DSP16xxx Targets Communications Apps

New Lucent Design Extends Conventional Techniques to Improve Performance

by Jeff Bier, Berkeley Design Technology, Inc.

Lucent Technologies has launched the first challenge to Texas Instruments' recently introduced ultra-high-performance TMS320C62xx DSP. Lucent's new family, the fixed-point DSP16xxx, is the first new DSP family launched by Lucent since its spinoff from AT&T. While TI's radical 'C62xx (see MPR 2/17/97, p. 14) breaks the mold of the past decade's DSPs by adopting a VLIW- and RISC-like architecture that targets maximum performance, Lucent's DSP16xxx remains true to its DSP heritage, achieving strong performance and competitive cost by sacrificing generality and ease of programming. Indeed, the DSP16000 core bears a strong resemblance to Lucent's DSP1600 core (introduced in 1990), but it adds significant capabilities in the data path and instruction set and boosts on-chip memory bandwidth.

With the 16xxx, Lucent continues its successful strategy of specialization. While TI dominates the $3 billion market for programmable DSPs with a 45% share (according to Forward Concepts) by offering an extremely broad product line, Lucent has managed to carve out the number-two position with a healthy 30% share by focusing almost exclusively on telecommunications applications.

The 16210 will be the first member of the new DSP family. According to Lucent, the 16210 will begin sampling in December. Alpha-version development tools, including a cycle-accurate instruction-set simulator, are available now. Lucent plans to begin volume shipments of the 16210 in mid-1998.

First in a Family

The company hopes the 16xxx family will appeal to designers of communications infrastructure equipment, such as cellular-telephone base stations and modem banks for Internet service providers. In such applications, increased performance allows more channels per processor. Lucent expects the 16210 to operate at 100 MHz (and 100 native MIPS) at 3.0 volts using a 0.35-micron three-layer-metal process. Thus, Lucent is initially positioning the 16xxx against high-end DSPs intended for line-powered applications, such as TI's 'C62xx and Motorola's DSP563xx.

In addition, Lucent hopes to extend the 16xxx family to appeal to designers of low-power but performance-hungry embedded products, such as advanced cellular phones. For these applications, Lucent plans to provide versions at lower voltages that will still maintain strong performance. These versions, which Lucent plans to begin sampling in mid-1998, will put the 16xxx in competition with processors such as Motorola's DSP566xx and TI's TMS320C54x.

According to Lucent, its older 16xx family will continue to be expanded and will maintain its focus on low-power, single-processor applications that don't demand the higher performance of the 16xxx.

Extending the Conventional DSP

The 16xxx is designed to operate primarily on 16-bit data but uses 32-bit buses internally, as Figure 1 shows, and a mixture of 16- and 32-bit instructions. Like most of today's DSPs, the 16xxx relies on complex instructions that can encode many parallel operations, and it issues these instructions at a maximum rate of one per cycle. Registers are dedicated to specific uses, addressing modes are oriented toward vector access, and hardware is included for accelerating program loops. Dedicated hardware is provided for communications-oriented processing, such as Viterbi decoding (used for decoding error-control-coded signals, for example).

But unlike most of today's DSPs, the 16xxx data path includes *two* $16 \times 16 \rightarrow 32$-bit multipliers, as Figure 2 shows. Dual multipliers are also used in the Lode DSP core (developed by TCSI and recently acquired by Atmel). Unlike Lode, however, the multipliers in the 16xxx are independent—each can draw its input operands from the upper or lower 16-bit halves of the 32-bit X and Y multiplier input registers. This independence allows the multipliers to be used more flexibly and thus allows more applications to utilize both multipliers.

Also unusual among fixed-point DSPs, the 16xxx sports an extra three-input, 40-bit adder (separate from the ALU), a

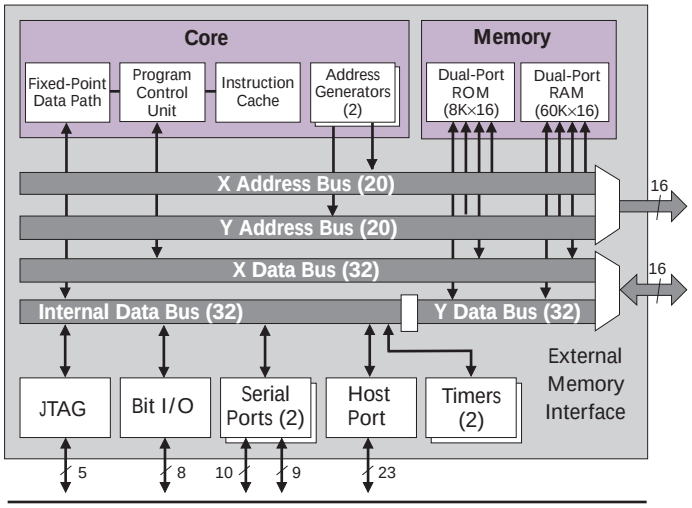

Figure 1. Top-level view of the DSP16210. At this level, the DSP16210 is very similar to the earlier DSP1620—the main difference being the DSP16210's 32-bit data buses, expanded from 16 bits on the older part. The internal data bus becomes the Y data bus before connection to memory.

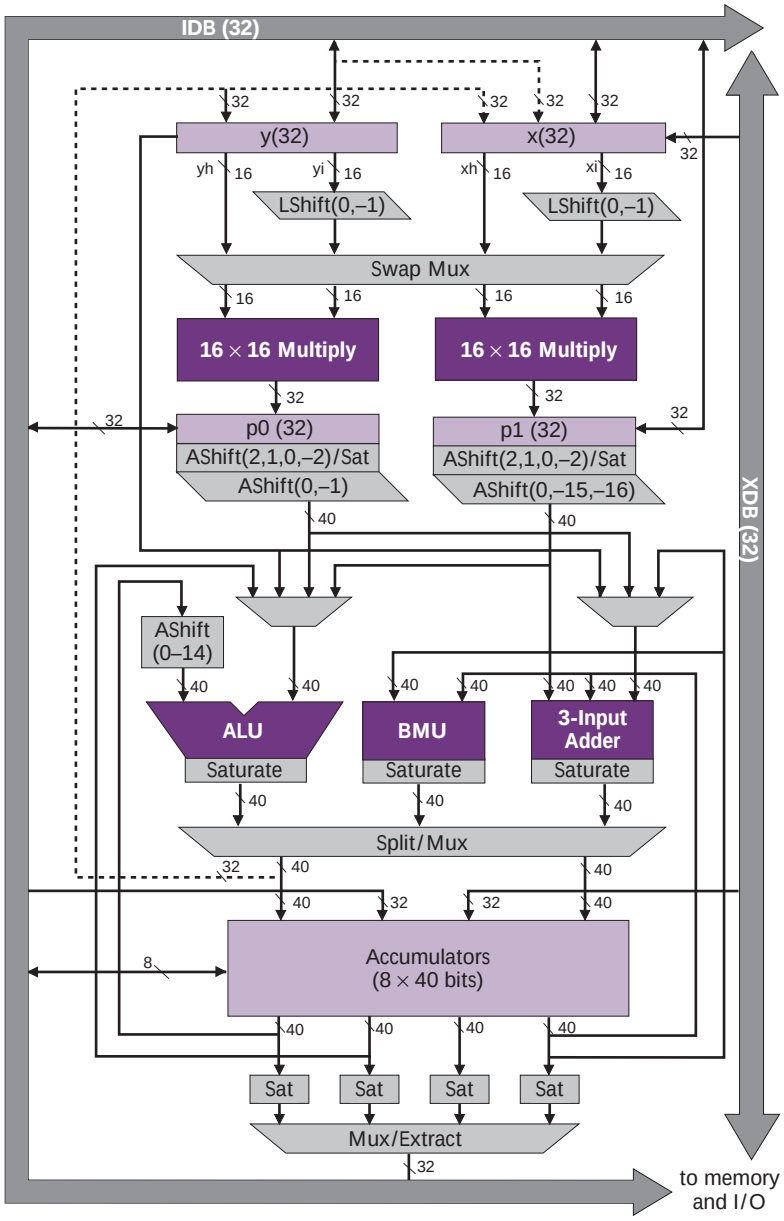

Figure 2. The DSP16xxx data path plainly reveals the processor's DSP orientation with its dual 16-bit multipliers, dedicated registers, large number of specialized shifters, and numerous saturation units. LShift denotes a logical shift, while AShift denotes an arithmetic shift. A negative shift amount denotes a right shift; a positive amount denotes a left shift. Dashed lines denote mode-controlled paths.

powerful bit-manipulation unit (BMU), and eight 40-bit accumulators. All execution units have single-cycle throughput. These resources will be useful for a wide range of DSP applications, but especially for the communications equipment targeted by Lucent.

Data Path Is Key

The 16xxx's data path and its specialized, complex instructions allow numerous operations to be encoded into a single instruction and executed in parallel. As Table 1 shows, several combinations of ALU, BMU, and adder operations can be executed in parallel, in some cases combined with two multiplications. Up to two 32-bit data transfers can be combined with these operations. Multiply-accumulate operations—which form the heart of many key performance-hungry DSP algorithms, such as filtering, transforms, and correlation—can thus be executed at a rate of two per cycle, including the associated data moves. This is twice the rate of conventional DSPs.

While more flexible than those found in Atmel's Lode DSP, the 16xxx's dual multipliers are less adaptable than those in TI's 'C62xx. The 16xxx multipliers are limited to selecting their operands from the upper or lower halves of two dedicated 32-bit multiplier input registers, X and Y. The X and Y registers can both be loaded from memory in parallel with the execution of one or two multiply operations, but this requires the needed 16-bit operands to be arranged as pairs in memory. This restriction is an example of Lucent's decision to sacrifice generality in exchange for performance and simple hardware—a common strategy for DSPs, but one that stands in sharp contrast to TI's approach on the 'C62xx.

The ALU supports 16-bit, 32-bit, and 40-bit operands—though not all operand sizes are supported by all operations. SIMD-style operations allow the ALU to perform two 16-bit additions or two 16-bit subtractions in parallel. To accelerate Viterbi decoding, the ALU can perform specialized minimum and maximum operations.

In addition to the ALU, the 16xxx data path also provides a standalone three-input adder. The adder allows a full-width (40-bit) addition or subtraction to proceed in parallel with another that uses the ALU.

As Figure 2 illustrates, a variety of limited-capability shifters are included in the data path to facilitate the scaling operations that are often necessary in fixed-point DSP algorithms, and to ease the implementation of multiprecision arithmetic. Saturation hardware is also provided at several points in the data path. Lucent designed these features to simplify the implementation of the bit-exact algorithms in some telecommunications standards, such as the speech-compression algorithms used in GSM cellular telephones.

Large Register Set Reduces Memory Accesses

The 16xxx uses a simple, three-stage pipeline with few hazards. In the absence of wait states, all instructions execute in one cycle except for branches, which take three cycles. Wait states can be incurred as a result of an off-chip memory access (due to slow external memory or the need to make two 16-bit accesses to read or write a 32-bit value) or an

attempt to perform two simultaneous accesses to one on-chip memory bank (banks are 1K×16 each).

The main registers in the data path of the 16xxx are the eight 40-bit accumulators A0–A7. Most data-path operations use accumulator registers for at least one operand. Eight accumulators are a significant improvement over the 16xx's two and several more than typically found in fixed-point DSPs. The extra accumulators reduce the need to swap values back and forth to memory, conserving valuable cycles in critical inner loops.

An unusual feature of the 16xxx data path allows results of an arithmetic operation to be simultaneously deposited into an accumulator and into one of the multiplier input registers. This feature is enabled via a mode bit. When enabled, results directed to accumulators A6 or A7 by any arithmetic instruction are also routed to a multiplier input register implicitly, as the dotted line in Figure 2 shows. This feedback feature allows programmers to avoid an explicit move instruction in cases where the result of an ALU or adder operation is to be used as an input to a subsequent multiply. This feature can shave a cycle or two in certain critical inner loops—although it wouldn't be necessary at all in an architecture based on general-purpose registers, such as TI's 'C62xx.

High Bandwidth to On-Chip Memory

Two on-chip bus sets, X and Y, each include a 20-bit address bus and a 32-bit data bus. Each can access almost any area of on-chip memory. The X bus is used for instructions and data reads; the Y bus is used for data reads and writes. The 32-bit X and Y buses can move pairs of 16-bit operands, assuming the operands are arranged as pairs in memory. Although 32-bit memory accesses are not required to be aligned on 32-bit boundaries (providing important flexibility for applications like filter-delay lines), unaligned accesses may incur a one-cycle penalty.

In general, the 16xxx fetches a 16- or 32-bit instruction via the X bus and reads or writes a 16- or 32-bit data word via the Y bus in each instruction cycle. By using its specialized on-chip instruction cache, however, the 16xxx can avoid the need for repetitive instruction fetches in small loops, enabling the X bus access to be used for a second data load. Simultaneous memory accesses must be directed to separate on-chip memory banks or suffer a one-cycle penalty.

For off-chip memory accesses, the two on-chip 32-bit buses are multiplexed onto a shared, 16-bit off-chip interface capable of a maximum of one transfer per clock cycle, for a peak bandwidth of 100 million 16-bit words per second at 100 MHz. As is common among newer DSPs, the ratio of on-chip memory bandwidth to off-chip memory bandwidth on the 16xxx (4:1 for reads) is higher than in older DSPs. This makes it increasingly important that programmers make good use of on-chip memory to tap the processor's performance potential. As Figure 1 shows, the 16210 will include 60 Kwords (120 Kbytes) of dual-ported on-chip RAM. This is more RAM than found on most currently available DSPs.

Operation (execution unit)	Combinations supported in a single instruction				
Add/Subtract (ALU)	●				
Round (ALU)		●			
Absolute value (ALU)			●		
Minimum (ALU)					●
Compare (ALU)				●	
Add/Subtract (Adder)	●			●	
2 Multiplications (Mul)	●				
Shift (BMU)		●	●		
Exponent (BMU)					●
2 Data transfers	●	●	●	●	●

Table 1. Combinations of operations supported in a single instruction. Each column represents a valid combination of operations that can be executed in parallel as part of a single instruction.

Instruction Cache Increases Data Bandwidth

The specialized on-chip instruction cache of the 16xxx is very similar to that of the 16xx but not like the caches found in general-purpose processors. In contrast to general-purpose processors, where caches are used to speed access to slower off-chip memory, the 16xxx instruction cache is used to free the X bus to be used for a second data access, enabling added on-chip data bandwidth and reducing power consumption.

The 16xxx cache is controlled explicitly via software using the DO instruction, which invokes a hardware-assisted loop. Up to 31 instructions can follow the DO instruction; these instructions will be repeated a specified number of times with no overhead cycles required for the loop itself.

During the first iteration of the loop, the instructions are executed and simultaneously copied to the cache. In this first iteration, the X memory bus is occupied with fetching instructions, so instructions that perform X memory data reads incur a one-cycle penalty. During subsequent iterations of the loop, however, the instructions are fetched from the cache, freeing the X bus to perform a data read in parallel with a second data access on the Y bus. Code already loaded into the cache can later be re-executed using the REDO instruction.

This cache design allows Lucent to achieve three-bus memory bandwidth in small inner loops (which account for a sizable fraction of the computation load in typical DSP applications) without the expense of a third bus or a more complex cache. The main limitation of Lucent's approach is that loop nesting is forbidden in the cache, as are most control-flow instructions. These restrictions complicate programming.

In addition to the DO instruction, which provides zero-overhead hardware looping support, two counter registers automatically increment when tested, providing support for low-overhead nested loops. The 16xxx allows conditional execution of many instructions, improving performance in many decision-intensive applications.

Mixed Instruction Lengths, Modes

In an attempt to maintain good code density while improving the architecture's per-cycle efficiency for communications

Operation	Execution Unit				Operation	Execution Unit			
	ALU	MPY	ADD	BMU		ALU	MPY	ADD	BMU
Arithmetic					**Bit Manipulation**				
Add	●				Insert bit field				●
			●		Extract bit field				●
Subtract	●				Shift left				●
			●		Shift right				●
Increment	●				**Program Control**				
Decrement	●				Uncond sub call				
Round	●				Uncond sub return				
Negate	●				Cond sub call				
Compare	●				Cond sub return				
Absolute value	●				Uncond branch				
Multiply					Cond branch				
Multiply		●			Do/Redo (cache)				
Multiply-accum	●	●			**Data Move**				
		●	●		Reg-to-reg move				
Multiply-subtract	●	●			Copy accumulator				
		●	●		Load from mem				
Logic					Store to mem				
AND	●				**Miscellaneous**				
OR	●				Divide-step	●			
XOR	●				Exponent detect				●
NOT	●				Normalization				●

Table 2. The DSP16xxx instruction set is in most ways similar to those of existing fixed-point DSPs. The main differences are the DSP16xxx's more extensive parallel operations (summarized in Table 1), its bit-field insert and extract instructions, and its SIMD-style add and subtract operations (described in the text).

applications, Lucent uses a mixture of 16- and 32-bit instructions and relies on mode bits to configure many aspects of the processor's data path. Generally, 16xxx instructions that are compatible with the earlier 16xx are 16 bits wide, while more powerful or flexible instructions unique to the 16xxx are 32 bits wide. Like the 32-bit instructions, the 16-bit instructions can encode many parallel operations, but the combinations of operations and operands supported by 16-bit instructions are severely limited. The 16xxx instruction set is summarized in Table 2.

Mode bits control shifting, saturation, and data-routing options in the data path. For example, mode bits select one of four scaling shift options at the output of each multiplier. The multitude of mode bits on the 16xxx has the potential to significantly complicate programming, but it does allow Lucent to achieve a more efficient instruction set without bloating code size—important for many cost-sensitive DSP applications, where die sizes are increasingly dominated by RAM and ROM.

```
// Get first inputs and coefficients
                                        y=*r0++    x=*pt0++

// First two multiply-accumulates
                p0=xh*yh    p1=x1*y1    y=*r0++    x=*pt0++
a0=a5+p0+p1     p0=xh*yh    p1=x1*y1    y=*r0++    x=*pt0++

// Multiply-accumulates
do  ((nTaps/2)-2)  {
  a0=a0+p0+p1  p0=xh*yh    p1=x1*y1    y=*r0++    x=*pt0++
}

// Last sum;  reset coefficient and input pointers
a0=a0+p0+p1                             y=*r0++j   x=pt0++h
```

Figure 3. Assembly code for an FIR filter inner loop on Lucent's DSP16xxx. Each line is a single instruction. The DO instruction invokes a zero-overhead hardware loop containing one instruction. The instruction inside the DO loop performs seven operations: a three-input add, two multiplications, two data reads, and two pointer increments. (Source: BDTI)

Pin- But Not Binary-Compatible

To facilitate system upgrades, the 16210 will be pin-compatible with Lucent's 1620. Although much DSP software continues to be developed in assembly code to obtain the best speed and minimal memory usage, the 16xxx assembly language is not compatible with that of the 16xx. However, 16xx programmers should be comfortable with the 16xxx assembly language—many 16xxx instructions are very similar to or even identical with those of the 16xx.

Lucent plans to provide a tool to translate 16xx assembly source code to 16xxx assembly. Some code will require manual translation, however, and translated code will require reoptimization to achieve maximum performance.

Lucent is hoping the similarities between the 16xxx and its predecessor, along with the translator tool to ease the transition, will help the 16xxx leverage the success of the 16xx in the communications equipment market. This strategy has a precedent in Texas Instruments' transition from the 320C5x (introduced in 1989) to the 320C54x family (introduced in 1994). Despite their very similar names and similar architectures, the 'C54x assembly language is not compatible with that of the 'C5x. In TI's case, this similar-but-not-compatible approach has succeeded; the 'C54x is in widespread use, including applications that formerly relied on the 'C5x. In contrast, Motorola's DSP563xx family (introduced in 1995) is binary compatible with the predecessor DSP560xx, introduced in 1987.

Programming Is Complicated

Like the 16xx before it, the 16xxx is a complicated device to program. While the addition of 32-bit instructions has allowed Lucent to remove some painful restrictions of the 16xx, the increased parallelism and expanded instruction set of the 16xxx add complexity. On the whole, assembly-language programmers will find the 16xxx similar to the 16xx in terms of programming complexity. Specialized, complex instructions and restrictions on register access are the main factors detracting from orthogonality and thus ease of assembly programming. The multitude of mode bits will also keep programmers on their toes. Figure 3 shows the 16xxx assembly-language source code for an FIR filter inner loop, illustrating both the complexity and the power of the 16xxx assembly language.

Currently, only limited, prerelease versions of the 16xxx software tools are available. In the past, software tools have been a weakness of Lucent's DSPs, compounded by limited third-party support. Lucent's tool plans for the 16xxx are ambitious, however; if the vendor can deliver, it will help offset the inherent complexity of the processor for assembly-language programmers.

The 16210 will include a sophisticated on-chip debugging module that is a significant improvement over those in earlier Lucent DSPs; it should simplify debugging in scenarios where the processor must be operated at full speed.

To support system integration, the 16210 includes a bit-I/O port, two timers, two serial ports (one with DMA controller), and a parallel host port with DMA controller—a configuration very similar to other high-end fixed-point DSPs. The bit-I/O port contains eight lines that can be individually configured as inputs or outputs; it also provides pattern masking and matching hardware.

Performance Shows Improvement

Berkeley Design Technology, Inc. (BDTI), has implemented its widely used DSP benchmark suite on the 16210, using Lucent's cycle-accurate instruction-set simulator. The results suggest that, cycle for cycle, the 16xxx will be noticeably—but not dramatically—more efficient on typical DSP algorithms than today's mainstream, conventional fixed-point DSPs. On representative benchmarks, the 16xxx's cycle counts are roughly 25% below the average for processors in this group. The improvement is more pronounced when compared with Lucent's own 16xx; the older Lucent processor suffered from relatively poor per-cycle efficiency due to its highly constrained architecture. In comparison, the 16xxx accomplishes about twice as much work per cycle.

Lucent's target of 100-MHz operation for the initial 16xxx is competitive, but not industry leading. Nevertheless, due to its relatively high per-cycle efficiency in DSP algorithms, the 16xxx—were it available today—would be faster in typical DSP applications than all of its conventional-DSP competitors, according to BDTI's projections. The 16xxx's speed does not approach that of the TI 'C62xx, however, since the VLIW-like TI chip achieves even better per-cycle efficiency and a significantly higher 200-MHz clock rate.

As DSPs increasingly find use in portable devices, power consumption becomes a more critical issue. Lucent projects typical power consumption for the 16210 of 294 mW at 100 MHz and 3.0 volts (with no I/O pin activity). This is somewhat high for a 16-bit DSP, especially considering the 16210's lower-than-average supply voltage. The processor's strong per-cycle efficiency, however, means that its projected benchmark energy consumption (an estimate of the amount of energy required by the processor to complete a certain amount of work) is competitive with that of most currently available fixed-point DSPs intended for portable applications. Thus, if Lucent is able to execute its plan to provide versions of the 16xxx at lower voltages that maintain strong performance, the 16xxx should compete well in power-sensitive applications.

Despite the significant gain in per-cycle efficiency and expansion of the instruction set compared with the older 16xx, the 16xxx achieves good memory efficiency on BDTI's benchmarks. The 16xxx's use of program memory in both DSP algorithm kernel and decision-making code appears comparable to those of its conventional-DSP competitors.

At $100 in 1,000-piece quantities, the 16210 is expensive, even considering its strong performance. Thus, its cost/performance results on the BDTI benchmarks lag those of its competitors—in some cases, by 2× or more. Of course, Lucent's highly focused market strategy means that few 1,000-piece orders will be solicited, and prices for very large orders will be negotiated, making comparisons tricky.

Price & Availability

The DSP16210 will begin sampling to customers in December, according to Lucent. The part will be available in 144-TQFP and 132-BQFP packages. Production is scheduled for mid-1998, and pricing is projected to be $100 in quantities of 1,000. For more information, contact Lucent (Allentown, Penn.) at 800.372.2447 or *www.lucent.com/micro*.

Maintaining a Strong Position

While the speed of the 16xxx cannot compete with that of TI's latest chip, Lucent's new architecture promises to significantly boost performance for its DSP product line while maintaining power consumption and memory usage levels in line with those of current-generation conventional DSPs. Hence, the 16xxx will appeal to designers seeking a balance among speed, power consumption, and memory use. In contrast, the new TI part sacrifices power consumption and memory efficiency to obtain maximum performance, making it attractive for applications where power and cost are not critical.

Lucent has maintained a strong position in DSPs over the past decade by specializing—in particular, focusing on telecommunications applications and targeting a small number of very large customers. The 16210 represents a continuation of that strategy. If Lucent can deliver on its plans, the 16210 should appeal to many of the same communications equipment vendors that now use the 16xx. This will help Lucent maintain its significant market share in the programmable DSP market—a market that is growing at more than 30% per year. MPR

Jeff Bier is general manager of Berkeley Design Technology, Inc. (BDTI), and coauthor of Inside Lucent's 16xxx, *a technical analysis report including independent benchmark results. For more information, visit BDTI on the Web at* www.bdti.com. *Ole Wolf and Jennifer Eyre of BDTI also contributed to this article.*