-case results from all simulations for dozens of timing constraints. While analyzing the complete circuit, we batch and send all simulations to our compute farms, utilizing hundreds to thousands of machines worldwide.

Transparently clocked designs provide greater tolerance to clock skew on silicon, which is a significant portion (15-20%) of the FCLK cycle time. The LVSTNT required times are merged into our normal static timing tools, enabling us to take full advantage of transparency through latches and domino state elements. To provide a transparent timing interface, DCN selects and dual rail data inputs were ideally designed to be precharged. During evaluate, the DCN select gate inputs and data and data# inputs would transition monotonically; if this occurs after the thru-gate opens, then we have a nicely transparent timing path. This elegant interface is not feasible when just single rail data inputs are available because both data and data# cannot be precharged to ground. Generating data# locally, results in either data or data# starting out high before evaluate, and as the first pass-gate opens, the DCN would start developing the opposite logic value before developing the correct small signal waveform. This posed significant simulation modeling challenges. Aiming for a robust design, we decided to prevent generating this wrong differential by adding an extra clocked n device, the thru-gate, and then requiring that data be set up to the thru-gate opening. While the thru-gate intrinsically slows the circuit, due to the additional n device in series, this greatly simplified the timing complexities of both dynamic and static analysis, enabling robust tools.

Determining whether the circuit operated or failed raised many questions in our challenge to enable HVM. Sense-amplifiers in the ideal world of a dynamic simulator resolve with just a few electrons. On silicon, the coupling noise to signal waveforms and power supply alone contribute significant complexity to an ideal model. A few failure criteria are described below.

An initial failure criterion used during pre-silicon verification was the magnitude of the CDL output glitch. If this noise glitch propagated to a subsequent domino stage or state element, a logic error or severe speedpath could occur. This provided an easily implementable pass/fail criterion that was based upon an observable

circuit failure. In standard CMOS designs, noise tools verify that the circuit will not falsely discharge a domino node. In LVS designs, we not only guarantee this will not happen, but this failure point can directly dictate the required set-up times to the sense-amplifier. Traditional static and dynamic circuits never attempt such closely intertwined timing and noise requirements.

A second failure criterion mandated a minimum differential voltage. The total requirements ranged from 50-100 mV, or roughly 5-10% of VCC. Device variations due to process (Le, Vt, dual-Vt, etc.) was analyzed for our library of sense-amplifiers, and accounted for nearly half of the signal voltage requirement, with the remaining attributable to incomplete precharge, noise, and the inherent differential needed to sense correct data. This requirement enhances confidence in the design, and it covers several corner cases and simulation artifacts not caught by the CDL failure criterion.

A third criterion avoids designing non-full-rail static signals, which static timing tools ignore, but are very easy to create at such extremely high frequencies. While a dynamic simulator shows circuit robustness with non-full-rail input signals, real silicon in HVM would not be nearly as forgiving. An additional motivation in avoiding non-full-rail signals is the inability to define them when we translate waveforms back to the static timing tool environment.

Functional race analysis across process corners poses additional failure, modeling, and runtime concerns. Our implementation contains just one functional race, the sense-amplifier enable assertion to the DCN reset clock. Minimizing functional races (mindelays) is important, because if they fail to make speed, the part will not function at any frequency. If mindelays had occurred, they would have required significant timing guard band (impractical at these frequencies), and/or significant effort to simulate the circuit (with additional timing paths) across process variation. In light of design for debug and testability, we added software controllable circuitry to vary our functional race margins.

Merging Static and Dynamic Timing Tools

Creative solutions enabled interfacing our dynamic timing tools with standard project tools into a seamless tool flow that could be batched. Specific attention was paid to drawing a precise boundary of what was dynamically analyzed: for dynamic simulation. We automated netlist extraction to form a black box containing just the data inverters, DCN, sense-amplifiers, and CDL. LVSTNT provided the minimum number of timing edges, leaving the project static timing tools to analyze all but the small signal development and failure criteria. To enable the static tools to analyze the black box for the remaining edges, we fully automated the generation of timing tool assertions. For example, the set-up of static data falling would be checked against the thru-gate clock rising. Outside the black box, static timing tools analyze the select and data timings and convert their timings into waveform inputs to the dynamic simulation. Black box interface timings come from a combination of the dynamic simulator, the static simulator, or a worst-case merging of max and min timing. (Providing details on over 30 classes of signal types, domino, static, etc., and edges, rise, fall, lead, trail, is beyond the scope of this paper.) The fully automated performance verification tool suite provides a huge return on investment, given the design size, complexity, and multi-year life cycle of a production processor. This automation greatly increased our productivity, providing consistency of assumptions among our designs, and it enabled high-quality audits of the correctness and thoroughness of our checks.

Dynamic Noise Analysis of LVS

Noise easily overwhelms the tens of millivolts of signals inspiring us to support dynamic noise simulation directly within our LVSTNT timing tool. Project noise analysis tools provided the input waveforms, based on the circuits (static or LVS) driving into the LVS block. LVSTNT super-imposes DC and pulse-wise-linear noise waveforms onto the DCN data and select gates. Logically off devices will result in their gates being slightly turned on, with the device's source tied to the rail that provides worst-case leakage, with respect to the signal being developed. The methodology and algorithms are fairly complicated and posed several logic and circuit challenges. A few early designs proved highly susceptible to leakage constraints, so we formed several guidelines on circuit topologies and sizing to deal with these problems.

Layout Rules and Matching

High-quality layout was a key enabler to not just functional first silicon, but to HVM. Sensing signals on the order of 50-100 mV (~5% of vcc) demanded diligent elimination of all noise, be it from residual precharge, leakage, gate to drain coupling, wire coupling, or non-common mode noise caused by mismatched layout or process variation. Careful, up-front attention to layout enabled early identification, elimination, or mitigation of noise sources.

The creation of a custom LVS Layout Rule Checking tool enabled us to create correct-by-construction layout by highlighting non-common mode geometries. The LRC tool helped guarantee that all differential paths were matched in terms of layout geometries. It analyzed all pertinent device and metal layers, handling process patterning and variation issues, enforcing consistent shielding, and ensuring all signal attackers (cross

capacitance) were common mode. This correct-by-construction layout was absolutely necessary for high-volume production and for the creation of a database of this size. SRAM and analog designs deal with similar layout matching issues, but on a much smaller scale; e.g., SRAMs involve just one arrayed memory cell. LVS circuitry contained hundreds of thousands of unique layout geometries and timing paths, and ensuring matched nlayout alone was a feat never before accomplished on this scale.

To date, no tool or methodology holes have led to functional, yield, or speed issues on silicon.

CONCLUSION

Low-Voltage Swing circuit technology utilizes custom tools and methodologies to implement random small signal logic at an unprecedented scale. Our 2x frequency integer core implementation on Intel's 90nm process meets the present Pentium® 4 processor product demands. With process and post silicon optimizations the design will support increasingly higher frequency processors.

ACKNOWLEDGMENTS

Innumerable thanks to Matt Morrise, who worked side-by-side with us to create our custom dynamic simulator tool. Thanks to Nick Kuhlman for developing and supporting countless methodology and tool issues. Edmund Pierzchala enabled our layout design rule checker tools. Dan Milliron provided our noise analysis tools and automation tools. Many thanks to our crack layout design team, for ensuring high-quality, low-noise layout. Andy Soelburg and Sean Mirkes provided a comprehensive methodology to interface static and dynamic timing analyses.

REFERENCES

[1] Dan Deleganes, et. al., "Low-voltage-swing logic circuits for a 7Ghz x86 integer core," *IEEE ISSCC*, February 2004.

[2] S. Thompson, et. al., "A 90nm technology featuring 50nm strained silicon channel transistors, 7 layers of Cu interconnects, low k ILD, and 1 um2 SRAM cell," *2002 IEDM Digest*, Dec. 2002, pp. 21-64.

[3] Pereira, R., and Mitchell, J.A., and Solana, J.M., "Fully pipelined TSPC barrel shifter for high speed applications," *IEEE Journal of Solid-State Circuits*, Vol. 30, No. 6, June 1995.

[4] Stevens, K. and Morrise, M., "Algorithm for finding vectors to stimulate all paths and arcs through an LVS gate," U.S. Patent 6557149.

AUTHORS' BIOGRAPHIES

Daniel J. Deleganes is the project manager of the Intel Pentium 4 processor integer execution cluster described. Daniel received a B.S.E.E. degree from the University of Washington in 1988 and an M.S.E.E. degree from Cornell University in 1989. At Intel he contributed to high-speed Intel i486™ and Intel Pentium processor 2nd level cache SRAM families, several Pentium processor generations, and all Pentium 4 processor generations. His e-mail is daniel.j.deleganes at intel.com.

Micah Barany is the microarchitecture manager of the Pentium 4 processor integer execution cluster described. Micah received a B.S. degree in Engineering Physics from the University of California, San Diego in 1989, and an M.S.E.E. degree from Stanford University in 1993. At Intel he contributed to Intel i386™ SX and Intel i486 SL microprocessor families, and several Pentium and Pentium 4 processor generations. His e-mail is micah.barany at intel.com.

Daniel Chow is a memory design lead and integrator in the Execution Cluster. He joined Intel in 1996 and has been involved with the methodology definition and implementation of high speed circuits in all Pentium 4 processor generations. Prior to Intel, he worked at Motorola as the primary memory designer for the MC68060. He received his MSEE and BSEE from Oregon State University. His e-mail is daniel.c.chow at intel.com.

Thomas D. Fletcher directs circuit methodology and research for new microprocessors in DPG. He has worked at Intel since 1991 and is a senior principal engineer. He was the clock unit owner for the Pentium Pro processor and has steered early circuit design and research for several generations of the Intel Pentium 4 processor. He is listed as inventor or co-inventor for 45 Intel patents and has 5 IEEE publications. His e-mail is tom.fletcher at intel.com.

George L. Geannopoulos co-managed the LVS design team and is currently managing the PLL team in LTD. He joined Intel in 1994. His interests include high-speed circuits, PLL, clock generation and analog design. Prior to joining Intel, he worked at Bipolar Integrated Technologies as a design manager designing VLSI ECL RISC FPUs FPCs and register files. He also worked at MMI/AMD designing programmable array logic (PALs). He received a B.S.E.E. degree from the University of

---

™ i486 and i386 are trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Illinois, Champaign Urbana. His e-mail is george.geannopoulos at intel.com.

Kurt Kreitzer currently manages the LVS design team. He recently worked to spec and develop quality LVS design tools and methodologies. Kurt received a B.S.C.E. degree from Oregon State University in 1994. After working on the Pentium® Pro, he created the modular SRAM array used throughout the Pentium 4 and other projects and implemented the Pentium 4 Trace Cache. His e-mail is kurt.kreitzer at intel.com.

Anant P. Singh is a senior designer on the LVS team. His recent focus is in mixed signal design where he has worked to define, implement and productize LVS circuits in the dual pumped execution core of the next-generation Pentium 4 processor. He has also designed circuits to enable the backside bus logic on Pentium® III processors. Anant has an M.S.E.E. degree. from the University of Washington and a B.S.E.E. degree from Delhi University. Prior to Intel, Anant worked in the field of control systems and automation. His e-mail is anant.p.singh at intel.com.

Sapumal B. Wijeratne co-designed the LVS AGU/ALUs. Prior to this work Sapumal held numerous technical leadership positions including methodology lead for domino circuits and register files. He is currently co-managing the next lead processor's integer execution core design team in LTD. Sapumal received a B.S.E.E. from Lafayette College in 1984 and a M.S.E.E. from Purdue University in 1986. His e-mail is sapumal.wijeratne at intel.com.

Copyright © Intel Corporation 2004. This publication was downloaded from http://developer.intel.com/.

Legal notices at http://www.intel.com/sites/corporate/tradmarx.htm.

® Pentium is a trademark of Intel Corporation or its subsidiaries in the United States and other countries.

THIS PAGE INTENTIONALLY LEFT BLANK

For further information visit:

developer.intel.com/technology/itj/index.htm

Copyright © 2004 Intel Corporation. All rights reserved.
Intel is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.
For a complete listing of trademark information visit: www.intel.com/sites/corporate/tradmarx.htm