The NVAX CPU Chip: Design Challenges, Methods, and CAD Tools

By Dale R. Donchin, Timothy C. Fischer, Thomas F. Fox, Victor Peng, Ronald P. Preston, and William R. Wheeler

1 Abstract

The NVAX CPU chip is a 1.3 million transistor, VAX microprocessor designed in Digital's 0.75-micrometer CMOS-4 technology. It has a typical cycle time of 12 ns under worst-case operating conditions. The goal of the chip design team was to design a high-performance, robust, and reliable chip, within the constraints of a short schedule. Design strategies were developed to achieve this goal, including organization of a chip design flow and new implementation and verification methods. New proprietary CAD tools also played important roles in the chip development.

2 Introduction

The NVAX CPU chip is a 1.3 million transistor, VAX microprocessor designed in Digital's 0.75-micrometer fourth-generation complementary metal-oxide semiconductor (CMOS-4) technology. The implementation of the NVAX CPU chip posed many design and complexity management challenges. The combination of the chip performance goal, the high level of integration, and the small feature sizes of the CMOS-4 technology increased the severity of on-chip electrical issues and the difficulty of verifying the physical design. These challenges were intensified by a short design schedule. This paper describes some of the design strategies, methods, techniques, and proprietary computer-aided design (CAD) tools used by the chip design team, which were instrumental in meeting these challenges.

3 Chip Overview

In order to appreciate the design challenges that were faced, it is necessary to understand the size and complexity of the design. The NVAX CPU chip has a macropipelined microarchitecture and implements the VAX instruction set.[1] Because it is a complex instruction set computer (CISC) architecture, the VAX architecture implements variable length instructions with complex addressing modes, and precise traps and exceptions. The chip is composed of five subchips, or functional boxes, called the I-box, E-box, F-box, M-box, and C-box.

The I-box fetches, parses, and decodes instructions, and predicts conditional branches. The E-box runs under microprogrammed control and executes all instructions, except floating-point and longword integer multiply instructions, which are executed by the F-box. The M-box processes memory references, including virtual-to-physical address translation. The C-box controls the off-chip backup cache (the second-level cache for data and third-level cache for instructions) and contains the bus interface

unit. The chip also has a direct-mapped 2 kilobyte (KB) virtual instruction cache (VIC), a two-way, set-associative 8KB data and instruction primary

cache (P-cache), a 12KB control store read-only memory (ROM), a 96-entry, fully associative translation buffer, and a 512-bit by 4-bit conditional branch history random-access memory (RAM) for branch prediction. The chip photomicrograph, with box and large array boundaries outlined, is shown in Figure 1.

The NVAX chip's layout database is composed of over 4,200 unique custom cells, and has a total transistor count of 1.3 million. It was the first product chip to be implemented in Digital's 0.75-micrometer, three metal layer, 3.3-volt (V) CMOS-4 technology.[2] The chip's typical cycle time under worst-case operating conditions is 12 nanoseconds (ns) or 83.3 megahertz (MHz), and it dissipates 14 watts (W). A summary of the chip features is given in Table 1.

NOTE

Figure 1 (NVAX Chip with Boxes and Large Arrays Outlined) is a photograph and is unavailable.

Note: *Through-hole pin grid array.

4 Design Goals and Constraints

Our design goal was to develop a high-performance chip that is electrically robust and reliable. We had to accomplish this within the CMOS-4 process constraints and the design time allotted by the development schedule. Our initial implementation schedule was based on scheduling metrics from previous designs. Due to competitive marketing pressures, this schedule was substantially reduced, making it significantly more aggressive than for previous designs. Our cycle time was constrained to a maximum of 14 ns under worst-case conditions. Electrical reliability had to be guaranteed for cycle times down to 10 ns under worst-case conditions. Based on CMOS-4 process limits, the chip die size was constrained by a maximum diagonal length of 21.8 millimeters (mm).

The trade-offs between design time and chip characteristics, such as performance, area, and function, were the dominant issues throughout the project.

5 Design Strategy and challenges

To achieve the goal of a 14-ns cycle time, we designed custom circuitry and layouts, including dynamic, asynchronous, and differential logic. To deal with the size and complexity of the chip, together with the schedule constraint, our strategy called for a large design team. Complex, custom very large-scale integration (VLSI) chip designs inherently have high levels of design schedule risk. To reduce our exposure to schedule

slips, we made several high-level design decisions early in the project. Wherever possible, we avoided using circuit structures that were time-consuming to analyze. We defined and followed detailed design guidelines

to ensure design consistency, robustness, and reliability. We used a top-down design flow and made extensive use of proprietary CAD tools that were expressly developed for high-performance custom VLSI design. Lastly, we minimized chip area by hand-crafting layout in sections of the chip where the leverage on reducing overall die size was significant; or when critical path node had to be minimized to satisfy the path timing constraints.

The floor plan was accurately monitored throughout the project. This was essential because initial die estimates indicated that the chip size was close to the maximum size that the CMOS-4 technology could support. These strategic decisions reduced design time and allowed us to focus on achieving a fast CPU cycle time without compromising the quality of the design.

In addition to minimizing risks to the schedule, we had to solve several global design and verification problems to achieve the cycle time of 14 ns. The design team assumed a 10-ns cycle time when it addressed problems that are exacerbated by a faster cycle time, such as interconnect reliability and signal integrity. Some of the key concerns were on-chip power, ground, and low skew clock distribution, and the routing of long signal interconnects. Verification of the custom layout was another challenge, particularly given the schedule constraints. The use of CAD tools was a significant benefit in development of the chip. These issues are addressed in more detail in the remaining sections of this paper.

6 Design Flow and Management

The chip design team was organized into five semiautonomous groups, each of which focused on the design of a functional unit (C-box, E-box, F-box, I-box, M-box). To ensure design compatibility and consistency across the chip, each team adhered to the same design guidelines and methods. For example, box-level interfaces were rigorously defined, a consistent register transfer level (RTL) modeling style was used, and circuit noise margins, transistor sizing, and other circuit and layout guidelines were observed. The design team followed the top-down, hierarchical design flow depicted in Figure 2, but there was much overlap between the activities. Complexity was managed by thoroughly reviewing and testing the design at each level of abstraction (microarchitecture performance model, RTL model, logic, circuit, and layout), and by using CAD tools to verify that all the design representations were consistent across the levels of abstraction. When making design decisions, we considered the implications across the levels of abstraction. For example, when we made microarchitectural trade-offs, we considered the implications for power consumption, logic complexity, circuit speed and cycle time, layout, die size and, of course, schedule.

The NVAX CPU Chip: Design Challenges, Methods, and CAD Tools

Performance Model and Microarchitecture Specification

The NVAX performance model is a software program that models the microarchitecture of the NVAX chip. The architecture team used the model to study the effect of various microarchitectural factors on overall CPU performance and to define the chip's microarchitecture. The performance model was updated as the microarchitecture evolved so that the team could assess the impact of design changes on performance.

The chip microarchitects wrote a textual specification of the chip that documented its function and microarchitecture. As the details of the design were resolved, the specification was updated and expanded to reflect the actual implementation. The functional design verification team used the specification to develop implementation-specific tests.

RTL Model and Floor Plans

The design team developed a detailed RTL model of the chip once the chip specification was completed. This model was written in Digital's proprietary BDS hardware description language. As the BDS code was being written, many logic/circuit feasibility studies were spawned. The model was used to verify that the proposed microarchitecture executed VAX code correctly. It also served to guide logic implementation and was used by system design teams to develop modules based on the NVAX microprocessor.[3,4] The RTL model was updated as the project progressed.

The chip floor plan was devised early in the design to estimate and track the die size and to provide area-impact data for microarchitectural trade-offs. Once the chip-level floor plan was stable, the box design teams developed more detailed box-level floor plans. All floor plans were entered directly into the layout database and maintained throughout the project. Tracking the floor plan at several levels eased the difficulty of integrating the box layouts to form the full-chip composite late in the project. More details on floor plans and the use of CAD tools are described in the section Floor Plan Techniques.

RTL Verification and Schematic Design

We verified the RTL model by running a combination of pseudorandom tests, standard VAX architecture tests, and handwritten implementation-specific tests. In order to identify flaws before time-consuming schematic and layout changes were implemented, we ran regression tests on the model whenever we made changes to the design. To track design changes and issues, designers posted a description of the changes and issues along with the ramifications for other parts of the design in an electronic bulletin board. Each new entry to the bulletin board was mailed electronically to every member of the team. This tracking procedure helped reduce design

iterations caused by stale information.

We used the RTL model as a specification for logic/circuit design. To synthesize logic directly from the RTL model for circuits with less critical area and speed constraints, we used a CAD tool, OCCAM. Because these constraints were stringent for a large portion of the chip, engineers designed most of the circuits. We developed a library of common circuit and layout structures to reduce the design and verification effort. We defined and followed detailed design guidelines to ensure design consistency. More information on the types of circuits used on the NVAX chip is being published in "A 100 MHz Macropipelined VAX CMOS Microprocessor."[5]

Schematic Verification and Layout Design

We held design reviews and used CAD tools to check schematics for unintended deviations from the design guidelines. We performed extensive logic simulation on the schematics. We used SPICE for accurate critical path timing analyses, and a static circuit timing analyzer, NTV, to detect unidentified slow paths.[6] (NVAX timing verification is described in a later section.) Figure 3 is a histogram of slow paths as a function of cycle time that was generated by NTV about two months before we taped out the first pass of the chip. Because NTV does not predict circuit delays as accurately as SPICE, all questionable paths flagged by NTV were simulated using SPICE. Those that were found to be slow were redesigned to meet the target cycle time.

Logic and circuit changes resulting from these analyses and the impact of these changes on other design representations and verification tests were tracked on the electronic bulletin board. Since many people were working on the design simultaneously, detailed tracking of changes and open issues proved crucial to meeting our schedule. A single change often required modification and reverification of the RTL model, schematics, layout, verification tests, and in some cases the chip textual specification.

Layout design proceeded on a subbox, or section, basis. A typical section consisted of approximately ten related schematics. Each section was checked for connectivity and correctness after its layout was completed. Sections within a box were then integrated and verified (boxes contained from 5 to 15 sections) before the complete chip composite was assembled.

Schematic verification and layout design were performed concurrently during much of the project. Although this overlap led to design changes that increased the amount of layout rework, the chip development schedule would have been significantly longer if schematic verification and layout design had been done serially. Layout design took about a year to complete.

Layout Verification

Section- and box-level layout verification was performed in parallel with the layout design. Once layout design was completed, the final stages of layout verification ensured that the assembled chip layout met reliability and integrity guidelines for global nodes such as power, clocks, and signals. Most of the layout checks were performed by CAD tools that used the CMOS-4 layout design rules and NVAX-specific electrical rules. More details on layout verification are given in the section Layout Verification Tools.

7 Floor Plan Techniques

With 1.3 million transistors, nearly half a million signal nodes, and over 16 million polygons on the NVAX die, precise monitoring of the floor plan was critical. From the earliest area estimates, it was clear that the chip size was close to the maximum that the CMOS-4 technology could support. We had to ensure that the die did not exceed the technology limit.

Area Estimation and Placement

Preliminary estimates of the die area were made by extrapolations from previous CPU designs.[7,8] Better area estimates were developed for regular structures, such as the cache arrays and data paths, by creating test layout structures. More accurate floor plans of the control sections of the chip were developed after the RTL model was written. In most cases, the partitioning of the RTL model was a close approximation to the desired schematic and layout partitioning. To estimate the area of random control logic, we used the OCCAM logic synthesis tool, and in many cases Digital's proprietary layout synthesis tool, CLEO.

As seen in the chip photomicrograph in Figure 1, the clock generator and drivers (CLKGEN) were distributed in a narrow north-south channel near the center of the chip. That location was chosen to minimize clock skew. The I-box, E-box, M-box, and C-box subchips are located on the right side of the chip. Their relative positions were chosen to accommodate the flow of the pipelined data in the main data path, which runs north-south just to the right of CLKGEN. The VIC, F-box, and P-cache were placed on the left-hand side of the chip, adjacent to the boxes with which they communicate.

Interconnect Planning and Routing

After we determined the initial placements of all major control sections, we used a new two-level block router called GLOW to route the layout for the interbox and intrabox signals. Routing was performed at the same time schematics for the control areas were in design. Since layout did not exist for the control blocks, GLOW was allowed to route signals over the blocks with some restrictions, as well as route signals in channels. We influenced how GLOW routed signals by specifying some areas of blocks as opaque (no over-the-block routing permissible) and some as porous (over-the-block routing is permissible if channels are full). Typically, cell blocks that contained regular arrays (such as programmable logic arrays) or critical circuitry were identified as opaque, whereas most random control areas were identified as porous.

The capacitance values of the interbox and intrabox signal interconnects were extracted from the layout and used in circuit simulations of critical paths. In some cases, the placement of sections and the signal routing were altered to reduce interconnect capacitance on critical signals. The use of synthesis tools such as GLOW, OCCAM, and CLEO allowed a much more detailed floor plan to be developed than was typical for prior designs.[7,8] The ability to feed capacitance information from floor plan routing back into SPICE simulations proved invaluable.

8 Third Metal layer

The top aluminum interconnect layer (M3) in the CMOS-4 process was specifically designed to meet the electrical requirements of the NVAX chip. The third metal layer was designed for a low sheet resistivity and high current capacity in order to handle the electrical problems associated with the power grid, clock distribution, critical signal routing, and large array design.

Power and Ground Distribution

When the NVAX microprocessor is run at maximum speed, it draws a direct current of about 5 amperes. Due to CMOS switching transients, the alternating current peaks are considerably higher.

Individual cell layout did not contain M3. The power, ground, and clock connections for a cell were routed by short vertical M2 lines inside each cell. These M2 lines were connected to the M3 grids automatically by a CAD tool.

On-chip Clock Distribution

In order for us to meet the performance goals, it was critical to keep clock skews small and edge rates sharp across the chip. As shown in Figure 5, special attention was given to the clock distribution scheme. Differential outputs from an off-chip oscillator were supplied to a receiver located at the top of the chip. The output of the receiver was routed to the global clock generator (CLKGEN), which was placed at the center of the chip to reduce clock skew. The outputs of the global clock generator were buffered by four inverters to increase their driving capability. The clocks were then distributed, using the low-resistance third metal layer (17 milliohms per square), from the top to the bottom of the central clock routing channel that spans the chip.

Clocks were supplied to the different functional boxes by locally tapping off the central clock routing and buffering each signal with four inverters to further increase the signal's driving capability. This buffering helps to minimize the capacitive loads seen by the clock phases in the central routing channel in which the RC delays are held to 30 picoseconds (ps). To reduce distribution skew between the global clock lines, loading on each global line was balanced by adding dummy loads to the more lightly loaded lines. The buffered clock phases were distributed to the east and west of the central clock routing channel, again using M3 to reduce RC delay. The east-west clock routing was strapped with M2 as shown in Figure 5. These straps were not allowed to cross box boundaries. Box-level clock skew was reduced by using a common section buffer design and layout, and by carefully tuning the buffer drive capability to the clock load in each

section.

Finally, before the clocks were used by the logic, the clock signals were locally buffered. These final stages of local buffering served two purposes: they reduced the gate loading on the east-west clock routing, and they helped to sharpen the clock edges seen by the logic.

The global clock routing network was spaced so that the RC delays of local clock branches would never exceed a negligible 125 ps. We calculated the RC delays of local clock branches using the WAWOTH layout interconnect analyzer (described in the section New Proprietary CAD Tools) and, where necessary, rerouted branches to meet the 125-ps design goal. A sample RC plot, generated by WAWOTH for a section of local clock routing, is given in Figure 6. The clock skews and edge rates across this 1.62-centimeter chip are less than 0.5 ns and 0.65 ns, respectively.

Microcode Control Store

The design of the 12KB ROM control store was simplified by dividing it into four subarrays. Each subarray has its own M1 bit lines. The M1 bit lines from the subarrays are cascaded onto low-capacitance M3 super bit lines that extend over all four subarrays. Since the capacitance of the M3 super bit lines is low, the access time is very fast, obviating the need for sense amplifiers and voltage reference generators. This substantially reduced the time required to design and verify the control store ROM.

Primary Cache

A similar technique was used in the 8KB P-cache to ease the timing requirements. The three high-order P-cache address bits must be translated and consequently become valid later than the untranslated lower-order bits. By dividing the P-cache into eight subarrays, each with its own sense amplifiers, the cache subarray access can be started before the three translated address bits are valid. When the last three address bits become valid, the outputs of the subarrays are multiplexed onto the M3 super bit lines, resulting in a faster cache access time.

9 Layout Verification tools

Verifying the NVAX chip layout presented several CAD software challenges. Prior to the NVAX design, the existing layout verification tools were able to extract full-chip netlists from layout for all large designs in a single batch process. However, the existing layout netlist extractor could not handle designs such as NVAX with over one million transistors. Also, a more accurate capacitance extraction algorithm was required to calculate side-to-side and fringing capacitance, which became significant effects in the small physical dimensions in CMOS-4. Furthermore, accurate interconnect resistance extraction was needed for NVAX. A combination of new CAD tools (see Figure 7) and design methods was employed to meet the NVAX layout

verification requirements.

Partitioning Using "Clean Belts"

To address the problem of extracting parasitic capacitance data from such a large design, the NVAX chip layout was constructed so that each chip partition could be independently extracted without introducing inaccuracies in the results. The chip was partitioned into nonoverlapping regions, each of which had a rectilinear annulus or "clean belt" around its periphery. A clean belt is a rectangular region that contains only metal lines and satisfies a number of layout design rules beyond those set by the technology. The clean belt layout rules prevented design rule violations within the clean belt and between adjacent clean belts. The rules also ensured that extracting parasitic capacitance from a region enclosed by a clean belt could be done accurately regardless of the materials that border the region. Partitioning the chip in this manner made it easier to locate global wiring errors.

Hierarchical Netlist Extraction

A new netlist extractor, HILEX, was used to meet the high data capacity requirements of the NVAX microprocessor. HILEX is more efficient than the previous in-house netlist extractor because it recognizes layout cell instances and in many cases needs to extract cell-only definitions. In contrast, the previous netlist extractor "flattened" the layout data into one nonhierarchical cell and, therefore, extracted all data without reusing previously extracted cell definitions. The actual performance improvement realized by HILEX depends on the hierarchy of the chip layout design. If very few cells are replicated, or cells are replicated in a way that requires the extractor to explode the cells (i.e., create more flat data) to extract them properly, then minimal performance improvements are seen. An example of the latter situation is an array of a repeated pair of overlapping cells that forms one or more transistors due to the overlap (one cell contains diffusion areas that become source and drain regions when overlapped with the other cell, which contains polysilicon lines that become transistor gates).

Several layout design guidelines were defined to ensure that performance improvements from using HILEX would be realized. Adherence to the guidelines minimized situations that require HILEX to explode cells and encouraged the use of hierarchy in the layout. However, since it was not always possible to adhere to these design recommendations, HILEX was designed to handle large amounts of flat data.

The chip netlist was extracted from the complete chip layout prior to tape out. This presented quite a challenge since even with the use of HILEX, extracting the chip netlist from the 225MB chip layout file in one pass required more than the maximum of two million virtual pages of memory allowed by the VMS operating system architecture. To go beyond the VMS

virtual memory limit, the internal memory management routines within HILEX were modified to allocate additional heap from the process stack (in P1 space) when the VMS memory allocation routines indicated that P0 memory space was exhausted. This technique was used to allocate the 2.5 million virtual pages required for full-chip connectivity extraction.

Netlist Comparison

A utility called WLC was used to verify that netlists extracted from layout by HILEX matched netlists generated from the chip schematics. Since the NVAX schematic hierarchy rigorously matched the layout hierarchy only at certain levels, the connectivity comparison was performed flat. WLC employed a graph-building and graph-traversing algorithm that worked well for comparing less than 500,000 device connections. However, substantial paging occurred when comparing larger netlists. Since the NVAX chip contained 1.3 million devices, the performance of WLC was inadequate for full-chip netlist verification.

To improve the elapsed time of netlist comparison batch jobs, multiprocessing was employed. HILEX was modified to write the extracted netlist of the clean belt partitions. Each partition was then compared by WLC, in parallel on multiple CPUs, to its equivalent schematic-generated netlist. This approach reduced the total elapsed time for the NVAX chip netlist comparison from more than three days to seven hours. Cross-reference files output by WLC and the schematic netlists were processed by a separate program, MATCH_CHECKER, to verify the connectivity of nodes that crossed partition boundaries. This additional step added only eight minutes to the total elapsed time for comparing chip netlists.

Capacitance Extraction

The small spacing dimensions of the submicron CMOS-4 process caused fringing and lateral capacitances to contribute significantly to the total nodal capacitance. The existing layout extraction tool only extracted overlapping parallel plate capacitance. Thus, a new layout capacitance extractor, CUP, was written to accurately extract fringing, lateral, and area capacitances.

CUP extracted interconnect capacitance from layout by decomposing interconnect layout into pieces of uniform layout cross sections. The geometry of each interconnect piece, and its distance from layers above, below, and adjacent to it, are used to calculate its area, fringing, and lateral components of capacitance. The empirical formulae used to calculate the capacitive components were based on curves of two-dimensional electrostatic simulation data of various layout cross sections. This technique produced accurate internodal and total interconnect capacitance data. This accuracy resulted in CUP being very compute intensive.

Multiprocessing was employed again to reduce the elapsed turnaround time for capacitance extraction batch jobs. CUP sectioned the layout database into fixed-size stripes, which were inserted into the batch queues of multiple CPUs. This method reduced the data complexity and allowed as many parallel computations as there were processors. During NVAX chip

design, capacitance extraction was partitioned across as many as 20 CPUs. Multiprocessing reduced, for example, the NVAX I-box capacitance extraction from 26 hours to just 8 hours using 4 processors. Extraction of the E-box took 40 hours using one processor, but only 12 hours with 4 CPUs. Table 2 shows the device and node counts of the NVAX boxes (excluding the caches),

and the CUP extraction run times on a VAX 6000 Model 500. Each box run resulted in approximately 500,000 extracted parasitic capacitors.

Resistance Extraction

Verifying the NVAX power, ground, and clock networks, and long signal lines required accurate extraction of interconnect resistance from layout. To meet this requirement, the REX resistance extractor was developed.[9] REX processed the output of HILEX to produce a series and parallel combination of resistors that modeled a node's interconnect. The resistor network was generated by fracturing the node layout into polygons based on changes in the layout geometries (width, length, bends) of the node or the occurrence of contacts. The effective resistance of each polygon and contact, or cluster of contacts, was then determined from technology parameters and the polygon geometries.

The power and ground resistor networks were extracted for individual boxes rather than the entire chip. The resulting networks were still quite large due to the fine granularity of the REX extraction process. Table 3 shows the extraction times for a REX job running on a VAX 6000 Model 500 and the total number of resistors extracted from each box.

10 New Proprietary CAD Tools

Several other novel CAD tools were specifically designed for the NVAX chip. These tools provided practical solutions to verification and analysis problems that were previously unmanageable or intractable.

CHANGO Logic Simulator

CHANGO was an important development for NVAX functional verification because it allowed significantly more simulation cycles and functional verification tests to be performed from the NVAX transistor-level description than was previously possible. CHANGO is a two-state gate-level logic simulator designed to maximize performance through compiled, straight-line simulation. Electrical issues such as gate delay and charge sharing were not modeled since CHANGO was used for functional, not timing, verification. CHANGO's parallel simulation capability allowed the simultaneous execution of 13 different NVAX model simulations on one CPU, which resulted in an eightfold increase in simulation performance. Overall, CHANGO has been shown to accelerate simulation one to two orders of magnitude over traditional event-driven gate-level simulators. Its high throughput enabled us to boot the VMS operating system twice (75 million cycles) prior to tape out.

To create a CHANGO model, a transistor-level netlist description of the design was input to a preprocessor called GEN_MODEL. GEN_MODEL transformed

the netlist into a logical description of the design, consisting of simple Boolean elements, D-type latches, and SR flops. CHANGO transformed this logical description into a highly optimized simulation stream of VAX assembly code.

CHANGO achieved its high simulation throughput in many ways. Conditional branch latency penalties were largely avoided because CHANGO simulation code is designed to execute in a straight-line fashion. Due to the high switching event densities we observed on NVAX, 18 percent on average, this straight-line compiled approach to simulation was more efficient than event-driven simulators, which typically fail to compete when event densities increase beyond 3 to 5 percent. The CHANGO translation process further optimized the simulation by partitioning the simulation according to signals that should be evaluated during each particular clock phase. This avoided processing signals during clock phases when signal transitions could not occur. Further, evaluation of a switching event was only performed when the signal could affect the evaluation of some other signal. This prevented simulation of unimportant switching events that were ignored by the remaining design. Redundant signals (i.e., nodes with the same logical behavior) were grouped together as a list of synonym signals in order to model multiple nodes by only one simulation event.

NTV Timing Verifier

NTV is a static timing verification tool developed for use on the NVAX microprocessor.[10] NTV processed 350,000 circuit paths and checked 42,000 timing constraints on the NVAX design. NTV eliminated the need for the pattern-dependent dynamic speed verification strategy used by other chip designs and significantly reduced the extensive speed verification work needed for SPICE simulations. It identified critical paths that would have otherwise remained undetected due to the complexity and size of the NVAX design.

NTV read multiple flat transistor netlists with or without parasitics and automatically identified circuit structures such as complementary, dynamic, and cascode gates as well as several types of latches. Based on the classification of these structures, NTV identified timing constraints. For example, NTV checked that the latch storage nodes become valid before the latches closed. NTV also read user-specified timing for primary inputs and propagated node timing throughout the design based on when signals arrived at gate inputs, the drive capability of each gate, and its output loading.

NTV has three delay models that were used for calculating gate delay: (1) unit delay was used for an initial rough timing estimate before real parasitics were known, (2) a SPICE-calibrated lumped RC model was used for delay calculation of complementary gates, and (3) an Elmore-distributed RC model was used for other structures.[11] NTV flagged circuits that failed to meet the identified timing constraints within a user-specified time tolerance. Like other static timing verifiers, some paths identified by NTV were "don't cares" or were logically impossible. The user eliminated these false paths by deleting timing constraint checks or by specifying mutual

exclusivity between specified groups of nodes.

WAWOTH Interconnect Analyzer

Traditional manual techniques for checking RC delay, IR noise, and electromigration (EM) were impractical for NVAX due to the size and complexity of the design. A suite of CAD tools called WAWOTH was developed to perform these checks automatically, more accurately, and in far less time than would otherwise have been possible.

During EM and IR analysis, current sources representing gate-switching events were added to a REX-generated resistor network. The magnitude of each current source was calculated based on the average switching frequency of the gate, the load it drove, and whether average or peak current was desired for the current analysis mode. The network node voltages were then solved through LU decomposition. Peak voltages were flagged for IR analysis, and average and peak current densities were calculated for each resistor element and checked against EM limits.

During RC analysis, node capacitance was proportionately distributed along the resistor network. The resulting RC network was processed by Carnegie-Mellon's AWE algorithm to generate a close approximation of the transfer function for the network.[12] From this, the step response RC delay was calculated and the delay response to any specified edge was found through convolution of the transfer function.

Since it was neither possible nor necessary to perform RC and EM analysis on all signal nodes, WAWOTH contained a number of tools that identified only those nodes that would have some chance of failing these checks. To decrease run time, we reduced the size of the files that were input to WAWOTH by eliminating any devices and parasitics that were not related to the node under examination. Noteworthy were the large data requirements met by WAWOTH. For example, WAWOTH calculated the current through the 719,000 resistive elements that compose the power and ground nodes of the E-box. Current stimulus of the network was derived from average node-switching frequencies calculated from logic simulation data. Over 1,800 signal nodes were also analyzed by WAWOTH.

11 Conclusions

Our design strategies, methods, and CAD tools allowed us to complete the NVAX CPU chip design in 30 percent less time than our initial schedule had required. Typical parts operate at 83.3 MHz (a 12-ns cycle time) under worst-case conditions for temperature and power supply. This is 2 ns better than our maximum cycle time design constraint. The chip booted the VMS operating system ten days after the first prototype wafers were available, and booted the ULTRIX system a few days later. Multiprocessing was running within weeks. Fifteen obscure logic bugs were found in the first-pass chips, but none of them impeded system debug or prototype development.

No circuit design bugs were found. Only one design revision was needed prior to volume chip manufacturing.

Careful design, complexity management, and proprietary CAD tools targeted to large custom CMOS integrated circuits played crucial roles in the successful design of the NVAX microprocessor.

12 Acknowledgements

We would like to acknowledge the contributions of the following people: W. Anderson, E. Arnold, R. Badeau, I. Bahar, M. Benoit, B. Benschneider, D. Bernstein, D. Bhavsar, L. Biro, M. Blaskovich, P. Boucher, W. Bowhill, L. Briggs, J. Brown, S. Butler, R. Calcagni, S. Carroll, M. Case, R. Castelino, A. Cave, S. Chopra, M. Coiley, E. Cooper, R. Cvijetic, M. Delaney, D. Deverell, A. DiPace, D. DuVarney, R. Dutta, J. Edmondson, J. Ellis, H. Fair, G. Franceschi, N. Geagan, S. Goel, M. Gowan, J. Grodstein, P. Gronowski, W. Grundmann, H. Harkness, W. Herrick, R. Hicks, C. Holub, A. Jain, R. Khanna, R. Kiser, D. Koslow, K. Ladd, S. Levitin, S. Martin, T. McDermott, K. McFadden, B. McGee, J. Meyer, M. Minardi, D. Miner, T. O'Brien, J. Pan, H. Partovi, N. Phillips, J. Pickholtz, R. Razdan, S. Samudrala, J. Siegel, K. Siegel, C. Somanathan, J. St. Laurent, R. Stamm, R. Supnik, M. Tareila, S. Thierauf, M. Uhler, N. Wade, S. Watkins, and Y. Yen.

13 References

1. G. Uhler et al., "The NVAX and NVAX+ High-performance VAX Microprocessors," Digital Technical Journal, vol. 4, no. 3 (Summer 1992, this issue): 11-23.

2. B. Zetterlund, J. Farrell, and T. Fox, "Microprocessor Performance and Process Complexity in CMOS Technologies," Digital Technical Journal, vol. 4, no. 2 (Spring 1992): 12-24.

3. J. Crowell, K. Chui, T. Kopec, S. Nadkarni, and D. Sovie, "Design of the VAX 4000 Model 400, 500, and 600 Systems," Digital Technical Journal, vol. 4, no. 3 (Summer 1992, this issue): 60-72.

4. L. Chisvin, G. Bouchard, and T. Wenners, "The VAX 6000 Model 600 Processor," Digital Technical Journal, vol. 4, no. 3 (Summer 1992, this issue): 47-59.

5. R. Badeau et al., "A 100 MHz Macropipelined VAX CMOS Microprocessor," Accepted for publication in IEEE Journal of Solid State Circuits, vol. 27, no. 11 (November 1992).

6. SPICE is a general-purpose circuit simulator program developed by Lawrence Nagel and Ellis Cohen of the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley.

7. T. Fox, P. Gronowski, A. Jain, B. Leary, and D. Miner, "The CVAX 78034 Chip, a 32-bit Second-generation VAX Microprocessor," Digital Technical Journal, vol. 1, no. 7 (August 1988): 95-108.

8. W. Durdan, W. Bowhill, J. Brown, W. Herrick, R. Marcello, S. Samudrala, G. Uhler, and N. Wade, "An Overview of the VAX 6000 Model 400 Chip Set," Digital Technical Journal, vol. 2, no. 2 (Spring 1990): 36-51.

9. J. Hwang, "REX-A VLSI Parasitic Extraction Tool for Electromigration and Signal Analysis," 28th Design Automation Conference (1991): 717-722.

10. J. Pan, L. Biro, J. Grodstein, B. Grundmann, and Y. Yen, "Timing Verification on a 1.2M-Device Full-Custom CMOS Design," 28th Design Automation Conference (1991): 551-554.

11. W. Elmer, "The Transient Response of Damped Linear Networks with Particular Regard to Wide-band Amplifiers," Journal of Applied Physics, vol. 19 (1948): 55-63.

12. V. Raghavan and R. Rohrer, "A New Nonlinear Driver Model for Interconnect Analysis," 28th Design Automation Conference (1991): 561-566.

14 Biographies

Dale Donchin Dale Donchin manages schematic entry and layout verification CAD tool development in the Semiconductor Engineering Group. He facilitated the use of CAD tools for NVAX design, primarily for layout and physical chip verification. Dale is presently performing in a similar capacity for a new microprocessor design based on the Alpha AXP architecture. He joined Digital in 1978, and was previously a development manager in the RSX operating system group. Dale holds a B.S.E.E. (1976, honors) and an M.S.E.E. (1978) from Rutgers University College of Engineering and is a member of IEEE and ACM.

Timothy C. Fischer Tim Fischer is a senior hardware engineer with the Semiconductor Engineering Group. He is working on the design of a floating-point unit for a future high-performance microprocessor based on the Alpha AXP architecture. Prior to this work, Tim was a member of the E-box design team and contributed to the design of global clock generation and distribution for the NVAX microprocessor. He also worked on the design of the bus interface unit on the NVAX+ chip. Tim joined Digital in 1989 after receiving his M.S. in computer engineering from the University of Cincinnati.

Thomas F. Fox Frank Fox, a consulting engineer in the Semiconductor Engineering Group, co-led the implementation of the NVAX microprocessor and consulted with the Advanced Semiconductor Development Group on the design of the CMOS-4 technology. He joined Digital in 1984 and worked on the implementation of the CVAX microprocessor. Frank received a B.E. degree from University College Cork, National University of Ireland

(1974), and a Ph.D. degree from Trinity College, Dublin University (1978), both in electrical engineering. He has published papers on ultrasonic instrumentation, MRI scanners, and VLSI design.

Victor Peng Victor Peng is a consultant engineer in the Semiconductor Engineering Group. He received his B.S. in electrical engineering from Rensselaer Polytechnic Institute in 1981, and his M.S. in electrical engineering from Cornell University in 1982. Victor joined Digital in 1982. He was a member of the design team for the VAX 8200/8300 memory interface chip and patchable control store chip. He led the implementation of the VAX 6000 Model 400 floating-point chip and was co-manager of the NVAX chip design team.

Ronald P. Preston Ronald Preston is a principal engineer in the Semiconductor Engineering Group. Since joining Digital in 1988, he has worked on the design of several microprocessors. Ron was the circuit design and implementation leader for the E-box on the NVAX microprocessor. He is currently designing the instruction issue logic for a superscalar RISC microprocessor. Prior to joining Digital, he worked on the design of CMOS microcontrollers for Signetics Corporation. Ron received his B.S.E.E. in 1984 and his M.S.E.E. in 1988 from Rensselaer Polytechnic Institute. He is a member of Eta Kappa Nu and IEEE.

William R. Wheeler A principal engineer in the Semiconductor Engineering Group, Bill Wheeler performed architectural definition work for the NVAX microprocessor and was project leader of the NVAX M-box. Prior to this work, he designed the I-box unit for the third-generation VLSI VAX implementation. He is currently involved in advanced development work for improving custom design tools and methods. Before joining Digital in 1983, Bill received both his B.S.E.E. (1982) and his M.S.E.E. (1983) from Cornell University. He has coauthored three papers on VLSI designs and holds three patents.

15 Trademarks