Design of the Two-Core x86-64 AMD "Bulldozer" Module in 32 nm SOI CMOS

Hugh McIntyre, *Member, IEEE*, Srikanth Arekapudi, *Member, IEEE*, Eric Busta, *Member, IEEE*, Timothy Fischer, *Member, IEEE*, Michael Golden, *Member, IEEE*, Aaron Horiuchi, *Member, IEEE*, Tom Meneghini, *Member, IEEE*, Samuel Naffziger, *Senior Member, IEEE*, and James Vinh, *Senior Member, IEEE*

***Abstract*—This paper describes key circuit innovations in a new x86-64 micro-architecture [1] AMD code-named "Bulldozer" [2], [3]. It is implemented in 32 nm high-K metal gate SOI CMOS. It occupies 30.9 mm ^{2}, contains 213 million transistors, reduces the number of F04 gates per cycle by more than 20% compared to a previous processor in the same technology [4], and demonstrates superior frequency scaling across voltage. The module includes two independent integer cores but shares the fetch, decode, floating-point, and L2 cache units to maximize single-threaded performance and multi-threaded throughput while significantly improving power and area efficiency compared to fully replicated CPU cores. The design includes a new soft-edged flop (SEF) family to enable high frequency and low power. Achieving power efficiency in combination with high-frequency design is a particular challenge, and this paper describes several of the unique approaches to power optimization that have been employed in the design. The gate-count reduction and power optimization enable faster frequencies in the same power envelope compared to previous designs.**

***Index Terms*—Array design techniques, clocks, flip-flops, circuit design, low power, clock power reduction, power management, register files, energy-efficient circuits, idle power, multi-core, clocked storage elements, floating point, x86-64, SOI, dynamic circuit design, design productivity, low voltage design, soft errors, CMT, 64-bit architecture, 8T RAMcell.**

I. Introduction

AMD's two-core "Bulldozer" module is a new x86-64 processor designed from the ground up to meet the demanding compute needs of both client and server workloads. Built in 11-level metal 32 nm HKMG SOI CMOS, the processor module contains many circuit innovations to improve performance while optimizing power efficiency and frequency. This paper focuses on circuit innovations used to meet aggressive design goals.

This new micro-architecture [1] contains two processor cores that implement chip-level multi-threading (CMT). Rather than sharing unused resources as in simultaneous multi-threading (SMT), Bulldozer's CMT provides dedicated compute resources to each CPU to maximize single-threaded performance and multi-threaded throughput while significantly improving power and area efficiency compared to fully replicated CPU cores. The result is improved performance and frequency and reduced area and power compared to a previous AMD x86-64 CPU built in the same 32 nm process [4]. Frequency at constant voltage is improved by more than 20% (Fig. 4) while the dual-core switching capacitance is reduced to 84% of two previous cores.

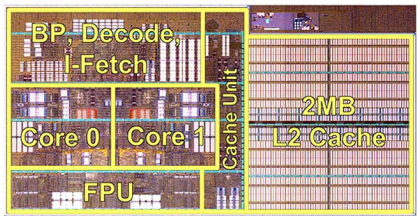


Fig. 1. Module photo.

Physically, the two-core Bulldozer module contains 213 million transistors and is designed to operate from 0.8–1.3 V. To achieve aggressive frequency goals, the design reduces the number of fan-out-4 inverter delays per cycle (FO4) by more than 20% to 12–14 usable gates between flops, achieving higher frequencies in the same power envelope even with increased core counts.

As shown in Fig. 1, the Bulldozer module includes two cores with shared L2 cache and occupies 30.9 mm^2. Each core is dedicated to a thread and contains a four-wide integer execution unit, issue/retire logic, 16-KB four-way set associative L1 data cache, and a load/store unit. A separate dual-threaded floating-point unit is shared between the integer cores. Of the total module floorplan area, the two cores occupy approximately 21% and the floating-point unit occupies another 9%.

A 40-entry out-of-order (OOO) integer instruction scheduler accepts instructions dispatched from the front end and issues up to four instructions per cycle to two arithmetic logic unit (ALU) and two address-generate unit (AGU) pipes. The integer execution units execute most instructions in one cycle and are fully bypassed for minimum latency. Logical register numbers are mapped into an integer physical register file (PRF) containing 96 entries of 64 bits each, physically constructed as two replicated 4R/4W arrays that together support eight reads and four writes per cycle in total.

The 16 KB L1 write-through D-cache has a four-cycle load-use latency and is backed by a shared 1- or 2-MB L2 cache having 18/20-cycle load-to-use latencies, respectively. The 2-MB L2 cache shown in Fig. 1 occupies about 35% of the module area, and is physically shorter than the rest of

Manuscript received May 02, 2011; revised June 20, 2011; accepted July 20, 2011. Date of publication October 25, 2011; date of current version December 23, 2011. This paper was approved by Guest Editor Alice Wang.

H. McIntyre, S. Arekapudi, M. Golden, and J. Vinh are with AMD, Sunnyvale, CA 94088 USA (e-mail: hugh.mcintyre@amd.com).

E. Busta, T. Fischer, A. Horiuchi, and S. Naffziger are with AMD, Fort Collins, CO 80528 USA.

T. Meneghini is with AMD, Boxborough, MA 01719 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSSC.2011.2167823

the module because the height is set by the SRAM cell. The resulting notch is used for logic at the SOC level, part of which is visible in Fig. 1. A cache unit provides the interface between the L2 cache and the rest of the module. Besides the L2 cache, cache unit, and floating-point unit, the two cores in each module also share the instruction front-end logic, which consists of instruction fetch (containing a 64 KB, two-way instruction cache), decode, and branch prediction units. The instruction front-end comprises about 14% of the module area.

The floating-point (FP) unit contains a dedicated 60-entry instruction scheduler and four FP pipelines containing two complete 128-bit FMACs that are new on Bulldozer. The FP unit is fully dual-threaded and can issue and execute instructions from both threads simultaneously. The FP unit design provides sufficient compute bandwidth for both threads, but also enables all resources to be used together to maximize single-thread performance. The 10R/6W FP PRF is 160 entries $\times$164 bits and is split horizontally into two arrays. This arrangement dramatically improves power efficiency and timing, and is described in more detail later in this paper. Due to the distances between the register file and FP pipes, two levels of bypasses between FP pipelines are supported: 0-cycle latency in the datapaths and one-cycle latency in the FP PRF.

The Bulldozer module contains 84 unique custom macros and 317,000 scannable flops. Module-level VSS power gating (C6) is used to reduce leakage power by approximately 95% when both cores are idle [4]. The 32 nm SOI process provides three transistor V_T types (low, regular, and high), with longer channel lengths used to achieve even finer-grained trade-offs between leakage and delay. V_T's used across the design consist mostly of regular (47%) and long-channel regular (46%), with less than 1% low-V_T used for the most critical paths.

II. Power

In contrast to second- or third-generation power-optimized CPU designs [4], [5], Bulldozer's ground-up design required co-development of power efficiency, timing, and functionality. One of the greatest challenges with such a complex, high-frequency design is to achieve a truly power-efficient implementation. Generally, CPU power efficiency is achieved through careful analysis of the power consumption of the completed implementation to identify waste and reduction opportunities, which are then implemented in a subsequent version of the core. With a ground-up design such as Bulldozer, this sort of analysis loop was not possible: power efficiency had to be designed simultaneously with convergence on timing and functionality.

Another challenge to power efficiency was presented by the high-frequency target for the design. As is well known [6], it is much more difficult to achieve power efficiency with a high-frequency design than a low-frequency design due to higher flop counts, larger gates to make timing, tighter skew requirements on clocks, etc. Despite these challenges, the Bulldozer team achieved the goals through a relentless focus on power through all stages of the design while simultaneously reducing the cycle time from past designs.

First, wide communication paths were minimized in the micro-architecture and floorplan and low-power features included. Power evaluation of these features was accomplished

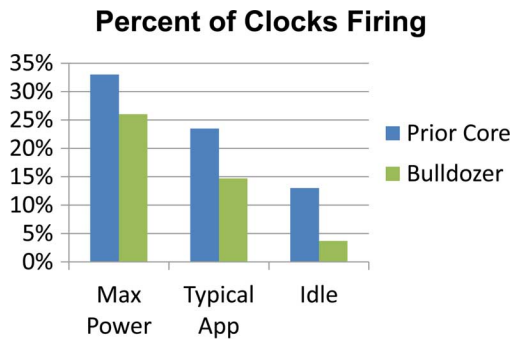


Fig. 2. Clock activity factors.

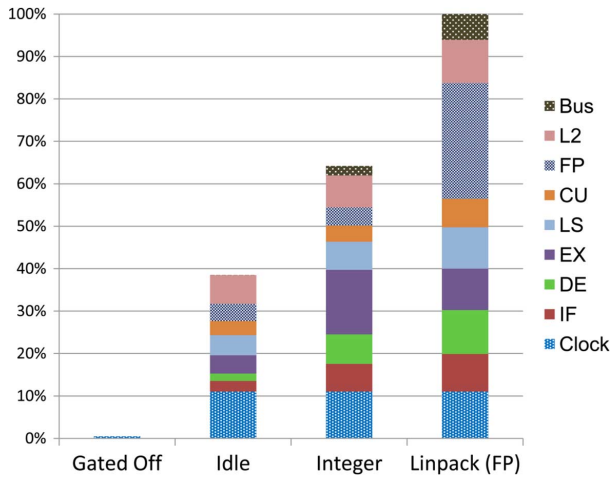


Fig. 3. Power by unit.

by instrumenting the high-level performance model with power information so feature evaluations could include the energy cost. Second, as the RTL was written but before schematics and layout were complete, the team used detailed RTL-based clock and flip-flop activity analysis as a proxy for switching power, tracking these weekly and driving the team to reach aggressive clock gating goals. Third, the team developed a new power analysis flow that enabled early mixed schematic and layout analysis of transistor-level power consumption, including full-custom parts of the design. This roll-up provided the basis for a round of aggressive power optimizations (clock gating, gate downsizing, pre-charger optimization, etc.) while the implementation was still somewhat malleable.

As shown in Fig. 2, the percentage of clocks firing while executing code is significantly lower than in the legacy design. A focus on turning off clocks when the core is halted (easy to analyze) also paid dividends by reducing clock activity factors while executing code. This can be seen in Fig. 3, which shows the power consumed by the large floating-point unit is extremely low when executing integer code. The team invested heavily in multi-level clock gating to ensure no additional circuit was clocked beyond what is necessary to accept a new instruction should one come down the pipe. Combining these approaches with the lower-power circuit methodologies and the inherent efficiency of the two-core Bulldozer module architecture resulted in a design that consumes less energy per operation while still operating at a high frequency.

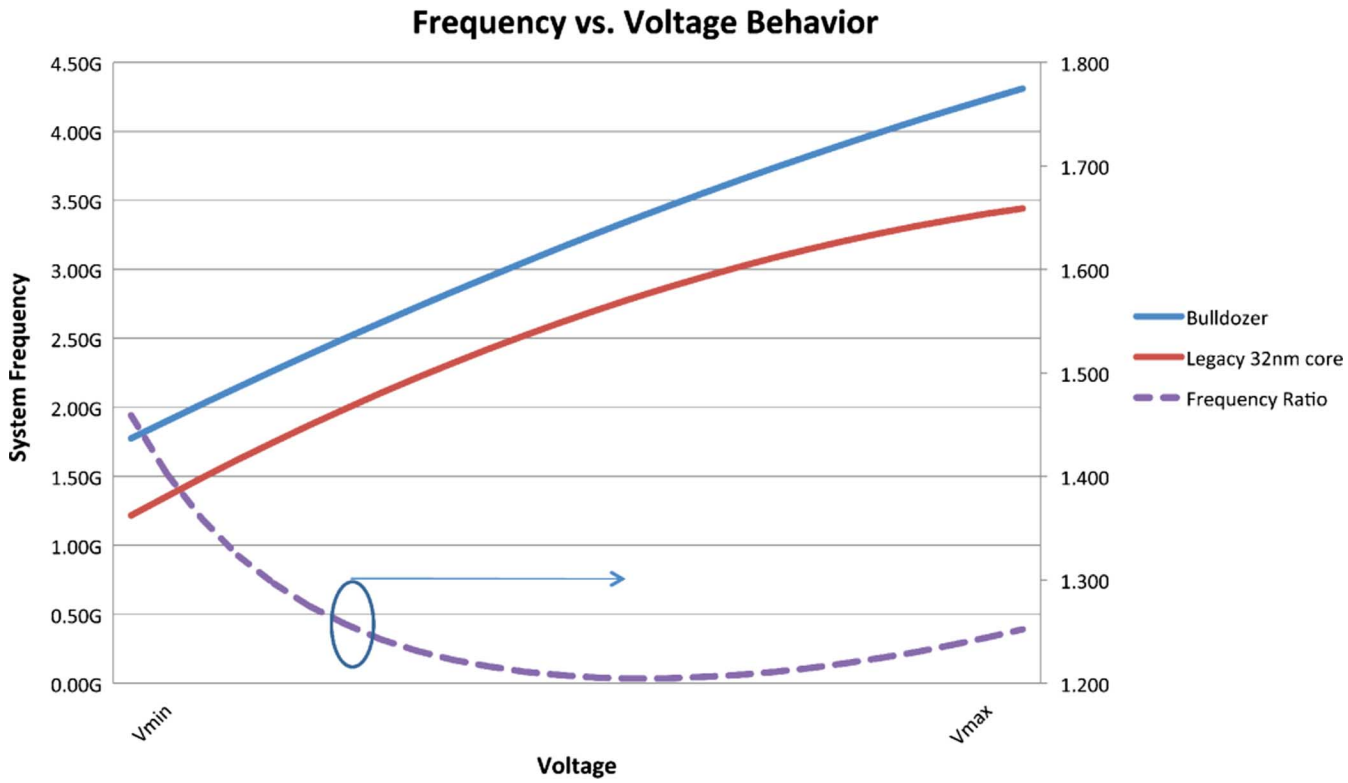


Fig. 4. Bulldozer core frequency achievement.

Another key objective of the core design was to enable very high frequency (at high voltage) while also running efficiently at low voltage. Energy per operation is minimized at low voltage, so the team needed to optimize the circuits for that operating point. This required two separate timing corners with equally aggressive cycle time goals, one at 0.8 V the other at 1.3 V. As can be seen in Fig. 4, this two-point timing optimization enables the core to scale better than the legacy design across the full operating range. Both gate-dominated timing paths and paths with high RC content have been tuned to provide better scaling than the prior design, which optimized timing at just a single nominal voltage. This enables the core to thoroughly exploit boost capability and ACPI P-states [7] by scaling frequency very efficiently with voltage based on compute requirements. To further exploit this broad operating range, Bulldozer combines an integrated digital power monitor [4] with chip-level power management logic to dynamically manage the operating point of the cores based on the workload.

III. Soft-Edged Flop

The twin goals of low power and high frequency drove the choice of a soft-edged master-slave flop (SEF) for the design, in contrast to legacy AMD designs that used a combination of conventional MSFF latch-based flops (slow but power-efficient) and sense-amp-based flops (faster but high-power and high-area) for critical circuits. SEFs enable low insertion delay and efficient low-power operation, but offer the same robustness across process variation and voltage range as a latch-based MSFF. This avoids the need for poorly scaling circuit techniques such as pulse-flops, which suffer from a dual-edged timing constraint of hold time versus set time that imposes too large a burden on tools and methodology for a design destined for 14 nm technology and beyond.

Enhanced performance in the SEF is achieved by use of a two-clock design instead of the single clock in a legacy MSFF. Output data is launched from the early clock (ECLK) while data capture (setup and hold) are relative to the late clock (LCLK). This creates a soft timing edge, spreading variation effects across two pipeline stages and helping mitigate the effects of clock jitter and process variation on critical timing paths. Cycle stealing was limited by taxing setup time to the LCLK master latch and rebating setup time to the ECLK slave latch, then choosing the worst of the two, as shown in Fig. 5. This enables the use of standard CAD tools for timing without the complexity of multi-cycle paths. In addition, for silicon-limited critical paths, fuse-programmable LATE_X control bits allow further extension of the transparency window by approximately half a gate delay for debug and tuning. Fig. 6 shows the clock gater schematic, which generates LCLK two gates after ECLK.

When setup to ECLK fails, a late-arriving D input pushes out the ECLK to QB delay of the flop to the downstream logic cone. However, since jitter and variation are hidden by the SEF transparency, there is no need to add the ECLK to QB transparency push-out to pre-silicon timing provided the timing rebate is less than a jitter/variation allocation, which in our design was 5% of the cycle time.

The base SEF (HPSEF) has two potential costs, however, that need to be addressed. First, the transparency window (typically 10% of the cycle time) increases hold time because hold time is measured to LCLK while data is launched early

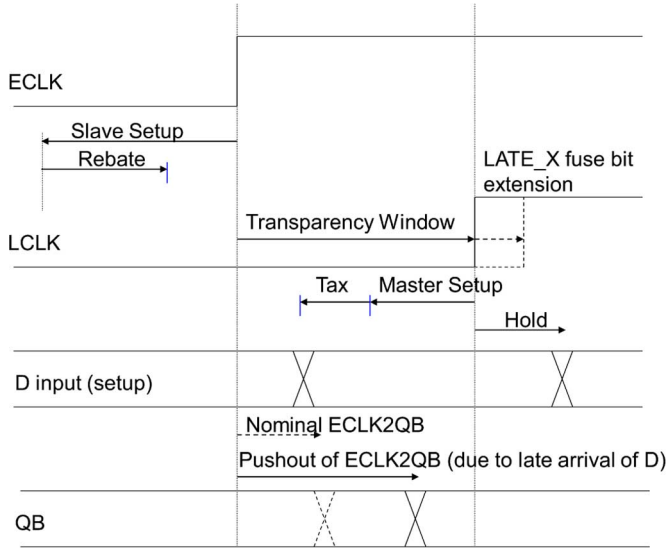


Fig. 5. SEF timing diagram.

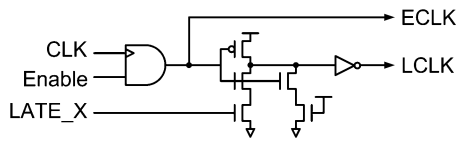


Fig. 6. Clock gater with LATE_X function.

on ECLK. Second, the improved performance of the HPSEF costs increased power because of the need to route two clock signals (ECLK and LCLK). To address this last issue, we set targets and design checks for the team to ensure that the average number of flops connected to a clock driver (gater) is eight or higher. This ensures that the additional wire cap is offset by amortizing the ECLK → LCLK delay element power (a tunable AND gate) across many flops rather than having it replicated inside each flop.

Hold time was addressed by creating a non-soft-edged low-hold variant (LHSEF, or LPLHSEF when combined with the low-power optimization described later) to minimize gate insertion for hold time needs. In this flop, the ECLK and LCLK are internally OR'ed together into the master latch in the flop so the data is both launched and captured on ECLK (Fig. 7). Apart from the loss of transparency (inevitable with a low-hold flop), one limitation of this flop is the clock duty-cycle compression of the internal master latch caused by OR'ing ECLK and LCLK. The result is compression of the master latch pulse by two gates, requiring latch optimization for process robustness.

Power efficiency was improved by designing a low-power SEF (LPSEF) with an internally gated slave latch clock. In the LPSEF flop, the internal slave latch clock does not switch when the master stores logic 0 and slave logic 0. In the LPXSEF, the internal slave latch clock is gated when a logic 0 or 1 is stored in both master and slave (Fig. 7). Simulated benchmarks showed an average activity factor of 5% on the D inputs for a majority of flops; thus, for a majority of time, the internal slave clock does not fire. This also allowed us to aggressively optimize the HPSEF for setup and hold timing performance without concern about increasing power dissipation. As shown in Fig. 8, at predicted data switching activity factors, the LPSEF saved on average 21% and LPXSEF saved 34% switching capacitance relative to the HPSEF. The majority of flops (76%) used the LPSEF/LPXSEF (51%) or LPLHSEF (25%) with the HPSEF only on timing-critical paths, delivering a good power saving overall.

Fig. 7 shows a comparison of the different SEF types and relative performance metrics, normalized to 1.0 for the worst flop type for each metric. With the exception of the LPXSEF, the SEF variants are layout-footprint compatible to enable automated flop swapping for power/performance. A small area penalty is incurred compared to the legacy flop to achieve swappability.

IV. Floating-Point Unit and Register File

As described in the Introduction, Bulldozer includes a new floating-point unit that adds many features relative to previous AMD x86-64 processors. A split datapath (Fig. 9) is used to improve data locality while supporting high throughput. This split uses a highly ported register file (physically ten read ports and six write ports), which uses pipe-sharing techniques to accommodate 13 read and 7 write busses. These ports support operand delivery and reception from either side of the FPRF.

Fig. 9 highlights the FPRF split into two 91-bit/73-bit low/high arrays, each associated with their respective data paths. The arrays are nearly identical, differing only in the number of data and parity bits. The array sizes are larger than standard IEEE formats because of the inclusion of predecode, control, and parity bits as well as data. The L/H array split supports the operand precision needed for each data path, with the low side also supporting x87 (80-bit) extended precision. This also enables power savings of 45% by shutting off one of the two arrays and its associated data path for instructions not needing the full-width data path. Power savings are dependent on workload, but instructions that enable this optimization include x87 operations and scalar SSE/AVX instructions. The use of two arrays also allows data-path placement on both sides of each array, significantly shortening operand/result bus paths for speed and power efficiency, particularly for SIMD instructions.

Reads use a hybrid two/three-level hierarchical bitline structure of local and cascaded super bitlines. The local bitline is a pre-charged dynamic M02 node. The pre-charged dynamic super bitline is a split, cross-coupled structure that controls data propagation in both directions between left and right banks through the EN_L and EN_R signals (Fig. 10). This arrangement splits the super bitline route, which reduces overall RC delay by 50% and reduces average super bitline power consumption by 25%.

Functional units provide zero-cycle bypass of results to consumer units that are physically close, during which the RF is fully disabled. The RF provides one-cycle bypass (read-after-write) between all functional units, implemented by pull-downs on left and right super bitlines. This circuit provides full bypass from any of the seven results onto any of the ten read ports.

Write data is driven from SEFs outside the array macro, routed and repeated to the array center on global write bitlines and multiplexed with write data from units on the other side of the array as necessary (Fig. 11). Selected write data is then driven from the array center using local write bitlines to all

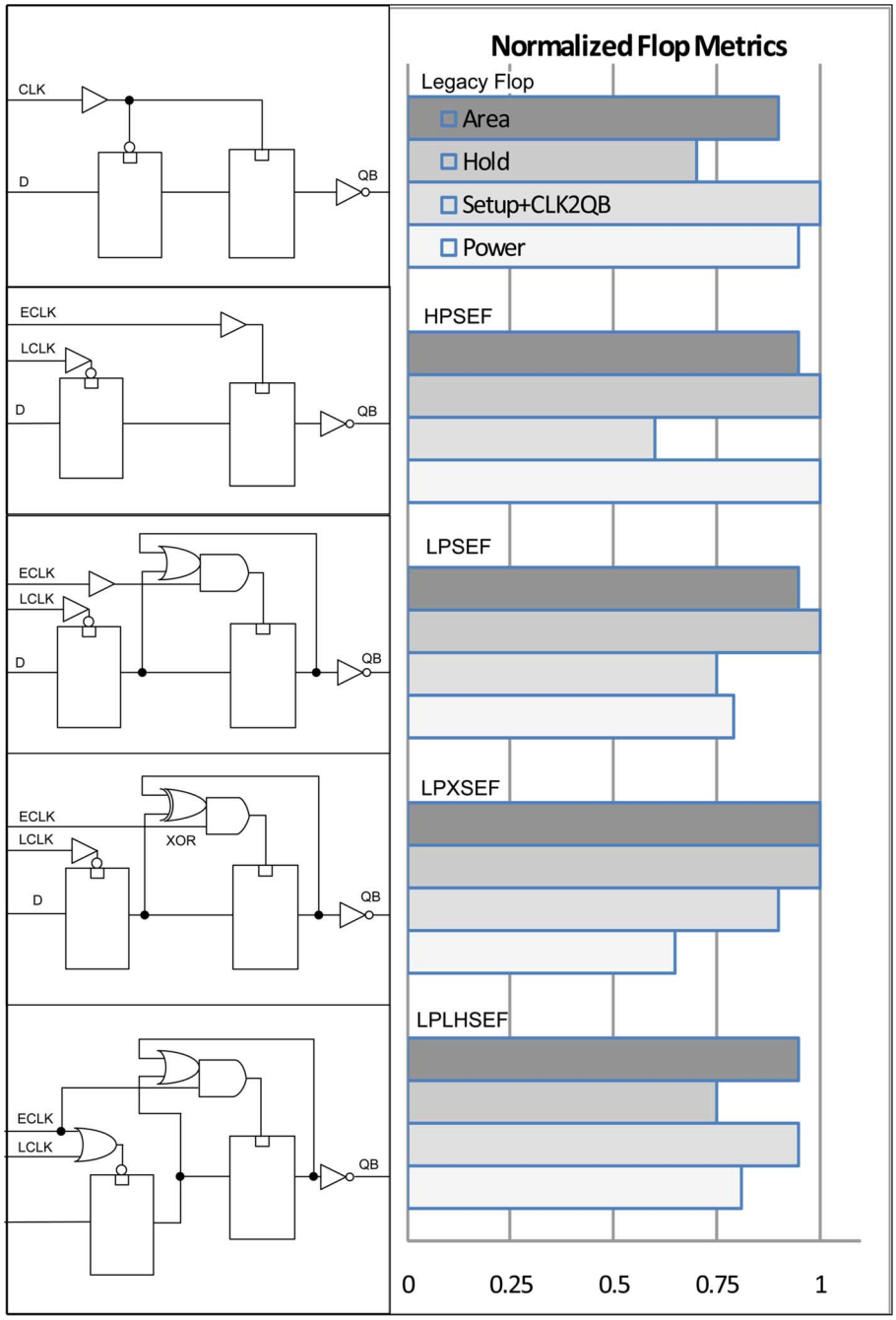


Fig. 7. Normalized SEF Metrics (lower value is better).

bitcell write ports. While the use of global and local bitlines requires additional route resources, it also enables balancing the setup and hold requirements for write data. This RC-balance technique reduces clock power by eliminating the need for write latches while simplifying array construction.

V. Integer Unit

The 40-instruction OOO unified integer scheduler issues up to four operations per cycle, to two ALUs and two AGUs, and supports single-cycle wake-up, bypass, and execution of dependent operations. AGU operations include increment/decrement (INC), address-generate, and x86-64 LEA instructions. Timing-critical paths, including the ready operation wake-up and pick loop (Fig. 12), broadcasting the picked instruction from the scheduler to the execution unit, and operand bypass between execution units, drive integer unit floorplanning (Fig. 13).

To achieve the high frequency required of the core, a low-latency edge-triggered clock gater is used in critical paths (Fig. 14), including the wake-pick loop and the execute-bypass loop. Setup time requires only that the input and output of the data inverter are equal before clock switches. Cross-coupled PMOS keeper devices allow the dynamic node to recover from false evaluation before the input switches full-rail and do not fight the NMOS evaluation devices. A self-resetting NMOS keeper holds data for a full clock phase. The fast combination

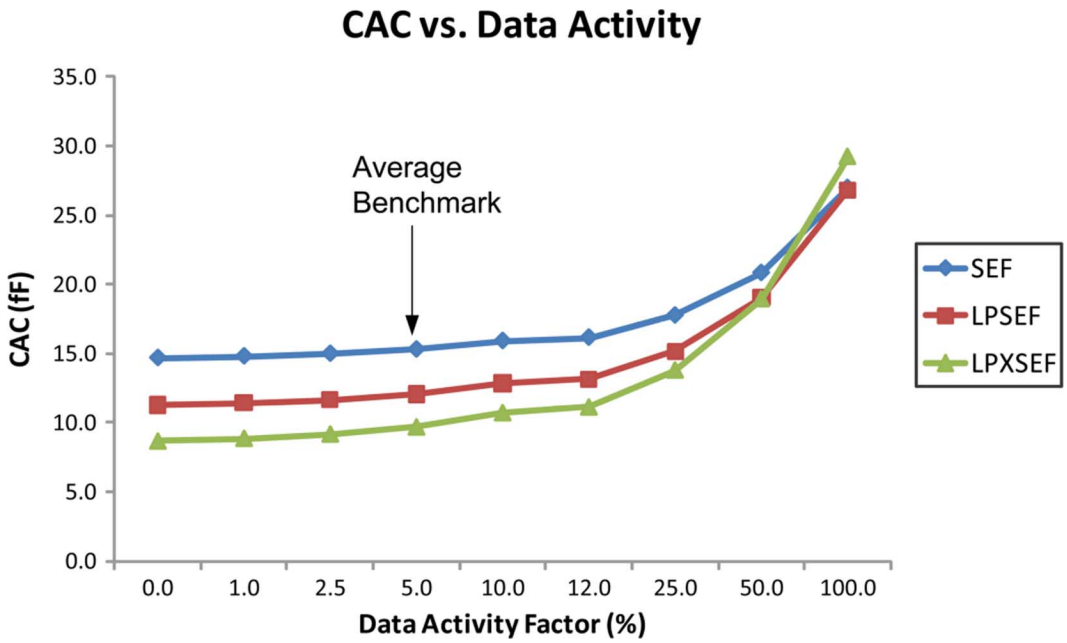


Fig. 8. Simulated flop power versus data input activity factor (CAC = capacitance × clock enable activity factor = $\mathrm{P}/(\mathrm{V}^{2*}\mathrm{f})$).

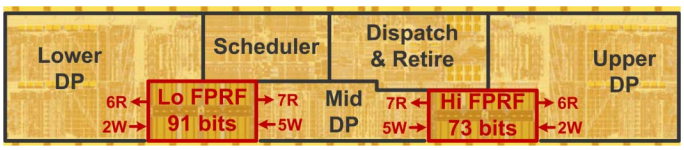


Fig. 9. Floating-point unit floorplan.

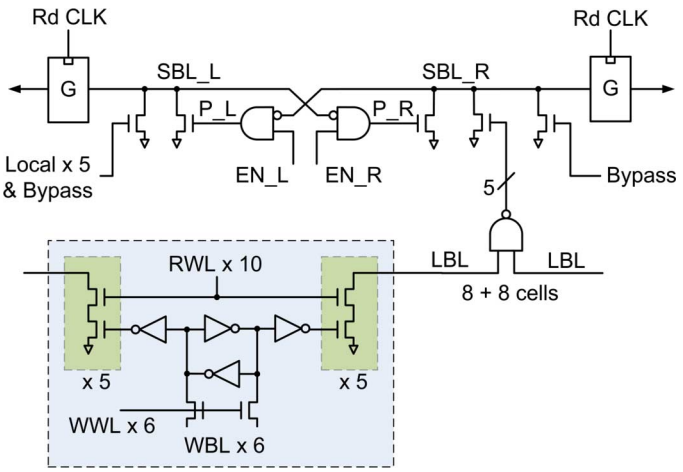


Fig. 10. FPRF read scheme.

of a two-high NMOS stack followed by an inverter determines the clock-to-output latency.

To remove wire delay and congestion from the critical register file read and bypass paths, a four-read/four-write register file is placed twice in the design to serve as an eight-read/four write physical register file (PRF). The critical half of the PRF, further from the output driver, has fewer pull-downs on the local bitline, reducing slew rates on critical words with minimal area impact. The INC unit is redundantly duplicated, reducing bypass latency to the ALUs. Load and store addresses never have to be driven into the bypass network, so address generation is less critical. ALU results have to bypass into both AGUs; because the increment operation is simple, the extra wire delay does not limit timing.

Critical paths, including post-wake and pick logic and the ALU adders, are implemented with static CMOS logic, with beta ratios skewed higher (1.5×) or lower (0.6×) than the natural process beta ratio (Fig. 15). Skewed gates favor the evaluate edge of the set-dominant RS latch output relative to the reset edge. Although wide OR functions in the post-wake and pick logic are amenable to domino logic, re-clocking the RS latch outputs would be susceptible to duty-cycle variation. When area constraints permit, full-custom arrays are converted to standard cell designs. An array of transparent-low latches (B-latches) replaces bitcells as storage elements. A logical AND function of the latch output with clocked wordlines produces local bitlines. The bitlines are driven into wide static OR functions. The critical edge is always driven into a one-high transistor stack; stacked devices drive only the non-critical reset edge. Low-$\mathrm{V_T}$ gates, normally forbidden in domino gates due to leakage concerns, can be used. Placement flexibility allows floorplanning optimizations, including placing a regular array into an irregular region. These techniques yield almost all the speed of a full-custom design, reduce design effort, and allow the exploration of a large design space in short time.

Although most register-file bit reads in the payload are full-swing and single-ended, some critical bits are read dual-rail (Fig. 16). This has several advantages: it allows the use of cross-coupled keepers, against which evaluate pull-downs do not have to fight; it eliminates inverters in data paths; it permits the use of skewed logic gates for both true and complementary data; and, it increases tolerance of jitter duty-cycle variation by allowing

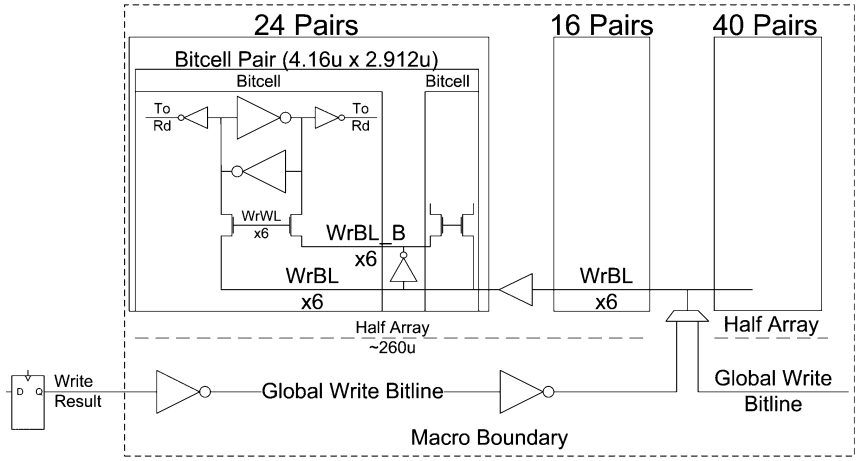


Fig. 11. FPRF write data distribution.

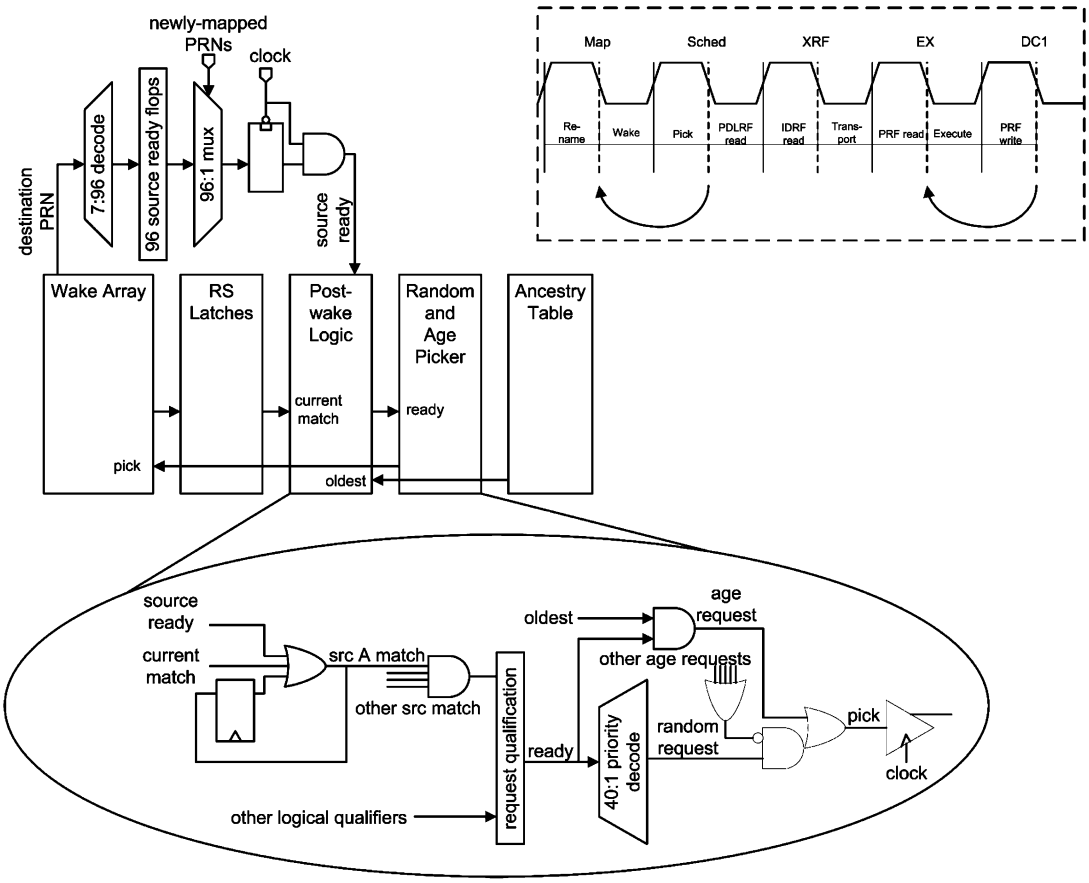


Fig. 12. Picker logic.

late-arriving bits to flow through clock gaters. Because dual rail-reads are used for only the most critical bits, the affected array read power increases by only 5%–18% depending on workload, without significant power increase outside the arrays.

To reduce active power, the scheduler stores only tags: physical register numbers (PRN) of destination and source operands. All structures except the PRF and functional units manipulate 8-bit PRNs (seven bits plus parity) instead of 64-bit instruction data. Instead of a shifting, collapsing structure to preserve age information, an ancestry table keeps track of the oldest instruction in the scheduler. The oldest instruction is picked, if ready. If not, a priority encoder scans ready instructions in physical order, which does not correspond to age order. When an operation is picked for execution, the destination PRN is read from a register file and broadcast to a fully associative content-addressable memory (CAM), which stores up to four source operands for every instruction in the scheduler. Pairs of destination bits are decoded before broadcast, reducing the number of switching bitlines. Rather than pulling down on a "mismatch" signal if a single bit mismatches in the CAM, a logical AND structure is

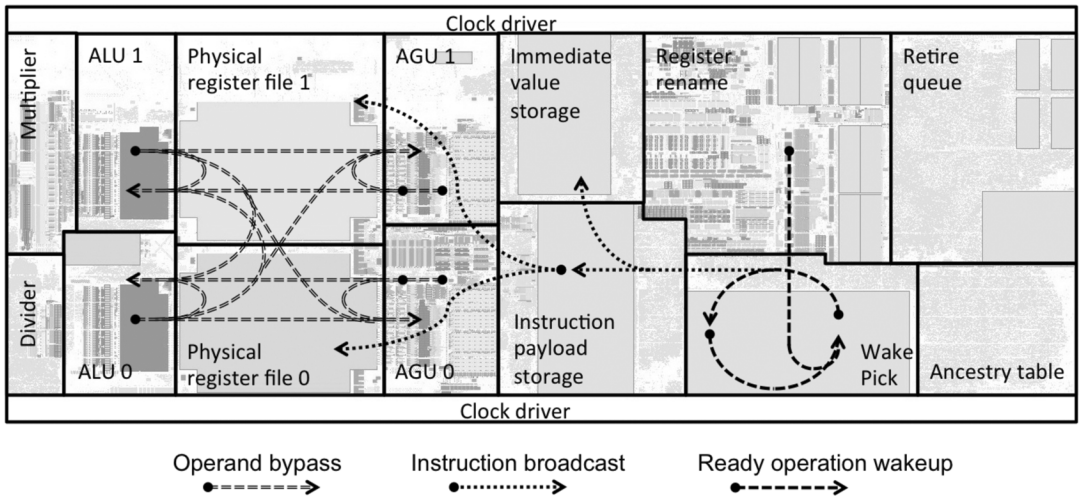


Fig. 13. Integer unit floorplan showing critical paths.

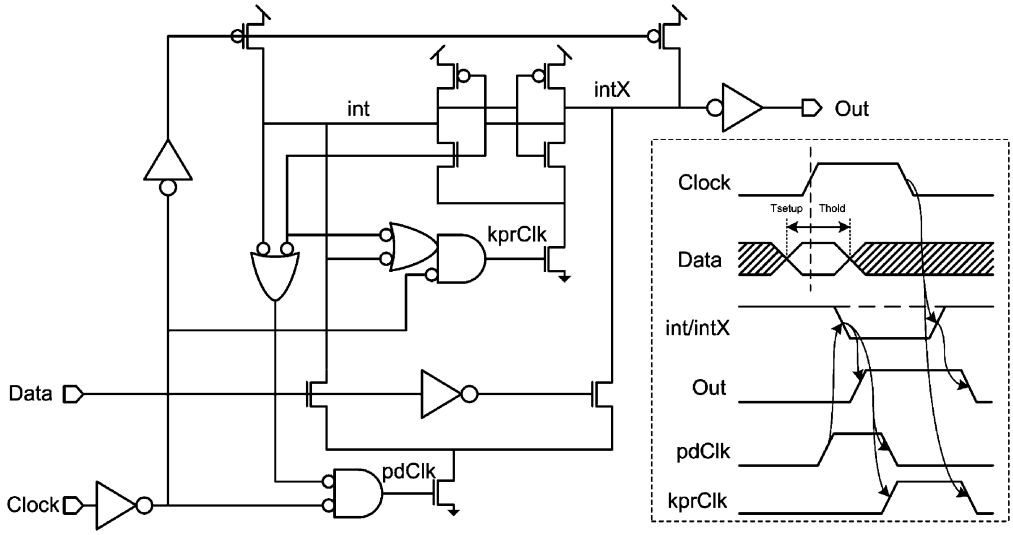


Fig. 14. Edge-triggered low-latency clock gater.

used (Fig. 17). As a result, fewer CAM structures switch because match is less likely than mismatch.

VI. SRAM and Register File Methodology

All L1 SRAMs and register files use strongly write-optimized cells to enable robust low-voltage write operation. The design removes write column muxing on all arrays outside the L2 cache, so all of the bits on a local write wordline always write. This removes the "dummy read" disturb seen by non-selected columns in a column-muxed SRAM, thereby avoiding the requirement for the cell to have a stable static noise margin with the write wordline on and the bitlines starting close to V_{DD}. Instead, the write bitlines can be fully static, providing a useful power saving, and the cell is more robust. As well as improving V_{DDMIN}, the write-optimized cell also improves write speed at nominal voltages. In practice, the removal of column muxing was only difficult for a single array that writes only two bits at a time, for which the solution was read-modify-write. In contrast, few register files would use column muxing even if available.

One potential concern with the removal of column muxing is spacing for multi-bit soft errors. The solution was to add multiple interleaved parity or ECC bits as required to maintain the same SER spacing. For example, if the multi-bit SER rule called for parity with a column spacing of two, the solution was an even parity bit for bits 0, 2, 4, etc., and an odd parity bit for bits 1, 3, 5, etc. For almost all arrays with 20–30 columns or more, the penalty because of extra parity bits is minimal.

The second array design goal was effort reduction. Wherever possible, we replaced custom array macros with scanned flop or unscanned transparent-low B-latch arrays. These standard cell arrays were hand-placed but gained from the benefits of static design, standard cell design efficiency, and the ability to use flexible placement to fit in non-rectangular placement areas. The flop arrays also allow standard scan-based testing and data dump for debug rather than MBIST. The much smaller B-latch arrays, in contrast, give a useful area saving but require MBIST or a modified sequential scan-based test mode [8] whereby all entries are written to one data value in cycle N, then one entry is written to the complementary value in cycle $N + 1$.

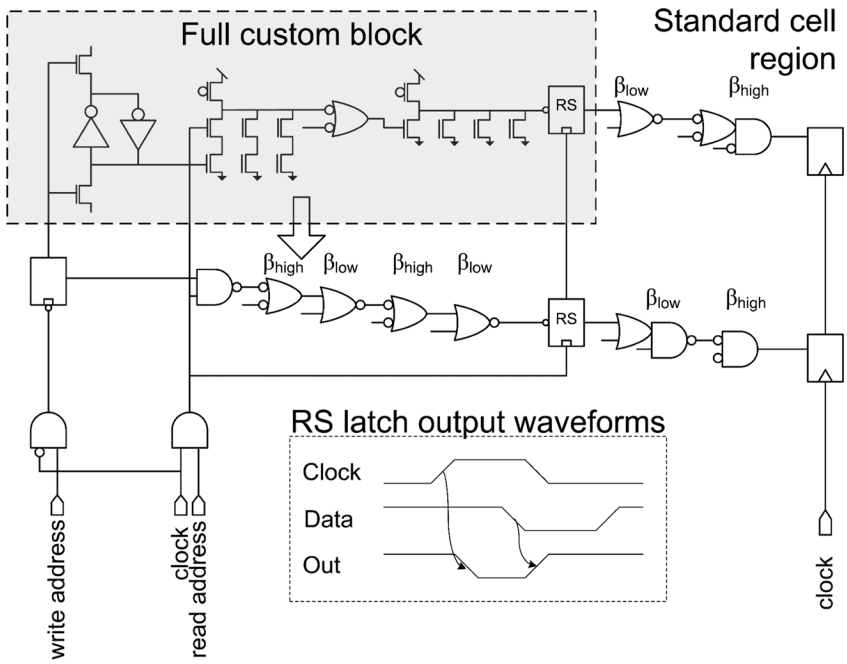


Fig. 15. Array implemented as standard cells.

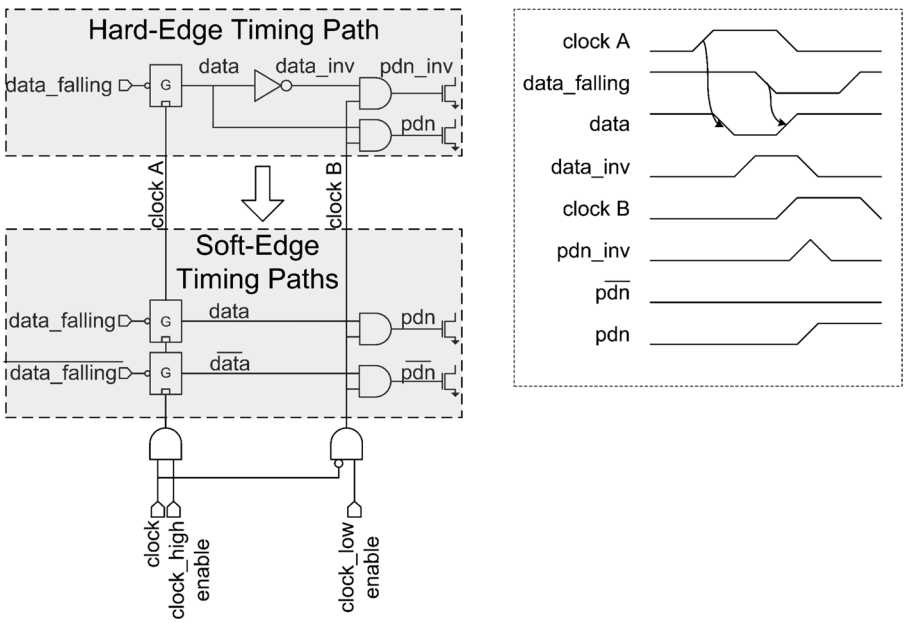


Fig. 16. Eliminating hard timing edges and phase paths.

Overall, 36% of the unique array designs are flop or B-latch arrays (30% and 6%, respectively), while a further 28% of arrays use a custom design style but are built from a register file compiler. Excluding the ROMs, this means only 23% of arrays are custom register files or L1 or L2 SRAMs. However, 75% of the storage bits are in the dense L2 SRAMs and a further 11% each are in L1 SRAMs and ROMs. The compiled and custom register files comprise 3% of the bits, and the flop and B-latch arrays are only 0.4% and 0.2% of the bits. This means the area cost of standard cell and compiled arrays is minimal.

All of the L1 SRAMs use a 0.294-μm^2 8T cell with single-ended sensing, eight cells per local bitline, and delayed-onset keepers as described in [4]. There is no self-timed or low-swing logic in the core outside the L2. The L2 SRAMs, in contrast, use a dense 0.258-μm^2 6T cell, conventional full-swing sensing, and column muxes to minimize area.

The final array design goal was power saving and avoidance of architectural critical paths. For all of the L1 SRAMs except the D-cache, the 8T SRAMs are used in single port mode (read or write per cycle, but not both), enabling denser array layouts because of R/W decoder sharing. The D-cache, in comparison,

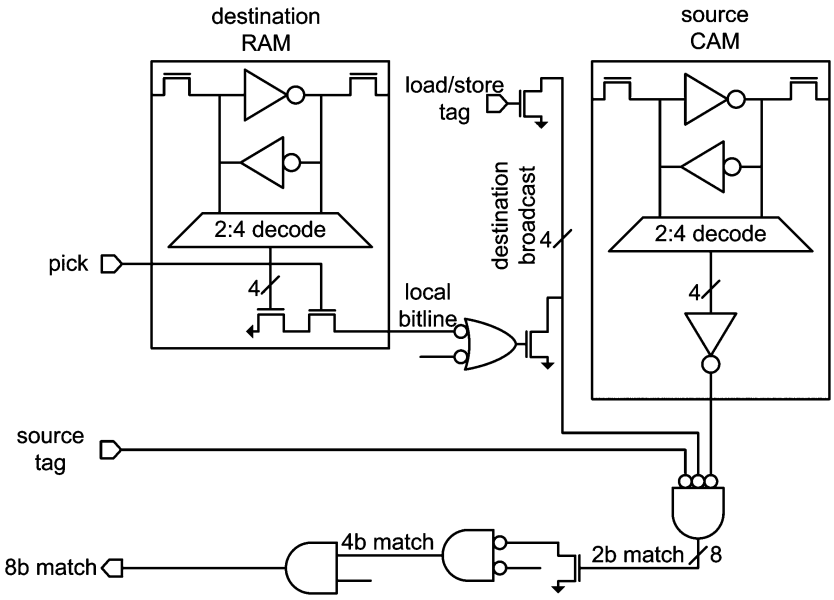


Fig. 17. Wake-up array logic.

is sized at 16 KB per core but supports two reads and a B-phase write per cycle. This achieves the same one-of-16 bank conflict rate as a previous 64 KB design by performing micro-banking at a 32-row level. The 8T cell and a change to a write-through cache removed a difficult read-modify-write critical path from prior designs, further helping performance by removing read/write bank conflicts.

Finally, the new pipeline was used as an opportunity to replace dynamic mismatch CAMs with lower-power static or match structures, remove power-intensive shifting queues, and minimize unnecessary instruction retries via wider usage of wake-up CAMs. As an example of this, the I-TLB and D-TLB use static CAMs, which avoid power consumption if the page address is unchanged, and a power filter to avoid reading the dynamic RAM in this case. The static CAM consumes approximately 40% as much power as a dynamic CAM, and the power filter avoids RAM reads in approximately half of the TLB accesses.

VII. Clocking

The high frequency and power saving goals for Bulldozer put a premium on efficient yet low-skew clock distribution. With clock skew accounting for a significant portion of total cycle time, clock-skew minimization is even more important for high-performance microprocessors [9]. At the same time, the clock power including ungated portions of the gater represents 11% of the peak power and up to 29% of idle power, so power reduction is also important.

The overall architecture of the clock distribution on AMD's Bulldozer core is shown in Fig. 18. The final level of clock distribution uses a single clock grid for low skew, supplied by vertical M11 spines that run along the vertical extent of the core on the top-most M11 layer, driven from five horizontal banks of clock drivers whose size was tuned based on the load. Horizontal clock stubs then extend from these spines over standard cell clock rows that contain the clock gaters, which are the final stage that drives the flops.

Clock power-reduction efforts targeted two areas: 1) reduction of parasitic clock distribution wire capacitance, and 2) reduction of end-load capacitance. Since end loading reductions have a multiplicative effect because they also reduce the demands on clock distribution wiring infrastructure, special attention was given to the overall design and stage gain of the clock gaters used in the design. By establishing a minimum stage gain design guideline for the clock gaters, the end loading was reduced by an average of 3% per gater over a previous project [4]. Fan-outs in the pre-clock H-tree were also optimized to minimize H-tree power. Meanwhile, grid capacitance was reduced by doubling the number of standard cells per clock row to 36 compared to a legacy clock grid (thereby reducing the horizontal spine count) and clustering the clock gaters closer to the vertical clock spines, thereby reducing the horizontal wire load. This design choice places more of the clock skew budget on the post-clock gater side of the distribution, which can be designed to compensate with increased wire sizes. Since the post-gater routes benefit from the clock-enable activity factor reduction, overall power was reduced compared to the denser clock grid option. The final grid capacitance was balanced with 48% wire and 52% gate load. The overall grid skew is less than or equal to 8 ps, with a further 4 ps budget from the final clock gater to the flops.

Once the end loads were known in both capacitance value and location within the clock grid the clock wire tuning effort was started. First, grid wiring was optimized assuming ideal clock drivers. Second, final clock driver drive strengths were determined and binned to available sizes in the clock library. Third, final analysis using real clock drivers verified the performance. One complicating matter was the fact that L/R ratio of the 16× metal layers employed in the clock grid was high enough for those layers to exhibit significant transmission line behavior, as illustrated in Fig. 19. The model for Fig. 19 was a uniformly loaded clock spine with a total span of 1.1 mm, and M11 wire width of 1.2 μm and spacing of 1.4 μm. The illustrated waveforms were at the final clock driver output and the approximate mid-point in the clock grid. To address this chal-

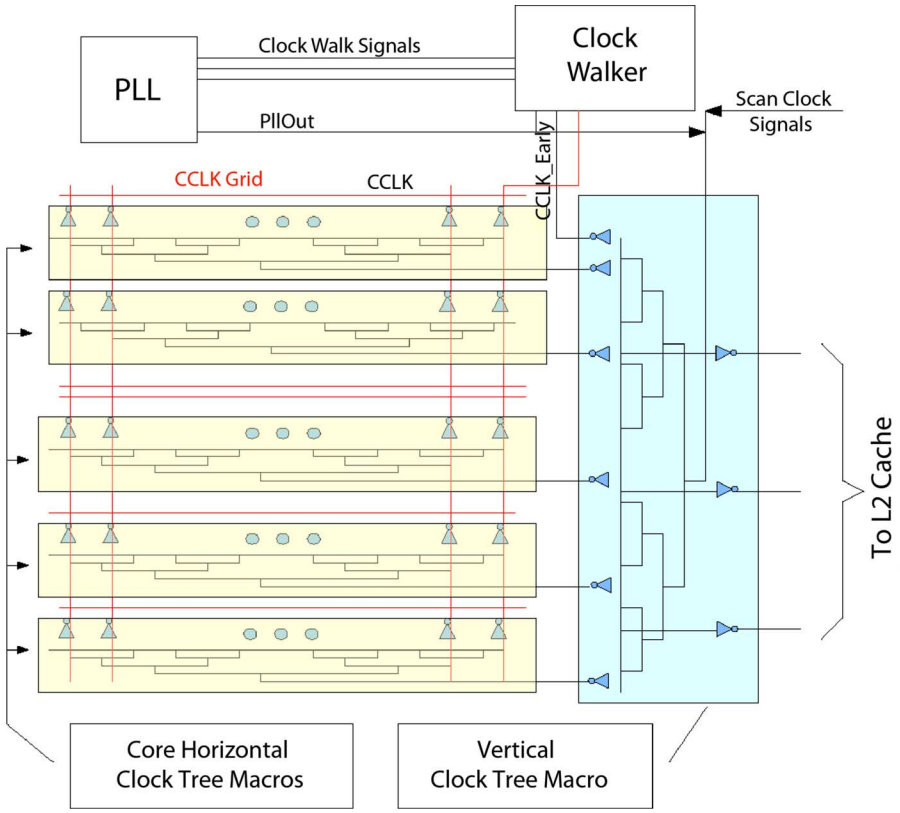


Fig. 18. Core clock distribution architecture.

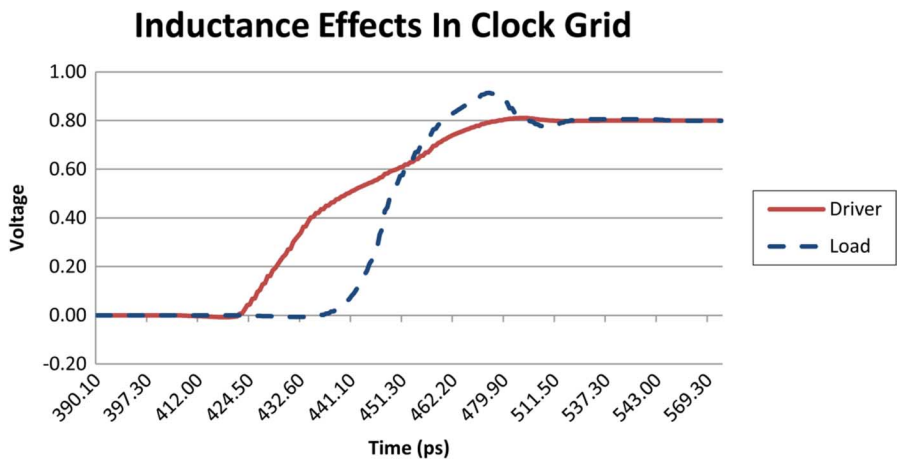


Fig. 19. Transmission line effects in clock network.

lenge, the wire-tuning algorithms were modified to avoid wire width and space combinations that would lead to signal integrity problems in the clock grid.

VIII. Conclusion

This paper describes key circuit innovations in AMD's Bulldozer, a new x86-64 micro-architecture implemented in 32 nm high-K metal gate SOI CMOS. Timing optimization at both high and low voltages results in reduced frequency roll-off at high and low voltages compared to previous core designs while achieving significant frequency improvement across the operating voltage range. The design has more than 20% fewer F04 gates per cycle; this paper describes several of the circuit innovations required to implement this. Achieving power efficiency in combination with high-frequency design is a particular challenge, and this paper also describes several unique approaches to power optimization that have been employed in the design. The gate count reduction and power optimization enable faster frequencies in the same power envelope compared to previous designs.

Acknowledgment

The authors gratefully acknowledge the work of the dedicated and talented members of the design teams in Sunnyvale, CA, Fort Collins, CO, Boxboro, MA, Austin, TX, and Bengaluru, India, as well as the many team members in Product Engineering, CAD, and DFT who helped to bring the design into production.

REFERENCES

[1] M. Butler *et al.*, "Bulldozer: An approach to multithreaded compute performance," *IEEE Micro*, vol. 31, no. 2, pp. 6–15, Mar./Apr. 2011.

[2] T. Fischer *et al.*, "Design solutions for the Bulldozer 32 nm SOI 2-core processor module in an 8-core CPU," presented at the IEEE Int. Solid-State Circuits Conf. (ISSCC), San Francisco, CA, Feb. 2011.

[3] M. Golden *et al.*, "40-entry unified out-of-order scheduler and integer execution unit for the AMD Bulldozer x86-64 core," presented at the IEEE Int. Solid-State Circuits Conf. (ISSCC), San Francisco, CA, Feb. 2011.

[4] R. Jotwani *et al.*, "An x86-64 core in 32 nm SOI CMOS," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 162–172, Jan. 2011.

[5] N. A. Kurd *et al.*, "Westmere: A family of 32 nm IA processors," presented at the IEEE Int. Solid-State Circuits Conf. (ISSCC), San Francisco, CA, Feb. 2010.

[6] V. Srinivasan *et al.*, "Optimizing pipelines for power and performance," presented at the 35th Int. Symp. Microarchitecture (MICRO-35), Istanbul, Turkey, Nov. 2002.

[7] Advanced Configuration & Power Interface Specification Rev 4.0a. Apr. 2010 [Online]. Available: http://www.acpi.info/spec.htm

[8] M. Yilmaz *et al.*, "The scan-DFT features of AMD's next-generation microprocessor core," presented at the 2010 Int. Test Conf., Austin, TX, Paper 2.1.

[9] P. Restle and A. Deutch, "Designing the best clock distribution network," in *1998 Symp. VLSI Circuits Dig. Papers*, 1998, pp. 2–5.

Hugh McIntyre (M'92) received the B.A. and M.A. degrees in engineering and electrical and information sciences in 1987, and the Diploma in computer science in 1988 from Cambridge University, Cambridge, U.K.

He is currently a Principal Member of Technical Staff at AMD, where he has worked since 2005 on design of x64 CPUs, including SRAM, custom macro and optimization of standard cell designs, including the Load-Store Unit for Bulldozer. He is currently working on custom designs for future AMD processors. From 1996 to 2005, he worked at Sun Microsystems on SRAMs and custom circuits, including memory compilers, for multiple UltraSPARC processors. Prior to joining Sun, he was with Inmos Limited, then ST Microelectronics, working on high-speed SRAMs, Flash EPROMs, and a media processor.

Mr. McIntyre holds seven U.S. patents and has three other IEEE publications.

Srikanth Arekapudi (S'98–M'04) received the B.Tech. degree in electronics and communication engineering from the National Institute of Technology Warangal in 1999, the M.S. degree in electrical and computer engineering from the University of Massachusetts at Amherst in 2001, and the D.Eng. degree in electrical engineering from Stanford University, Stanford, CA, in 2004.

He worked at Agilent Technologies and Silicon Graphics Inc. before he joined Advanced Micro Devices in 2003. At AMD he contributed to several custom circuit and stdcell designs for K8 and Bulldozer based projects. In the past, he served as a lead for Integer Execution Unit and High Speed Clock Design. He is now involved in the design of next-generation x86 processors.

Eric Busta (M'11) received the B.S. degree in electrical engineering with the Computer Science option from the University of Wisconsin—Madison in 1995, and the M.S. degree in electrical engineering from the University of Wisconsin—Madison in 1998.

He worked for Rockwell-Collins in Cedar Rapids, IA, for a year, working on native Java microprocessor technology. He joined Hewlett Packard in 1999 to work on high performance integer datapath design. He then joined the Itanium2 Joint Development Team with Intel Corporation, where he continued to work on integer datapath design and post silicon debug. He joined Advanced Micro Devices in Fort Collins, CO, at the Mile High Design Center in 2006 to work on Floating Point datapath and register file design.

Tim Fischer (M'06) has worked in high-speed digital circuit design for 22 years. He received the Masters degree in computer engineering from the University of Cincinnati, Cincinnati, OH, in 1989. He then worked for Digital Equipment in Hudson, MA, until 1998 designing VAX and Alpha CPUs. From 1998 to 2006, he worked on Itanium CPU design with Hewlett-Packard and Intel in Fort Collins, CO. Since 2006, he has been an AMD Fellow working on CPU circuit design and methodologies. He is currently involved in circuit design methodology for future CPUs.

Michael Golden (M'11) received the B.S. degree in computer engineering in 1990 and the M.S. degree in electrical engineering in 1991 from the University of Illinois at Urbana-Champaign. In 1995, he received the Ph.D. in computer science and engineering from The University of Michigan, Ann Arbor, MI. Both his M.S. thesis and his Ph.D. dissertation involved work with compilers.

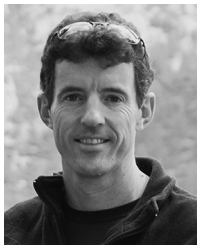
He joined AMD in 1995 and has been working there on CPU design ever since. At AMD he has helped develop CPU performance models, designed high-performance full-custom register files and SRAMs, led medium-sized design teams, and worked on timing and power methodology.

Aaron Horiuchi (S'99–M'03) received the B.S.E. with Electrical specialty degree from the Colorado School of Mines, Golden, CO, in 2001 and the M.E. degree from Colorado State University, Fort Collins, CO, in 2007. He received his Colorado state P.E. license in 2007.

Mr. Horiuchi joined Hewlett Packard in 2001, working on PA-RISC and Itanium processor and chipset development on high-speed digital links, SRAM design, and electrical test and debug. In 2007 he joined AMD, where he has worked on L3 cache and cache controller design, circuit methodology, and floating point design for next generation x86 processors. He holds two issued patents.

Tom Meneghini (M'88) received the B.S.E.E. degree from University of Lowell, Lowell, MA, in 1986, and the M.S.E.E. degree from University of Massachusetts, Lowell, MA, in 1992.

He joined Honeywell in 1986 to work on PCs and minicomputers before returning to school for his Master's degree in 1990. In 1992, he joined Hewlett Packard and worked on power systems until 1995 when he relocated to Fort Collins, CO to work on PA-RISC microprocessor design. In 2000, he joined Sun Microsystems to work on the Millennium processor project. In 2004, he joined Advanced Micro Devices to work as a Member of the Technical Staff where he worked on arrays and other macros. Most recently, he designed and implemented the clocks for the Bulldozer core module, and continues to focus on clock design. He holds three patents in the area of circuit design, and has one other IEEE publication.



Samuel Naffziger (M'02–SM'10) received the B.S.E.E. degree from the California Institute of Technology, Pasadena, CA, in 1988, and the M.S.E.E. degree from Stanford University, Stanford, CA, in 1993.

He has over 22 years experience in microprocessor design, having led the implementation of PA-RISC, Itanium and AMD processors working for Hewlett Packard, Intel, and AMD. He joined AMD in 2006, helping start the Mile High Design Center in Fort Collins, CO, where he has been responsible for power and frequency optimization of mainstream processors and is the power efficiency architect for AMD's products. He holds 96 U.S. patents on processor circuits and architecture and has over 25 IEEE publications and presentations. He is a Corporate Fellow at AMD.

Mr. Naffziger chaired the Digital subcommittee of the IEEE International Solid-State Circuits Conference for five years. He has been an Associate Editor for the JSSC.

James Vinh received the B.S.E.E. degree from the University of Washington in 1988.

He has over 20 years of experience in high speed circuit design, going through successful designs of multiple microprocessor architectures including Sparc, Mips and X86 at Hal Computer, Mips Technology, and AMD. He enjoyed working on sophisticated designs from scratch, and that was one of the main reasons he joined AMD in 2005 to help kickstart the Bulldozer project. His interest in extremely fast but noise proof circuit design techniques led to four issued patents and five other recent patent applications. He currently manages a design team to own various units including the Integer Scheduler, Instruction Decoder, and the Branch Prediction Units in AMD next-generation microprocessor projects.