

TFP Designed for Tremendous Floating Point

New MIPS Processor has Dual FPUs, High Cache Bandwidth

By Linley Gwennap

MIPS Technologies, Inc. (MTI) has received first silicon on its ambitious TFP processor. The chip set initially will be deployed in a Silicon Graphics (SGI) parallel supercomputer (see [070202.PDF](#)). Toshiba will manufacture TFP for SGI and also plans to sell the chip set to other companies; several supercomputer vendors are reportedly interested in the new processor, due to its eye-popping peak performance of 300 MFLOPS and 1.2-Gbytes/s bandwidth to its large external cache.

TFP can issue up to four instructions per cycle using any combination of its two integer units, two FPUs, two load/store pipes, and one branch unit. Unlike the super-pipelined R4000 chips, TFP uses a shorter five-stage pipeline but rearranges the stages to reduce the load-use penalty. As this reordering increases the mispredicted-branch penalty, the chip uses a large branch-prediction cache to reduce the number of mispredicted branches.

The off-chip cache acts as a large first-level cache for floating-point data as well as a second-level cache for instructions and integer data. The tag and data SRAMs are interleaved, allowing the processor to reach its bandwidth goal. The cache requires custom tag RAMs and uses synchronous SRAMs for the data. Because of the extra expense of the two-chip processor and high-performance cache, TFP will not replace the R4400 at the high end of the MIPS family; instead, it will satisfy SGI's need for maximum floating-point performance until T5 is available around the end of 1994.

MTI has not yet tested the first chips but expects that the 75-MHz TFP will match the integer performance of a 150-MHz R4400 and more than double its floating-point performance. This translates to nearly 100 SPECint92 and well over 200 SPECfp92. The floating-point rating could exceed all current processors; DEC's 200-MHz 21064 is the leader at 200.4 SPECfp92. Given the complexity of verifying this processor, MTI now admits that TFP may not ship until 2Q94. The expected high price of the chip set will limit it to only the highest-end applications.

MIPS IV Extensions

TFP is the first processor to implement the MIPS IV architecture. It includes all of the 64-bit extensions of the R4000's MIPS III. Most of the new extensions are focused on improving floating-point performance.

One key to TFP's high peak MFLOPS rating is the ability to issue an FP multiply and FP add in a single in-

struction. MIPS IV includes a new MADD instruction to accomplish this. This is a three-operand multiply-add ($A \times B + C \rightarrow D$) rather than the four-operand independent multiply-add ($A \times B \rightarrow C$; $D + E \rightarrow E$) implemented in PA-RISC. The MADD instruction assists the large number of scientific algorithms that calculate sums of products. Like POWER's FMADD, it is not compliant with the IEEE standard because it does not do rounding between the multiply and the add. The lack of rounding, however, allows both higher precision and higher performance.

MIPS IV adds a new addressing mode, register+register, but only for FP loads and stores. Most other RISC architectures (except Alpha) already include some form of register+register addressing for both integer and FP memory accesses.

The new architecture also includes four conditional move instructions similar to those in SPARC version 9 and Alpha. These operators allow many IF-THEN clauses to be coded without branches. Eliminating branches becomes increasingly important as the number of potential issue slots per cycle grows.

Chip Set Partitioning

Figure 1 (see below) shows a block diagram of the TFP processor. The integer unit (IU) is the more complex chip; it handles instruction dispatch and all integer arithmetic and also contains the on-chip data and instruction caches. The floating-point chip (FPC) performs all floating-point functions. The IU generates all addresses and control signals for the external cache, but the FPC is the only chip that can write to that cache.

The IU includes a 1024-entry branch address cache that contains the predicted target address (if any) for each group of four instructions in the 16K instruction cache. Instructions are read, four at a time, into the instruction buffer before being dispatched to the appropriate functional unit.

The general-purpose (integer) register file implements nine read ports and four write ports, as shown in Figure 1. Only one of these read ports supplies information to the data cache, limiting the processor to one store operation per cycle. The register file can retire results from two loads and two math operations on each clock.

The integer arithmetic unit contains two ALUs for general math, one shifter, and one multiply/divide unit. This means that the CPU cannot issue two shifts or two multiply/divide instructions on the same cycle but can handle most other combinations of integer math operations. Integer multiply and divide are multiple-cycle op-

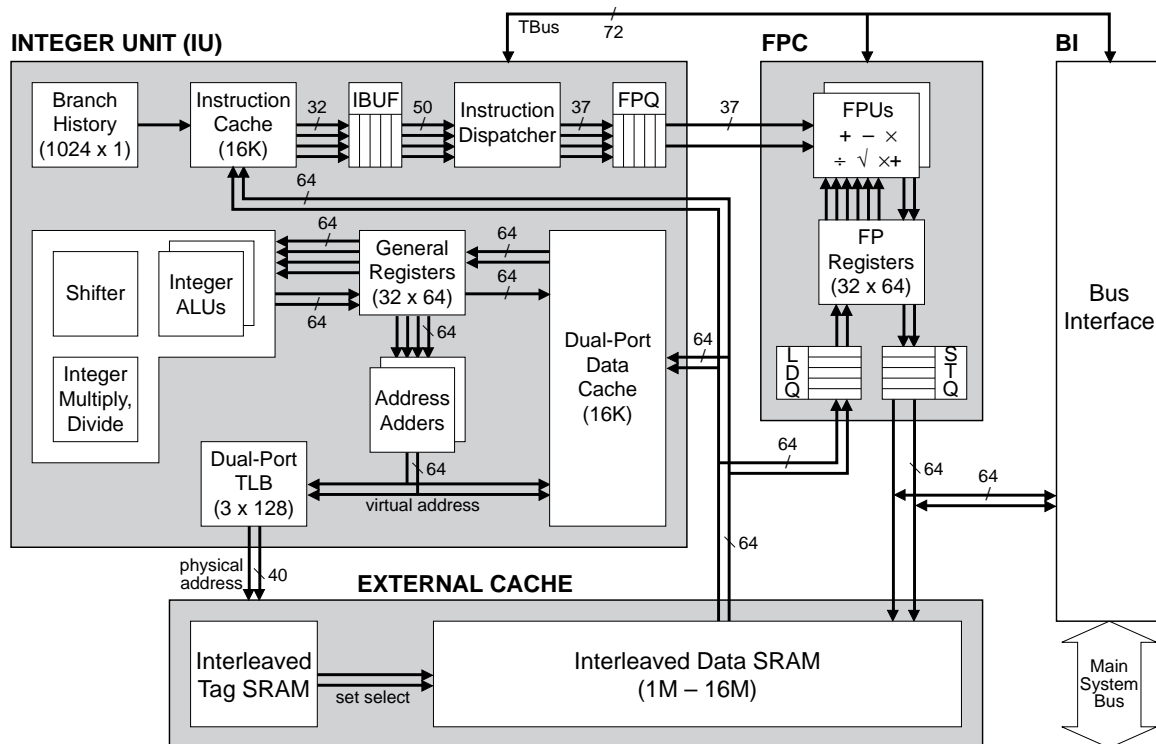


Figure 1. The TFP processor requires two chips plus special tag RAMs and synchronous SRAMs for the external cache.

erations and only one can be in progress at a time.

The two address-generation units can each calculate one virtual address per cycle. The 16K data cache is dual-ported, providing two 64-bit values from independent addresses. Unlike Pentium's dual-ported cache, there are no load restrictions due to bank conflicts. The TLB is also dual-ported, translating two addresses per cycle. It has a total of 384 entries, far more than most microprocessors, to handle large data sets. Making such a large TLB fully associative is difficult, so the TLB is divided into three sets of 128 entries each.

The FPC contains the floating-point register file, which has eight read ports and four write ports. Six of the read ports are needed to launch two MADD instructions, while the remaining ports allow two load/store operations per cycle. The dual FPUs perform all of the standard IEEE operations as well as MADD.

MTI calls the external cache a "streaming" cache because of its high bandwidth. It can deliver two 64-bit values per cycle at 75 MHz for a total of 1.2 Gbytes/s. The external cache can be as large as 16M. The bus interface (BI), which is not part of the TFP chip set, connects to main memory and the system bus.

Aggressive Superscalar Dispatch

At four instructions per cycle, TFP has a higher issue rate than any implementation of a popular RISC architecture except IBM's RIOS processor (see μ PR

8/21/91, p. 10). RIOS can also issue four instructions per cycle, but only if they consist of an integer operation (including loads and stores), an FP instruction, a condition-code instruction, and a branch. As a result, the number of cycles in which RIOS reaches its peak issue rate is vanishingly small.

TFP should be much more successful in this regard due to its larger number of execution units. The most common operations are integer math, loads, and stores; TFP can issue two math instructions and two loads (or one load and one store) per cycle. The loads and store can be for either integer or floating-point data. Up to two floating-point math instructions can be issued as well, either or both of which can be MACs. The dispatcher can also issue one branch per cycle.

TFP uses a six-entry buffer (IBUF in Figure 1) and complex alignment logic to ensure that the dispatcher always has four consecutive instructions from which to choose. The dispatcher looks for data dependencies and resource conflicts in the set of four instructions and issues as many as possible. Instructions are always issued in order, so any instruction that cannot be issued holds up all successive instructions. In some cases, the first available instruction is stalled due to a cache miss or long-latency math operation, and no instructions are issued on that cycle.

TFP includes a 15-entry FP-instruction queue (FPQ in Figure 1) that decouples FP execution from the inte-

ger pipeline. If an FP operation is waiting for data from the external cache, it is placed in the queue until the data is available; successive operations are also queued to maintain in-order execution. Once the data is available, instructions are issued from the queue. In this way, integer execution can continue even if FP execution is stalled due to cache or FPU latencies.

Five-Stage Pipeline with a Twist

The processor uses a five-stage pipeline, as shown in Figure 2. The basic sequence of operations is quite similar to traditional RISC processors such as the R3000. One major difference is that ALU operations are performed in the fourth stage rather than the third stage. Since these calculations are in the same stage as the data-cache access, data can be loaded on one cycle and immediately used in the next cycle, eliminating the load-use penalty.

Figure 3 shows a load-use code sequence, which is very common; with TFP's high issue rate, even a one-cycle load-use delay would require four instructions between the load and the use to avoid a penalty. MTI believes that the compiler would be unable to fill most of these slots; such a design would also reduce TFP's performance on uncompiled R4000 code.

Rearranging the pipeline does have its drawbacks, which is why other processors have not taken this approach. The TFP pipeline creates an address-use penalty not found in most processors. This penalty occurs when an address is calculated by one instruction and is needed by the following instruction, as shown in Figure 3. This situation causes a one-cycle interlock in TFP, but MTI believes that the address-use sequence is much less common than the load-use sequence. The new register+register addressing mode will help recompiled code reduce the need to precalculate addresses, at least for floating-point loads and stores.

Prediction Eases Branch Penalty

The modified pipeline has an additional drawback: it extends the branch penalty by one cycle, since the branch condition is not resolved until the fourth stage. TFP reduces the effect of this penalty by using branch prediction. When the prediction fails, however, it takes three cycles to restock the pipeline. Since up to four instructions could have been issued on each of those cycles, as many as 12 instruction slots are lost on a mispredicted branch.

The branch cache increases prediction accuracy. Each entry in the branch cache contains a target address and a valid bit. When the prefetch buffer loads a block of four instructions from the instruction cache, it checks the corresponding entry in the branch cache; if the valid bit is set, the next set of instructions is taken from the indicated target address. If the prediction is correct, execu-

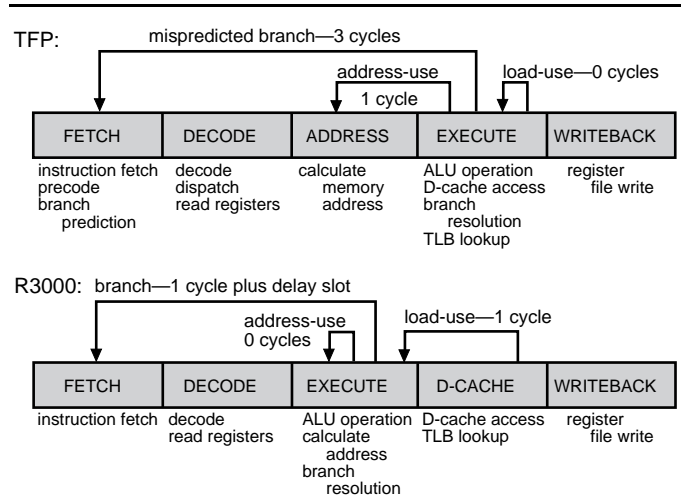


Figure 2. TFP uses a five-stage pipeline but its execute stage is later than in most other processors, including the R3000.

tion continues without any delay, although some dispatch slots are lost if the target is not quadword-aligned.

MTI selected the single-bit algorithm used by Alpha rather than the two-bit algorithm used by Pentium (see *070402.PDF*). Although the two-bit algorithm can yield higher accuracy, it requires a two-ported RAM that can be read and written at the same time. TFP updates its branch cache only in the case of a mispredicted branch, which allows the cache to be written during the three-cycle penalty. The single-ported RAM cell is smaller and aligns perfectly with the instruction cache RAM. This allows for a 1024-entry branch cache that provides about the same prediction accuracy as Pentium's 256-entry, two-bit branch cache, according to MTI.

All things considered, delaying ALU operations in the pipeline makes sense for TFP because branch prediction reduces the branch penalty while the four-instruction dispatch exacerbates the potential load-use penalty. This pipeline sequence may appear in other highly superscalar processors in the future.

TFP Emphasizes FP Issue Rate

Although the processor has been optimized for floating-point calculations, the FPU (designed by Weitek) is unimpressive. Basic operations—add, multiply, convert, and even multiply-accumulate—can be issued in a single cycle but have a four-cycle latency. At 75 MHz, this

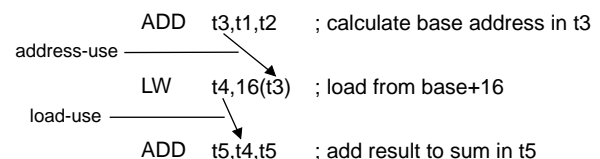


Figure 3. TFP's unusual pipeline eliminates the traditional load-use penalty but creates a less common address-load penalty.

| Operation | Issue Rate | | Latency | |
|-----------------|------------|-----------|-----------|-----------|
| | SP | DP | SP | DP |
| Load | 1 cycle | 1 cycle | 5 cycles | 5 cycles |
| Store | 1 cycle | 1 cycle | n/a | n/a |
| Add/Subtract | 1 cycle | 1 cycle | 4 cycles | 4 cycles |
| Convert | 1 cycle | 1 cycle | 4 cycles | 4 cycles |
| Multiply | 1 cycle | 1 cycle | 4 cycles | 4 cycles |
| Multiply-Add | 1 cycle | 1 cycle | 4 cycles | 4 cycles |
| Divide | 11 cycles | 17 cycles | 14 cycles | 20 cycles |
| Sq. Root | 11 cycles | 20 cycles | 14 cycles | 23 cycles |
| Reciprocal | 5 cycles | 11 cycles | 8 cycles | 14 cycles |
| Recip. Sq. Root | 5 cycles | 14 cycles | 8 cycles | 17 cycles |

Table 1. Each FPU can issue common FP operations in a single cycle with a four-cycle latency, but other operations take longer.

works out to 53 ns of latency, compared to 20 ns for the PA7100 and 30 ns for DEC's 21064. Latency is only a factor when an instruction requires the result of a calculation that is still in progress; MTI points out that modern compilers generally avoid such data dependencies through loop unrolling and other techniques.

The TFP design emphasizes the issue rate of FP operations over latency. By including two pipelined FPUs, the processor can sustain two FP operations per cycle for the basic operations listed previously. As shown in Table 1, operations such as divide and square root take longer to calculate. These operations lock up an FPU for several cycles, although the four-stage execution pipeline allows a new operation to be issued three cycles before the current calculation is completed.

Pipelined Cache Design

The IU contains 32K of cache split evenly between instructions and data. Both the instruction and data caches are direct-mapped and virtually indexed. The instruction cache is virtually tagged, speeding accesses by eliminating the need for address translation. The data cache uses a write-through protocol and physical tags to simplify coherency with the external cache. Instruction-cache coherency is not maintained in hardware, although the instruction cache does maintain a process identifier for each cache line so flushing is usually not required on a process switch. Neither on-chip cache includes parity protection.

TFP uses a two-level cache design but, as with the pipeline, there is a twist. The on-chip data cache stores only integer data; all FP data is kept in the external cache, which also acts as a second-level cache for both instructions and integer data. Because the FP register file is on a separate chip from the data cache, single-cycle accesses to the data cache would not be possible at 75 MHz. Most floating-point programs use large data sets, reducing the effectiveness of a 16K first-level cache; for these programs, the multi-cycle overhead of checking the smaller cache outweighs the benefit of an occasional cache hit.

FP loads go directly to the external cache, completing in five cycles. The use of synchronous SRAMs for both tag and data allow the load path to be fully pipelined, as shown in Figure 4. This pipelining, along with the FP instruction queue, eliminates most load delays.

The cache tag RAM was designed by MTI and is fabricated by Toshiba. It contains 8K entries organized in four sets. Each entry includes 20 bits of tag address and 8 state bits, which maintain a four-state MESI-like protocol for each of the four blocks in the cache line. The tag RAMs implement address-compare logic, as well as input and output latches that pipeline the accesses.

The cache data RAMs are standard synchronous SRAMs that MTI expects to be available from multiple sources. Unlike SuperSPARC or Pentium, TFP requires "separate I/O" synchronous SRAMs, which have separate read and write ports. Although only one operation (read or write) can be performed at a time, the split buses eliminate the turnaround time that occurs when a read is followed immediately by a write on a single bus.

Interleaving Increases Bandwidth

The external cache is two-way interleaved and can deliver two results on each cycle. Data is organized so consecutive 64-bit values are in alternate banks. Both the IU and FPC have two 64-bit buses, one for each bank. Each bank uses its own tag RAM, although both tag RAMs contain the same information. A typical 4-Mbyte cache design uses 32 SRAMs, each 256K × 4. Two additional parts are needed for parity. Since the data RAMs are synchronous, they must be accessible within the 13.3-ns cycle time; in practice, 12-ns parts are used.

The interleaved cache can sustain two accesses per cycle only if each pair of accesses consists of any even address and any odd address. Pairs of even or odd addresses can reduce an interleaved cache to one access per cycle, but MTI has added an "address bellow" that, coupled with the load and store queues, helps solve alignment problems. If a pair of addresses both need the same bank, only one can be issued, but on the next cycle, the bellow tries to pair the remaining address with one from the next queue entry. As a result, consecutive accesses to odd and even banks are always paired, even if they weren't issued on the same cycle.

The external cache is four-way set-associative and uses physical indexes and tags. The line size is programmable up to 512 bytes; long lines can improve performance for large data sets. Each line is split between the two banks.

The pipelined cache allows integer loads to be overlapped just like FP loads. There is no integer instruction queue, however, so all integer data dependencies stall the instruction pipeline until the necessary data is returned from the cache. The integer load penalty, including the time needed to restart the pipeline, is eight cy-

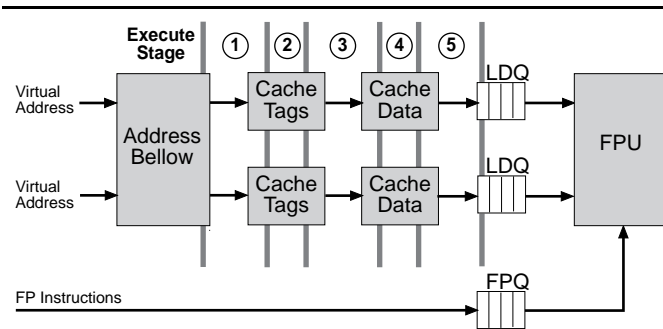


Figure 4. Using pipelined SRAMs for tag and data, TFP's external cache can service two 64-bit loads every cycle. Data and instruction queues help hide the five-cycle latency.

cles. For integer stores that miss the on-chip cache, the data must be passed to the FPC using the TBus, since the IU has no direct write access to the external cache.

On any accesses that miss the external cache, the bus interface (BI) is responsible for fetching the data from main memory and loading it into the external cache. MTI is not providing a BI chip for TFP, as most high-end system vendors use proprietary buses and want to design their own ASIC. The BI communicates status to TFP through the TBus and can read and write the external cache directly. MTI recommends that the BI use a third tag RAM so it can service snoop requests without interrupting the processor.

Large Die, Large Package

The dual cache buses increase TFP's pin count dramatically. Both the IU and FPC have two 64-bit read-data buses (one for each bank) along with the 72-bit TBus. The IU also has 52 bits of address and control for each of the two tag RAMs. The FPC doesn't drive the tag RAMs, but it has two 64-bit write-data buses for the interleaved cache.

Both the IU and FPC use a 591-pin ceramic PGA with 382 signal pins. Even in a 0.7-micron CMOS process with three metal layers, both chips have a 298-mm² (17.2 × 17.3 mm) die, which is larger than Pentium and near the limit of a 25-mm reticle. The IU has 2.6 million transistors, while the FPC has just 830,000; the FPC could use a smaller die, but is limited by the large number of pads.

The two chips operate at 3.3V and still dissipate about 15 watts each at 75 MHz. All of the interfaces use low-voltage TTL (LVTTTL) signals but also tolerate the use of 5V signals.

Based on the μ PR Cost Model (see *071004.PDF*), we estimate the manufacturing cost of the two chips to be \$850 total. This compares to an estimated cost of \$480 for Pentium and just \$185 for the R4400. The TFP cost is very high due to its two-chip design, enormous dice, high-pin-count packages, and expected low volumes. Toshiba has not yet announced pricing for TFP; MTI hopes that the price will be around \$2000–\$3000.

Price and Availability

The TFP chip set will be marketed by Toshiba, but that company has not yet announced price and availability. For more information, contact Jean-Claude Toma at Toshiba America, 9775 Toledo Way, Irvine, CA 92718; 714/455-2227, fax 714/859-3963.

Flaming Arrow Hits Niche Target

TFP is a technology *tour de force* that includes several interesting features. The dual integer units, dual load/store pipes, and dual FPUs make this the most flexible superscalar processor yet revealed, although the imminent RIOS-2 will give it stiff competition. TFP's pipelined cache delivers high bandwidth with relatively inexpensive 12-ns SRAMs, allowing a large yet fast external cache. The PA7100, by contrast, achieves similar bandwidth but requires leading-edge 9-ns SRAMs that limit the cache size to 512K.

The new processor is clearly optimized for scientific applications that can take advantage of the dual FPUs and have large data sets that need a big, fast cache. For computational fluid dynamics, weather simulation, and similar applications, TFP should outperform any current microprocessor. In fact, no processor in the next year is likely to offer a similar set of features. TFP's major competition in this area will come from traditional supercomputers and minisupercomputers, along with emerging massively-parallel systems using proprietary and merchant-market microprocessors. Compared to these systems, TFP should have a significant cost advantage.

The high-end scientific market is a small one that measures annual sales in thousands of units, not millions. TFP will also be attractive for technical applications that are floating-point intensive, such as CAD and 3-D modeling. These applications, however, are less sensitive to cache bandwidth than scientific code, and next-generation processors from DEC, HP, and IBM are likely to match TFP in SPECfp92 performance. Furthermore, many technical users are sensitive to cost; TFP will be at a significant cost disadvantage compared to single-chip microprocessors.

MIPS will compete in these cost-sensitive markets with the R4400 and eventually with T5, which should match TFP's floating-point throughput with greatly improved integer performance all on a single-chip. SGI hopes that TFP, by reducing the cost of supercomputer performance, will increase the size of the ultra-high-end market and spawn future TFP chips with even better performance. TFP will never be a mainstream product, but it could redefine the concept of a supercomputer and put competitive pressure on Cray and others. ♦