

Philips Hopes to Displace DSPs with VLIW

TriMedia Processors Aimed at Future Multimedia Embedded Apps

by Brian Case

Philips Semiconductors, a division of the Dutch conglomerate Philips Electronics, has announced a new embedded microprocessor architecture called TriMedia. (Philips absorbed the old-line U.S. semiconductor maker Signetics in 1978, and the TriMedia group is located in a former Signetics building in Sunnyvale, Calif.) TriMedia is a VLIW architecture and is the successor to the LIFE microprocessor experiment conducted at Philips in the late 1980s (see MPR 8/8/90, p. 6).

The current announcement is of the technology and broad product plans only—the first TriMedia chips will not be available in volume until the middle of 1996. The announcement was prompted by leaks and rumors, plus Philips' desire to kindle early customer interest.

Philips hopes to establish TriMedia as the industry-standard architecture for multimedia applications. The company views multimedia as the combination of high-end graphics, hi-fidelity audio, and full-motion video (the three technologies that give rise to the name TriMedia). In addition, Philips believes communications will be an indispensable part of many multimedia applications. Video conferencing, TV set-top boxes, and entertainment and education devices are seen by the TriMedia group as prime applications for its chips.

Figure 1 shows a time line of the history and near-term plans for the TriMedia product line. The TriMedia group was formed in early 1994 after a market and technology study was done at Philips Research Palo Alto (PRPA). Tape out of the first microprocessor is planned for the middle of 1995, with volume production following about a year later. TriMedia has had an unusually long gestation period—10 years!—if the original LIFE investigation is considered the start of the project.

Figure 2 shows a more detailed product roadmap. The first-generation chips will sport a direct PCI interface for a low-cost, high-performance PC multimedia card. The target clock rate for these chips is 100 MHz, which will yield a peak performance of 2.5 billion opera-

tions/s. A derivative of this first chip will use a slower clock rate in a videophone application. The second generation will be a compatible performance upgrade. The third generation will allow more design freedom to take advantage of experience with the first chips and then-current IC technology.

Minimizing Cost Is Key

High-quality multimedia applications need enormous amounts of processing power. Currently, a single general-purpose microprocessor is unable to simultaneously render real-time 2D or 3D graphics, generate CD-quality sound, recognize speech, and decode full-screen, full-motion video. If these functions are implemented at all in today's systems, DSPs and/or specialized chips—such as dedicated MPEG decoders—usually handle each task separately. For some functions, like MPEG encoding, specialized chips are still exotic. Also, in a dedicated (i.e., not a PC-based) system, a separate general-purpose processor is usually required to control the specialized chips and DSPs and perform general system functions.

One of the goals of TriMedia is to be able to implement the functions of a collection of DSPs, specialized chips, and a general-purpose processor with a single chip. Using a single microprocessor for all functions can lower the total cost, size, and power requirements of the system—three important considerations for consumer products. A single microprocessor requires only one programming model and one set of development tools. With a single processor chip, overall system design is simpler.

Philips does not expect TriMedia to displace an x86 processor in general-purpose computer designs. TriMedia can eliminate the need for a separate general-purpose processor in an embedded application, such as a video-game machine, but in PC applications, a TriMedia chip would complement the host processor.

VLIW Yields Simple Hardware

The key to extremely high performance is parallelism, i.e., multiple operations per cycle. For existing

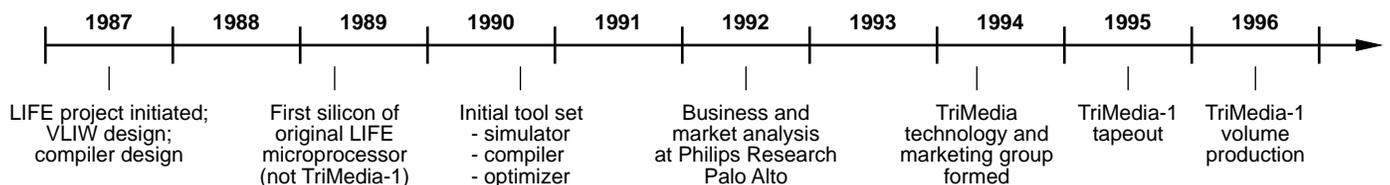


Figure 1. Time line of TriMedia history and near-term plans. The original LIFE project proved the value and feasibility of a VLIW microprocessor, but the LIFE design had some drawbacks.

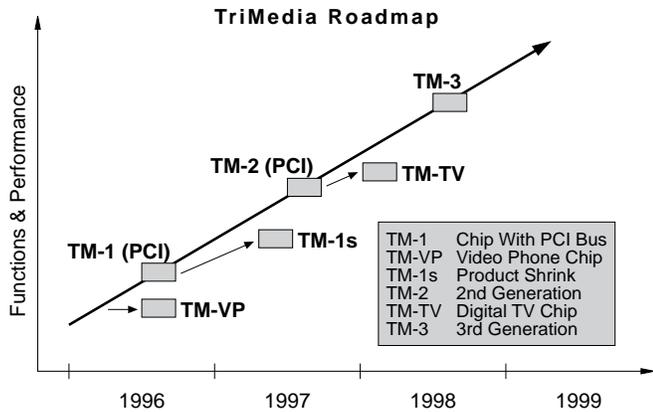


Figure 2. Initial TriMedia roadmap. The first chip will implement a direct PCI interface for use on a PC multimedia plug-in card and operate at 100 MHz. A lower-frequency derivative of this chip (sans PCI interface) will power video phones. The second-generation chip will increase performance and spawn a microprocessor for digital TV. For the third generation, design decisions will be completely re-evaluated.

CISC and RISC instruction sets, this can be achieved only with superscalar implementations. For new instruction sets, VLIW is the least complex way to achieve superscalar performance (see *080205.PDF*).

In a sense, traditional DSP architectures are like narrow, specialized VLIWs: they combine two or three independent operations into a single instruction. For example, a DSP instruction might perform two separate address calculations, memory accesses, and computations. The data operations and addressing modes of a DSP instruction set are tailored to the peculiarities of the intended application domain—signal processing algorithms. Some DSPs have addressing modes that perform the core data-indexing operation of Fourier transforms (bit reversal).

A VLIW architecture has a big hardware advantage over superscalar implementations. Where superscalar chips need hardware to scan the instruction stream looking for independent instructions that can be executed simultaneously, VLIW chips blindly fetch one wide instruction per cycle and execute all operations specified in the instruction. The processor can be this simple because the compiler performs operation scheduling normally done at run time in a superscalar processor. A VLIW microprocessor requires the compiler to perform static instruction scheduling; a superscalar machine performs dynamic instruction scheduling in hardware.

TriMedia = VLIW + DSP

The TriMedia architecture attempts to combine the benefits of VLIW and DSP architectures. The first generation has an instruction with up to five independent operations. If TriMedia were simply a five-wide VLIW, however, it would have only a cost advantage over superscalar implementations of general-purpose micro-

$$\text{me8: } |a - e| + |b - f| + |c - g| + |d - h|$$

— 4 subtracts, 4 absolute values, 3 adds

$$\text{quadavg: } \frac{(a + e + 1)}{2} + \frac{(b + f + 1)}{2} + \frac{(c + g + 1)}{2} + \frac{(d + h + 1)}{2}$$

— 7 adds, 1 divide

Figure 3. Examples of compound, single-cycle TriMedia operations that accelerate multimedia functions. These instructions fetch 32-bit operands, but work on 8-bit quantities. Thus, “a” and “e” are the most-significant bytes of the two operands, “b” and “f” the next-most-significant bytes, etc.

processors. By the time TriMedia arrives, superscalar machines capable of executing five instructions per cycle will probably be fairly common (Even today, IBM’s Power2 can execute six).

The key to TriMedia’s performance advantage is its specialized operations for multimedia algorithms. Figure 3 shows a couple of examples. The me8 (8-bit motion estimation, similar to UltraSparc’s PDIST) (see *081604.PDF*) and quadavg (sum of four rounded averages) instructions implement functions useful in MPEG decoding. These single-cycle instructions fetch two 32-bit operands but operate on four pairs of bytes.

The peak performance of 2.5 billion operations/s is calculated as follows. Each TriMedia-1 instruction has up to five operation slots. Two of those slots can be filled with me8 and quadavg operations. The remaining three slots can each hold a single operation plus guard condition (explained below). Thus, the maximum number of primitive operations per instruction is 25 (11 + 8 + 3 × 2). At 100 MHz, this yields the claimed peak performance.

This performance is clearly not the same as 2.5 billion instructions/s for a general-purpose microprocessor, but for TriMedia’s intended applications it is meaningful. The TriMedia group has used its cycle-accurate simulator to evaluate a mixed multimedia application scenario consisting of MPEG-1 decoding (at 30 frames/s) plus superimposed texture-mapped 3D graphics. For a 100-MHz TriMedia-1 processor, the MPEG-1 decoding consumes 22% of available processor cycles and 12% of memory bandwidth. The remaining resources can render 150,000, 50-pixel triangles/s (texture-mapped, Gouraud shaded, α -blended, fogged, Z-buffered) with a 32-bit SDRAM interface or 250,000 triangles/s with 64-bit SDRAM. TriMedia-1 can also implement MPEG-2 decoding and the H.261 codec (coder/decoder).

They Sought a Better LIFE

One of the things Philips learned from the LIFE experiment was that raw, verbose VLIW instructions are wasteful, especially in a machine with a large number of execution units. While the tight inner loops of performance-critical applications often result in full use of available instruction slots, no-ops must be inserted into

Uncompressed Instruction

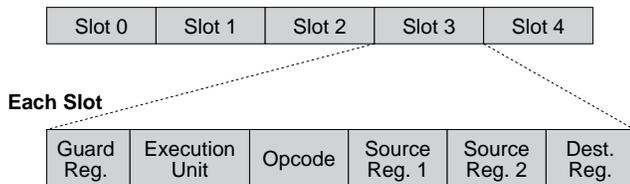


Figure 4. TriMedia verbose instruction format. Actual instructions are described by a format field that saves space by not encoding slots that would simply hold no-ops. Philips is not ready to reveal the instruction encoding in more detail.

some slots most of the time. The result is suboptimal use of bus bandwidth and instruction-cache capacity.

To combat this problem, TriMedia uses a modest (for a VLIW processor) instruction size—five operation slots—even though the first chip will implement 25 execution units. In addition to a relatively small number of operation slots, TriMedia uses a compression technique to avoid wasting space on no-ops.

Figure 4 shows the general TriMedia instruction format with its five operation slots. Each slot has six fields: an opcode, three register operands, an identifier for the execution unit this slot targets, and a guard register. The value in the guard register determines whether or not the operation will be executed, that is, execution of the operation is conditional.

Guarded execution (which was part of the original LIFE architecture) has two important benefits. First, as with conditional execution in the ARM architecture, it eliminates many conditional branches. Instead of a conditional branch around one or more operations, those operations are simply guarded with the opposite of the condition that would have been used in the branch.

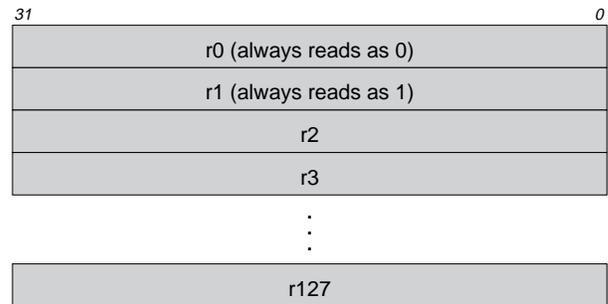
Second, guarding dramatically improves the compiler's ability to fill branch delay slots. As with the first RISC architectures and implementations, TriMedia uses delayed branches (instead of hardware branch prediction) to cover branch latency. Eliminating branch prediction saves hardware; also, the compiler is able to do good static branch prediction.

The actual binary encoding of a given instruction is not as simple as Figure 4 would suggest. Instructions are compressed by removing no-ops and encoding the most frequently occurring operations with the fewest bits. Instructions are byte aligned, and each instruction is described by a short header that tells how many operation slots are in used in the instruction and what encoding is used for each slot.

Hardware Rich in Execution Resources

Figure 5 shows the TriMedia programming model. The architecture provides 128 general-purpose registers, with r0 and r1 containing 0 and 1, respectively. The source, destination, and guard fields in an operation slot

General Registers



Destination Program Counter



Program Control and Status Word

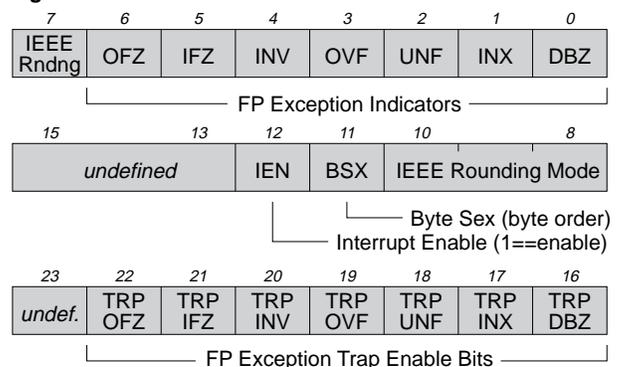


Figure 5. TriMedia Programming model. There are 128 general-purpose registers, but r0 and r1 contain fixed constants 0 and 1 to serve as TRUE and FALSE for use as guard conditions.

can name any of the general-purpose registers. Only the LSB of a register is used for a guard condition; thus, r1 can be used as an unconditional guard if necessary.

The large number of registers provides the compiler with sufficient temporaries to hold speculative values computed by operations in branch delay slots. If a branch takes the direction opposite the one predicted by the compiler, the values computed by the operations in the branch delay slots will simply be ignored by the code along the unpredicted path. Thus, the compiler can use the large number of registers to implement the essential function of a reorder buffer (found in some superscalar implementations) without the hardware cost. This is another way that VLIW (at least in the case of TriMedia) is like a simplified superscalar processor. Instead of requiring hardware for dynamic scheduling of reorder-buffer entries, the compiler statically schedules the "reorder buffer" in a large number of registers.

Figure 6 shows a block diagram of the TriMedia-1 chip. Compared to a raw VLIW, TriMedia has extra hardware for the instruction-decompression logic, the operation-routing network, and the register-routing network. These circuits are not too complex, however.

For More Information

To get more information or arrange a technology presentation, contact Philips Semiconductors' TriMedia group at 408.991.3838; fax 408.991.3300.

Also, the restriction to five operation slots is what makes it possible for TriMedia to use a traditional register file. Without the restriction, the need for 75 register ports (3 for each of 25 execution units) would have been prohibitive, and the designers would have had to resort to a less efficient operand storage structure, such as the "funnel files" of the original LIFE implementation. The register file and operand-routing network make the compiler's job of scheduling operation slots simpler.

TriMedia-1 has 25 execution units, including constant generators, several integer ALUs, DSP execution units (to, for example, execute the me8 and quadavg instructions mentioned above), integer multipliers, integer shifters, branch units, load-store units, and floating-point units. Specialized TriMedia chips may omit some of these units to reduce cost for specific markets.

TriMedia has a branch delay of three cycles, requiring the compiler to generate three delay-slot instructions. To completely fill this delay, the compiler must find 15 useful operations.

Compiler Strategy

The TriMedia group has developed its own compiler strategy based on decision-tree grafting and profile-driven recompilation. Tree grafting provides the ability to generate the large basic blocks needed to fill operation slots in critical loops. Profile-driven recompilation allows the compiler to predict branches accurately and arrange code to speed up the most likely execution paths.

The compiler is responsible for managing processor resources. For example, due to unequal execution-unit latencies, it is possible to issue operations so that more than five results are produced in a given cycle. The compiler knows all latencies and refrains from emitting code that violates basic resource constraints.

Philips Experienced in Consumer Markets

Very low cost is the key to mass-market, high-volume multimedia products, and implementing all computing requirements with a single inexpensive microprocessor is one way to bring costs down. Philips also stresses the value of a unified programming environment for all compute-intensive functions. It says its potential customers that are currently using DSPs rarely use compiler-generated code in DSP applications. When Philips describes the TriMedia programming strategy to experienced DSP programmers, they are either very en-

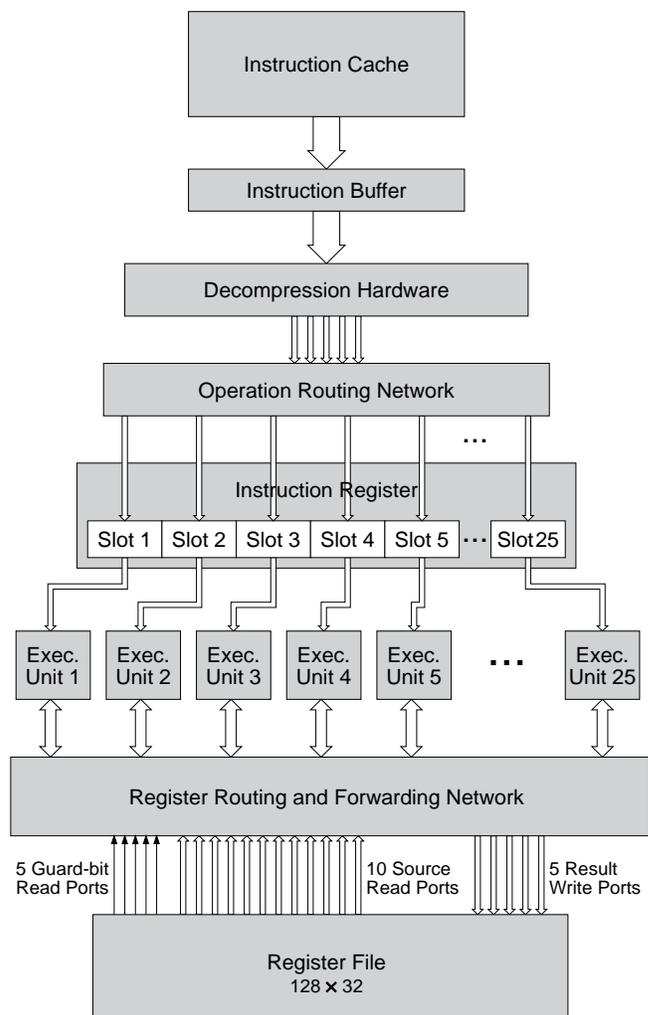


Figure 6. High-level block diagram of the TriMedia-1. Bytes from the instruction stream are queued before being decompressed. The operation-routing network directs the operations to the correct execution units. With a maximum of five operation slots in any instruction, the register file needs fifteen 32-bit ports and five 1-bit ports.

thusiastic or very skeptical.

Philips is convinced of the benefits of using a single processor for several different types of multimedia data, but some industry watchers believe different data types demand different processors. In other markets, several cheap CPUs are still used where one powerful chip could do the job; for example, VCRs often have several 4-bit chips instead of one 16-bit CPU. Philips must demonstrate compelling advantages for TriMedia to succeed.

It is far too early to predict whether or not TriMedia chips will be an industry standard—or even qualify as successful—in delivering low-cost multimedia functions. If Philips can deliver its promises on target, TriMedia will at least have a chance to be an important player. Certainly, no other reasonably priced microprocessor yet announced will reach the multimedia performance levels planned for TriMedia. ♦