The Intel Pentium Pro processor is a high-performance Intel Architecture microprocessor designed for desktop, workstation, and server applications. Its microarchitecture forms the basis of the company’s future high-volume microprocessor portfolio. The initial design has also been augmented with enhancements such as MMX technology and compacted onto newer fabrication processes to create the Pentium II processor product line.

The original Pentium Pro processor design, known initially as the P6, introduced several performance features, including:

- register and flag renaming
- speculative and out-of-order dispatch and execution of instructions
- reordered memory access
- multiple branch prediction
- a full-speed second-level (L2) cache accessed through a dedicated 72-bit backside bus
- a 64-bit, transaction-oriented, pipelined front-side bus, which operates at GTL voltage levels and directly supports four-way multiprocessing systems

The design was first implemented in Intel’s proprietary 0.6-μm, four-metal BiCMOS process. The CPU die measured 683 mils (17.3 mm) on a side and contained 5.5 million transistors. Designers created two versions of the shipped product by combining the CPU with either a 15.5-million transistor, 256-kbyte L2 cache or a 31-million transistor, 512-kbyte L2 cache. In either case, the CPU and L2 cache dies were mounted in a 387-pin, dual-cavity, ceramic PGA package, with wire-bonded connections to the pad rings.

DFT design requirements

The challenges associated with production test and debug on the processor were considerable, due to its tight release schedule, the enormous complexity of its circuit design and microarchitecture, and the complexity of the system platform. In addition to these issues, Intel’s unique business requirement of simultaneously meeting very high production, performance, and test quality targets strongly influenced its design-for-test direction. This set of constraints limits Intel microprocessor design teams’ ability to use DFT and test generation techniques (full or partial scan and scan-based BIST), which are commonly used in other microprocessors in the industry. Within these constraints, the design team optimized the design for low die area, high performance, low power dissipation, high test quality, and low test cost.

Die area. Raw fab output is a function of wafer capacity and product die size. In addition to the die area impact on yield, more fabs must be built to ship a larger design, assuming a fixed quantity of product. An independent study has estimated that a die area increase of 15% in one particular Pentium processor design would have cost Intel the construction of a new multibillion-dollar fab. For this reason, die area plays an overwhelming role in the economics of high-
volume microprocessor production. As a result, the DFT features implemented in any new Intel processor must be as area-efficient as possible.

**CPU performance.** While fab economics constrain high-volume product die sizes, microprocessor architects must spend more and more transistors to excite customers for each new-generation CPU. Most customers are willing to invest in a new CPU only if its performance or feature set is substantially better than that of existing products already being fabricated in the latest process technology.

For a given process technology, microarchitecture is the largest determinant of performance, and the number of transistors available for performance-enhancing features is proportional to die area. Furthermore, all proliferation products are direct modifications of the lead processor design. If the lead design of a new product family fails to include all possible performance-enhancing features, the entire product line is at risk of having lower performance than would have been otherwise possible.

The need to produce this customer-perceived value for the product line creates a strong pull to make the new-generation die as large as possible. Unfortunately, this pull directly opposes the business need to keep die area small for fab capacity and yield reasons in a high-volume market segment. This results in a constant battle for die area between performance and other features such as DFT.

**Power delivery and cooling.** Die area also plays a role in a relatively new design constraint for desktop CPUs: power dissipation. While the square of the supply voltage is the most significant factor in the equation, power does scale linearly with die area. Platform costs increase rapidly with CPU power dissipation, and since desktop platform pricing is relatively inflexible, wattage has become a limiting factor in desktop CPU design. This in turn creates additional pressure to reduce die area as much as possible, and a desire to avoid design techniques (DFT or otherwise) that interfere with power-saving measures or sink unnecessary current.

**Test quality and cost.** Customers require product quality levels well below a few hundred defective parts per million (DPM) shipped, and this cannot be economically achieved for a high-volume part without DFT and a good test strategy. While zero DFT would result in unacceptable test quality, any die area spent on DFT is area unavailable for CPU performance and customer-visible features. Some DFT approaches also affect power and circuit performance. For these reasons, we decided that a number of common industry DFT techniques such as scan and scan-based BIST were impractical for the Pentium Pro processor. In general, any DFT technique that enables high test quality with low die area overhead is more desirable. Efficient DFT techniques provide good test quality over the whole design, with little degradation in die area, performance, or power.

In a qualitative sense, the product of performance-based customer value and test-based customer value has an optimal point at which the needs of test quality and product performance are balanced, as shown in Figure 1. For Intel's lead CPUs, the optimum point is skewed well to the left.

There is a natural and powerful tension between the competing interests of die area, performance, power, and production test quality. Design teams that fail to understand the cost of poor test quality will naturally create designs that are difficult to test. Designers who spend a disproportionate amount of die area on DFT are at risk of delivering parts that have marginal appeal in the marketplace. Design teams that comprehend both issues, and are rewarded for optimizing over this full space, will be able to achieve a healthy balance and a better product.

**DFT design principles**

We distilled the design constraints just discussed into a short list of guiding principles that drove the definition and design of DFT features for the Pentium Pro processor. The most important of these principles were that DFT must

- have zero performance impact
- have minimal die area impact
- be multiuse features wherever possible (supporting component debug, production test, and so on)
- be designed in from the start (that is, coded and validated in the register-transfer-level model)

Due to the aggressive frequency and volume targets for this processor, we viewed the first two rules as hard constraints. The principle of designing multiuse DFT features...
further supported the goal of minimizing area impact. Where area trade-offs had to be made, a multiple-use DFT feature almost always won out over a special-purpose feature.

The fourth principle was a key to the success of the program. Just like any other feature in the design, we coded the DFT features into the RTL model, verified their schematics using schematic formal verification tools, and verified their functionality through RTL simulation. One validator was assigned full time to the DFT features for an entire year preceding tape-out. We wrote a set of powerful new tools to support accessing these features through the JTAG test access port interface in the RTL simulator and to check the test results. As a result, all of the major DFT features were shown to be functional before tape-out.

DFT features implemented

Table 1 summarizes the major Pentium Pro processor DFT features, their costs (in die area and person-months of design time), and the capabilities for which each feature was most useful. Area numbers are shown for the CPU and 256-Kbyte L2 dies.

In keeping with the third DFT design principle, we designed each group of features listed in the table to target as many of the following capabilities as possible:

- component debug on an ATE tester
- system platform debug
- high-quality production test

Our processor includes the DFT features described here.

<table>
<thead>
<tr>
<th>DFT feature</th>
<th>Design cost</th>
<th>Debug benefit</th>
<th>Production test benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>JTAG private instructions</td>
<td>300 sq. mils, 4</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Scan-out</td>
<td>5,200 sq. mils, 4</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DFT control registers</td>
<td>7,700 sq. mils, 13</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Array freeze/dump</td>
<td>1,900 sq. mils, 2</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Internal breakpoint mechanism</td>
<td>400 sq. mils, 2</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Debug mode</td>
<td>250 sq. mils, 12</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Performance counters, last branch records, and branch trace messages</td>
<td>100 sq. mils, 8</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Physical DFT</td>
<td>0 sq. mils, 3</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>LFSR BIST</td>
<td>1,200 sq. mils, 2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>M icrocoded BIST</td>
<td>800 sq. mils, 4</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Other CPU test and debug logic</td>
<td>4,500 sq. mils, 9</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>L2 cache PO ST, PBIST, SMURFL</td>
<td>18,000 sq. mils, 36</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**JTAG test access port.** The Pentium Pro processor test access port (TAP) is a compliant implementation of the IEEE 1149.1 (JTAG) standard. We implemented seven public TAP instructions. In addition, many more private instructions provide access to the majority of the test and debug features described in this article.

**Scan-out.** Scan-out is a patented observation-only, scan-like technique that provides observability of internal nodes in the design. Intel designs have used scan-out as a debug and test feature since the i960 and original Pentium processor designs. The Pentium Pro processor CPU die includes over 2,000 scan-out nodes. We chose this number to balance the amount of key architectural and microarchitectural state needed for debug versus the amount of silicon area we were willing to spend on the feature.

Scan-out can work in two modes. Snapshot mode takes a single sample of the key processor state and shifts it out through the JTAG port while the processor is running. Signature mode enables on-the-fly continuous sampling and signature compression of the same key state elements in the design. Both modes can be used while the processor is running normally at full clock frequency, without affecting its operation. We used snapshot mode in the component debug environment to dump a complete cycle-by-cycle map of the sampled internal state. We used signature mode to increase observability during production test.

**Control register bus access.** An internal 32-bit bus in the Pentium Pro processor provides access to special "control" registers. We used this bus to provide direct access to unit-specific testability and performance-monitoring features. A
private mechanism enables access to these features during debug and production test.

**Array freeze and dump.** The Pentium Pro processor itself contains 39 internal arrays. These range in size from the 8-Kbyte, single-ported instruction and data caches down to highly multiported register files with only a few hundred storage elements. Virtually all arrays contain state information, the visibility of which was critical to debug success. Therefore, we devised a cost-efficient debug mechanism to provide observability access to the array contents.

Every key array for debug responds to a freeze command broadcast in the processor. Upon receiving this command, an array immediately disables its write logic, so that its contents freeze even as the processor clock remains running. Special enabling mechanisms at the arrays dump the frozen state off the processor.

**Debug mode.** This mode allows a running processor to be stopped in its tracks and placed in an interactive testing mode. After testing operations have been completed, the processor can be released into normal execution mode again.

**Internal breakpoint mechanism.** All of the previous debug features needed a precise mechanism to trigger their state capture and dump in a system debug environment. We created an internal breakpoint mechanism to provide this capability. This mechanism can be set up to respond to any of a variety of internal microarchitectural events, with any of a set of debug actions to be taken after a programmed internal delay.

**Microcode update.** This feature allows temporary modification of sections of microcode for test and debug purposes. It provided a very powerful feature for diagnosis and the workaround of bugs found in early silicon debug.

**Performance counters, last branch records, and branch trace messages.** These debug hooks, implemented in dedicated hardware and accessed through special transactions on the bus, provided additional mechanisms through which system debuggers could monitor performance and software flow in a running system.

**Physical debug features.** In addition to the logical and architectural debug features just described, the Pentium Pro processor included physical DFT rules and structures to support probing and efficient workarounds in the course of silicon debugging. Specific physical DFT structures include e-beam probe sites, contact probe sites, focused ion beam (FIB) editable structures, spare gates and wires to be used for FIB edits, and layout support for quick metal-only mask edits.

**Clock bypass mode.** To support clock shrink-and-stretch capability for silicon debug, we implemented a clock bypass mode in which the PLL could be shut off and the core clocks driven directly to the die from the debug tester.

**L_{test} mode.** This testing mode allows a private TAP instruction to be used to disable all devices that draw static current. Test engineers could use this mode in conjunction with clock bypass to create L_{test} tests for the processor.

**Built-in self-test.** The Pentium Pro processor contains a full BIST routine that is primarily targeted at achieving high toggle coverage for burn-in testing. We implemented most BIST functionality in microcode, but hardware linear feedback shift registers (LFSRs) covered several inaccessible programmable logic arrays (PLAs).

**Backside bus connectivity check.** Since the Pentium Pro processor consists of two dies (CPU and separate L2 cache chip) mounted in a dual-cavity package, this package presented a specific testing problem. We had to ensure that the bond wires between the two dies were correctly formed. A feature in both dies directly tests this connectivity, by driving a pattern of alternating 0s and 1s across the bus. This captures the values in a test register in the L2 cache and shifts the result back through the processor’s JTAG port.

**L2 power-on self-test (POST).** We implemented a 13N, partial moving inversion POST for all arrays (data, tag, and LRU) in the L2 die. This test runs at every power-on of the die and serves as an effective early filter in the production test flow.

**L2 programmable BIST.** In addition to POST, we implemented a sophisticated stand-alone test feature, dubbed PBIST, in the L2 die. This feature provides an on-chip, fully programmable array test pattern generator, including a small vector memory and a programming language of looping constructs and address counter control. The PBIST feature provides the main mechanism through which the L2 cache die is tested in production.

**L2 speed measurement using a RAM feedback loop (SMURFL).** To address speed test of the L2 die, we implemented SMURFL. This feature implements a ring oscillator type of structure using the whole array access path. By measuring the speed of this path as it executes 128 consecutive array reads on a low-resolution tester, the production test program can accurately measure array access time. This feature plays a major role in enabling speed-binning of L2 cache dies in the production test flow.

**Production test development methodology and process**

To meet the high test quality goals for the project, we decided to handwrite the majority of production tests for the processor. We set the following test-writing goals (in priority order):

- maximize test quality
- maintain vector memory depth within tester limitations
- minimize test runtime
Ultimately, test quality is measured not by coverage achieved against some fault model, but by the defective parts per million (DPM) actually shipped to the customer. We took as a key activity, therefore, to educate the design team on this distinction, and to structure the whole test-writing and fault-grading methodology to support the ultimate goal of minimizing DPM.

We focused the test-writing effort not only at high traditional stuck-fault coverage but also at array test pattern coverage, transition fault coverage, bridging fault sensitization within dense data path segments, and targeting known speed path and circuit sensitivities. We ensured test quality by establishing a rigorous multistep process of fault grading, documentation, and review of all tests and fault coverage results, and closely tracking partial and full-chip coverage numbers across the chip.

To optimize both results and productivity of the test-writing effort, we set test quality goals differently for each area of the chip, based on the following categories (in priority order):

- embedded caches/arrays
- the largest, densest data path structures
- all other data path structures
- hand-drawn control logic
- synthesized control logic

Caches, embedded arrays, and large data path structures on the Pentium Pro processor accounted for 69% of the CPU transistor count. These design areas were also the densest, and therefore the most sensitive to manufacturing defects. Our strategy was to ensure that the highest quality testing techniques were applied to these structures, and to require specific at-speed memory algorithmic tests for every cache and embedded array on the chip.

To ensure high-quality testing of the most sensitive non-array parts of the design, we prioritized and listed the design’s largest, densest data path structures and held these data paths to higher test quality standards. These included establishing an (effectively) 100% stuck-fault coverage goal and specific goals for transition fault coverage and bridging fault sensitization within the data path, and targeted testing of all multiplexers in the data path.

For data path structures below this cut-off point and for hand-drawn control blocks, we relaxed the test requirements to a minimum of achieving 90% stuck-fault coverage. For these blocks, we educated the designers on the high-quality test-writing practices used for the large data path structures. Where it was easy to do, designers took the initiative to apply the better test-writing techniques.

Finally, we accepted a lower stuck-fault coverage goal for all synthesized control logic in the design. This was a calculated risk/effort trade-off. Several factors (most notably the absence of schematics and the amount of effort therefore required to prove individual faults untestable) conspired to make test-writing for this kind of logic very unproductive beyond a certain point.

**Results**

The Pentium Pro processor saw first silicon in December 1994. After only eight months of debug, test development, and production readiness work, the design team released the design to the manufacturing team in August 1995 to begin work on volume production. The initial design and its first shrink onto Intel’s newer 0.35-µm fabrication process were formally introduced and became publicly available in November 1995. Since that time, Intel has shipped several million units of the Pentium Pro processor with virtually no customer returns for manufacturing defects.

**Design results.** Our goals for the DFT design approach were to enable fast debug and high-quality, cost-effective production test with no impact on performance, virtually no impact on power, and minimal die area impact. The total die area used for DFT features was 4% of the CPU die and 6% of the L2 die. We validated almost all features before initial tape-out, and consequently all were usable on first silicon. DFT features had no negative impact on processor performance, either in clock frequency or instructions per clock.

**Silicon debug results.** All new microprocessor designs have design flaws that are discovered only during silicon debug, and our processor was no exception. However, as a result of the debug features built into the design, we could diagnose all significant errata and could work around nearly all of them on first silicon in both tester and system debug environments. The ability to work around errata greatly accelerated the time to market and reduced the number and frequency of new tape-outs needed.

Figure 2 shows the time to market achieved from initial tape-out for Intel’s most recent lead CPUs. The DFT features include
enabled major milestones to be met quickly, such as boot-
ing DOS and Windows, which was achieved within about a
week of receiving first silicon. Production-worthy silicon re-
quired only three metal-only steppings and one full-layer
stepping. Without its DFT features, product introduction
would have been delayed by many months.

Production test results. Designers completed structur-
al test writing and most of the fault grading for the Pentium
Pro processor test suite by November 1995 and final fault
grading by May 1996. Fault simulation work was stopped
when the full-chip fault grade, measured against the single
stuck-fault model, reached 96%. Coverage of transition and
bridging faults was ensured by test code reviews, and not
simulated. In a similar fashion, designers wrote speed-path
tests and reviewed them for the worst paths found in silicon.

The ultimate indicator of initial production test quality is
customer returns in the first million parts shipped. Against
this metric, the Pentium Pro processor was a factor of 5 to 20
better than previous Intel products. Internal indicators in
the production test process correspondingly show current DPM
results well within desired targets.

Our team’s commitment and planning for DFT features,
clean execution in the design, and follow-through with fo-
cused attention to test-writing proved very successful. The
Pentium Pro processor team got a state-of-the-art, high-
performance microprocessor through debug and into pro-
duction with a high-quality test set within approximately 11
months from first silicon. We achieved this while minimiz-
ing DFT area cost and with no performance or power im-
pact to the design.

Acknowledgments
We gratefully acknowledge the assistance of Mark Waggoner
and Mark Balmer in providing information on the L2 DFT features.
We also thank Bob Colwell, Wayne Needham, Randy Steck, and
Keshavan Tiruvallur for their review of this article.

References
1. R. Colwell and R. Steck, “A 0.6um BiCMOS Processor with Dy-
namic Execution,” Digest of Tech. Papers, Int’l Solid-State Cir-
pentiumii/home.htm.
4. D. Bhavsar and J. Edmondson, “Alpha 21164 Testability Strat-
5. A. Crouch, M. Pressly, and J. Circello, “Testability Features of
and Ad-Hoc Design for Test: Testing the PowerPC 603™ Mi-
pp. 76-83.
Press, 1993, pp. 764-772.
8. K. Thompson, “Intel and the Myths of Test,” IEEE Design & Test
9. IEEE Std. 1149.1-1990, Standard Test Access Port and Bound-
ary-Scan Architecture, (includes IEEE Std 1149.1a-1993), IEEE,
1993.
10. A. Carbine, “Scan Mechanism for Monitoring the State of In-
5,253,255.

Adrian Carbine, a VLSI design manager at Intel since the early 90
microprocessors, was a senior design manager for the Pentium Pro
processor. His responsibilities included the logic, circuit, and layout design of one fifth of
the CPU in addition to DFT, test development,
and silicon debug. Currently, he manages the
design automation team working on a next-generation IA-32 mi-
croprocessor and is involved in DFT at the corporate level. Carbine
holds a BS degree in electrical engineering from Brigham Young
University in Utah. He is a member of the IEEE.

Derek Feltham is the DFT methodology man-
ger for Intel’s next-generation IA-32 micro-
processor design. He worked on DFT feature
design for the Pentium Pro processor. Feltham
received his BASc and MASc degrees from the
University of Toronto, and his PhD in electrical
and computer engineering from Carnegie
Mellon University. He has been a member of the IEEE since 1986.

Direct questions concerning this article to Adrian Carbine, In-
tel Corporation, 5200 NE Elam Young Parkway, Hillsboro, OR
97124; adrian@ichips.intel.com.