

# MICROPROCESSOR REPORT

THE INSIDERS' GUIDE TO MICROPROCESSOR HARDWARE

VOLUME 9 NUMBER 16

DECEMBER 4, 1995

## IBM Extends DSP Performance with Mfast

Powerful Chip Uses Mesh Architecture to Accelerate Graphics, Video



by Dave Epstein

Aiming to push DSP performance to new levels, IBM Microelectronics is developing a powerful chip that will serve as a general-purpose multimedia accelerator. The Mfast architecture will extend IBM's Mwave DSP line with an initial implementation that combines twenty 32-bit processors on a single chip. The chip is designed to sustain 10 billion operations per second (BOPS), all with 16-bit precision, at 50 MHz.

Today, the Mwave family provides high-quality audio and fax/modem services in products such as IBM's ThinkPad and Aptiva PCs. The company, however, purchases graphics and video chips from other vendors. The Mfast (Mwave Folded Array Signal Transform) processor will eventually handle these functions, acting as a complete media processor similar to those recently announced by Chromatic and Philips. But the Mfast architecture is still under development, and IBM does not expect to see Mfast chips in PCs until mid-1997.

The company has taken a different approach than its competitors, designing the transform algorithms first, then building a very powerful engine that executes the complex, yet primitive, operations quickly and efficiently. Mfast is a sophisticated architecture intended to address video teleconferencing, MPEG-1, -2, and -4, and standard 2D and 3D graphics acceleration.

### Mesh Architecture Is Alive and Well

IBM's Gerald Pechanek described the Mfast engine as a highly scalable processor architecture at the Microprocessor Forum in October. The presentation described a  $4 \times 4$  mesh (interconnected two-dimensional array) of 32-bit processors controlled by four sequential processors, all on a single chip. Pechanek described the advantages of the architecture in terms of the folded nature of the mesh. Folding is a method of reducing the wire crossings of a connected grid. The  $4 \times 4$  array is "folded" along its diagonal by taking the upper-right-hand element and

dragging it over to the lower-left-hand corner, as shown in Figure 1 (see below). The same basic design can be applied to a single chip with a  $2 \times 2$  or  $4 \times 4$  mesh for high-performance video and graphics, or to a multichip solution for broadcast-quality video and superior 3D graphics. This article focuses on the  $4 \times 4$  chip, which will be the first silicon and probably the initial product.

The performance of 10 BOPS seems low compared with the 20 BOPS claimed by Chromatic for its Mpact chip (see [091404.PDF](#)). Most of Chromatic's operations, however, are for just a handful of bits; Mpact achieves only 2 BOPS for the 16-bit operations performed by IBM's design. Also, Pechanek points out that 10 BOPS is not a theoretical never-to-exceed rate but rather a sustained rate while executing a 2D DCT (discrete cosine transform) or IDCT (inverse DCT) function, which are common in video compression and decompression, respectively.

As Figure 2 shows, the chip is partitioned into three areas: the processor core, the I/O interfaces, and the synchronous DRAM (SDRAM) controller. The Mfast chip will have a PCI interface and various graphics, video, audio, and communication ports. The SDRAM interface will be 64 or 128 bits wide to deliver the necessary bandwidth for high-end performance. IBM believes that the availability and cost of SDRAMs will be superior to Chromatic's RDRAM approach, even with the downside of requiring more pins and more DRAM chips. A future low-end Mfast might use a 64-bit SDRAM interface and a  $2 \times 2$  mesh.

The processor section is composed of a  $4 \times 4$  mesh of processor elements (PEs) fed by four sequence processors (SPs), as Figure 2 shows. All PEs and SPs have 32-bit processor cores with a  $32 \times 32$ -bit register file, ALU, shifter, and multiply unit. The architecture allows floating-point support, but that is likely to be in a follow-on product. Each SP has its own separate 1K instruction and 1K data caches as well as a memory-control unit, while each PE has mesh-control switches and its own small buffer memories.

## Fold, Bend, and Mutate

The mesh, consisting of an array of 16 PEs, is perhaps the most interesting part of the design. IBM has taken a  $4 \times 4$  mesh with nearest-neighbor connectivity, then combined and deleted connections by folding the array three times, thereby eliminating a great deal of wiring and bus crossings. Figure 1 shows the first fold in the sequence. After the folding, some of the new neighbors were directly connected to make the data flow actually more efficient than the original mesh connectivity for DCTs, IDCTs, and fast Fourier transforms (FFTs).

Busés can be eliminated by recognizing that only some of the connections are used during these operations. For instance, in a SIMD (single instruction, multiple data) array operation, processing elements that accept a result from a North connection usually use the South connection to deliver a result to its neighbor. Most algorithms utilize this single-direction style of communication, which allows the South and East connections, as well as the North and West connections, to be combined. The third fold, which further groups the PEs, totally eliminates the wraparound buses.

The PEs are laid out by hand and optimized for density and connectivity, then replicated 16 times for the mesh. The resulting array is a compact cluster of processors that executes transformation and butterfly operations as efficiently as a fully connected mesh.

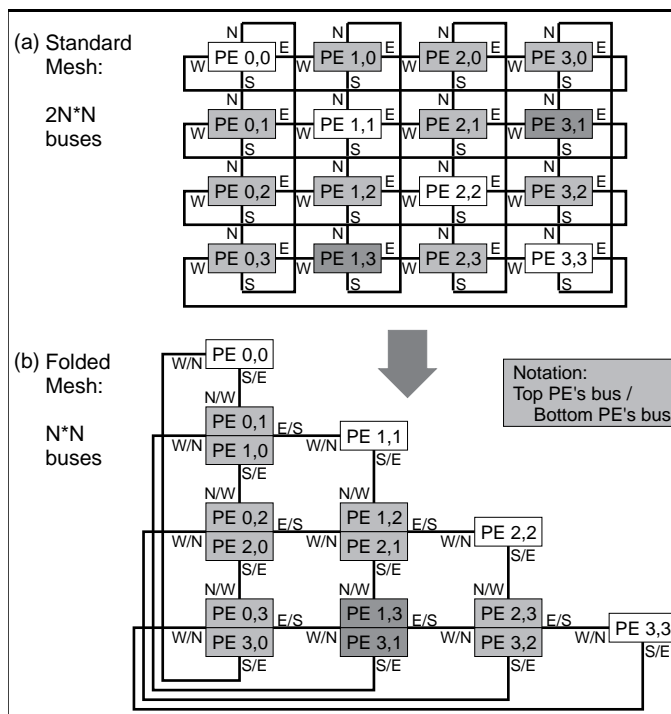


Figure 1. IBM folded the  $4 \times 4$  array of processor elements (PEs), combining the north (N) bus of the top PE of each pair with the west (W) bus of the bottom PE. Likewise, the south (S) and east (E) buses are combined, cutting the total number of buses in half.

## VLIW Processing Elements

A single PE, shown in Figure 3, is a flexible design that can execute either simple 32-bit sequential instructions or 160-bit eVLIW (encapsulated very long instruction word) instructions, which look much like horizontal microinstructions. Each eVLIW is broken into five 32-bit fields, which directly control each of the three function blocks in the PE plus one load and one store operation. The eVLIW instructions are stored in a 16-entry writable VIM (VLIW instruction memory) within each PE. The parallel operations generated by eVLIW instructions allow Mfast to reach its peak performance.

Instructions arrive at the PE by way of the 32-bit Instr-N bus. The sequential instructions are executed directly, but some instructions, which IBM calls surrogate execute-VLIW-indirect instructions, cause an eVLIW instruction to be dispatched from the VIM. The surrogate addresses the VIM and can specify modifiers for the eVLIW instruction, based on the PE's mesh location, for source and destination information.

The VIM must be initialized for each function to be executed, but it typically will contain some frequently used routines. The IDCT, for instance, requires just four VIM locations and is used so often in video decoding that it is left resident most of the time. Other functions may come and go as needed. Loading the 160 bits is costly, since only 32 bits can be loaded at a time, but it can be done in the background while the PE is executing other VIM-resident functions or, for the tricky programmer, from locations that are being modified on the fly.

Data arrives over the 32-bit Data-In-N bus, multiplexed with a local 128-entry scratchpad memory that holds 32-bit constants or temporary data. A 32-entry register file with six write and nine read ports provides data to the MAU, a multiply/add unit; the DSU, or data selector unit, for shifting and multiplexing; and the ALU, for logicals, adds, and subtracts. In a future implementation, the MAU and ALU could be modified to perform floating-point operations.

Results are fed simultaneously to the PE's local register file and to a connected neighbor's register file. Each PE can receive one external write to its register file per cycle. The 32-bit Data-Out-N bus can return data from the register file or the PE's scratchpad memory to an SP.

Conditionals are handled by a conditional move in the DSU at the PE site or by reporting condition codes back to the dispatching SP. Pechanek is currently investigating the usefulness of conditional execution within a PE.

The PE has a simple four-stage pipeline: fetch, decode, execute/writeback, and conditional return. The register files receive results at the end of the execute cycle from the PE's internal operations and possibly those of connected neighbor PEs. This tight register file interconnection provides zero latency communications between

PEs and allows algorithms to be executed quickly and efficiently while being relatively easy to program. Data going back to the SP is delayed by one cycle, since it must first be written into the PE's register file before making it to the Data-Out-N bus.

One might think the data buses that connect PEs to SPs would be connected by rows or columns. Once again, however, IBM turned to folding to match the row and column wiring with the access requirements of the algorithms. Figure 2 shows that this arrangement, along with a switch placed at each symmetrical pair (for example PE 1,3 and PE 3,1), allows SP number N to access column N or row N without committing a lot of silicon to wiring. Note that this fold is for the SP-to-PE data and instruction bus wiring and shouldn't be confused with the PE-to-PE result bus folding discussed previously. IBM has optimized each of these topological folds for physical and algorithmic efficiency.

### Sequence Processors in Control

The SPs are general-purpose 32-bit processors that handle sequential (nonmesh) operations as well as dispatching and controlling the parallel transforms. IBM chose to use the same core as for the PEs, since it was already a compact execution unit with robust facilities. IBM added a memory interface consisting of separate instruction and data memories, each 1K in size, and a connection to the external memory interface shared among all SPs. Each SP has a VIM, although it is typically loaded with more straightforward and primitive operations than those contained in the PEs.

Each SP fetches instructions independently from either its own I-Mem-N, which acts as a cache, or main memory. These instructions are distributed to the SP, and to the column of PEs directly connected to it, by way of the Instr-N bus. SP-0 has a special mode to control the entire array, allowing two-dimensional SIMD operation. This mode must be set up in advance, so the other SPs do not interfere. The SP pipeline is the same as for the PEs, and all must execute in lock step for array operations.

Like most DSPs, the memory interface uses physical addressing and is straightforward. Addresses are 32 bits in length, using base-plus-offset, immediate, or indexed modes. Future implementations could add virtual addressing if necessary. The chip supports up to 16M of SDRAM; a typical system may have 2M configured with four 256K $\times$ 16 SDRAMs to achieve a 64-bit interface. The 128-bit interface would require either four  $\times$ 32 parts or eight of the  $\times$ 16 parts. The interface will support DMA functions, block accesses with a stride, and four queued requests to support the four SPs. The requests can be simultaneously queued and are pipelined to utilize the full bandwidth of the SDRAM.

The instruction and data memories are software managed, although the data memory does perform some

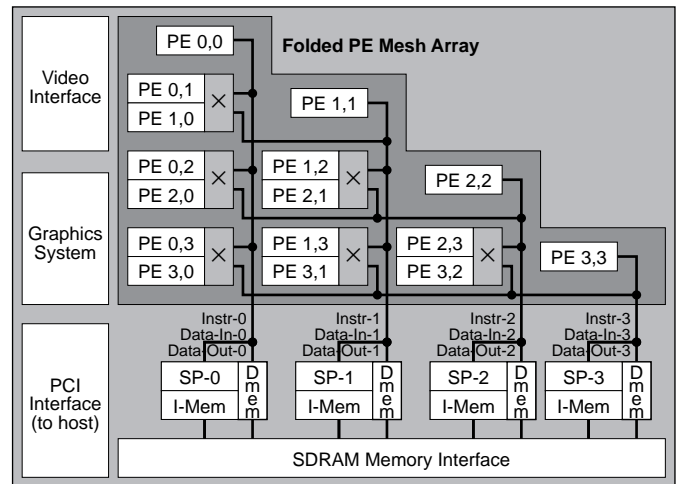


Figure 2. The Mfast processor core contains a folded 4  $\times$  4 mesh of 16 processor elements (PEs) controlled by four sequence processors (SPs). Each SP has its own instruction memory (I-Mem) and data memory (D-Mem). The chip connects directly to video, graphics, SDRAM, and PCI bus.

automatic prefetching. This gives all the flexibility (and burden) to the programmer to optimize memory allocation on each of the SPs.

### “Reasonable Die Size”

Although IBM is not supplying specifics, Pechanek indicated that the Mfast die size, using IBM's 0.35-micron CMOS-5X technology, is reasonable and competitive, even with the 4  $\times$  4 configuration. This is no simple feat, given the twenty 32-bit processors on board. Process shrinks and a 2  $\times$  2 array version should eventually provide a more cost-competitive solution. A catch may be the price of the SDRAM, which must come down for sub-system costs to be competitive.

Although no silicon exists, a fully functional VHDL model is running. The model is being used to write and debug algorithms for MPEG and graphics acceleration while still allowing some tweaks to the hardware before committing to silicon. IBM is developing a compiler, as-

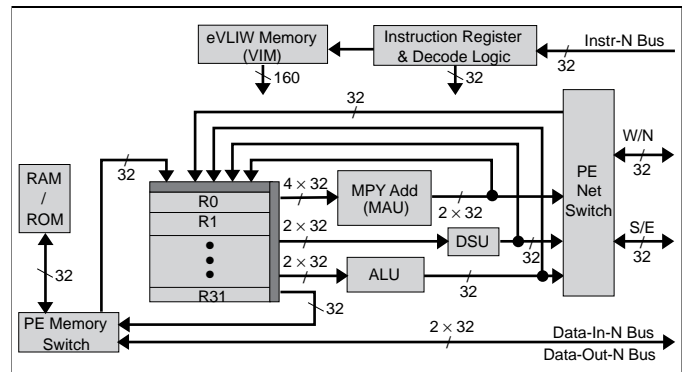


Figure 3. Each processor element (PE) executes 160-bit VLIW instructions and has a peak throughput of five 32-bit operations per cycle: one each from the MAU (multiply-add unit), DSU (data shift unit) and ALU, plus one load and one store.

sembler, and software support tools to aid in debug and data-flow analysis. The PE and SP modules are in design and layout with the tape-out date remaining unspecified. A good guess at the schedule would put prototype availability in 2H96, leading to full production in mid-1997.

The familiar API software model is used for Mfast as it is with Mwave today, as well as by Nvidia and Chromatic. IBM will supply drivers for standard Windows APIs such as GDI and DirectX (see MPR 11/13/95, p. 3). For existing applications, the separate Mwave chip can supply Sound Blaster compatibility. The drivers interface with the Mwave Manager, running on the host processor, which is in direct communication with the Mwave/OS running on the Mwave and Mfast processors. The Mwave and Mfast chips each have their respective local memories as well as analog hardware interfaces.

### IBM on the High Road

IBM has chosen to build what it is best at: complex and high-performance processors that are based on fundamental algorithmic technology. Instead of attacking the end-user problem from the top, it has focused on the primitive functions that must be executed quickly and efficiently to get the best performance. Although this could lead to expensive mainframe-like solutions, advances in manufacturing technology hold the promise of making these designs cost-effective. IBM's competitors could find themselves accelerating yesterday's applications, providing lower performance, albeit less expensive, solutions. The IBM approach could provide the best performance at a small premium.

The interim solution of delivering Mwave and Mfast as a two-chip design could give IBM the edge. The Mwave audio and communications capabilities, including Sound-Blaster compatibility, are already proven. The company can concentrate the initial Mfast on video and graphics without the burden of audio integration and software



**Mfast architect Gerald Pechanek of IBM explains the advantages of a folded mesh design.**

MICHAEL MUSTACCHI

## For More Information

Mfast is not a product, but Mwave information can be received by calling IBM Microelectronics fax service at 415.855.4121. IBM Microelectronics' home page is located at [www.chips.ibm.com](http://www.chips.ibm.com).

interaction problems. This, in fact, may be a more natural partition, since communications cannot drop any data, and human audio perception is more discerning than video. Therefore, Mfast can use different tradeoffs of algorithmic speed vs. absolute real-time accuracy. The two-chip solution, however, may be more expensive than a single media processor.

The delivery schedule is the key to success. IBM is behind first-generation solutions such as chips from Nvidia and Chromatic. The company is now in a race to see whether Mfast can intercept these lower-end solutions as they migrate upward, delivering better performance at a similar price. IBM's engine may be more powerful, but it is also more difficult to program. Like other media processors, Mfast also faces the software compatibility challenge.

By the time Mfast debuts, Nvidia and Chromatic will be delivering improved versions of their current devices, Philips' Trimedia processor will be established in the market, and companies such as S3 will have their own multimedia accelerators. To gain attention in such a crowded market, Mfast must deliver outstanding performance at a reasonable price. The initial design shows that IBM is on the right track to outperform competing devices. Now, the company must bring the product to market at a competitive price. ♦

*Dave Epstein was previously VP of engineering at NexGen. He is now an independent consultant and can be reached at 415.493.8332.*