TFP MICROPROCESSOR CHIP SET

mips
Open RISC Technology

PRELIMINARY
PRODUCT INFORMATION

October, 1993

- ❏ Advanced Superscalar Architecture
  - Supports Four Instructions per Cycle
  - Two Load/Store Instructions per Cycle
  - Two Integer and Two Floating Point Execute Instructions per Cycle
- ❏ High Performance Design
  - 75 MHz Clock Rate
  - 3.3 Volt Technology
  - Separate Integer and Floating Point Chips
  - 300 Double Precision MFLOPS Peak
  - On-Chip Floating Point Instruction Queue
  - Separate 64 bit Load and Store Data Busses
  - Implements MIPS IV Instruction Set
- ❏ High Integration Chip-Set
  - Integer Unit Contains:
  - 16 KByte Dual Ported Data Cache
  - 16 KByte Single Ported Instruction Cache
  - 384 Entry Dual Ported Translation Lookaside Buffer
  - 1K Entry Branch Prediction Cache
  - Second Level Cache Support
- ❏ Optimized for Floating Point Performance
  - Separate-Chip Floating Point Unit
  - Two Floating point Execution Units
  - Two Floating Point Arithmetic and Two Floating Point Memory Operations per Clock
  - Large Load/Store Data Queues
- ❏ Second Level Cache Support
  - Two 4-Way Set Associative Tag RAM Chips
  - Delivers Two 64-bit Operands to the Floating Point Unit Every Clock.
  - Each Tag RAM has a Dedicated Bus Interface to the Integer Unit.
- ❏ Compatible with Industry Standards
  - ANSI/IEEE Standard 754-1985 for Binary Floating Point Arithmetic
  - MIPS III Instruction Set Compatible
  - Conforms to MESI Cache Consistency Protocol
  - IEEE Standard 1149.1/D6 Boundary Scan Architecture

The MIPS TFP Microprocessor Chip Set from MIPS Technologies implements a super-scalar architecture, providing super-computer performance at a fraction of the cost. The 64 bit architecture of the MIPS TFP Microprocessor is implemented using separate integer and floating point devices. The impressive floating point performance of the TFP Microprocessor Chip Set makes it ideal for applications such as engineering workstations, scientific computing, 3-D graphics workstations, and multi-user systems. The high throughput is achieved through complete separation of the integer and floating point functions, the use of wide, dedicated data paths, and large on- and off- chip caches.

The TFP Microprocessor Chip Set implements the MIPS IV instruction set. MIPS IV is a superset of the MIPS III instruction set and is backward compatible. Implementing a 3.3 volt technology with a target frequency of 75 MHz, the TFP Microprocessor Chip Set delivers peak performance of 300 MIPS and 300 MFLOPS. The integer unit contains 2.6 million transistors. The floating point unit contains 830 thousand transistors. Each device is housed in a 591 pin PGA package and is fabricated using the VHMOSIII 0.8-micron silicon technology.

1.0 INTRODUCTION TO THE TFP MICROPROCESSOR CHIP SET

2.0 TFP FUNCTIONAL UNITS

2.1 INTEGER UNIT

- 2.1.1 Instruction Cache
- 2.1.2 Instruction Cache Tag RAM
- 2.1.3 Branch Cache
- 2.1.4 Instruction Queue
- 2.1.5 X-Bar
- 2.1.6 General Purpose Registers
- 2.1.7 Arithmetic Logic Units
- 2.1.8 Translation Lookaside Buffer
- 2.1.9 Data Cache
- 2.1.10 Data Cache Tag RAM
- 2.1.11 Data Cache Valid RAM
- 2.1.12 Integer Unit Bus Interface
  - 2.1.12.1 Floating Point Unit
  - 2.1.12.2 Even Tag RAM
  - 2.1.12.3 Odd Tag RAM
  - 2.1.12.4 Even Data RAM's
  - 2.1.12.5 Odd Data RAM's
  - 2.1.12.6 Cache Controller
- 2.1.13 The TBus
- 2.1.14 Clock Interface
- 2.1.15 Integer Unit Package Pinout

2.2 FLOATING POINT UNIT

- 2.2.1 FPU Operations
- 2.2.2 TBus Interface
- 2.2.3 FPU Bus Interface
  - 2.2.3.1 Integer Unit
  - 2.2.3.2 Even Data RAM's
  - 2.2.3.3 Odd Data RAM's
  - 2.2.3.4 Cache Controller
- 2.2.4 Clock Interface
- 2.2.5 FPU Package Pinout

2.3 TAG RAM

- 2.3.1 Tag RAM Organization
- 2.3.2 Cycle Types
- 2.3.3 Dirty Bit RAM
- 2.3.4 Tag RAM Bus Interface
  - 2.3.4.1 Even Tag RAM to IU
  - 2.3.4.2 Odd Tag RAM to IU
  - 2.3.4.3 Even Tag RAM to Even Data RAM's
  - 2.3.4.4 Odd Tag RAM to Odd Data RAM's
  - 2.3.4.5 Cache Controller
  - 2.3.4.6 Clock Interface
- 2.3.5 Tag RAM Package Pinout

2.4 STREAMING CACHE DATA RAM's

- 2.4.1 Data RAM Architecture
- 2.4.2 Streaming Cache System Architecture
- 2.4.3 Streaming Cache Bus Interface
  - 2.4.3.1 Even Data RAM to IU
  - 2.4.3.2 Odd Data RAM to IU
  - 2.4.3.3 Even Data RAM to Even Tag RAM
  - 2.4.3.4 Odd Data RAM to Odd Tag RAM
  - 2.4.3.5 Even Data RAM to FPU
  - 2.4.3.6 Odd Data RAM to FPU
- 2.4.4 Data RAM Package Pinout

3.0 RESPONSIBILITIES OF THE CACHE CONTROLLER

3.1 Overview of the Cache Controller

- 3.1.1 Streaming Cache Data Management
- 3.1.2 Tag RAM Management
- 3.1.3 TBus Interface
  - 3.1.3.1 IU to CC TBus Protocol
  - 3.1.3.2 CC to IU TBus Protocol

1.0 INTRODUCTION TO THE TFP MICROPROCESSOR CHIP SET

There are 7 basic parts to the system. Each is explained in more detail in the following pages:

1) Integer Unit
2) Floating Point Unit
3) Tag RAM (even addresses)
4) Tag RAM (odd addresses)
5) Streaming Cache SRAM (Even data)
6) Streaming Cache SRAM (Odd data)
7) Cache Controller

Both the Integer and Floating Point units have multiple execution units, allowing execution of 4 instuctions per clock; two load/store instructions and two register to register or floating point execute instructions. Separate integer and floating point units maximize floating point throughput and allow for simultaneous execution of integer and floating point instructions. The floating point unit contains two pipelines, each of which can perform a double precision multiply-add every cycle. Two identical tag RAM's store address information for the even and odd data banks of the interleaved streaming cache. The integer and floating point units interface only to the streaming cache. Updates to the tag RAM's as well as all transactions requiring interface to main system memory are handled by the cache controller. Separate load and store data busses on the floating point unit eliminate bus turnaround time and allow both loads and stores to second level cache to execute simultaneously. Multiple tag RAM's provide an interleaved caching scheme, allowing access times to the cache to be hidden and providing two 64 bit operands to the FPU every clock.. A separate dirty bit RAM within each Tag RAM allows for updating of the dirty bit status for one cycle at the same time as a Tag RAM access for another cycle.

A set of compound multiply-add instructions has been added, taking advantage of the fact that the majority of floating point computations use the chained multiply-add paradigm. The operator for the multiply-add instructions is not defined by the IEEE and does not perform intermediate rounding. Eliminating the intermediate rounding step allows for a lower inherent latency and has higher precision and higher performance than an operator which performs intermediate rounding.

A register + register addressing mode for floating point loads and stores has been added which eliminates the extra integer add required in many array accesses. Register + register addressing for integer memory operations is not supported.

A set of four conditional move operators allows IF statements to be represented without branches. 'THEN' and 'ELSE' clauses are computed unconditionally and the results placed in a temporary register. Conditional move operators then transfer the temporary results to their true register.

To allow the compiler flexibility in scheduling conditional moves and their related comparisons, the Condition Code register has been expanded to 8 bits.

Figure 1 shows a block diagram of the TFP Microprocessor Chip Set. The diagram is divided into two sections seperated by the thick dotted line. All components below the line constitute the TFP Microprocessor Chip Set. Those components above the line, the Cache Controller and the Data Buffer Chips, are not part of the TFP Microprocessor Chip Set. Consequently the Cache Controller is discussed only in terms of functionality required. Package characteristics and external pinout information are not discussed in this document. Refer to section 2.5 for more information on the responsibilities of the Cache Controller.

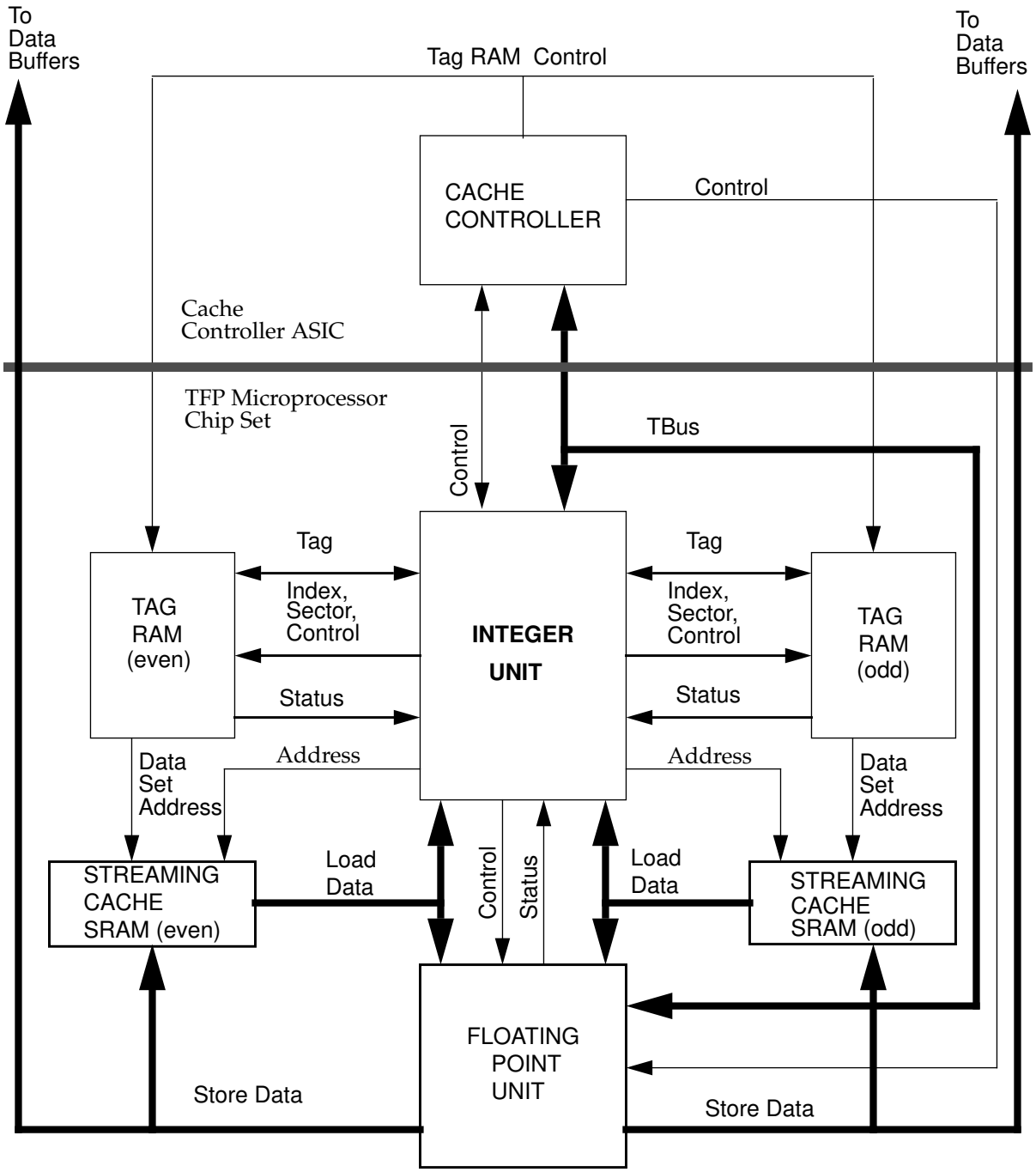

Figure 1: TFP Microprocessor Chip Set Block Diagram

2.0 TFP FUNCTIONAL UNITS

This section provides a more indepth look at the seven component parts of the TFP Microprocessor Chip Set. To provide the user with maximum flexibility in system design, the cache controller was not included as part of the integer unit. The Cache Controller is also not offered as part of the chip set and is discussed only in terms of functionality required.

2.1 INTEGER UNIT.

The Integer Unit chip (IU) is a 591 pin device and is the main computing component of the system. The high pin count is a result of the numerous dedicated busses provided by the IU. This dedicated bussing scheme helps to take full advantage of the multiple execution units within the IU, allowing each unit to run independently of the other, alleviating not only the need for multiplexing data and address, but also allowing loads and stores to the streaming cache to occur simultaneously.

The IU contains four caches and has dedicated interfaces to all components in the system. The IU performs address generation and provides address information for interfacing to the streaming cache via separate and dedicated address busses for the even and odd banks. Instruction and data interface to the Floating Point Unit (FPU) is via a dedicated 80 bit TBus, and addresses to the tag RAM's are provided via separate and dedicated tag, index, and sector busses for both the even and odd tag RAM's.

The IU contains two arithmetic logic units (ALU) as well as two address generation units, yielding a maximum of 4 instructions per cycle. The on-chip 16 KByte Data cache is dual ported and contains separate address and 64 bit data busses for each port. This allows multiple accesses to the cache to occur simultaneously. The 16 KByte instruction cache is 128 bits wide and single ported. Both caches are virtually indexed. The instruction cache is virtually tagged, alleviating the need for address translation on I-cache accesses. The data cache is physically tagged to maintain coherency with second level cache. Each 32 byte line in the I-cache contains a specific address space identifier (ASID). This value is assigned by the operating system and is process specific. There are at least two specific ASID values per process, one for the instruction cache, one for the TLB. The ASID helps to differentiate between multiple processes within the same cache and helps to reduce I-cache flushing by allowing the operating system to invalidate only those lines whose process is no longer valid. The operating system can also flush the I-cache when all 256 ASID values have been used.

In addition to the data and instruction caches, the IU also contains Branch and Translation Lookaside Buffer (TLB) caches. The Branch cache is accessed along with the instruction cache and is used to predict and modify the program counter on branch or jump instructions. The Branch Cache is a 15 bit field concatenated to each aligned 128 bit quadword of the I-cache. The Branch Cache implements a simple branch prediction mechanism which branches depending on the state of the predict bit associated with each Branch Cache entry.

The TLB cache is used to convert virtual addresses to physical addresses. A single TLB services both the data and instruction caches. The instruction cache only requires address translation on a miss. Similar to the instruction cache, each entry of the TLB also contains an ASID. However, this value is different from that contained in the I-cache. Having separate ASID values for each cache allows separate flushing of the Instruction and TLB caches. Below is a list of features of the four caches.

- Instruction cache;
  - 16 KBytes
  - Virtually indexed
  - Virtually tagged
  - Direct mapped, no hashing
  - Single ported, 128 bit path
  - Fetches 4 instructions (128 bits) per cycle
  -32 Byte line size
  - No parity
  - 11 cycle miss penalty to streaming cache
  - No coherency maintained with streaming cache
  - Alignment on 128-bit boundaries
  - Separate ASID values for I-cache tags
- Data Cache;
  - 16 KBytes
  - Virtually indexed
  - Physically tagged
  - Direct mapped, no hashing
  - Dual ported, 64 bit data paths
  - Two loads or one load and one store per cycle
  - 32 Byte line size
  - No parity
  - 8 cycle miss penalty to streaming cache
  - Coherency maintained with the streaming cache
  - Write through with allocate protocol
  - Separate ASID for D-cache tags
- Branch Cache;
  - 1K Entries, one entry per 4 instr.
  - Virtually indexed in parallel with Instruction cache
  - Direct mapped, no hashing
  - 3 cycle miss penalty
- TLB Cache;
  - Dual ported, 2 translations/clock

- 3-way set associative,
- 384 entries total (128 X 3 way)
- Implements random replacement algorithm
- Supports 4K,8K,16K,64K,1M,4M,16M page sizes
- Maps one virtual to one physical page Indexed by low-order 7 bits of virtual address
- Index is hashed by Exclusive-OR of low order 7 bits of TLB cache ASID.
- Software Refilled

Figure 2 shows a block diagram of the integer unit.

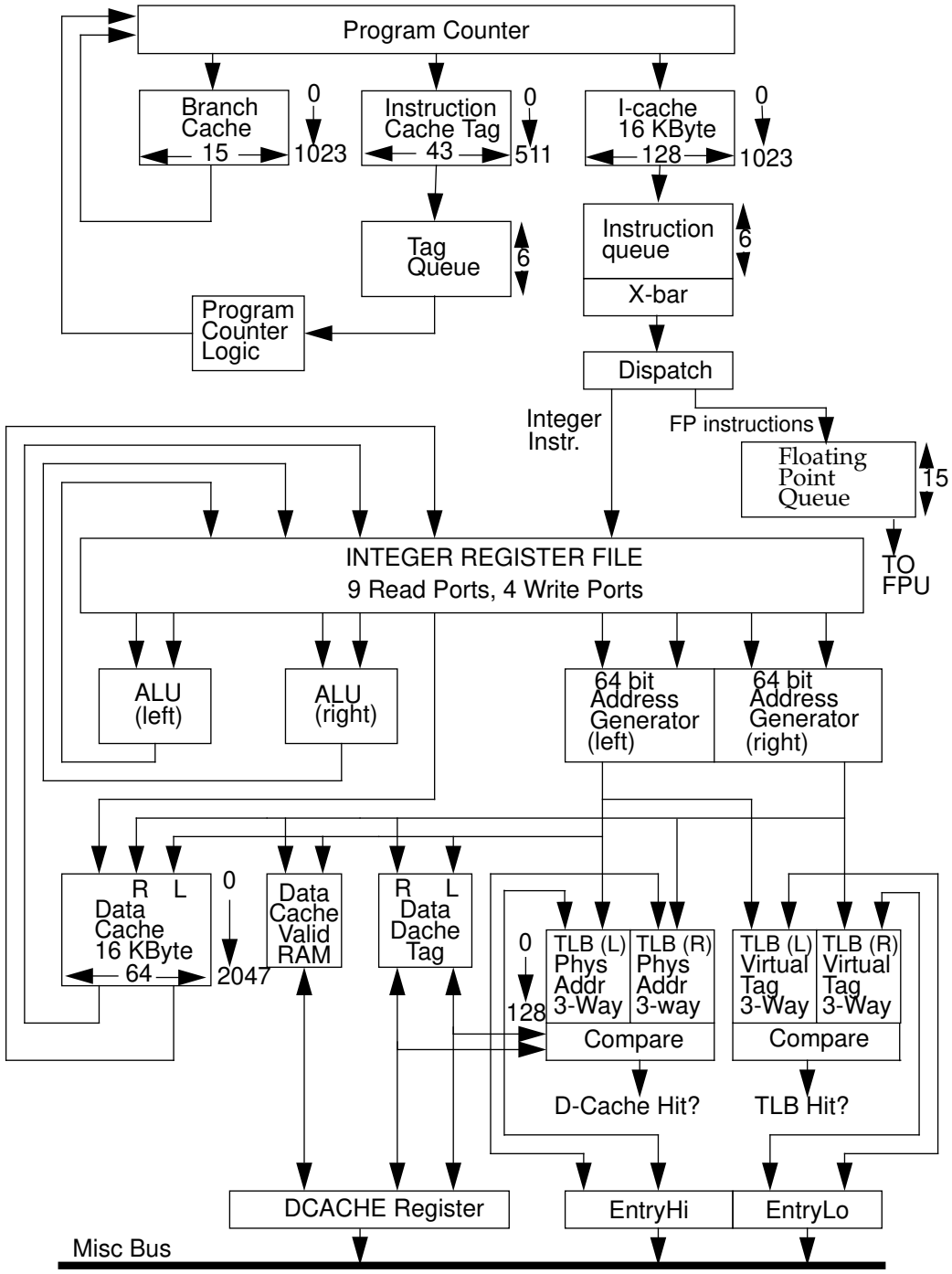

Figure 2: Integer Unit Block Diagram

2.1.1 INSTRUCTION CACHE

The instruction cache is 16 KBytes in size and is a direct-mapped architecture arranged as 1024 entries by 128 bits. Each entry contains four 32 bit instructions. Each access to the I-cache results in the fetching of four instructions. The I-cache is virtually indexed and virtually tagged, hence coherency between the I-cache and the physically tagged Data Cache is not maintined. Figure 3 shows the organization of the instruction cache.

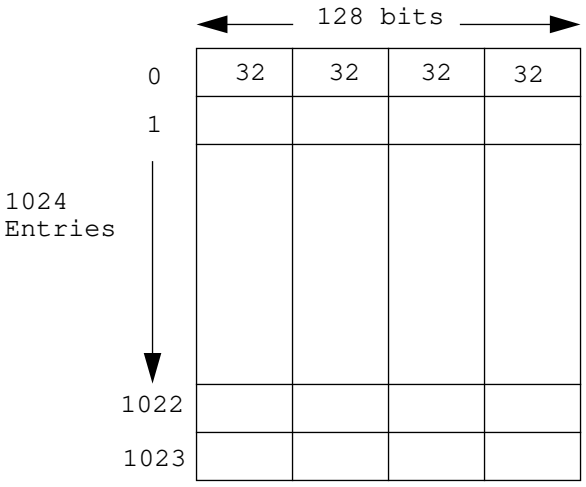

Figure 3: Instruction Cache Organization

2.1.2 INSTRUCTION CACHE TAG RAM

The I-cache line size is 32 bytes . There is one tag entry per I-cache line. Each entry in the I-cache tag RAM consists of a 34 bit tag, an 8 bit address space identifier (ASID), and a tag valid bit. The ASID differentiates between processes and allows tag addresses from different processes to reside in the I-cache tag at the same time.

The valid bit in the I-cache tag RAM defines whether the tag information is valid. If the address tag compare is valid but the valid bit for that entry is not set, an I-cache miss occurs. The valid bit is used on I-cache misses to validate I-cache entries as the instructions are written into the I-cache. The Region bits contain the values of bits 63:62 of the address bus and define one of the four addressing mode the procesor is operating in. Figure 4 shows a block diagram of the instruction cache tag RAM.

2.1.3 BRANCH CACHE

The Branch Cache works in conjunction with the instruction cache and implements a simple branch prediction mechanism. The Branch has 1024 entries, one entry for each 128 bit quadword in the instruction cache. The total width of the Branch cache is 15 bits, 10 of which are used to allow branching to any of the 1024 entries in the instruction cache.

The branch cache does not detect the presence of branches in a given I-cache line. Each branch cache entry has a predict bit associated with it. If the predict bit is on, the corresponding 10 bit branch target address is jammed into the program counter. The I-cache vectors to a new index location and resumes the fetching of instructions. Whether or not the branch prediction was correct is not known until 3 cycles later when the branch instruction is actually executed. If the prediction is correct, all of the I-cache entries in the pipeline following the branch instruction will be correct and no cycles are lost. If the prediction was incorrect, the pipeline is flushed. The new branch target address is calculated and three cycles later instruction execution resumes.

The two bit DELAY field associated with each branch cache entry encodes which of the four instructions in the I-cache entry contained the branch instruction.

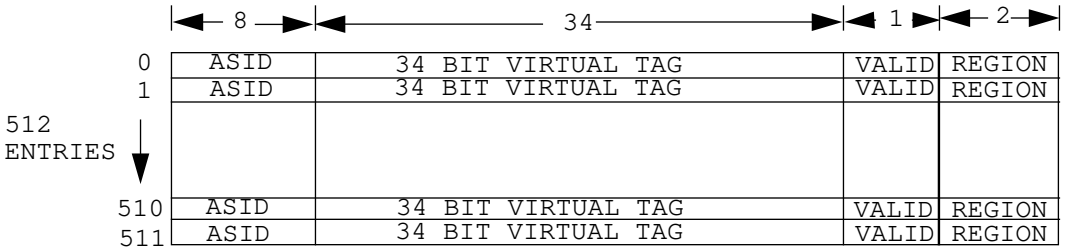

Figure 4: Instruction Cache Tag RAM

Since the compiler is not required to begin instruction execution on a 128 bit boundary, the two bit TARGET32 field is encoded to indicate where in the quadword the instruction execution began.There is one TARGET32 field per branch cache entry. Figure 5 shows a diagram of the branch cache organization.

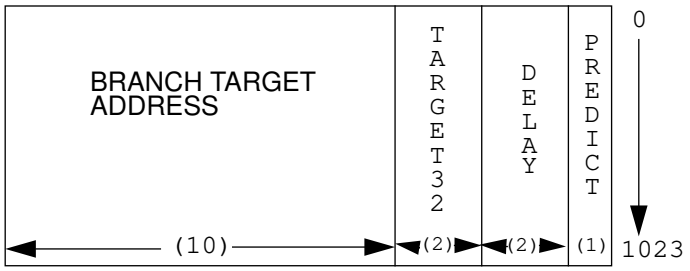

Figure 5: Branch Cache

2.1.4 INSTRUCTION QUEUE

Both the I-cache and the I-cache tag RAM have queues which act as temporary storage for instructions waiting to be executed. Each queue is six-deep. When instructions are fetched from the I-cache they undergo predecoding before being placed in the queue. The purpose of predecoding is to reduce instruction processing time in the decode stage of the pipeline. Eighteen additional characterization bits are added to each original 32 bit instruction.These bits are used for two cycles, afterwhich most of the bits are no longer needed. By the time the instructions reach the floating point queue they are 37 bits wide as five bits of the original 18 additional bits will be used by the FPU. Predecoding accomplishes three things:

1) Consistent alignment of the 5-bit destination field.
2) Instruction Category encoding.
3) Addition of timing critical bits.

Figure 6 shows a block diagram of the instruction and tag queues.

2.1.5 X-BAR

The X-BAR determines which of the four instructions to send depending on the resources available from cycle to cycle. This process is called Resource Modeling. The idea behind resource modeling is that instructions are not dispatched until there is sufficient resources available for them to complete. The X-bar monitors the status of each execution unit as well as determines interdependencies between any of the four instructions in a given line.

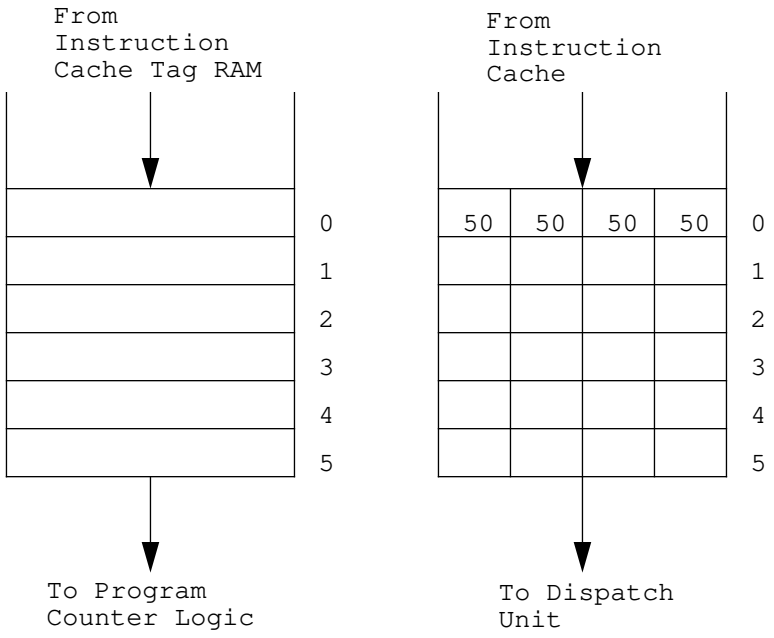

Figure 6: Instruction and Instruction Tag Queues

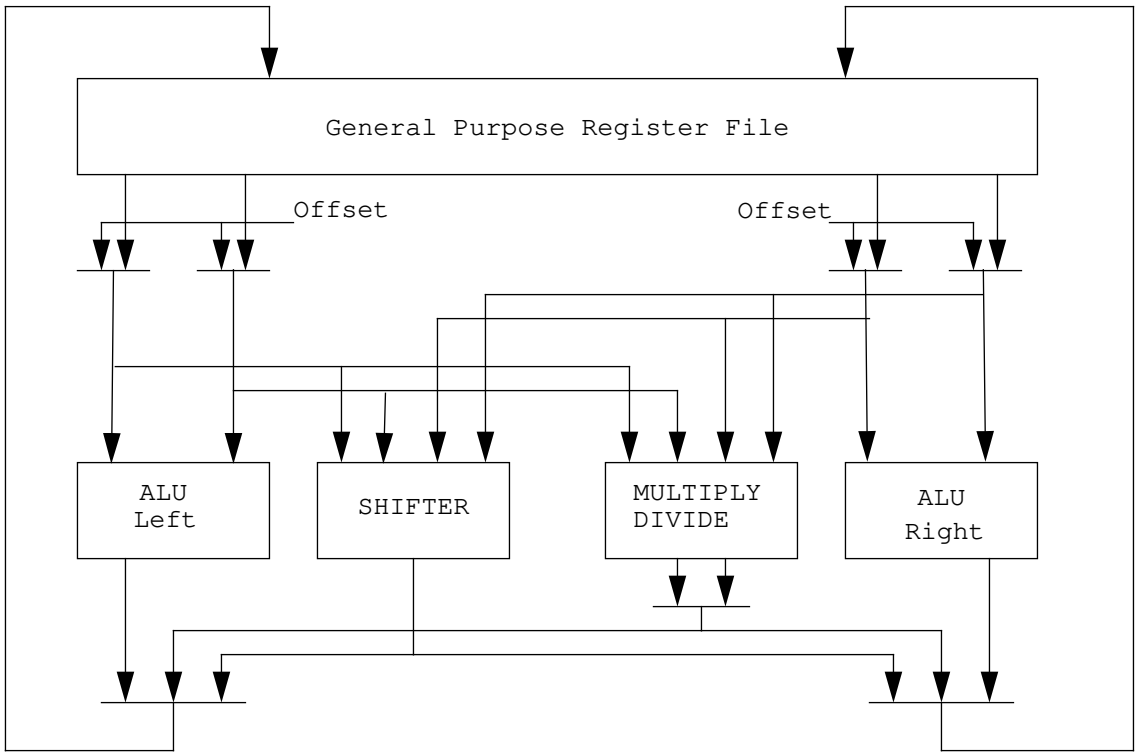

Figure 7: Arithmetic Logic Units

2.1.6 INTEGER REGISTER FILE

The integer register file in figure 1 consists of 9 read ports and 4 write ports. Four of the read ports are dedicated to ALU usage. Four others are used for address generation on integer or floating point memory operations. The ninth read port is used for data cache stores and to implement certain load alignment instructions. On D-cache stores the existence of the ninth port reduces the amount of sharing that would normally be required by those port(s) normally dedicated to address generation on memory operations.

2.1.7 ARITHMETIC LOGIC UNIT

The Arithmetic Logic Unit (ALU) consists of two 64 bit arithmetic units, one shifter, and one Multiply/Divide unit. The results of the operations are written back to the register file. All arithmetic operations occur in the execution stage (E-stage). Figure 7 above shows a block diagram of the ALU.

2.1.8 TRANSLATION LOOKASIDE BUFFER

The Translation Lookaside Buffer (TLB) is dual ported and is physically split into two halves. Each half contains 128 entries and is 3-way set associative, yielding a total of 384 entries each. One half contains the virtual tags (VTAGS), the other the actual physical address (PA) corresponsing to each virtual tag .

TLB, Data Cache, and Data Cache Tag RAM lookups are performed in the execution stage (E-stage) of the pipeline. The VTAG portion of the TLB is used to determine whether a certain range of addresses resides in the PA portion. If it is determined that the translation for the virtual address resides in the TLB, the contents of the PA portion is compared to that in the D-cache tag RAM, resulting in either a hit or a miss to the D-cache. Either a TLB or a D-cache miss initiates an external memory cycle. Figure 8 shows a block diagram of the Translation Lookaside Buffer.

2.1.9 DATA CACHE

The data cache is 16 KBytes organized as 2048 entries X 64 bits and is dual-ported. Line size is 32 bytes. Two separate address and data busses allow either two loads, or one store and one load to occur simultaneously. The D-cache is

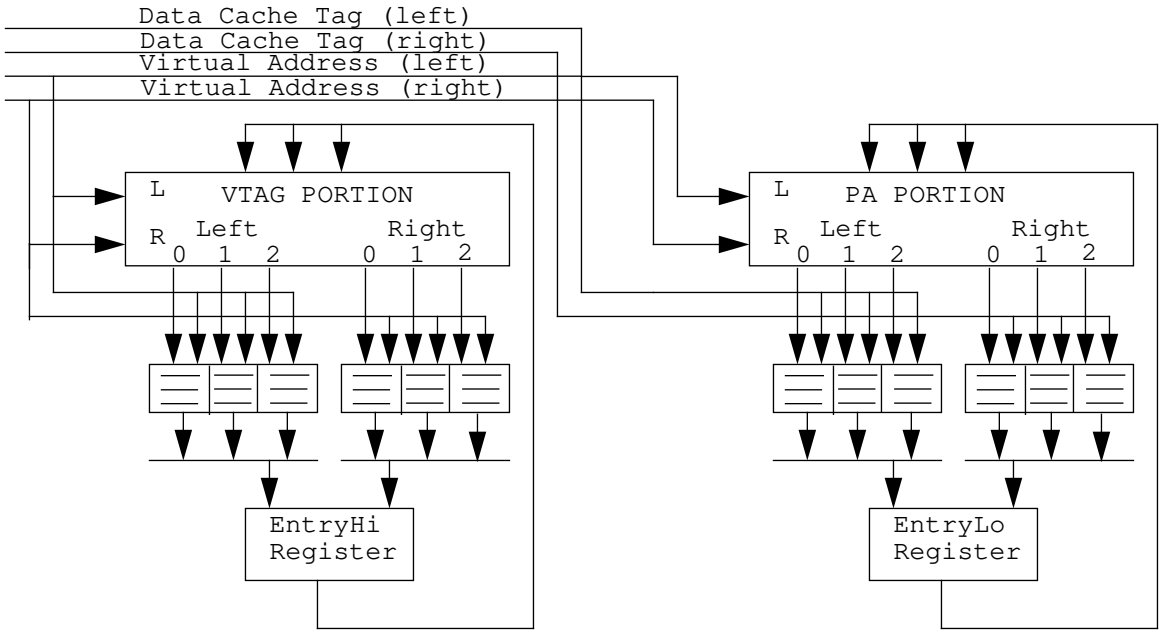

Figure 8: Translation Lookaside Buffer

virtually indexed and physically tagged and implements a write-through protocol. All writes to the D-cache are also written out to second level cache. Since only one store cycle can occur at a time, the D-cache has one store aligner in the write path between the register file and the D-cache. There are two load aligners in the read path. These aligners assure that, if the requested data is less than 64 bits, that data, regardless of where it is in the 64 bit doubleword, will be shifted to the proper place within the doubleword before being written to the register file. The store aligner performs the opposite function by shifting the requested data of the register to the desired location in the 64 bit doubleword before writing it to the D-cache.

One Byte is the smallest writeable quantity to the D-cache, meaning that the IU can write any or all bytes within a given 64 bit doubleword. Read cycles have the same resolution. Although the D-cache write size is variable, the streaming cache is not and requires a minimum of 32 bits on every write cycle. Therefore even though the IU may write less than 64 bits to the D-cache, it must assure that a minimum of 32 bits of the given doubleword are placed on the bus when the write to the streaming cache is initiated.

2.1.10 DATA CACHE TAG RAM

The data cache tag RAM contains 512 entries, one entry per 32 byte D-cache line. Each tag entry is 28 bits wide.

When the RAM is indexed, the corresponding physical tag is compared to the translated address from the TLB. On loads the tag RAM is accessed at the same time as the D-cache. There is no time lost waiting for the result of the TLB lookup. By the time the status of the TLB lookup is determined the D-cache has been indexed and the data read out. If it is determined that the requested address is in the cache, the data is written to the register file in the same clock and the cycle is completed. If the requested address is not in the cache, the buffers which allow the data to write the register file are simply not enabled. A D-cache miss occurs and a memory cycle to second level cache is initiated. Simultaneous access of the D-cache and the tag RAM allows for single clock reads when there is a D-cache hit.

D-cache stores are done the same way as loads. The TLB lookup and the accompanying store to the D-cache occur in the same clock. If either the virtual or physical address compare did not match, resulting in either a TLB or D-Cache miss, the store location in the D-cache is invalidated in the next clock by turning off the valid bit in the D-cache Valid RAM location corresponding to the store.

2.1.11 DATA CACHE VALID RAM

The data cache Valid RAM is 4 bits wide and contains 1024 entries. Each entry represents a 32 bit value, hence there are two valid bits per 64 bit doubleword. Each index to the

valid RAM corresponds to the status of two 64 bit Data cache words. There are two reasons for a separate valid RAM with individual bits:

The first reason is to alleviate invalidating entire lines of the cache when floating point data is found. Sometimes integer and floating point data reside in the same data cache (D-cache) line. Floating point loads and stores interface directly to the streaming cache and do not usually affect the contents of the IU data cache. However, the data cache of the IU must be kept coherent with the streaming cache. Therefore, if a FP store is done to a given location in the streaming cache which also resides in the D-cache, the D-cache entry must be invalidated. By having individual valid bits for each 32 bit word in the data cache, the mixing of floating point and integer data in a given D-cache line is better accomodated. This way if an integer load is done to that same location a D-cache miss occurs, forcing the IU to fetch the data from the streaming cache.

The second reason for having a separate valid RAM is to be able to easily invalidate the data for integer stores which miss in the D-cache. In the IU data is stored to the D-cache in the same cycle that the TLB and D-cache hit/miss status is determined. This is done so that the store data does not have to wait for the result of the lookup before it is written to the D-cache. If a store miss occurs the data is invalidated in the following cycle by turning the valid bit off for that D-cache entry.

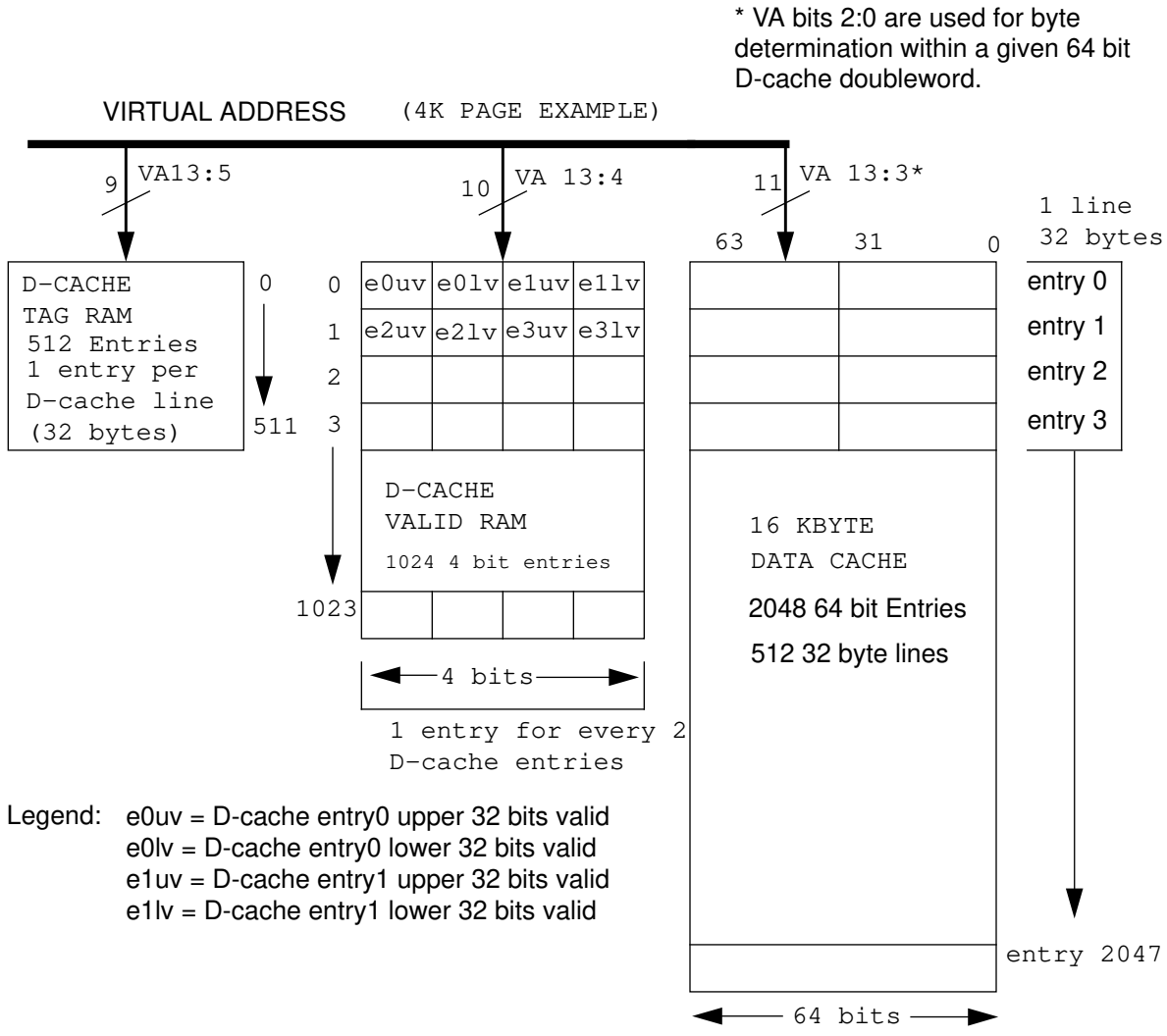

Figure 9: Data Cache, Data Cache Tag, Data Cache Valid RAM Organizaation

Valid bits are checked on both loads and stores. Because the D-cache contains only data it does not know whether a given location contains integer or FP data, only whether the entry is valid or not valid.

Figure 9 above shows the organization of the data cache.

2.1.12 INTEGER UNIT BUS INTERFACE

The IU is a 591 pin device and interfaces to all other components in the chip set. The following sections define the external pinout of the IU and are divided into specific component interfaces.

2.1.12.1 INTEGER UNIT TO FLOATING POINT UNIT INTERFACE

TBUS<79:72> (Tbus) Active High Output

Part of the 80 bit TBus. These bits are output only and connect directly to the TBUS<79:72> pins of the Floating Point Unit and are used for transferreing data and operations to the FPU. When TBUS<79:78> equals '10', the TBus is able to transfer 2 memory specifiers and 2 floating point operations to the FPU. When TBUS<79:78> does not equal '10', the TBus is used for special operations such as integer stores and moves to and from the FPU. TBUS<79:72> is valid whenever the FVALID signal is active. Refer to sections 2.1.12.6 and 2.1.13 for definitions of the remaining TBus bits.

PMODE (Parity Mode) Active High Output

The PMODE pin determines whether odd or even parity checking and generation is to be performed by the FPU and is connected directly to the PMODE pin of the FPU. When enabled, parity is checked on the even and odd load data busses of the FPU and is generated for the even and odd store data busses of the FPU for data going to the streaming cache.

PERRE (Parity Error Even) Active High Input

This signal is generated by the FPU and is asserted when a parity error is detected on the load data and load parity busses of the even bank. PERRE is connected directly to the PERRE pin of the FPU.

PERRO (Parity Error Odd) Active High Input

This signal is generated by the FPU and is asserted when a parity error is detected on the load data and load parity busses of the odd bank. PERRO is connected directly to the PERRO pin of the FPU.

FVALID# (FPU Valid) Active Low Output

This signal active indicates that the TBus has a valid FPU operation during the current cycle. FVALID is connected directly to the FVALID pin of the FPU.

TBUSOE# (Tbus Output Enable) Active Low Output

This signal is used as a tri-state enable for the TBus<63:0> pin connections to the FPU. TBUSOE# is connected directly to the TBUSOE# pin of the FPU. Two cycles after TBUSOE# is asserted the IU receives valid input data on the TBus.

ENQLE# (Enqueue Load Even) Active Low Output

This signal indicates when an even 64 bit doubleword of data from the streaming cache should be written into the load data queue and the enqueue load pointer incremented. ENQLE# is connected directly to the ENQLE# pin of the FPU.

ENQLO# (Enqueue Load Odd) Active Low Output

This signal indicates when an odd 64 bit doubleword of data from the streaming cache should be written into the load data queue and the enqueue load pointer incremented. ENQLO# is connected directly to the ENQLO# pin of the FPU.

ENQLP# (Enqueue Load Priority) Active Low Output

This signal indicates the order of execution between two loads to be enqueued. ENQLP asserted indicates that the odd 64 bit doubleword should be enqueued before the even 64 bit doubleword. ENQLP# is connected directly to the ENQLP# pin of the FPU.

DEQSE# (Dequeue Store Even) Active Low Output

This signal indicates when an even 64 bit doubleword of data should be read from the store data queue and the Dequeue Even Pointer updated. DEQSE# is connected directly to the DEQSE# pin of the FPU.

DEQSO# (Dequeue Store Odd) Active Low Output

This signal indicates when an odd 64 bit doubleword of data should be read from the store data queue and the Dequeue Even Pointer updated. DEQSO# is connected directly to the DEQSO# pin of the FPU.

DEBUG# (Debug Mode) Active Low Output

This signal indicates whether the results are written into the Floating Point Register File when an exception occurrs. When DEBUG is active (0), a floating point operation that raises the enabled exception does not write its result to the FPR. For a disabled exception the FPR is updated with the appropriate substitution value.When DEBUG is inactive (1), a floating point operation that raises an exception writes its appropriate substitution value to the FPR regardless of whether the exception is enabled or disabled. DEBUG# is connected directly to the DEBUG# pin of the FPU.

FCCL (Floating Point Condition Code Left) Active High Input

This signal is the condition code generated by the left FPU execution unit. The IU keeps track of the floating point operation, the CC destination, and the pipeline stage. FCCL is connected directly to the FCCL pin of the FPU.

FCCR (Floating Point Condition Code Right) Active High Input

This signal is the condition code generated by the right FPU execution unit. The IU keeps track of the floating point operation, the CC destination, and the pipeline stage. FCCR is connected directly to the FCCR pin of the FPU.

FPINTR# (Floating Point Interrupt) Active Low Input

This signal indicates that a floating point exception has occurred on either execution unit and the enable bit of the FSR for that type of exception was active. FPINTR# is connected directly to the FPINTR# pin of the FPU.

2.1.12.2 INTEGER UNIT TO EVEN TAG RAM

INDEXE<16:0> (Index Even) Active High Output

These bits form the index to the even tag RAM. This bus is output only. Connections to the Tag RAM are as follows:

INDEXE<1:0> Not Used. To be left unconnected.
INDEXE<3:2> connect to SECTOR<1:0> pins of the even Tag RAM.
INDEXE<13:4> connect to INDEX<9:0> of the even Tag RAM.
INDEXE<16:14> connect to INDEX<12:10> of the even Tag RAM through hardware jumpers.

TAGE<21:0> (Tag Address Even) Active High I/O

The Tag RAM address bus is bidirectional. The address is driven by the IU on lookup cycles, and driven by the CC on writes through the TBus. On read cycles the tag RAM drives the address information to the CC through the TBus as requested. Connections to the Tag RAM are as follows:

TAGE<0> Not Used. To be left unconnected.

TAGE<1> connects to TAG<17> of the even Tag RAM through a hardware jumper.
TAGE<3:2> connect to TAG<19:18> pins of the even Tag RAM.
TAGE<20:4> connect to TAG<16:0> of the even Tag RAM.
TAGE<21> connects to TAG<17> of the even Tag RAM through a hardware jumper.

ESAE<1:0> (External Set Addr. Even) Active High Output

When a tag RAM lookup is performed, logic inside the tag RAM determines which of the four ways compared correctly with the tag address on the bus and encodes this information as a two bit value called the 'match set address'. Oftentimes the IU will perform the tag lookup many cycles before the corresponding write to the data RAM's is done. When this occurrs there must be a mechanism to store the 2 bit set address value for use during the actual write. The lookup is performed ahead of time and the result is stored to the IU via the match set address (MCHSA) pins of the tag RAM. When the write is done this information is driven back through the tag RAM via the ESAE pins and becomes the upper two bits of the even data RAM address of the second level cache. ESAE<1:0> is connected directly to the ESA<1:0> pins of the even tag RAM.

ESASELE (External Set Address Select Even) Active High Output

This pin controls a multiplexor inside the tag RAM which allows the data set address for the data RAM's to be driven either directly from the tag RAM compare result logic or from the ESAE pins. If ESASELE is asserted the ESAE<1:0> values are driven out to the data RAM's. This pin is normally only asserted on write cycles when the tag RAM lookup is performed before the corresponding write. ESASELE is connected directly to the ESASEL pin of the tag RAM.

DBSETE# (Dirty Bit Set Even) Active Low Output

This pin is the dirty bit RAM write enable and is connected to the DBSET pin of the even tag RAM. The tag RAM contains a separate on-chip 16 bit wide dirty bit RAM which contains the status of each of the sectors for each of the 4 ways of the tag RAM. The dirty bit RAM is accessed on lookup cycles regardless of the state of the line. If the access is a hit and DBSETE is active, the dirty bit is written one clock after the lookup is done.

MATCHE# (Match Even) Active Low Input

This signal is an outpuT by the even Tag RAM. When the IU performs a lookup cycle to the tag RAM an index is supplied which corresponds to all four sets of the four way set associative even tag RAM. Assertion of this signal by the tag RAM indicates that the address on the bus compared to one of the four sets. MATCHE# is connected directly to the MATCH# pin of the even tag RAM.

MCHSAE<1:0> (Match Set Addr. Even) Active High Input

These two bits are used on lookup cycles and encode which of the four ways in the 4-way set associative even Tag RAM the compare occurred. Oftentimes the lookup for a store is performed many cycles before the actual store data is available. MCHSAE<1:0> allows the IU to store the result of the compare until the store data is available. The information is then sent back to the Tag RAM and is used to construct the upper two bits of the Data RAM address. MCHSAE<1:0> connect directly to the MCHSA<1:0> pins of the even tag RAM.

MCHVSE<3:0> (Match Virtual Synonym Even) Active High Input

These signals represent virtual address bits <15:12>. This four bit virtual synonym value is stored in the even Tag RAM along with each 20 bit tag address. Because the data cache is virtually indexed and is larger than some of the programmable page sizes, multiple virtually indexed locations in the D-cache can have the same physical address. To insure that not more than one physical location is active at a time, a check of virtual address bits <15:12> is done at the same time as the address compare to assure that the same virtual address is being accessed. These four virtual synonym bits tell of the last location in the D-cache to be used. MCHVSE<3:0> connect directly to the MCHVS<3:0> pins of the even tag RAM.

MCHSTE<1:0> (Match State Even) Active High Input

This two bit value is driven by the even Tag RAM on lookup cycles and indicates the state information for one of four 128 byte sectors in the streaming cache corresponding to the address which compared. This two bit value encodes which state a given line in the G-cache is in; Shared, Invalid, or Exclusive. For more information on Tag RAM architecture, refer to section 2.3. MCHSTE<1:0> connect directly to the MCHST<1:0> pins of the even tag RAM.

2.1.12.3 INTEGER UNIT TO ODD TAG RAM

INDEXO<16:0> Index Odd) Active High Output

These bits form the index to the odd tag RAM. This bus is output only. Connections to the Tag RAM are as follows:

INDEXE<1:0> Not Used. To be left unconnected.
INDEXE<3:2> connect to SECTOR<1:0> pins of the odd Tag RAM.
INDEXE<13:4> connect to INDEX<9:0> of the odd Tag RAM.
INDEXE<16:14> connect to INDEX<12:10> of the odd Tag RAM through hardware jumpers.

TAGO<21:0> (Tag Address Odd) Active High I/O

The Tag RAM address bus is bidirectional. The address is driven by the IU on lookup cycles, and driven by the CC on writes through the TBus. On read cycles the tag RAM drives the address information through the TBus to the CC as requested. Connections to the Tag RAM are as follows:

TAGE<0> Not Used. To be left unconnected.
TAGE<1> connects to TAG<17> of the even Tag RAM through a hardware jumper.
TAGE<3:2> connect to TAG<19:18> pins of the even Tag RAM.
TAGE<20:4> connect to TAG<16:0> of the even Tag RAM.
TAGE<21> connects to TAG<17> of the even Tag RAM through a hardware jumper.

ESAO<1:0> (External Set Addr. Odd) Active High Output

When a tag RAM lookup is performed, logic inside the tag RAM determines which of the four ways compared correctly with the tag address on the bus and encodes this information as a two bit value called the 'match set address'. Oftentimes the IU will perform the tag lookup many cycles before the corresponding write to the data RAM's is done. When this occurrs there must be a mechanism to store the 2 bit set address value for use during the actual write. The lookup is performed ahead of time and the result is stored to the IU via the match set address (MCHSA) pins of the tag RAM. When the write is done this information is driven back through the tag RAM via the ESAO pins and becomes the upper two bits of the odd data RAM address. ESAO<1:0> are connected directly to the ESA<1:0> pins of the odd tag RAM.

ESASELO (External Set Address Select Odd) Active High Output

This pin controls a multiplexor inside the tag RAM which allows the data set address for the data RAM's to be driven either directly from the tag RAM compare result logic or from the ESAO pins. If ESASELO is asserted the ESAO values are driven out to the data RAM's. This pin is normally only asserted on write cycles when the tag RAM lookup is performed before the corresponding write. ESASELO is connected directly to the ESASEL pin of the odd tag RAM.

DBSETO# (Dirty Bit Set Odd) Active Low Output

This pin is the dirty bit RAM write enable and is connected to the DBSET pin of the odd tag RAM. The tag RAM chip contains a separate 16 bit wide dirty bit RAM which contains the status of each of the sectors for each of the 4 ways of the tag RAM. The dirty bit RAM is accessed on lookup cycles regardless of the state of the line. If the access is a hit and DBSETO# is active, the dirty bit is written one clock after the lookup is done. DBSETO# is connected directly to the DBSET# pin of the odd tag RAM.

MATCHO# (Match Odd) Active Low Input

This signal is an output from the odd Tag RAM. When the IU performs a lookup cycle to the tag RAM an index is supplied which corresponds to all four sets of the four way set associative odd tag RAM. Assertion of this signal by the tag RAM indicates that the address on the bus compared to one of the four sets. MATCHO# is connected directly to the MATCH# pin of the odd tag RAM.

MCHSAO<1:0> (Tag RAM Match Set Address Odd) Active High Input

These two bits are used on lookup cycles and encode which of the four ways in the 4-way set associative odd Tag RAM the compare occurred. Oftentimes the lookup for a store is performed many cycles before the actual store data is available. MCHSAO<1:0> allows the IU to store the result of the compare until the store data is available. The information is then sent back to the Tag RAM and is used to construct the upper two bits of the Data RAM address. MCHSAO<1:0> are connected directly to the MCHSA<1:0> pins of the odd tag RAM.

MCHVSO<3:0> (Match Virtual Synonym Odd) Active High Input

These signals represent virtual address bits <15:12>. This four bit virtual synonym value is stored in the odd Tag RAM along with each 20 bit tag address. Because the data cache is virtually indexed and is larger than some of the progaammable page sizes, multiple virtually indexed locations in the D-cache can have the same physical address. To insure that not more than one physical location is active at a time, a check of virtual address bits <15:12> is done at the same time as the address compare to assure that the same vitrual address is being accessed. These four virtual synonym bits tell of the last location in the D-cache to be used. MCHVSO<3:0> are connected directly to the MCHVS<3:0> pins of the odd tag RAM.

MCHSTO<1:0> (Tag RAM Match State Odd) Active High Input

This two bit value is driven by the odd Tag RAM on lookup cycles and indicates the state information for one of four 128 byte sectors in the streaming cache corresponding to the address which compared. This two bit value encodes which state a given line in the G-cache is in; Shared, Invalid, or Exclusive. For more information on Tag RAM architecture, refer to the tag RAM section. MCHSTO<1:0> are connected directly to the MCHST<1:0> pins of the odd tag RAM.

2.1.12.4 INTEGER UNIT TO EVEN DATA RAMS

LDE<63:0> (Load Data Even) Active High Input

Load data bus between the IU and the even bank of the streaming cache. This bus is unidirectional and accepts load input data only. Store data is transferred through the FPU via the TBus. Data transfers between the streaming cache and the IU are synchronous and conform to a 5 cycle pipeline. When a streaming cache load is initiated by the IU, data is returned in 5 clocks. Streaming cache misses take approximately 50 clocks. There are two 32 bit wide SIMmodules on the even bank. LDE<63:32> are connected directly to the LD<32:0> pins of the upper 32 bit even data RAM module. LDE<31:0> are connected directly to the LD<32:0> pins of the lower 32 bit even data RAM module.

WEE#<1:0> (Write Enable Even) Active Low Output

The streaming cache data RAM's can be written either by the IU or by the CC. Writes by the CC are handled through the FPU via the TBus. WEE#<1> is connnected to the WE# pin of the RAM's containing the upper 32 bits of data for the even bank. WEE#<0> is connnected to the WE# pin of the RAM's containing the lower 32 bits of data for the even bank. Normally the IU drives these pins during write hits to the streaming cache. The CC can also write the streaming cache. This is normally done on streaming cache misses where the requested data has to be fetched from main memory. In this case the CC is responsible for fetching the data from main memory and writing the data to the streaming cache through the TBus. Bit 63 of the TBus is passed through the IU onto WEE#<1> to write the upper 32 bits of the even bank. Bit 62 of the TBus is passed through the IU onto WEE#<0> to write the lower 32 bits of the even bank.

ADDRE<17:0> (Address Even) Active High Output

These pins supply the address to the second level cache. Bits 17:0 of the IU connect directly to address pins 17:0 of both SIMmodules of the even bank .

2.1.12.5 INTEGER UNIT TO ODD DATA RAMS

LDO<63:0> (Load Data Odd) Active High Input

Load data bus between the IU and the odd bank of cache SRAM. This bus is unidirectional and accepts load input data only. Store data is transferred through the FPU via the TBus. Data transfers between the streaminc cache and the IU are synchronous and conform to a 5 cycle pipeline. When a streaming cache load is initiated by the IU, data is returned in 5 clocks.. Streaming cache misses take approximately 50 clocks. There are two 32 bit wide SIMmodules on the odd bank. LDO<63:32> are connected directly to the LD<32:0> pins of the upper 32 bit odd data RAM module. LSO<31:0> are connected directly to the LD<32:0> pins of the lower 32 bit odd data RAM module.

WEO#<1:0> (Write Enable Odd) Active Low Output

The streaming cache data RAM's can be written either by the IU or by the CC. Writes by the CC are handled through the TBus. WEO#<1> is connnected to the WE# pin of the SIMM containing the upper 32 bits of data for the odd bank. WEO#<0> is connnected to the WE# pin of the SIMM containing the lower 32 bits of data for the odd bank. Normally the IU drives these pins during write hits to the streaming cache. The CC also can write the streaming cache. This is normally done on streaming cache misses where the requested data has to be fetched from main memory. In this case the CC is responsible for fetching the data from main memory and writing it to the streaming cache through the TBus. Bit 61 of the TBus is passed through the IU onto WEO#<1> to write the upper 32 bits of the odd bank. Bit 60 of the TBus is passed through the IU onto WEO#<0> to write the lower 32 bits of the odd bank.

ADDRO<17:0> (Address Odd) Active High Output

These pins supply the address to the second level cache. Bits 17:0 of the IU connect directly to address pins 17:0 of both SIMmodules of the odd bank .

2.1.12.6 INTEGER UNIT TO CACHE CONTROLLER

TBUS<71:64> (Tbus Cache to Controller) Active High I/O

These pins form dedicated interface between the IU and the CC and connect directly to TBUS<71:64> pins of the Cache Conrtoller. TBUS<71:68> are reserved and are currently not used. TBCC<67:64> form the Function field. The function field is the only field of the TBus by which the CC changes the internal state of the IU chip itself rather than just the tag RAM's or the Data RAM's. The 4 bit field translates to 16 possible functions that the CC can perform on the IU. TBUS<71:64> connect directly to the TBUS<71:64> pins of the Cache Controller.

VALIDOUT# (Valid Out) Active Low Output

This signal is driven by the IU to indicate that the information on the TBus is valid for the Cache Controller to receive. VALIDOUT# is connected directly to the VALID_OUT# pin of the Cache Controller.

IUREL# (Integer Unit Release) Active Low Output

This signal is driven by the IU to control whether the TBus and other resources are under control of the IU or the CC. When asserted it indicates that the IU is giving control of the TBus to the Cache Controller. IUREL# is connected directly to the IU_REL# pin of the Cache Controller.

UPDE# (Update Data Even) Active Low Output

This signal is driven by the IU and indicates that store data associated with either an update or a non-cacheable write will be on the even store data bus on the next clock. When the CC is in control of the TBus, the IU asserts this signal low if the even store address queue contains an address for which bits <17:7> match the Tag RAM index bits <17:7> which were on the TBus four cycles earlier. The signal remains deasserted if no such match is detected. UPDE# should always be considered together with UPDO#. If either is asserted, a compare hit has occurred. Address comparisons are done on a 128 byte minimum quantity

and take one cycle. Address range comparisons larger than 128 bytes require multiple cycles. UPDE# is connected directly to the UPDE# pin of the Cache Controller.

UPDO# (Update Data Odd) Active Low Output

This signal is driven either by the IU and indicates that store data associated with either an update or a non-cacheable write will be on the odd store data bus on the next clock.

When the CC is in control of the TBus, the IU assertes this signal low if the odd store address queue contains an address for which bits <17:7> match the Tag RAM index bits <17:7> which were on the TBus four cycles earlier. The signal remains deasserted if no such match is detected. UPDO# should always be considered together with UPDE#. If either is asserted, a compare hit occurred. Address comparisons are done on a 128 byte minimum quantity and take one cycle. Address range comparisons larger than 128 bytes require multiple cycles. UPDO# is connected directly to the UPDO# pin of the Cache Controller.

CCREQ# (Cache Controller Request) Active Low Input

This signal is driven by the CC to control whether the TBus and other resources are under control of the IU of the CC. When asserted, it indicates that the CC either requests control or does not wish to give up control of the TBus. CCREQ# is connected directly to the CC_REQ# pin of the Cache Controller.

2.1.13 THE TBUS

TBUS<63:0> (TBus Data) Active Low BiDirectional

This bidirectional bus goes between the CC, the IU, and the FPU. The function of each bit changes depending on which device is driving. Normally the CC drives the TBus when reading or writing the tag RAM's or reading the Data RAM's and for general communication with the IU. The FPU uses the TBus to transfer Move data from the floating point register file (FPR) to the Integer Register File of the IU as requested by the IU. The IU uses the TBus for integer stores to the data RAM's, general communication with the CC and the FPU, and FPU to IU move instructions. TBUS<63:0> connects directly to TBUS<63:0> of the FPU as well as TBUS<63:0> of the Cache Controller.TBus connection between the CC and the FPU is by virtue of the fact the the IU communicates with both. There is no TBus communication protocol between the FPU and the CC.

2.1.14 CLOCK INTERFACE

CLK (Reference Clock) Active High Input

Master input clock to the Phase Lock Loop (PLL) circuitry of the IU. The output of the PLL is then used as the master clock for the chip. CLK is normally connected directly to the ouput of the clock driver. In most cases it is desirable to use the PLL circuitry, but for those applications which to not wish to use the PLL, the clock drivers should be connected to EXT_CLK.

EXT_CLK (External Clock) Active High Input

The EXT_CLK input allows the system designer to bypass the internal PLL of the IU and drive the chip directly from the system clock. When not in use this pin whould be tied to ground.

SYNC_IN (Synchronized PLL input) Active High Input

Sync_in is part of the PLL feedback path and must be connected to Sync_Out in order for the PLL circuitry to work correctly. The pins are made avaliable externally to allow the user to manually alter the phase of the PLL.

SYNC_OUT (Synchronized PLL input) Active High Output

Sync_Out is part of the PLL feedback path and must be connected to Sync_In in order for the PLL circuitry to work correctly. The pins are made avaliable externally to allow the user to manually alter the phase of the PLL.

LPF_OUT (Low Pass Filter Output) Active High Output

LFP_OUT is a special pin used to test the PLL circuitry during component test for monitoring the status of the low-pass filter. For normal operation this pin can be left unconnected.

VCC_PLL (Voltage Phase Lock Loop)

Voltage source for the phase lock loop circuitry. Connected to a regulated 3.3 volt source. VCC_PLL can be connected to GND_PLL through .1 and .015 microfarad capacitors in parallel.

GND_PLL (Ground Phase Lock Loop)

Ground source for the phase lock loop circuitry. GND_PLL can be connected to VCC_PLL through .1 microfarad and .015 microfarad capacitors in parallel.

RESET

Reset pin of the Integer Unit.

2.1.15 INTEGER UNIT PACKAGE PINOUT

Figure 10 shows a package pinout of the Integer Unit. Table 1 shows the signal names relating to the package pinout. Active low signals are denoted with an # at the end. Pins denoted NC are not connected internally.

Figure 10: Integer Unit Package Pinout

Table 1: Integer Unit Pinout

Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal
A03.....Vcc	B26.....TBUS<46>	D4.....LDE<1>	E27.....Vss	G5.....LDO<3>	H28.....Vss
A05.....TBUS<0>	B28.....TBUS<50>	D6.....TBUS<3>	E29.....Vcc	G7.....LDO<2>	H30.....Vss
A07.....TBUS<6>	B30.....TBUS<54>	D8.....TBUS<7>	E31.....Vss	G9.....LDE<0>	H32.....Vss
A09.....TBUS<12>	B32.....TBUS<58>	D10.....TBUS<11>	E33.....TBUS<67>	G11.....Vcc	H34.....Vss
A11.....TBUS<16>	B34.....TBUS<62>	D12.....TBUS<15>	E35.....TBUS<71>	G13.....Vcc	H36.....NC
A13..... ..TBUS<20>	B36.....TBUS<66>	D14.....TBUS<21>	E37.....TBUS<75>	G15.....Vcc	H38.....Vss
A15.....TBUS<24>	B38.....TBUS<72>	D16.....TBUS<25>	E39.....TBUS<79>	G17.....Vcc	H40.....Vcc
A17... ....TBUS<28>	B40.....TBUS<78>	D18.....TBUS<29>	E41.....X_SDI	G19.....Vcc	H42.....JTAG_R
A19.....TBUS<32>	B42.....NC	D20.....TBUS<33>	E43.....SYNC_OUT	G21.....Vcc	H44.....JTAG_O
A21.....TBUS<36>	B44.....Vss	D22.....TBUS<37>	E45.....CLK	G23.....Vcc	J1.....LDE<5>
A23.....TBUS<40>	C1.....Vcc	D24.....TBUS<43>	F2.....LDE<34>	G25.....Vcc	J3.....LDE<36>
A25.....TBUS<44>	C3.....Vss	D26.....TBUS< 47>	F4.....LDO<34>	G27.....Vcc	J5.....LDO<5>
A27.....TBUS<48>	C5.....LDO<0>	D28.....TBUS<51>	F6.....LDO<33>	G29.....Vcc	J7.....LDO<4>
A29.....TBUS<52>	C7.....TBUS<4>	D30.....TBUS<55>	F8.....Vss	G31.....Vcc	J39.....VCC_PLL
A31.....TBUS<56>	C9.....TBUS<10>	D32.....TBUS<59>	F10.....Vcc	G33.....Vcc	J41.....JTAG_S
A33.....TBUS<60>	C11.....TBUS<17>	D34.....TBUS<65>	F12.....Vss	G35.....Vcc	J43.....FCCR
A35.....TBUS<64>	C13.....TBUS<19>	D36.....TBUS<69>	F14.....Vcc	G37.....FVALID#	J45.....ADDRE<17>
A37.....TBUS<68>	C15.....TBUS<23>	D38.....TBUS<73>	F16.....Vss	G39.....Vcc	K2.....LDE<37>
A39.....TBUS<74>	C17.....TBUS<27>	D40.....TBUS<77>	F18.....Vcc	G41.....SYNC_IN	K4.....LDO<38>
A41.....TBOE#	C19.....TBUS<31>	D42.....RESET#	F20.....Vss	G43.....JTAG_I	K6.....LDO<6>
A43.....Vcc	C21.....TBUS<35>	D44.....X_SDO	F22.....Vcc	G45.....JTAG_C	K8.....Vss
A45.....Vss	C23.....TBUS<39>	E1.....Vss	F24.....Vss	H2.....LDE<4>	K38.....GND_PLL
B2.....Vss	C25.....TBUS<45>	E3.....LDE<2>	F26..... Vcc	H4.....LDO<36>	K40.....Vss
B4.....LDO<32>	C27.....TBUS<49>	E5.....LDO<1>	F28.....Vss	H6.....LDO<35>	K42.....FCCL
B6.....TBUS<2>	C29.....TBUS<53>	E7.....TBUS<1>	F30.....Vcc	H8.....Vcc	K44.....ADDRE<16>
B8.....TBUS<8>	C31.....TBUS<57>	E9.....TBUS<5>	F32.....Vss	H10.....LDE<32>	L1.....LDE<38>
B10.....TBUS<14>	C33.....TBUS<61>	E11.....TBUS<9>	F34.....Vcc	H12.....Vss	L3.....LDE<6>
B12.....TBUS<18>	C35.....TBUS<63>	E13.....TBUS<13>	F36.....Vss	H14.....Vss	L5.....LDO<8>
B14.....TBUS<22>	C37.....TBUS<70>	E15.....Vss	F38.....Vcc	H16.....Vss	L7.....LDO<37>
B16.....TBUS<26>	C39.....TBUS<76>	E17.....Vss	F40.....Vss	H18.....Vss	L39.....Vcc
B18.....TBUS<30>	C41.....FPINTR#	E19.....Vcc	F42.....EXT_CLK	H20.....Vss	L41....ADDRO<17>.
B20.....TBUS<34>	C43.....Vss	E21.....Vss	F44.....LPF_OUT	H22.....Vss	L43.....ADDRO<16>
B22.....TBUS<38>	C45.....Vcc	E23.....TBUS<41>	G1.....LDE<35>	H24.....Vss	L45.....ADDRE<15>
B24.....TBUS<42>	D2.....LDE<33>	E25.....Vss	G3.....LDE<3>	H26.....Vss	M2.....LDE<7>

Table 1: Integer Unit Pinout

Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal
M4.....LDO<41>	T38.....Vss	Y44.....ADDRE<6>	AE5.....LDO<48>	AJ41.....Vcc	AP2.....LDE<55>
M6.....LDO<39>	T40.....Vcc	AA1.....LDE<14>	AE7.....Vss	AJ43.....ESAO<0>	AP4.....LDO<23>
M8.....Vcc	T42.....ADDRO<10>	AA3.....LDE<45>	AE39.....Vcc	AJ45.....ESASELE	AP6.....LDO<55>
M38.....Vss	T44.....ADDRE<10>	AA5.....LDO<14>	AE41.....Vss	AK2.....LDE<52>	AP8.....Vss
M40.....Vcc	U1.....LDE<11>	AA7.....Vss	AE43....ADDRO<0>	AK4.....LDO<50>	AP38.....Vss
M42....ADDRO<15>	U3.....LDE<42>	AA39.....Vcc	AE45....ADDRE<1>	AK6.....LDO<21>	AP40.....Vss
M44....ADDRE<14>	U5.....LDO<43>	AA41.....Vss	AF2.....LDE<49>	AK8.....Vss	AP42.....DEQSE#
N1.....LDE<8>	U7.....Vss	AA43...ADDRO<5>	AF4.....LDO<17>	AK38.....Vss	AP44.....ENQLE#
N3.....LDE<39>	U39.....Vcc	AA45....ADDRE<5>	AF6.....Vcc	AK40.....Vss	AR1.....LDE<24>
N5.....LDO<9>	U41.....Vss	AB2.....LDE<46>	AF8.....Vss	AK42.....ESAO<1>	AR3.....LDE<56>
N7.....LDO<7>	U43.....ADDRO<9>	AB4.....LDO<15>	AF38.....Vss	AK44.....ESAE<0>	AR5.....LDO<56>
N39.....Vcc	U45.....ADDRE<9>	AB6.....LDO<46>	AF40.....Vss	AL1.....LDE<21>	AR7.....LDO<57>
N41....ADDRO<14>	V2.....LDE<43>	AB8.....Vss	AF42.....WEO<1>	AL3.....LDE<53>	AR39.....Vcc
N43....ADDRO<13>	V4.....LDO<13>	AB38.....Vss	AF44....ADDRE<0>	AL5.....LDO<52>	AR41.....MCHSAO0
N45....ADDRE<13>	V6.....Vcc	AB40.....Vss	AG1.....LDE<18>	AL7.....LDO<54>	AR43.....CCREQ
P2.....LDE<40>	V8.....Vss	AB42....ADDRO<4>	AG3.....LDE<50>	AL39.....Vcc	AR45.....ENQLP#
P4.....LDO<11>	V38.....Vss	AB44....ADDRE<4>	AG5.....LDO<18>	AL41.....Vss	AT2.....LDE<25>
P6.....LDO<40>	V40.....Vss	AC1.....LDE<15>	AG7.....Vss	AL43.....IUREL#	AT4.....LDO<25>
P8.....Vss	V42.....ADDRO<8>	AC3.....LDE<47>	AG39.....Vcc	AL45.....ESAE<1>	AT6.....LDO<26>
P38.....Vss	V44.....ADDRE<8>	AC5.....LDO<47>	AG41.....Vss	AM2.....LDE<22>	AT8.....Vcc
P40.....Vss	W1.....LDE<44>	AC7.....Vss	AG43.....WEO<0>	AM4.....LDO<20>	AT38.....MCHSTO1
P42.....ADDRO<12>	W3.....LDE<12>	AC39.....Vcc	AG45.....WEE<1>	AM6.....LDO<22>	AT40.....Vcc
P44.....ADDRE<12>	W5.....LDO<44>	AC41....ADDRO<3>	AH2.....LDE<19>	AM8.....Vcc	AT42.....MCHSAO1
R1.....LDE<41>	W7.....Vss	AC43....ADDRO<2>	AH4.....LDO<49>	AM38.....Vss	AT44.....DEQSO#
R3.....LDE<9>	W39.....Vcc	AC45....ADDRE<3>	AH6.....LDO<51>	AM40.....Vcc	AU1.....LDE<57>
R5.....LDO<42>	W41.....Vcc	AD2.....LDE<16>	AH8.....Vcc	AM42.....UPDE#	AU3.....LDE<26>
R7.....Vss	W43.....ADDRO<7>	AD4.....LDO<16>	AH38.....Vss	AM44.VALIDOUT#	AU5.....LDO<58>
R39.....Vcc	W45.....ADDRE<7>	AD6.....Vcc	AH40.....Vcc	AN1.....LDE<54>	AU7.....LDO<59>
R41.....Vss	Y2.....LDE<13>	AD8.....Vcc	AH42.....ESASELO	AN3.....LDE<23>	AU39.....MCHVS0
R43.....ADDRO<11>	Y4.....LDO<45>	AD38.....Vss	AH44.....WEE<0>	AN5.....LDO<53>	AU41.....MATCHO#
R45.....ADDRE<11>	Y6.....Vcc	AD40.....Vcc	AJ1.....LDE<51>	AN7.....LDO<24>	AU43.....MCHSAE0
T2.....LDE<10>	Y8.....Vcc	AD42...ADDRO<1>	AJ3.....LDE<20>	AN39.....Vcc	AU45.....INTMODE
T4.....LDO<12>	Y38.....Vss	AD44...ADDRE<2>	AJ5.....LDO<19>	AN41.....ENQLO#	AV2.....LDE<58>
T6.....LDO<10>	Y40.....Vcc	AE1.....LDE<48>	AJ7.....Vss	AN43.....DEBUG#	AV4.....LDO<27>
T8.....Vcc	Y42.....ADDRO<6>	AE3.....LDE<17>	AJ39.....Vcc	AN45.....UPDO#	AV6.....LDO<28>

Table 1: Integer Unit Pinout

Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal
AV8.....Vss	AW33.....Vcc	BA13.....TAGE<17>	BB38...INDEXE<4>	BD18.....TAGO<9>	BE43.....Vcc
AV10.....LDE<31>	AW35.....Vcc	BA15.....Vss	BB40..INDEXE<2>	BD20.....TAGO<7>	BE45.....Vss
AV12.....Vss	AW37..INDEXE<0>	BA17.....Vcc	BB42....MCHVSE3	BD22.....TAGO<5>	
AV14.....Vss	AW39.....Vcc	BA19.....Vss	BB44....MCHVSE2	BD24.....TAGO<3>	
AV16.....Vss	AW41....MCHVSO1	BA21.....Vss	BC1.....Vcc	BD26.....TAGO<1>	
AV18.....Vss	AW43.....MCHSTE0	BA23.....TAGE<3>	BC3.....Vss	BD28INDEXO<16>	
AV20.....Vss	AW45.....MATCH#	BA25.....Vss	BC5.....LDO<31>	BD30INDEXO<14>	
AV22.....Vss	AY2.....LDE<28>	BA27.....Vcc	BC7.....PMODE	BD32INDEXO<12>	
AV24.....Vss	AY4.....LDE<29>	BA29.....Vss	BC9.....TAGO<19>	BD34INDEXO<10>	
AV26.....Vss	AY6.....LDO<61>	BA31.....Vss	BC11.....TAGE<15>	BD36..INDEXO<8>	
AV28.....Vss	AY8.....Vcc	BA33..INDEXE<7>	BC13.....TAGE<14>	BD38..INDEXO<5>	
AV30.....Vss	AY10.....Vss	BA35..INDEXE<5>	BC15.....TAGE<12>	BD40..INDEXO<2>	
AV32.....Vss	AY12.....Vcc	BA37..INDEXE<3>	BC17.....TAGE<10>	BD42.....DBSETO#	
AV34.....Vss	AY14.....Vss	BA39..INDEXE<1>	BC19.....TAGE<8>	BD44.....Vss	
AV36.....BDSETE#	AY16.....Vcc	BA41.....MCHVSO3	BC21.....TAGE<6>	BE1.....Vss	
AV38.....Vss	AY18.....Vss	BA43.....MCHVSE1	BC23.....TAGE<4>	BE3.....Vcc	
AV40.....Vss	AY20.....Vcc	BA45.....MCHVSE0	BC25.....TAGE<1>	BE5.....LDO<63>	
AV42.....MCHSTO0	AY22.....Vss	BB2.....LDE<30>	BC27.INDEXE<16>	BE7.....TAGO<21>	
AV44.....MCHSAE1	AY24.....Vcc	BB4.....LDE<62>	BC29.INDEXE<14>	BE9.....TAGO<18>	
AW1.....LDE<27>	AY26.....Vss	BB6.....PERRE	BC31.INDEXE<12>	BE11.....TAGO<16>	
AW3.....LDE<59>	AY28.....Vcc	BB8.....TAGE<20>	BC33.INDEXE<10>	BE13.....TAGO<14>	
AW5.....LDO<60>	AY30.....Vss	BB10.....TAGE<18>	BC35...INDEXE<9>	BE15....TAGO<12>	
AW7.....LDO<29>	AY32.....Vcc	BB12.....TAGE<16>	BC37...INDEXO<6>	BE17.....TAGO<10>	
AW9.....LDE<63>	AY34.....Vss	BB14.....TAGE<13>	BC39...INDEXO<3>	BE19.....TAGO<8>	
AW11.....Vcc	AY36.....Vcc	BB16.....TAGE<11>	BC41...INDEXO<0>	BE21.....TAGO<6>	
AW13.....Vcc	AY38.....Vss	BB18.....TAGE<9>	BC43.....Vss	BE23.....TAGO<4>	
AW15.....Vcc	AY40.....Vcc	BB20.....TAGE<7>	BC45.....Vcc	BE25.....TAG0<2>	
AW17.....Vcc	AY42.....MCHVSO2	BB22.....TAGE<5>	BD2.....Vss	BE27.....TAGO<0>	
AW19.....Vss	AY44.....MCHSTE1	BB24.....TAGE<2>	BD4.....LDO<62>	BE29.INDEXO<15>	
AW21.....Vcc	BA1.....LDE<60>	BB26.....TAGE<0>	BD6.....PERRO	BE31.INDEXO<13>	
AW23.....Vcc	BA3.....LDE<61>	BB28.INDEXE<15>	BD8.....TAGO<20>	BE33.INDEXO<11>	
AW25.....Vcc	BA5.....LDO<30>	BB30.INDEXE<13>	BD10....TAGO<17>	BE35..INDEXO<9>	
AW27.....Vcc	BA7.....NC	BB32.INDEXE<11>	BD12....TAGO<15>	BE37..INDEXO<7>	
AW29.....Vcc	BA9.....TAGE<21>	BB34...INDEXE<8>	BD14.....TAGO<13>	BE39...INDEXO<4>	
AW31.....Vcc	BA11.....TAGE<19>	BB36...INDEXE<6>	BD16.....TAGO<11>	BE41....INDEX0<1>	

2.2 FLOATING POINT UNIT

The Floating Point Unit (FPU) is a 591 pin device which performs all floating point functions for the TFP Microprocessor Chip Set. The FPU has two execution units, allowing two arithmetic and two Floating Point memory operations to be executed every clock. The Floating Point Register File contains 8 read ports and 4 write ports. Large load and store data queues, each 32 entries deep, allow for a pipelined interface between the IU and the FPU, streamlining the flow of data and minimizing wait time. With a target frequency of 75 MHz, the Floating Point Unit offers a peak performance of 300 MFLOPS.

The FPU has no on-chip cache and uses the streaming cache, which is the second level cache of the IU, as its memory. Dedicated load and store data busses to both the even and odd banks of streaming cache allow a read or write operation to be performed every clock. An 80 bit TBus interface forms the control bus for the FPU and allows the FPU to interface to both the IU and the Cache Controller (CC). Normally the FPU is controlled by the IU. Dispatching of instructions, floating point loads and stores to the streaming cache, integer stores to the streaming cache, etc. are all under control of the IU. Cycles which miss in the streaming cache and require interface to the main memory are handled by the Cache Controller. For these cycles the FPU is used only to transfer data from the load data bus to the store data bus.

Floating point instructions are received from the integer unit through the TBus. The instructions are executed and the result written back to the FP register file. Floating Point data is retrieved from the streaming cache on the load data pins and then placed in the Load Data Queue. For store operations data from the result is placed in the store data queue. As soon as the corresponding address information from the Tag RAM is made available, the data is written out to the streaming cache. In addition to floating point operations, the FPU is also used during integer stores to the streaming cache, handled by the IU, as well as stores to main memory, handled by the CC.

A set of compound multiply-add instructions has been added, taking advantage of the fact that the majority of floating point computations use the chained multiply-add paradigm. The operator for the multiply-add instructions is not defined by the IEEE and does not perform intermediate rounding. Eliminating the intermediate rounding step allows for a lower inherent latency and has higher precision and higher performance than an operator which performs intermediate rounding.

Figure 11 shows a block diagram of the floating point unit.

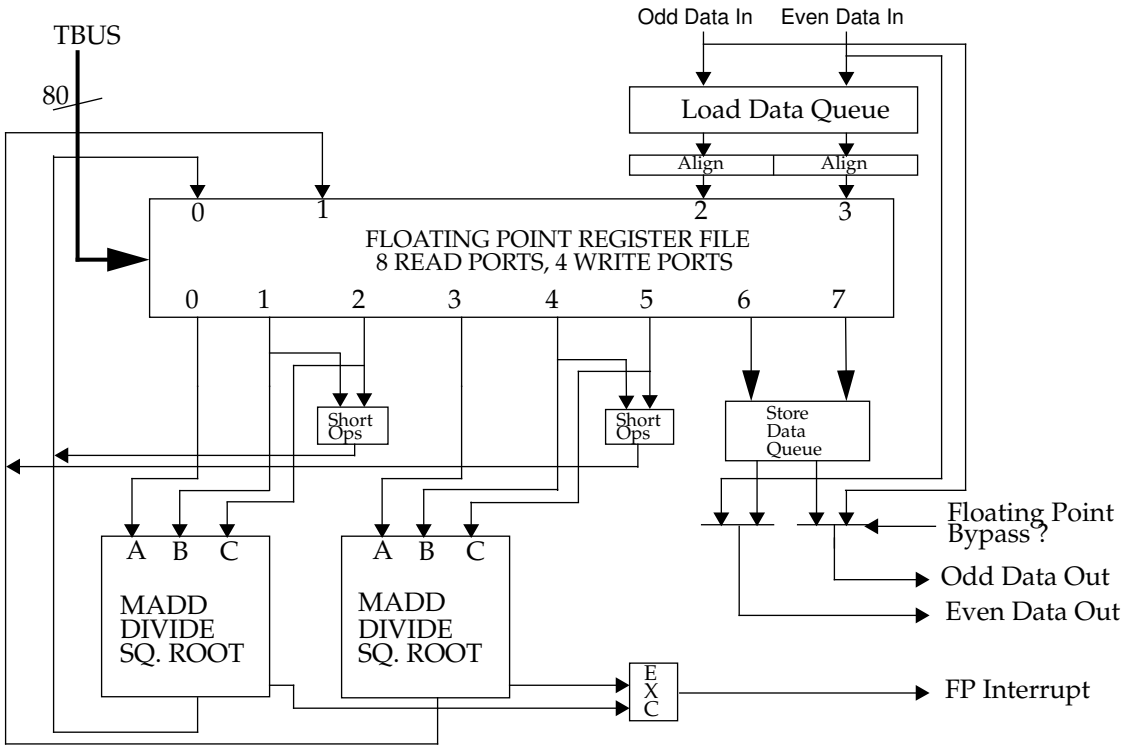

Figure 11: Floating Point Unit Block Diagram

2.2.1 FPU OPERATIONS

The FPU performs three basic types of arithmetic operations denoted as Short, Regular, and Long. Each operation has some number of clocks associated with its staging as well as latency. Staging defines how long the unit is busy. Latency defines the amount of delay clocks before the results are available. Each execution unit is fully pipelined and can begin a new operation on every clock.

Short operations have a one clock staging and one clock latency and include such move operations as MOV, MOVC, MOVZ, MOVN, NEG, ABS, and C. Except for Compares (C), short operations can be issued while long operations are executing. Compare instructions target the condition code field of the ControlStatus Register.

Regular operations have one cycle staging and four cycle latency. These include add, subtract, and multiply operations such as ADD, SUB, MUL, MADD, MSUB, CONVERT. Regular operations can require either 1, 2, or 3 source operands from the FP register file. These operands are read from the FPR. Each of the operations takes four cycles to complete. On the next clock the result is written back to the FPR. On-chip bypass logic allows the results of regular operations to be written to the FPR and at the same time back into the executions units. When the result of one operation is required for the operation immediately following, the bypass logic allows the result to be used in the next clock instead of having to write to the FPR and then read the information back out in the next clock.

Long opertaions have variable latency and staging times and include such operations as DIV, SQRT, RECIP, RSQRT. These times are shown in table 2 below.

Operation	Latency	Staging
DIV.S	14	11
DIV.D	20	17
SQRT.S	14	11
SQRT.D	23	20
RECIP.S	8	5
RECIP.D	14	11
RECIPSQRT.S	8	5
RECIPSQRT.D	17	14

Table 2: Long Operation Latency

2.2.2 TBUS INTERFACE

Instructions are dispatched by the IU to the FPU through the TBus. There are four basic types is transmissions which are differentiated by encoding the uppermost two bits of the TBus:

Normal - A normal dispatch contains two FP arithmetic operations, each 28 bits wide, and two FP memory operations, each 9 bits wide. There are roughly 30 FP arithmetic operations which can be dispatched by the IU. The 28 bit TBus format for arithmetic operations is different for each operation. For FP memory operations, the 9 bit value contains Floating Point Register destination as well as data alignment information. Each of the four potential instructions contains a valid bit associated with it. These four valid bits form the upper bits of the TBus. Setting this bit indicates to the FPU that a given instruction is valid and should be executed.

Move From - MoveFrom is similar in format to Normal mode except that the FP memory operation normally on TBus bits 73:64 is substituted with a move specifier. This operation moves data from a FP register to a general purpose register (GPR) in the IU and is the only time which the FPU drives the TBus.

IntStore - The IntStore operation supports integer stores to the streaming cache. As shown in figure 1, there is no direct path for integer stores from the IU to the streaming cache. Instead they are transmitted across the TBus and out onto the store data pins of the FPU. In IntStore mode the TBus contains the 64 bit integer data along with some store alignment information.

MoveTo - This operations moves data from a General Purpose Register (GPR) in the IU to a Floating Point Register. The MoveTo format is similar to the IntStore format except that instead of store alignment information, TBus bits 73:65 contain the FPR destination. The 64 data is transmitted on TBus pins 63:0.

Table 3 shows the TBus format for each of the four basic transmissions.

2.2.3 FLOATING POINT UNIT BUS INTERFACE

With the exception of the Tag RAM's, the FPU has dedicated interfaces to all other components in the system. The following sections define the external pinout of the FPU and are divided into specific component interfaces.

Table 3: TBUS Format

TBus Bits	7978	77	76	75	74	73 65	64	63 56	55 28	27 0
Normal	1 0	Vma	Vmb	Vfa	Vfb	MSpecA	MSpecB		FpOP-A	FpOP-B
MoveFrom	0 1	Vmf	Vmb	Vfa	Vfb	MfSpec	MSpecB		FpOP-A	FpOP-B
IntStore	0 0	1	----	---	---	IStSpec	-	Data		
MoveTo	1 1	1	----	---	---	MtSpec	-	Data		

2.2.3.1 FLOATING POINT UNIT TO INTEGER UNIT INTERFACE

The signals in this section comprise the interface between the Integer Unit and the Floating Point Unit. These signals are described further in IU section 2.1.12.1.

TBUS<79:72> (Tbus) Active High Input

Part of the 80 bit TBus. These bits are input only and connect directly to the TBUS<79:72> pins of the Integer Unit and are used for transferring data and operations to the FPU. TBUS<79:72> is valid whenever the FVALID# signal is active.

PMODE (Parity Mode) Active High Input

The PMODE pin determines whether even or odd parity checking and generation is to be performed by the FPU. PMODE is connected directly to the PMODE pin of the IU.

PERRE (Parity Error Even) Active High Output

This signal is generated by the FPU when PMODE is active and is asserted when a parity error is detected on the load data and load parity busses of the even bank. PERRE is connected directly to the PERRE pin of the IU.

PERRO (Parity Error Odd) Active High Output

This signal is generated by the FPU and is asserted when a parity error is detected on the load data and load parity busses of the odd bank. PERRO is connected directly to the PERRO pin of the IU.

FVALID# (FPU Valid) Active Low Input

This signal active indicates that the TBus has a valid FPU operation during the current cycle. FVALID# is connected directly to the FVALID# pin of the IU.

TBUSOE# (Tbus Output Enable) Active Low Input

This signal is used as a tri-state enable for the TBus<63:0> pin connections to the FPU. TBUSOE# is connected directly to the TBUSOE# pin of the IU.

ENQLE# (Enqueue Load Even) Active Low Input

This signal indicates when an even 64 bit doubleword of data from the streaming cache should be written into the load data queue and the enqueue load pointer incremented. ENQLE# is connected directly to the ENQLE# pin of the IU.

ENQLO# (Enqueue Load Odd) Active Low Input

This signal indicates when an odd 64 bit doubleword of data from the streaming cache should be written into the load data queue and the enqueue load pointer incremented. ENQLO# is connected directly to the ENQLO# pin of the IU.

ENQLP# (Enqueue Load Priority) Active Low Input

This signal indicates the order of execution between two loads to be enqueued. ENQLP# asserted indicates that the odd 64 bit doubleword should be enqueued before the even 64 bit doubleword. ENQLP# is connected directly to the ENQLP# pin of the IU.

DEQSE# (Dequeue Store Even) Active Low Input

This signal indicates when an even 64 bit doubleword of data should be read from the store data queue and the Dequeue Even Pointer updated. DEQSE# is connected directly to the DEQSE# pin of the IU.

DEQSO# (Dequeue Store Odd) Active Low Input

This signal indicates when an odd 64 bit doubleword of data should be read from the store data queue and the Dequeue Even Pointer updated. DEQSO# is connected directly to the DEQSO# pin of the IU.

DEBUG# (Debug Mode) Active Low Input

This signal indicates whether the results are written into the Floating Point Register File when an exception occurrs. When DEBUG is active (0), a floating point operation that raises the enabled exception does not write its result to the FPR. For a disabled exception the FPR is updated with the appropriate substitution value.When DEBUG is inactive (1), a floating point operation that raises an exception writes its appropriate substitution value to the FPR regardless of whether the exception is enabled or disabled. DEBUG# is connected directly to the DEBUG# pin of the IU.

FCCL (Floating Point Condition Code Left) Active High Input

This signal is the condition code generated by the left FPU execution unit. The IU keeps track of the floating point operation, the CC destination, and the pipeline stage. FCCL is connected directly to the FCCL pin of the IU.

FCCR (Floating Point Condition Code Right) Active High Input

This signal is the condition code generated by the right FPU execution unit. The IU keeps track of the floating point operation, the CC destination, and the pipeline stage. FCCR is connected directly to the FCCR pin of the IU.

FPINTR# (Floating Point Interrupt) Active Low Output

This signal is driven by the FPU and indicates that a floating point exception has occurred on either execution unit and that the enable bit of the FSR for that type of exception was active. FPINTR# is connected directly to the FPINTR# pin of the FPU.

2.2.3.2 FLOATING POINT UNIT TO EVEN BANK DATA RAM's

LDE<63:0> (Load Data Even) Active High Input

Load data bus between the FPU and the even bank of the streaming cache. This bus is unidirectional and accepts load input data only. Even bank store data is transferred via the dedicated SDE bus. There are two 32 bit wide SIM-modules on the even bank. LDE<63:32> are connected directly to the LD<32:0> pins of the upper 32 bit even data RAM module. LDE<31:0> are connected directly to the LD<32:0> pins of the lower 32 bit even data RAM module.

LPE<3:0> (Load Parity Even) Active High Input

Load parity bus between the FPU and the even bank of the streaming cache. This bus is unidirectional and accepts load parity data only. Even bank store data is transferred via the dedicated store parity bus. There are two 32 bit wide SIM modules on the even bank. LPE<3:0> are connected directly to the LP<3:0> pins of the upper 32 bit even data RAM module. Parity is checked and generated in 16-bit quantities. Hence only the lower two parity bits (LP<1:0>) are currently used. LP<3:2> are routed to the SIM module connector but are not connected.

SDE<63:0> (Store Data Even) Active High I/O

Store data bus between the FPU, the even bank of the streaming cache, and the external data buffers. The bus itself is bidirectional, although it is unidirectional from the FPU's point of view. When the FPU is in control the bus becomes unidirectional and drives store output data only. Even bank load data is via the dedicated LDE bus. There are three basic uses of the store data bus; Floating Point stores, Integer stores, and streaming cache data transfers to/from main memory. On Floating Point stores the IU instructs the FPU to drive data from a register file out onto the store data bus. Address information for the store is provided by the IU via a dedicated address bus between the IU and the even bank of the streaming cache. Integer Store

data and alignment information is transferred via the TBus to the FPU. Control signals from the IU allow the FPU to pass the data from the TBus to the SDE pins. When data from the streaming cache is to be transferred to main memory, the Cache Controller initiates a streaming cache load. The load data is then transferred on the LDE pins to the FPU. The CC controls the flow of data through the FPU asserting the FOE# and BYPASS# signals to the FPU. When the CC is in control of the bus during a cache line fill, the SDE<63:0> bus is driven by the external data buffers. The SDE pins on the FPU side are tri-stated to allow the cache line fill to occur. Refer to section 2.2.3.4 for more information in FOE# and BYPASS#.

SPE<3:0> (Store Parity Even) Active High Input

Store parity bus between the FPU and the even bank of the streaming cache. This bus is unidirectional and transfers store parity data only. Even bank load data is transferred via the dedicated load parity bus. There are two 32 bit wide SIM modules on the even bank. SPE<3:0> are connected directly to the SP<3:0> pins of the upper 32 bit even data RAM module. Parity is checked and generated in 16-bit quantities. Hence only the lower two parity bits (SP<1:0>) are currently used. SP<3:2> are routed to the SIM module connector but are not connected.

2.2.3.3 FLOATING POINT UNIT TO ODD BANK DATA RAM's

LDO<63:0> (Load Data Odd) Active High Input

Load data bus between the FPU and the odd bank of the streaming cache. This bus is unidirectional and accepts load input data only. Odd bank store data is via the dedicated SDO bus. There are two 32 bit wide SIMmodules on the odd bank. LDO<63:32> are connected directly to the LD<32:0> pins of the upper 32 bit odd data RAM module. LDO<31:0> are connected directly to the LD<32:0> pins of the lower 32 bit odd data RAM module.

SDO<63:0> (Store Data Odd) Active High I/O

Store data bus between the FPU, the odd bank of the streaming cache, and the external data buffers. This bus itself is bidirectional, although it is unidirectional from the FPU's point of view. When the FPU is in control the bus becomes unidirectional and drives store output data only. Odd bank load data is via the dedicated LDO bus. There are three basic uses of the store data bus; Floating Point stores, Integer stores, and streaming cache data transfers to/from main memory. On Floating Point stores the IU instructs the FPU to drive data from a register file out onto the store data bus. Address information for the store is provided by the IU via a dedicated address bus between the IU and the even bank of the streaming cache. Integer Store data and alignment information is transferred via the TBus to the FPU. Control signals from the IU allow the FPU to pass the data from the TBus to the SDO pins. When data from the streaming cache is to be transferred to main memory, the Cache Controller initiates a streaming cache load. The load data is then transferred on the LDE pins to the FPU. The CC controls the flow of data through the FPU asserting the FOE# and BYPASS# signals to the FPU. When the CC is in control of the bus during a cache line fill, the SDO<63:0> bus is driven by the external data buffers. The SDO pins on the FPU side are tri-stated to allow the cache line fill to occur.Refer to section 2.2.3.4 for more information in FOE# and BYPASS#.

2.2.3.4 FLOATING POINT UNIT TO CACHE CONTROLLER

TBUS<71:0> (TBus interface) Active High I/O

TBus <71:0> are connected between the CC, IU, and FPU. There is no communication protocol between the FPU and the CC. The TBus of the FPU and the CC are connected only because the IU must communicate with both.

FOE# (Floating Point Output Enable) Active Low Input

Assertion of FOE# and BYPASS# on streaming cache transfers to main memory enables the output drivers of the FPU which allow data to be driven directly from the LDE<63:0>/LDO<63:0> pins to the SDE<63:0>/ SDO<63:0> pins of the FPU respectively. When data from the streaming cache is to be transferred to main memory, the Cache Controller initiates a streaming cache load. The load data is then transferred on the LDE or LDO pins to the FPU. The CC controls the flow of data through the FPU by asserting the FOE# and BYPASS# signals to the FPU, allowing data to be driven onto the SDE or SDO busses. FOE# deasserted allows incoming FPU data to be written to a register and used internally.

BYPASS# (Floating Point Bypass) Active Low Input

Assertion of BYPASS# allows data from the Load data pins to bypass the internal circuits of the FPU and be transferred directly to the store data output buffers. The assertion of FOE# enables the buffers and allows the streaming cache load data to be driven out onto the store data pins. Bypass deasserted allows data from within the registers of the FPU to be driven out onto the SDE or SDO pins (assuming FOE# is active).

2.2.4 FLOATING POINT UNIT CLOCK INTERFACE

CLK (Reference Clock) Active High Input

Sync_in is part of the PLL feedback path and must be connected to Sync_Out in order for the PLL circuitry to work correctly. The pins are made avaliable externally to allow the user to manually alter the phase of the PLL.

SYNC_OUT (Synchronized PLL input) Active High Output

Sync_Out is part of the PLL feedback path and must be connected to Sync_In in order for the PLL circuitry to work correctly. The pins are made avaliable externally to allow the user to manually alter the phase of the PLL.

LPF_OUT (Low Pass Filter Output) Active High Output

LFP_OUT is a special pin used to test the PLL circuitry during component test for monitoring the status of the low-pass filter. For normal operation this pin can be left unconnected.

VCC_PLL (Voltage Phase Lock Loop)

Figure 12: Floating Point Unit Package Pinout

Table 4: Floating Point Unit Pinout

Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal
A03.....Vcc	B26.....TBUS<46>	D4.....LDE<1>	E27.....Vss	G5.....LDO<3>	H28.....Vss
A05.....TBUS<0>	B28.....TBUS<50>	D6.....TBUS<3>	E29.....Vcc	G7.....LDO<2>	H30.....Vss
A07.....TBUS<6>	B30.....TBUS<54>	D8.....TBUS<7>	E31.....Vss	G9.....LDE<0>	H32.....Vss
A09.....TBUS<12>	B32.....TBUS<58>	D10.....TBUS<11>	E33.....TBUS<67>	G11.....Vcc	H34.....Vss
A11.....TBUS<16>	B34.....TBUS<62>	D12.....TBUS<15>	E35.....TBUS<71>	G13.....Vcc	H36.....NC
A13.....TBUS<20>	B36.....TBUS<66>	D14.....TBUS<21>	E37.....TBUS<75>	G15.....Vcc	H38.....Vss
A15.....TBUS<24>	B38.....TBUS<72>	D16.....TBUS<25>	E39.....TBUS<79>	G17.....Vcc	H40.....Vcc
A17.....TBUS<28>	B40.....TBUS<78>	D18.....TBUS<29>	E41.....X_SDI	G19.....Vcc	H42.....JTAG_R
A19.....TBUS<32>	B42.....DEBUG	D20.....TBUS<33>	E43.....SYNC_OUT	G21.....Vcc	H44.....JTAG_O
A21.....TBUS<36>	B44.....Vss	D22.....TBUS<37>	E45.....CLK	G23.....Vcc	J1.....LDE<5>
A23.....TBUS<40>	C1.....Vcc	D24.....TBUS<43>	F2.....LDE<34>	G25.....Vcc	J3.....LDE<36>
A25.....TBUS<44>	C3.....Vss	D26.....TBUS< 47>	F4.....LDO<34>	G27.....Vcc	J5.....LDO<5>
A27.....TBUS<48>	C5.....LDO<0>	D28.....TBUS<51>	F6.....LDO<33>	G29.....Vcc	J7.....LDO<4>
A29.....TBUS<52>	C7.....TBUS<4>	D30.....TBUS<55>	F8.....Vss	G31.....Vcc	J39.....VCC_PLL
A31.....TBUS<56>	C9.....TBUS<10>	D32.....TBUS<59>	F10.....Vcc	G33.....Vcc	J41.....JTAG_S
A33.....TBUS<60>	C11.....TBUS<17>	D34.....TBUS<65>	F12.....Vss	G35.....Vcc	J43.....SDO<0>
A35.....TBUS<64>	C13.....TBUS<19>	D36.....TBUS<69>	F14.....Vcc	G37.....FVALID#	J45.....SDO<32>
A37.....TBUS<68>	C15.....TBUS<23>	D38.....TBUS<73>	F16.....Vss	G39.....Vcc	K2.....LDE<37>
A39.....TBUS<74>	C17.....TBUS<27>	D40.....TBUS<77>	F18.....Vcc	G41.....SYNC_IN	K4.....LDO<38>
A41.....TBOE#	C19.....TBUS<31>	D42.....RESET#	F20.....Vss	G43.....JTAG_I	K6.....LDO<6>
A43.....Vcc	C21.....TBUS<35>	D44.....X_SDO	F22.....Vcc	G45.....JTAG_C	K8.....Vss
A45.....Vss	C23.....TBUS<39>	E1.....Vss	F24.....Vss	H2.....LDE<4>	K38.....GND_PLL
B2.....Vss	C25.....TBUS<45>	E3.....LDE<2>	F26..... Vcc	H4.....LDO<36>	K40.....Vss
B4.....LDO<32>	C27.....TBUS<49>	E5.....LDO<1>	F28.....Vss	H6.....LDO<35>	K42.....SDE<0>
B6.....TBUS<2>	C29.....TBUS<53>	E7.....TBUS<1>	F30.....Vcc	H8.....Vcc	K44.....SDO<1>
B8.....TBUS<8>	C31.....TBUS<57>	E9.....TBUS<5>	F32.....Vss	H10.....LDE<32>	L1.....LDE<38>
B10.....TBUS<14>	C33.....TBUS<61>	E11.....TBUS<9>	F34.....Vcc	H12.....Vss	L3.....LDE<6>
B12.....TBUS<18>	C35.....TBUS<63>	E13.....TBUS<13>	F36.....Vss	H14.....Vss	L5.....LDO<8>
B14.....TBUS<22>	C37.....TBUS<70>	E15.....Vss	F38.....Vcc	H16.....Vss	L7.....LDO<37>
B16.....TBUS<26>	C39.....TBUS<76>	E17.....Vss	F40.....Vss	H18.....Vss	L39.....Vcc
B18.....TBUS<30>	C41.....NC	E19.....Vcc	F42.....EXT_CLK	H20.....Vss	L41.....SDE<32>
B20.....TBUS<34>	C43.....Vss	E21.....Vss	F44.....LPF_OUT	H22.....Vss	L43.....SDE<1>
B22.....TBUS<38>	C45.....Vcc	E23.....TBUS<41>	G1.....LDE<35>	H24.....Vss	L45.....SDO<33>
B24.....TBUS<42>	D2.....LDE<33>	E25.....Vss	G3.....LDE<3>	H26.....Vss	M2.....LDE<7>

Table 4: Floating Point Unit Pinout

Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal
M4.....LDO<41>	T38.....Vss	Y44.....SDO<6>	AE5.....LDO<48>	AJ41.....Vcc	AP2.....LDE<55>
M6.....LDO<39>	T40.....Vcc	AA1.....LDE<14>	AE7.....Vss	AJ43.....SDE<11>	AP4.....LDO<23>
M8.....Vcc	T42.....SDE<4>	AA3.....LDE<45>	AE39.....Vcc	AJ45.....SDO<42>	AP6.....LDO<55>
M38.....Vss	T44.....SDO<4>	AA5.....LDO<14>	AE41.....Vss	AK2.....LDE<52>	AP8.....Vss
M40.....Vcc	U1.....LDE<11>	AA7.....Vss	AE43.....SDE<9>	AK4.....LDO<50>	AP38.....Vss
M42.....SDE<33>	U3.....LDE<42>	AA39.....Vcc	AE45.....SDO<40>	AK6.....LDO<21>	AP40.....Vss
M44.....SDO<2>	U5.....LDO<43>	AA41.....Vss	AF2.....LDE<49>	AK8.....Vss	AP42.....SDE<14>
N1.....LDE<8>	U7.....Vss	AA43.....SDE<38>	AF4.....LDO<17>	AK38.....Vss	AP44.....SDO<13>
N3.....LDE<39>	U39.....Vcc	AA45.....SDO<38>	AF6.....Vcc	AK40.....Vss	AR1.....LDE<24>
N5.....LDO<9>	U41.....Vss	AB2.....LDE<46>	AF8.....Vss	AK42.....SDE<43>	AR3.....LDE<56>
N7.....LDO<7>	U43.....SDE<36>	AB4.....LDO<15>	AF38.....Vss	AK44.....SDO<11>	AR5.....LDO<56>
N39.....Vcc	U45.....SDO<36>	AB6.....LDO<46>	AF40.....Vss	AL1.....LDE<21>	AR7.....LDO<57>
N41.....SDE<2>	V2.....LDE<43>	AB8.....Vss	AF42.....SDE<41>	AL3.....LDE<53>	AR39.....Vcc
N43.....SDE<34>	V4.....LDO<13>	AB38.....Vss	AF44.....SDO<9>	AL5.....LDO<52>	AR41.....SDE<15>
N45.....SDO<34>	V6.....Vcc	AB40.....Vss	AG1.....LDE<18>	AL7.....LDO<54>	AR43.....SDE<46>
P2.....LDE<40>	V8.....Vss	AB42.....SDE<7>	AG3.....LDE<50>	AL39.....Vcc	AR45.....SDO<45>
P4.....LDO<11>	V38.....Vss	AB44.....SDO<7>	AG5.....LDO<18>	AL41.....Vss	AT2.....LDE<25>
P6.....LDO<40>	V40.....Vss	AC1.....LDE<15>	AG7.....Vss	AL43.....SDE<12>	AT4.....LDO<25>
P8.....Vss	V42.....SDE<5>	AC3.....LDE<47>	AG39.....Vcc	AL45.....SDO<43>	AT6.....LDO<26>
P38.....Vss	V44.....SDO<5>	AC5.....LDO<47>	AG41.....Vss	AM2.....LDE<22>	AT8.....Vcc
P40.....Vss	W1.....LDE<44>	AC7.....Vss	AG43.....SDE<10>	AM4.....LDO<20>	AT38.....BYPASS#
P42.....SDE<3>	W3.....LDE<12>	AC39.....Vcc	AG45.....SDO<41>	AM6.....LDO<22>	AT40.....Vcc
P44.....SDO<3>	W5.....LDO<44>	AC41.....SDE<39>	AH2.....LDE<19>	AM8.....Vcc	AT42.....NC
R1.....LDE<41>	W7.....Vss	AC43.....SDE<8>	AH4.....LDO<49>	AM38.....Vss	AT44.....SDO<14>
R3.....LDE<9>	W39.....Vcc	AC45.....SDO<39>	AH6.....LDO<51>	AM40.....Vcc	AU1.....LDE<57>
R5.....LDO<42>	W41.....Vcc	AD2.....LDE<16>	AH8.....Vcc	AM42.....SDE<44>	AU3.....LDE<26>
R7.....Vss	W43.....SDE<37>	AD4.....LDO<16>	AH38.....Vss...	AM44.....SDO<12>	AU5.....LDO<58>
R39.....Vcc	W45.....SDO<37>	AD6.....Vcc	AH40.....Vcc	AN1.....LDE<54>	AU7.....LDO<59>
R41.....Vss	Y2.....LDE<13>	AD8.....Vcc	AH42.....SDE<42>	AN3.....LDE<23>	AU39.....LPO<3>
R43.....SDE<35>	Y4.....LDO<45>	AD38.....Vss	AH44.....SDO<10>	AN5.....LDO<53>	AU41.....PMODE
R45.....SDO<35>	Y6.....Vcc	AD40.....Vcc	AJ1.....LDE<51>	AN7.....LDO<24>	AU43.....SDO<15>
T2.....LDE<10>	Y8.....Vcc	AD42.....SDE<40>	AJ3.....LDE<20>	AN39.....Vcc	AU45.....SDO<46>
T4.....LDO<12>	Y38.....Vss	AD44.....SDO<8>	AJ5.....LDO<19>	AN41.....SDE<45>	AV2.....LDE<58>
T6.....LDO<10>	Y40.....Vcc	AE1.....LDE<48>	AJ7.....Vss	AN43.....SDE<13>	AV4.....LDO<27>
T8.....Vcc	Y42.....SDE<6>	AE3.....LDE<17>	AJ39.....Vcc	AN45.....SDO<44>	AV6.....LDO<28>

Table 4: Floating Point Unit Pinout

Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal
AV8.....Vss	AW33.....Vcc	BA13.....SPE<1>	BB38.....SDE<49>	BD18.....SDO<28>	BE43.....Vcc
AV10.....LDE<31>	AW35.....Vcc	BA15.....Vss	BB40.....SDE<48>	BD20.....SDO<27>	BE45.....Vss
AV12.....Vss	AW37.....SDE<47>	BA17.....Vcc	BB42.....LPE<0>	BD22.....SDO<26>	
AV14.....Vss	AW39.....Vcc	BA19.....Vss	BB44.....LPE<1>	BD24.....SDO<25>	
AV16.....Vss	AW41.....LPO<2>	BA21.....Vss	BC1.....Vcc	BD26.....SDO<24>	
AV18.....Vss	AW43.....DEQSO#	BA23.....SDE<57>	BC3.....Vss	BD28.....SDO<23>	
AV20.....Vss	AW45.....NC	BA25.....Vss	BC5.....LDO<31>	BD30.....SDO<22>	
AV22.....Vss	AY2.....LDE<28>	BA27.....Vcc	BC7.....PERRE	BD32.....SDO<21>	
AV24.....Vss	AY4.....LDE<29>	BA29.....Vss	BC9.....SPO<2>	BD34.....SDO<20>	
AV26.....Vss	AY6.....LDO<61>	BA31.....Vss	BC11.....SDE<63>	BD36.....SDO<19>	
AV28.....Vss	AY8.....Vcc	BA33.....SDE<19>	BC13.....SDE<31>	BD38.....SDO<49>	
AV30.....Vss	AY10.....Vss	BA35.....SDE<18>	BC15.....SDE<30>	BD40.....SDO<16>	
AV32.....Vss	AY12.....Vcc	BA37.....SDE<17>	BC17.....SDE<29>	BD42.....FCCL	
AV34.....Vss	AY14.....Vss	BA39.....SDE<16>	BC19.....SDE<28>	BD44.....Vss	
AV36.....FCCR	AY16.....Vcc	BA41.....LPO<0>	BC21.....SDE<27>	BE1.....Vss	
AV38.....Vss	AY18.....Vss	BA43.....LPE<2>	BC23.....SDE<26>	BE3.....Vcc	
AV40.....Vss	AY20.....Vcc	BA45.....LPE<3>	BC25.....SDE<56>	BE5.....LDO<63>	
AV42.....DEQSE#	AY22.....Vss	BB2.....LDE<30>	BC27.....SDE<55>	BE7.....NC	
AV44.....NC	AY24.....Vcc	BB4.....LDE<62>	BC29.....SDE<54>	BE9.....SPO<1>	
AW1.....LDE<27>	AY26.....Vss	BB6.....ENQLE#	BC31.....SDE<53>	BE11.....SDO<63>	
AW3.....LDE<59>	AY28.....Vcc	BB8.....NC	BC33.....SDE<52>	BE13.....SDO<62>	
AW5.....LDO<60>	AY30.....Vss	BB10.....SPE<2>	BC35.....SDE<20>	BE15.....SDO<61>	
AW7.....LDO<29>	AY32.....Vcc	BB12.....SPE<0>	BC37.....SDO<18>	BE17.....SDO<60>	
AW9.....LDE<63>	AY34.....Vss	BB14.....SDE<62>	BC39.....SDO<48>	BE19.....SDO<59>	
AW11.....Vcc	AY36.....Vcc	BB16.....SDE<61>	BC41.....FPINTR#	BE21.....SDO<58>	
AW13.....Vcc	AY38.....Vss	BB18.....SDE<60>	BC43.....Vss	BE23.....SDO<57>	
AW15.....Vcc	AY40.....Vcc	BB20.....SDE<59>	BC45.....Vcc	BE25.....SDO<56>	
AW17.....Vcc	AY42.....LPO<1>	BB22.....SDE<58>	BD2.....Vss	BE27.....SDO<55>	
AW19.....Vss	AY44.....FOE#	BB24.....SDE<25>	BD4.....LDO<62>	BE29.....SDO<54>	
AW21.....Vcc	BA1.....LDE<60>	BB26.....SDE<24>	BD6.....ENQLO#	BE31.....SDO<53>	
AW23.....Vcc	BA3.....LDE<61>	BB28.....SDE<23>	BD8.....SPO<3>	BE33.....SDO<52>	
AW25.....Vcc	BA5.....LDO<30>	BB30.....SDE<22>	BD10.....SPO<0>	BE35.....SDO<51>	
AW27.....Vcc	BA7.....ENQLP#	BB32.....SDE<21>	BD12.....SDO<31>	BE37.....SDO<50>	
AW29.....Vcc	BA9.....PERRO	BB34.....SDE<51>	BD14..... SDO<30>	BE39.....SDO<17>	
AW31.....Vcc	BA11.....SPE<3>	BB36.....SDE<50>	BD16.....SDO<29>	BE41.....SDO<47>	

Voltage source for the phase lock loop circuitry. Connected to a regulated 3.3 volt source. VCC_PLL can be connected to GND_PLL through .1 and .015 microfarad capacitors in parallel.

GND_PLL (Ground Phase Lock Loop)

Ground source for the phase lock loop circuitry. GND_PLL can be connected to VCC_PLL through .1 microfarad and .015 microfarad capacitors in parallel.

RESET

Reset pin of the Floating Point Unit.

2.2.5 FLOATING POINT UNIT PACKAGE PINOUT

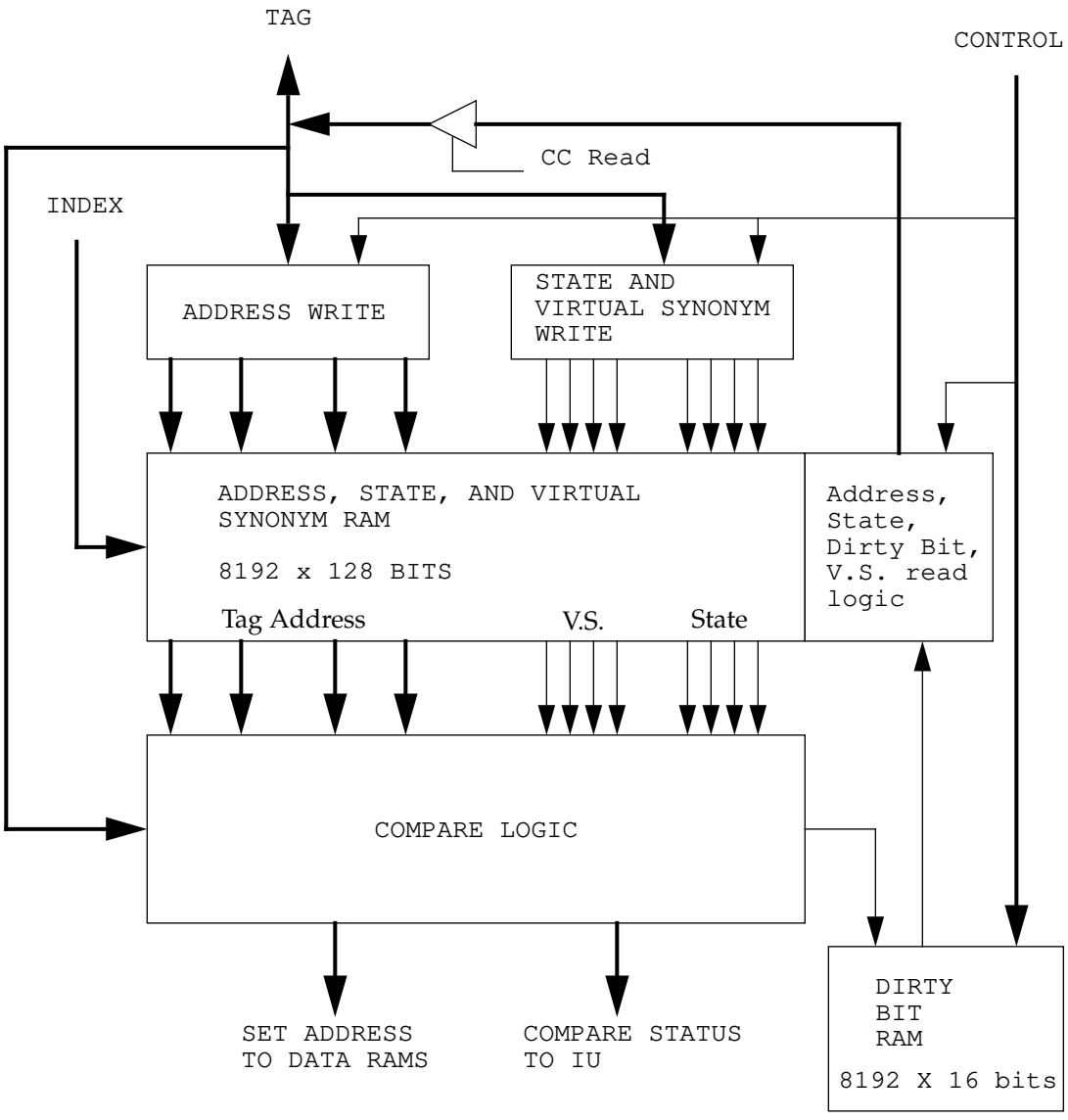

Figure 13: Tag RAM Block Diagram

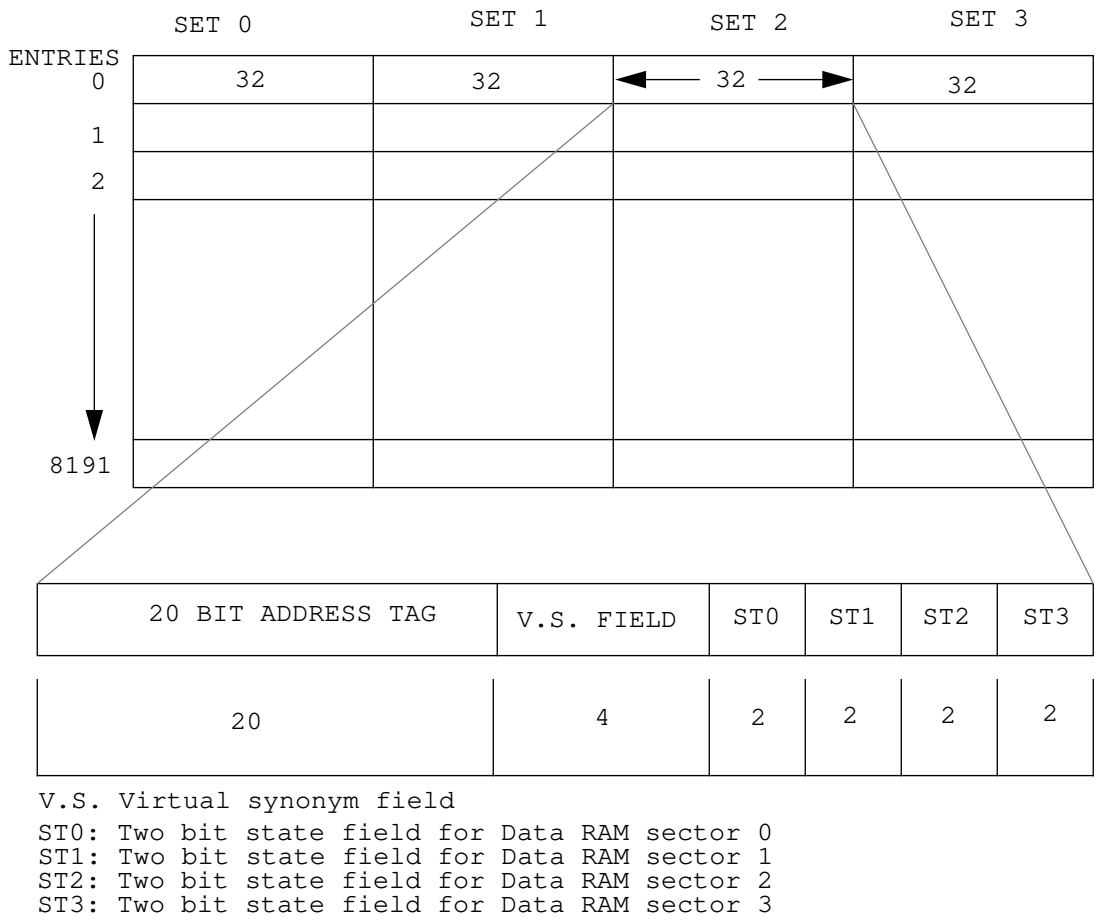

Figure 14: Tag RAM Bit Assignments

Figure 12 and Table 4 below show a package pinout of the Floating Point Unit. Active low signals are denoted with an # at the end. NC are pins which are not connected internally.

2.3 TAG RAM

Two identical tag RAM's are required in the TFP Microprocessor Chip Set in order to support the interleaved architecture of the second level streaming cache. Both RAM's contain the same information. One is used for the even bank, the other for the odd bank. The two banks are differentiated by the state of address bit A3. If this bit is low the access is to the even bank. A high on this pin enables the odd bank. Figure 13 shows a block diagram of the tag RAM.

2.3.1 TAG RAM ORGANIZATION

The Tag RAM is 4-way set associative and is organized as 8192 entries X 128 bits. Each entry contains 32 bits of information for each of the four ways, hence the 128 bit width. The 32 bits of each way is further broken down into a 20 bit tag address, 4 virtual synonym bits, and 8 state bits. Figure 14 shows the Tag RAM bit assignments.

Section 2.4.2 defines sectors, lines, etc. and has more information on the streaming cache architecture.

2.3.2 TAG RAM CYCLE TYPES

There are three basic cycles performed on the tag RAM:

1) Lookups
2) Reads

3) Writes

Lookups are performed by the IU and are done when the IU wants to either read or write second level cache. The IU supplies dedicated tag, index, sector, and control lines to each tag RAM. In a set associative cache implementation the IU does not know which of the sets of a given address the data is contained. The tag RAM compare supplies this information in the form of a 2 bit encoded value. This value forms the two upper bits of the streaming cache address and determines which of the sets the data resides. The IU does not read or write the tag RAM directly except for the dirty bit. If a write lookup gets a hit to the Tag RAM, indicating that the data exists in the streaming cache, the data is written out to the cache and the dirty bit is set on the next clock. If the line is in the exclusive state and it is unmodified (clean), setting of the dirty bit changes the state of the line from exlusive clean to exclusive dirty. If the line is already in the modified state (dirty) the dirty line must be written out to main memory before the write can be completed.

Read cycles are initiated by the Cache Controller (CC) and inform the CC of the status of a given entry. Information exchange is through the TBus. The CC supplies the 2 bit set address along with the proper control signals. The information is read from the read logic block attached to the tag RAM in figure 13.

Write cycles are also performed by the Cache Controller and are done when the CC wants to modify an entry. Refer to figure 13. On write cycles the tag, index, and sector information are generated by the CC and transferred to the IU on the TBus. The IU plays no part in a write cycle other than to pass the information from the TBus onto the appropriate dedicated busses of the even and odd Tag RAM's..

Note that the 20 bit tag field connects to both the address write and V.S. write blocks. Hence the tag pins contain different information based on whether the CC is writing the address, or the state and V.S. bits. Only one of these writes can be executed at a time.

2.3.3 DIRTY BIT RAM

The Dirty Bit RAM in figure 15 is organized as 8192 entries X 16 bits. The number of entries is the same as the tag RAM and is indexed in the clock after the Tag RAM access. The purpose of a separate Dirty Bit RAM is to allow the status of a given line to be changed during a Tag RAM access. Therefore if there is a write to the streaming cache and the dirty bit status for that write must be updated, the dirty bit can be set on the next clock, even though the Tag RAM has already started an access for the next cycle. The 2 bit state field in the tag RAM identifies the status of the entry; either invalid, shared, or exclusive. The exclusive line can either be clean or dirty. The Dirty Bit RAM maintains this information. In a sectored cache implementation there is one dirty bit for each sector. Therefore the dirty bits are not always the same for both the even and odd Tag RAM's. The address, state, and virtual synonym information is always the same for both Tag RAM's, but the dirty bit information can be different.

2.3.4 TAG RAM BUS INTERFACE

The Tag RAM bus interface section defines the signal pins of the even and odd Tag RAM's. The even and odd Tag RAM's have dedicated interfaces to all components in the system except the FPU. More information on these pins is available in sections 2.1.12.2 and 2.1.12.3.

2.3.4.1 EVEN TAG RAM TO INTEGER UNIT

INDEX<12:0> (Tag RAM Index)Active High Input

These pins form the index into the Tag RAM and are connected to the INDEXE pins of the Integer Unit. Refer to the INDEXE pin descriptions in section 2.1.12.2 for connectivity information.

SECTOR<1:0> (Tag RAM Sector) Active High Input

These two pins act as a 4:1 mux selector inside the Tag RAM. Each line of the Data RAM contains 4 sectors, with each sector containing sixteen 64 bit words, 8 in the odd bank and 8 in the even bank. The SECTOR<1:0> pins are used to select the correct state information for the sector to be accessed. Refer to the INDEXE pin descriptions in section 2.1.12.2 for more infomation.

TAG<19:0> (Tag RAM Tag address) Active High I/O

These pins form the tag address of the tag RAM. The TAG pins are written and read by the CC. The IU uses TAG<19:0> for lookup cycles. Reads and writes from the CC are passed onto the TAG pins through the TBus. Refer to the TAGE pin descriptions in section 2.1.12.2 for connectivity information.

MATCH# Active Low Output

This signal is an output from the even Tag RAM. When the IU performs a lookup cycle to the tag RAM an index is supplied which corresponds to all four sets of the four way set associative even tag RAM. Assertion of this signal by the tag RAM indicates that the address on the bus compared to one of the four sets. MATCH# is connected directly to the MATCHE# pin of the IU.

MCHVS<3:0> (Match Virtual Synonym) Active High Output

The bits correspond to virtual address bits 15:12 and form the virtual synonym entry in the Tag RAM. Refer to the MCHVSE <3:0> pin descriptions in section 2.1.12.2 for more information.

MCHSA<1:0> (Match Set Address) Active High Output

On a lookup cycle, these 2 bits are used form an encoded value of which of the four ways in the Tag RAM was a hit. Refer to the MCHSAE pin descriptions in section 2.1.12.2 for more information.

MCHST<1:0> (Match State) Active High Output

These 2 bits encode the state information of a given 128 byte sector in the data RAM's. The state can be shared, exclusive, or invalid. Refer to the MCHSTE <1:0> pin descriptions in section 2.1.12.2 for more information.

ESA<1:0> (External Set Address) Active High Input

These pins are connected directly to the ESAE<1:0> pins of the IU. Refer to the ESAE pin description in section 2.1.12.2 the IU for more information.

ESASEL (External Set Address Select) Active High Input

The ESASEL pin acts as a mux select to drive the ESA<1:0> bits through to the DATSA<1:0> bits.
Refer to the ESASELE pin description in section 2.1.12.2 for more information.

DBSET# (Dirty Bit Set) Active Low Input

The dirty bit RAM is written conditionally based on the state of DBSET#. Refer to the DBSETE# pin description in section 2.1.12.2 for more information.

2.3.4.2 ODD TAG RAM TO INTEGER UNIT

INDEX<12:0> (Tag RAM Index)Active High Input

These pins form the index into the Tag RAM and are connected to the INDEXO pins of the Integer Unit. Refer to the INDEXO pin descriptions in section 2.1.12.3 for connectivity information.

SECTOR<1:0> (Tag RAM Sector) Active High Input

These two pins act as a 4:1 mux selector inside the Tag RAM. Each tag RAM entry corresponds to four lines in the Data RAM's (streaming cache). Each line of the Data RAM contains 4 sectors, with each sector containing sixteen 64 bit double words, eight for the odd bank and 8 for the even bank.. The SECTOR<1:0> pins are used to select the correct state information for the sector to be accessed. Refer to the INDEXO pin descriptions in section 2.1.12.3 for more infomation.

TAG<19:0> (Tag RAM Tag address) Active High I/O

These pins form the tag address of the tag RAM. The TAG pins are written and read by the CC. The IU uses TAG<19:0> on lookup cycles. Reads and writes from the CC are passed onto the TAG pins through the TBus. Refer to the TAGO pin descriptions in section 2.1.12.3 for connectivity information.

MATCH# (Tag RAM Match) Active Low Output

This signal is an output from the odd Tag RAM. When the IU performs a lookup cycle to the tag RAM an index is supplied which corresponds to all four sets of the four way set associative even tag RAM. Assertion of this signal by the tag RAM indicates that the address on the bus compared to one of the four sets. MATCH# is connected directly to the MATCHO# pin of the IU.

MCHVS<3:0> (Match Virtual Synonym) Active high Output

The bits correspond to virtual address bits 15:12 and form the virtual synonym entry in the Tag RAM. Refer to the MCHVSO<3:0> pin descriptions in section 2.1.12.3 for more information.

MCHSA<1:0> (Match Set Address) Active High Output

On a lookup cycle, these 2 bits are used form an encoded value of which of the four ways in the Tag RAM was a hit. Refer to the MCHSAO pin descriptions in section 2.1.12.3 for more information.

MCHST<1:0> (Match State) Active High Output

These 2 bits encode the state information of a given 128 byte sector in the data RAM's. The state can be shared, exclusive, or invalid. Refer to the MCHSTO <1:0> pin descriptions in section 2.1.12.3 for more information.

ESA<1:0> (External Set Address) Active High Input

These pins are connected directly to the ESAE<1:0> pins of the IU. Refer to the ESAO pin description in section 2.1.12.3 for more information.

ESASEL (External Set Address Select) Active High Input

The ESASEL pin acts as a mux select to drive the ESA<1:0> bits through to the DATSA<1:0> bits.
Refer to the ESASELO pin description in section 2.1.12.3 for more information.

DBSET# (Dirty Bit Set) Active Low Input

The dirty bit RAM is written conditionally based on the state of DBSET#. Refer to the DBSETO# pin description in section 2.1.12.3 for more information.

2.3.4.3 EVEN TAG RAM TO EVEN DATA RAM INTERFACE

DATSAXE<1:0>, (Data Set Address X Even) Active High Output
DATSAYE<1:0>, (Data Set Address Y Even) Active High Output
DATSAWE#<1:0>, (Data Set Address W Even) Active Low Output
DATSAZE#<1:0>, (Data Set Address Z Even) Active Low Output

These two bits form the upper two address bits for the Even bank of the streaming cache. Each of the four sets of DATSA pins contains the exact same information and are buffered versions of one another. Two sets are inverted. Refer to section 2.4.2 for more information on DATSA connectivity.

2.3.4.4 ODD TAG RAM TO ODD DATA RAM INTERFACE

DATSAXO<1:0>, (Data Set Address X Odd) Active High Output
DATSAYO<1:0>, (Data Set Address Y Odd) Active High Output
DATSAWO#<1:0>, (Data Set Address W Odd) Active Low Output
DATSAZO#<1:0>, (Data Set Address Z Odd) Active Low Output

These two bits form the upper two address bits for the Odd bank of the streaming cache. Each of the four sets of DATSA pins contains the exact same information and are buffered versions of one another. Refer to section 2.4.2 for more information on DATSA connectivity.

2.3.4.5 TAG RAM TO CACHE CONTROLLER INTERFACE

The Cache Controller does not distinguish between even and odd Tag RAM's. Each interface pin of the CC goes to both the even and odd Tag RAM's.

RWSA<1:0> (Read Write Set Address) Active High Input

This two bit field is driven by the CC when the Cache Controller is reading or writing to the Tag RAM. They are used to choose between one of the 4 sets of the 4-way set associative tag RAM. The Dirty Bit RAM also uses these pins to update the correct dirty bit entry.

STRD# (State Read) Active Low Input

STRD# is a control pin used in the Cache Controller read logic block (see figure 13) inside the Tag RAM and is used to drive either State and V.S. information, or tag address information onto the bi-directional tag pins. At the same time the OE# pin must be asserted to enable this information onto the Tag address bus. The information enters the IU and is returned to the CC via the TBus.

OE# (Output Enable) Active Low Input
The OE# pin is driven by the CC when the CC is reading the Tag RAM. Assertion of OE# allows the information currently on the 20 bit Tag Address bus to be driven back to the IU where it is returned to the CC via the TBus.

STWE# (State Write Enable) Active Low Input

STWE# is asserted whenever the CC is writing state and Virtual Synonym information to the Tag RAM. Both the 20 bit tag address and the state and V.S. infomration are carried to the Tag RAM via the tag address bus. The CC tracks the information on the pins and assertes STWE# if state and V.S. information is to be written. TWE# is asserted if tag address information is to be written.

TWE# (Tag Write Enable) Active Low Input

TWE# is driven by the CC whenever tag address information is to be written to the Tag RAM. Refer to STWE# above.

2.3.4.6 CLOCK INTERFACE

CLK (Clock Input) Active high Input

This clock input should be used in designs which desire to bypass the on-chip phase lock loop logic. The input frequency is 75 MHz. This pin should be tied to ground when not in use.

PLLIN1 (Phase Lock Loop 1) Active High Input

There are two phase lock loop circuits on-chip. PLLIN1 is the 75 MHz clock input for the CMOS phase lock loop. This pin should be tied to ground when not in use.

PLLIN2 (Phase Lock Loop 1) Active High Input

PLLIN2 is the 75 MHz clock input for the Bi-Polar phase lock loop.This pin should be tied to ground when not in use.

PLLEN (Phase Lock Loop Enable) Active High Input

Enable pin for the bi-polar phase lock loop. Assertion of PLLEN enables the bipolar logic. Deassetion of this pin enables the CMOS phase lock loop logic.

JTAG_C (JTAG Clock) Active High Input

Input clock for the JTAG circuitry. Equivelent to the TCK pin in the IEEE 1149.1 JTAG specification.

JTAG_S (JTAG Control logic) Active High Input

JTAG control pin. Equivelent to the TMS (Test Mode Select) pin in the IEEE 1149.1 JTAG specification.

JTAG_I (JTAG In) Active High Input

JTAG serial input. Equivelent to the TDI (Test Data In) pin in the IEEE 1149.1 JTAG specification.

PLLOUT1 (Phase Lock Loop Out 1) Active High Output

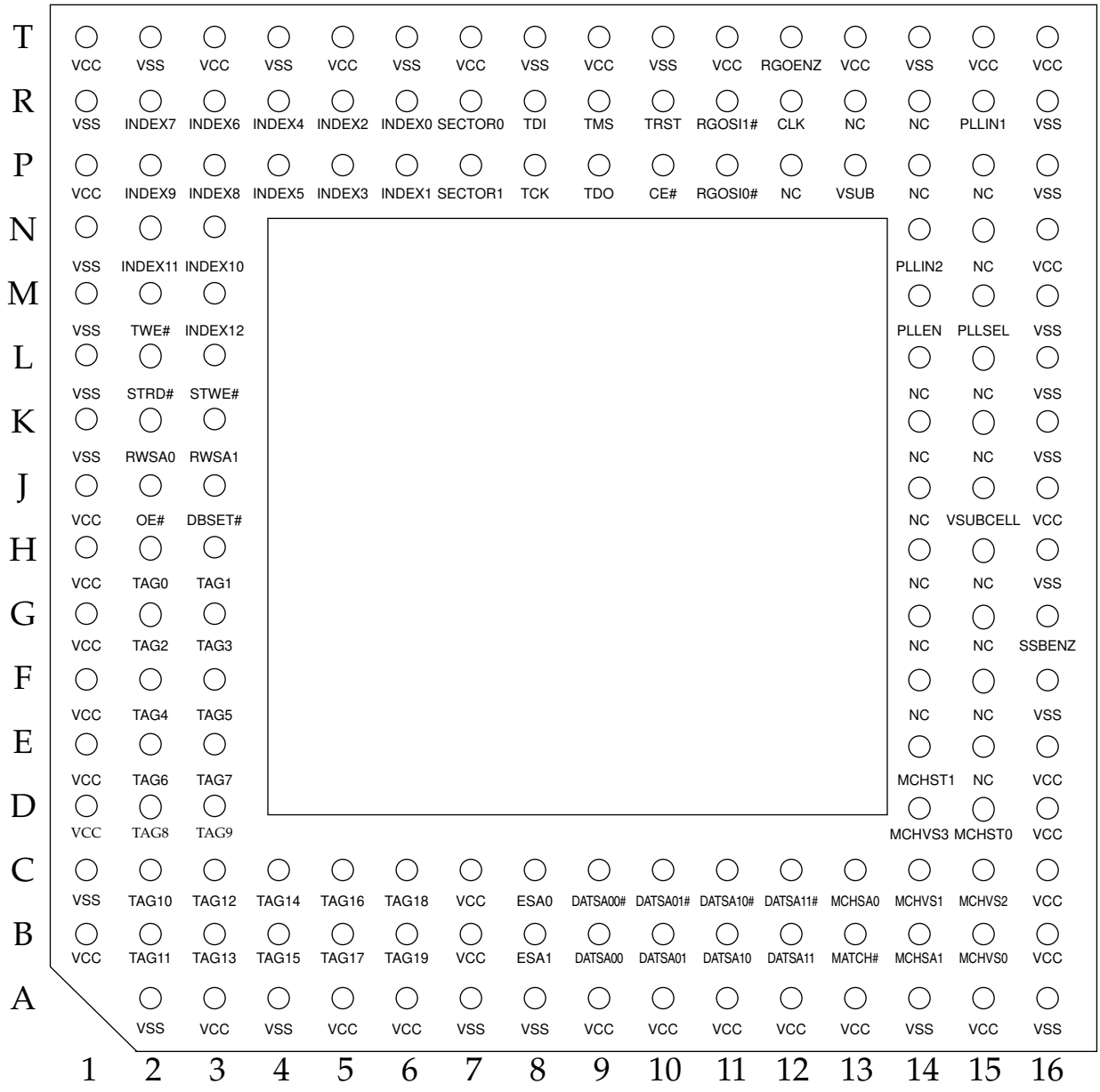

Figure 15: Tag RAM Package Pinout

Table 5: Tag RAM Pinout

Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal	Location .. .Signal
A2.....Vss	B12.....DATSAY<1>	D16.....Vcc	J2.....OE#	N14.....PLLIN2	R8.....TDI
A3.....Vcc	B13.....MATCH#	E1.....Vcc	J3.....DBSET#	N15.....NC	R9.....TMS
A4.....Vss	B14.....MCHSA<1>	E2.....TAG<6>	J14.....NC	N16.....Vcc	R10.....TRST
A5.....Vcc	B15.....MCHVS<0>	E3.....TAG<7>	J15.....VSUBCELL	P1.....Vcc	R11.....RGOSI1
A6.....Vcc	B16.....Vcc	E14.....MCHST<1>	J16.....Vcc	P2.....INDEX<9>	R12.....CLK
A7.....Vss	C1.....Vss	E15.....NC	K1.....Vss	P3.....INDEX<8>	R13.....NC
A8.....Vss	C2.....TAG<10>	E16.....Vcc	K2.....RWSA<1>	P4.....INDEX<5>	R14.....Vcc
A9.....Vcc	C3.....TAG<12>	F1.....Vcc	K3.....RWSA<0>	P5.....INDEX<3>	R15.....PLLIN1
A10.....Vcc	C4.....TAG<14>	F2.....TAG<4>	K14.....NC	P6.....INDEX<1>	R16.....Vss
A11.....Vcc	C5.....TAG<16>	F3.....TAG<5>	K15.....NC	P7.....SECTOR<1>	T1.....Vcc
A12.....Vcc	C6.....TAG<18>	F14.....NC	K16.....Vss	P8.....TCK	T2.....Vss
A13.....Vcc	C7.....Vss	F15.....NC	L1.....Vss	P9.....TDO	T3.....Vcc
A14.....Vss	C8.....ESA<0>	F16.....Vss	L2.....STRD#	P10.....CE#	T4.....Vss
A15.....Vcc	C9....DATSAW#<0>	G1.....Vcc	L3.....STWE#	P11.....RGOSI0	T5.....Vss
A16.....Vss	C10..DATSAW#<1>	G2.....TAG<2>	L14.....NC	P12.....NC	T6.....Vcc
B1.....Vcc	C11...DATSAZ#<0>	G3.....TAG<3>	L15.....NC	P13.....VSUB	T7.....Vss
B2.....TAG<11>	C12...DATSAZ#<1>	G14.....NC	L16.....Vss	P14.....NC	T8.....Vcc
B3.....TAG<13>	C13.....MCHSA<0>	G15.....NC	M1.....Vss	P15.....NC	T9.....Vss
B4.....TAG<15>	C14.....MCHVS<1>	G16.....SSBENZ	M2.....TWE#	P16.....Vss	T10.....Vss
B5.....TAG<17>	C15.....MCHVS<2>	H1.....Vss	M3.....INDEX<12>	R1.....Vss	T11.....Vcc
B6.....TAG<19>	C16.....Vcc	H2.....TAG<0>	M14.....PLLEN	R2.....INDEX<7>	T12.....RGOENZ
B7.....ESASEL	D1.....Vcc	H3.....TAG<1>	M15.....PLLSEL	R3.....INDEX<6>	T13.....Vcc
B8.....ESA<1>	D2.....TAG<8>	H14.....NC	M16..... Vss	R4.....INDEX<4>	T14.....Vss
B9.....DATSAX<0>	D3.....TAG<9>	H15.....NC	N1.....Vss	R5.....INDEX<2>	T15.....Vcc
B10.....DATSAX<1>	D14.....MCHVS<3>	H16.....Vss	N2.....INDEX<11>	R6.....INDEX<0>	T16.....Vcc
B11.....DATSAY<0>	D15.....MCHST<0>	J1.....Vcc	N3.....INDEX<10>	R7.....SECTOR<0>	

Output of the CMOS Phase Lock Loop circuitry. This input allows the user to modify the phase of the loop. This pin is to be left unconnected for normal operation when using the PLLIN1 clock input.

PLLOUT2 (Phase Lock Loop Out 2) Active High Output

Output of the Bi-Ploar Phase Lock Loop circuitry. This input allows the user to modify the phase of the loop. This pin is to be left unconnected for normal operation when using the PLLIN2 clock input.

JTAG_O (JTAG Out) Active High Output

JTAG serial output. Equivelent to the TDO (Test Data Out) pin in the IEEE 1149.1 JTAG specification.

2.3.5 TAG RAM PACKAGE PINOUT

Figure 15 and Table 5 below show a package pinout of the Floating Point Unit. Active low signals are denoted with an # at the end. NC are pins which are not connected internally.

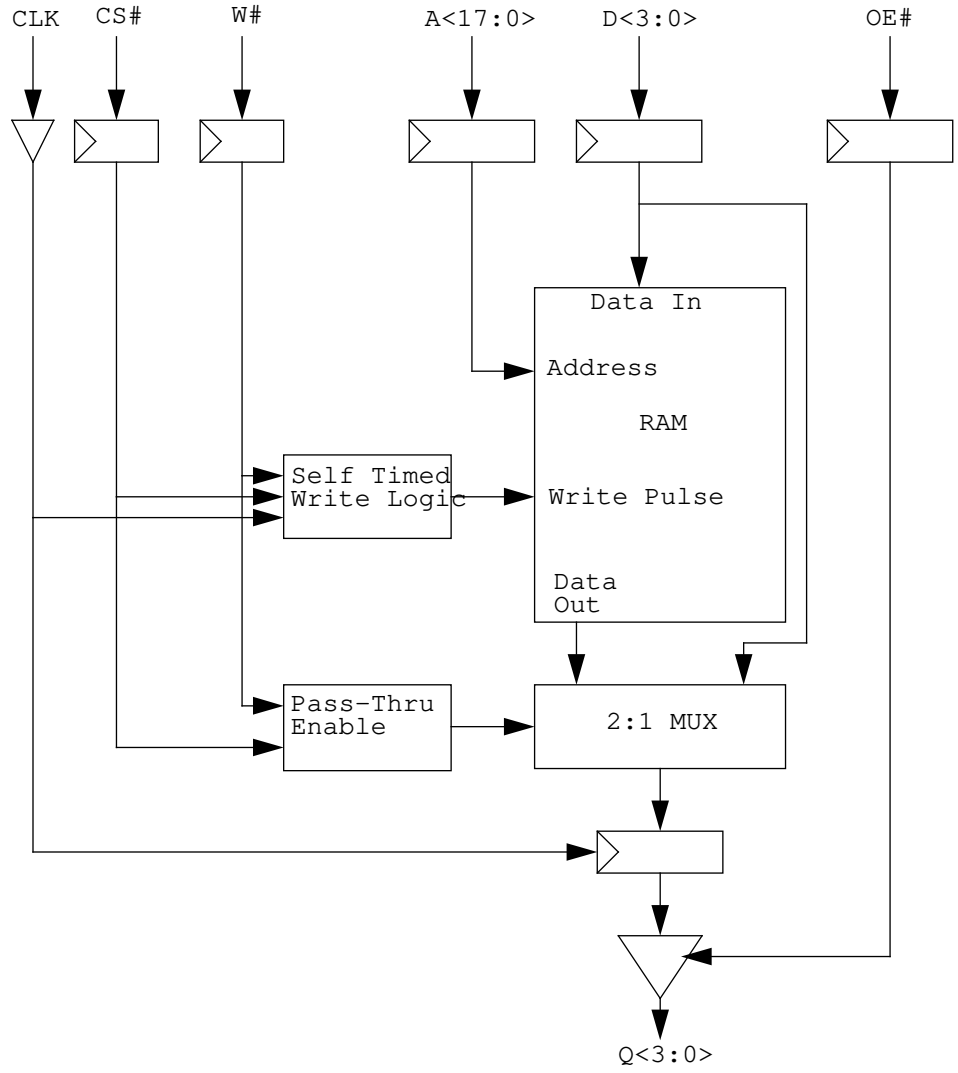

Figure 16: Data Ram Architecture Block Diagram

2.4 STREAMING CACHE DATA RAMS

The streaming cache data RAM's have separate load and store data busses. Although only one cycle can be performed by the data RAM's at a time, both read and write data can be on their respective busses at the same time. Having separate busses eliminates any bus turnaround time, which occurrs on back to back read followed by write cycles, and allows read and write data to be pipelined to the RAM, effectively allowing the RAM to perform a read or write operation every clock.

The total memory size is split between the even and odd banks. Each bank contains 2 MBytes and has a dedicated Tag RAM, allowing accesses to the banks to operate simultaneously and independently of one another. The RAM is buffered by input and output data registers. If the RAM is performing a read and write data appears on the input bus, the data is placed in the register. Self-timed write logic allows the RAM to write the data as soon as the RAM finishes the previous read cycle.

2.4.1 DATA RAM ARCHITECTURE

The 4 MByte streaming cache is implemented using 256 K X 4 Data RAM's. Each individual data RAM is be 256K x 4 bits wide and has separate read and write busses. Figure 16 shows a block diagram of the 256K X 4 Data RAM.
The Data RAM's must support four types of cycles which are decoded by the CS# and WE# bits:

1) /CS * /W = Write and Pass through
2) /CS * W = Read
3) CS * /W = Pass through
4) CS * W = Don't care

So each line contains 512 bytes, 256 bytes per bank. Since there are four sectors per line each sector contains 128 bytes.

The Data Set Address (DATSA) pins form the upper two bits of address for the cache and are used to differentiate between the four banks. Addressing the entire 4 MByte cache requires 22 address bits. The 256K RAM uses 18 of these bits. Accesses to the second level cache can either be in 32 or 64 bits quantities. Each 32 bit SIMmodule in the 4 MByte example has a dedicated write enable, allowing only 32 bits of the word to be written by the IU. The address breakdown is as follows:

DATSA<1:0>	2 bits - Decodes one of four sets in the cache.
A<19:9>	11 bits - Decodes one of 2048 lines per set.
A<8:7>	2 bits - Decodes one of four sectors per line.
A<6:4>	3 bits - Decodes one of eight 64 bit words per sector.
	-------
	18 address bits per RAM = 256K locations
A<3>	1 bit - Determines which bank is accessed.
A<2:0>	3 bits - Decodes a given byte within a 64 bit doubleword.
	-------
	22 address bits total = 4 MByte cache memory locations

Figure 17 shows a block diagram of the 4 MByte cache example.

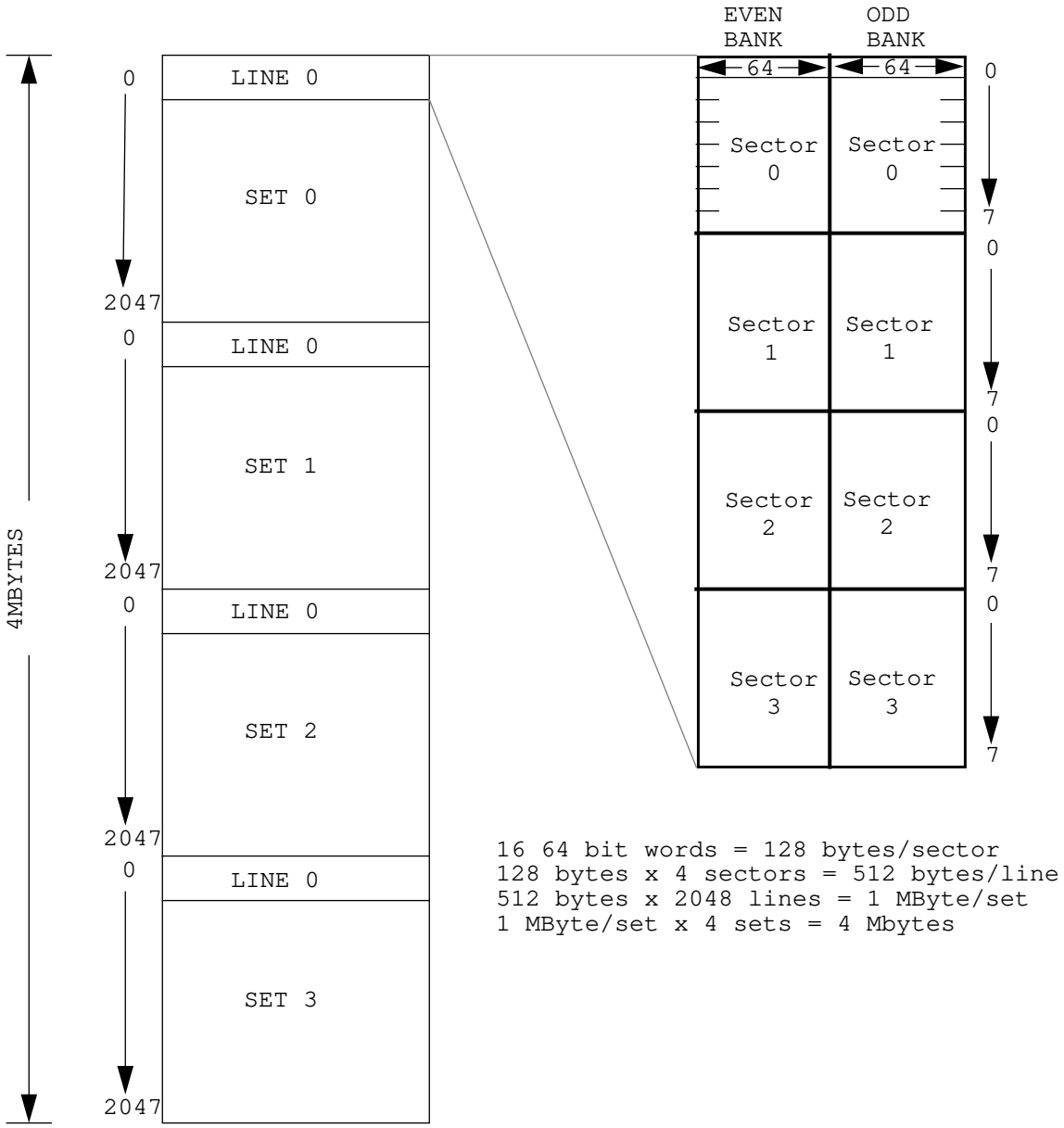

Figure 17: 4 MByte Streaming Cache Organization

2.4.3 DATA RAM BUS INTERFACE

The even bank of Data RAM's interfaces to the even Tag RAM. The odd bank of Data RAM's interfaces to the odd Tag RAM. Both banks interface to the IU and the FPU. The following sections define the external pinout of the even and odd data RAM's and are divided into specific component interfaces.

2.4.3.1 EVEN DATA RAM TO INTEGER UNIT

LDE<63:0> (Load Data Even) Active High Output

Load data bus between the IU and the even bank of the streaming cache. This bus is unidirectional and drives only load input data to the IU. Refer to the IU pin descriptions in section 2.1.12.4 for more infomration.

WEE#<1:0> (Write Enable Even) Active Low Input

The data RAM's can be written either by the IU or by the CC. The CC writes to the data RAM's through the TBus. The IU provides a total of four write enables, two for the odd bank and two for the even bank. Streaming cache designs must allow for 32 bit writeability. WEE#<1> is connnected to the WE# pin of the upper 32 bits of data for the even bank. WEE#<0> is connnected to the WE# pin of the lower 32 bits of data for the even bank. Refer to the IU pin descriptions in section 2.1.12.4 for more infomration.

ADDRE<17:0> (Address Even) Active High Input

These pins form the address to the even bank of the streaming cache. Bits 17:0 of the IU connect directly to address pins 17:0 of both SIMmodules of the even bank .

2.4.3.2 ODD DATA RAM TO INTEGER UNIT

LDO<63:0> (Load Data Odd) Active High Output

Load data bus between the IU and the odd bank of the streaming cache. This bus is unidirectional and drives only load input data to the IU. Refer to the IU pin descriptions in section 2.1.12.5 for more infomration.

ADDRO<17:0> (Address Odd) Active High Input

These pins form the address to the odd bank of the streaming cache. Bits 17:0 of the IU connect directly to address pins 17:0 of both SIM modules of the odd bank .

WEO#<1:0> (Write Enable Odd) Active Low Input

The data RAM's can be written either by the IU or by the CC. The CC writes to the data RAM's through the TBus. The IU provides a total of four write enables, two for the odd bank and two for the even bank. Streaming cache designs must allow for 32 bit writeability. WEO#<1> is connnected to the WE# pin of the upper 32 bits of data for the odd bank. WEO#<0> is connnected to the WE# pin of the lower 32 bits of data for the odd bank. Refer to the IU pin descriptions in section 2.1.12.4 for more infomration.

2.4.3.3 EVEN DATA RAM TO EVEN TAG RAM

DATSAXE<1:0>, (Data Set Address X Even) Active High Input
DATSAYE<1:0>, (Data Set Address Y Even) Active High Input
DATSAWE#<1:0>, (Data Set Address W Even) Active Low Input
DATSAZE#<1:0>, (Data Set Address Z Even) Active Low Input

These two bits form the upper two address bits for the Even bank of Data RAM's. Each of the four sets of DATSA pins contains the exact same information and are buffered versions of one another. Two of the sets are inverted.

2.4.3.4 ODD DATA RAM TO ODD TAG RAM

DATSAXO<1:0>, (Data Set Address X Odd) Active High Input
DATSAYO<1:0>, (Data Set Address Y Odd) Active High Input
DATSAWO#<1:0>, (Data Set Address W Odd) Active Low Input
DATSAZO#<1:0>, (Data Set Address Z Odd) Active Low Input

These two bits form the upper two address bits for the Odd bank of Data RAM's. Each of the four sets of DATSA pins contains the exact same information and are buffered versions of one another. Two of the sets are inverted.

2.4.3.5 EVEN DATA RAM TO FLOATING POINT UNIT

LD<63:0> (Load Data) Active High Output

The LD<63:0> pins of the even bank of the streaming cache connect directly to the LDE<63:0> busses of both the IU and the FPU.

LPE<3:0> (Load Parity Even) Active High Input

Load parity bus between the FPU and the even bank of the streaming cache. This bus is unidirectional and drives load parity data only. Even bank store parity data is transferred via the dedicated store parity bus. There are two 32 bit wide SIM modules on the even bank. LPE<3:0> of the FPU are connected directly to the LP<3:0> pins of the upper 32 bit even data RAM module. Parity is checked and generated in 16-bit quantities. Hence only the lower two parity bits (LP<1:0>) are currently used. LP<3:2> are routed to the SIM module connector but are not connected.

SD<63:0> (Store Data) Active High Output

The SD<63:0> pins of the even bank of the streaming cache connect directly to the SDE<63:0> pins of the FPU. There is no direct store data interface between the IU and the streaming cache data RAM's. Integer stores are handled through the FPU via the TBus. SD<63:0> are used on floating point stores, integer stores where the data is sent across the TBus to the FPU and out onto the SD pins, and streaming cache reads and writes to main memory. Both streaming cache writes to main memory are handled by the CC. The CC reads the data out of the streaming cache. Data is placed on the LD<63:0> pins. Control of the data through the FPU is handled by the CC. Control signals are asserted which allow the data to pass directly from the LDE bus onto the SDE bus and out to main memory.

SPE<3:0> (Store Parity Even) Active High Input

Store parity bus between the FPU and the even bank of the streaming cache. This bus is unidirectional and accepts store parity data only. Even bank load data is transferred via the dedicated load parity bus. There are two 32 bit wide SIM modules on the even bank. SPE<3:0> of the FPU are connected directly to the SP<3:0> pins of the upper 32 bit even data RAM module. Parity is checked and generated in 16-bit quantities. Hence only the lower two parity bits (SP<1:0>) are currently used. SP<3:2> are routed to the SIM module connector but are not connected.

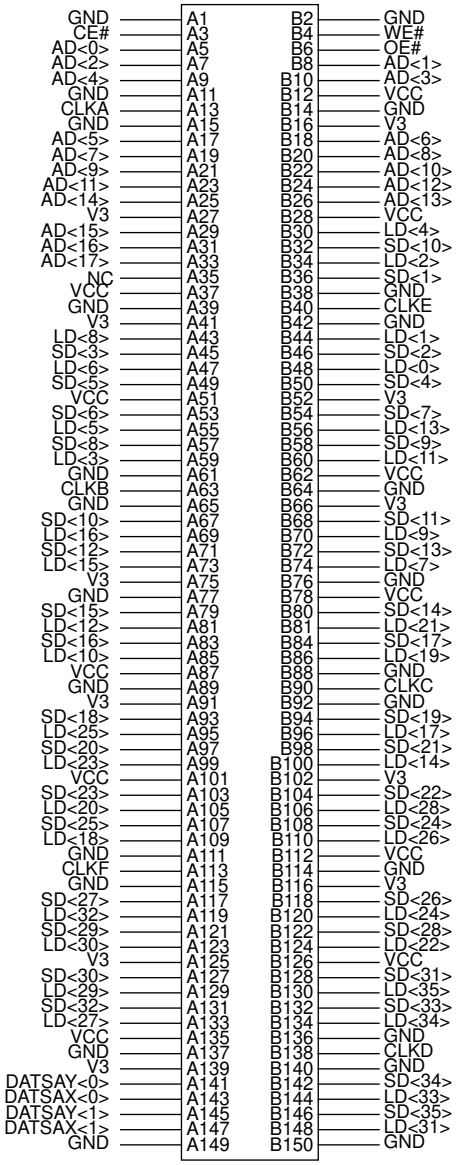

Figure 18: Streaming Cache Data RAM SIM Module

2.4.3.6 ODD DATA RAM TO FLOATING POINT

LDO<63:0> (Load Data Odd) Active High Output
The LD<63:0> pins of the odd bank of the streaming cache connect directly to the LDO<63:0> busses of both the IU and the FPU.

SD<63:0> (Store Data) Active High Output
The SD<63:0> pins of the odd bank of the streaming cache connect directly to the SDO<63:0> pins of the FPU. There is no direct store data interface between the IU and the Data RAM's. Integer stores are handled through the FPU via the TBus. SD<63:0> are used on floating point stores, integer stores where the data is sent across the TBus to the FPU and out onto the SD pins, and streaming cache reads and writes to main memory. Both s treaming cache reads and writes to mian memory are handled by the CC. The CC reads the data out of the streaming cache. Data is placed on the LD<63:0> pins. Control of the data through the FPU is handled by the CC. Control signals are asserted which allow the data to pass directly from the LDO bus onto the SDO bus and out to main memory.

2.4.4 DATA RAM PACKAGE PINOUT

Figure 18 above shows a pinout of the Data RAM SIM module.

3.0 RESPONSIBILITIES OF THE CACHE CONTROLLER

This section is intended to be an overview of the responsibilities of the cache controller and is not intended to be used as a design guide. The processor to cache interface is far more complex than what is described here. A more in-depth discussion of the processor/cache interface will be available in the forthcoming TFP User's Manual.

3.1 CACHE CONTROLLER OVERVIEW

The TFP Microprocessor Chip Set has large first and second level caches which minimize the need for interfacing to external main memory. Main memory is typically slow and when accessed frequently can degrade the overall performance of the processor. To maximize performance, the Integer Unit interfaces only to the 16 KByte on-chip cache and the streaming cache. The Floating Point Unit interfaces only to streaming cache. Neither device inititates cycles or interfaces directly to the main memory. However, there are times when interfacing to the external main memory is necessary.

The interface to external main memory is the responsibility of the Cache Controller. The Cache Controller (CC) performs the following duties:

1) Fetching data from main memory on a streaming cache miss.
2) The write-back of dirty data from the streaming cache to main memory.
3) Modification of streaming cache tags, state, and virtual synonym information in the Tag RAM.
4) Invalidation of the Data cache when the streaming cache is modified.
5) All coherence issues between streaming caches in a multiprocessor environment.
6) Filtering of coherence activity so that the Integer Unit is protected from unnecessary interruptions.
7) Checking of the store address queue so that it appears to the system as an extension of the streaming cache.
8) Local registers for interrupt prioritizing and management.

The Cache Controller is also required to handle booting from a ROM device and transferrence of ROM data into the streaming cache, control signal generation for the data buffers which interface to external main memory, and maintaining a bus copy in a third Tag RAM. The third Tag RAM is recommended and is used to monitor snooping activity on the bus. The two Tag RAM's which interface to the two banks of the streaming cache require all of their bandwidth to maintain a single cycle access rate. Use of the third Tag RAM eliminates these two Tag RAM's from having to monitor bus activity. This enhances performance and reduces Tag RAM overhead.

On a streaming cache miss, the IU transfers control of the bus to the CC. When control of the bus is transferred is determined by the CC state machine which has 5 full states and 4 transition states. While the CC is in control of the bus it must:

1) Update the streaming cache data.
2) Update the Tag RAM's.
3) Update the Store Address Queue.
4) Update the D-cache Valid bits.
5) Update the Interrupt pending flag.
6) Update the Sequential Update flag.

It is only after these duties are completed that control of the

bus is returned to the IU.

3..1.1 STREAMING CACHE DATA MANAGEMENT

On IU misses to the streaming cache the IU informs the CC that the requested data was not available. Control of the bus is then transferred to the CC. Since neither the IU or the FPU communicate directly with external main memory, it is the responsiblilty of the CC to fetch the requested data from main memory and place it in the streaming cache. Once the cycle is completed the CC relinquishes control of the bus back to the IU. The IU then fetches the requested data from the streaming cache and execution resumes.

In addition to streaming cache misses, the CC is responsible for monitoring the cache coherency attributes of each line in the cache. This is done via the 16 bit Dirty Bit RAM portion of the Tag RAM as well as the Tag RAM itself. If the state of the line in the cache changes, the state information in the Tag RAM must also be updated to reflect the change. If the state of the line remains the same and only the modified status changes, only the dirty bit is accessed. State information remains the same. Changing the state information is the responsibility of the CC. Updating of the dirty bit RAM is done by the IU.

Since the first level cache is write-through, all writes by the IU to the first level Data cache are also written out to the streaming cache. If a write is executed to a line which has already been modified, the CC will tell the IU to halt the cycle and take control of the bus. The modified data is then written out to main memory and control of the bus given back to the IU.

3.1.2 TAG RAM MANAGEMENT

The Cache Controller is responsible for monitoring and updating of the Tag RAM. The IU performs only Look-up cycles to the Tag RAM. These are done when the IU wishes to read or write the streaming cache and desires to know which set of the streaming cache the data resides in. Read and write cycles to the Tag RAM are performed by the CC.

The CC can write either Tag information, or state and virtual synonym information to the Tag RAM in a given cycle. Refer to figure 13. The Tag address is multiplexed with either tag or state and virtual synonym information. When the tag address is updated, the 20 bit tag bus contains all address bits. When the state and virtual synonym information is updated, the tag bus conforms to a specific bit orientation. The bit orientation changes for state and virtual synonym read cycles. Whenever there is a streaming cache miss, or when a line of the streaming cache is to be written out to main memory, the Tag RAM information must be updated to reflect this change. The CC must supply all necessary control signals to the Tag RAM.

3.1.3 TBUS INTERFACE

During normal operation the TBus is used for transferring information between the IU and the FPU. During this time the CC interfaces to the main memory backplane interface bus, as well as monitors the third Tag RAM and the hit/miss status of IU to streaming cache cycles. When a streaming cache miss or a coherence operation on the bus occurs, the TBus is used to transfer information between the IU and the CC. The bit orientation and definition of the Tbus changes depending on whether the IU or the CC is driving the TBus. The TBus for the system is 80 bits wide. However, the uppermost 8 bits are dedicated between the IU and the FPU. The IU to CC TBus is 72 bits.

3.1.3.1 IU TO CC TBUS PROTOCOL

When the IU is in control of the TBus, the orientation and definition of the TBus is as follows:

FIELD	WIDTH	TBUS BITS
Reserved	12	71:60
Command	4	59:56
Size	3	55:53
Coherence Protocol	3	52:50
No Match	2	49:48
Set Address	2	47:46
State	2	45:44
Virtual Synonym	4	43:40
Physical Address	40	39:0

The **Command** field is a four bit field which encodes the type of memory access the IU was doing which caused the transfer. Also included in the command field are the Cacheable and Non-cacheable coherence attributes of each access type.

The **Size** field is an encoded three bit value which indicates the size of the operation, one through eight bytes, for non-cacheable operations.

The **Coherence** Protocol field is a three bit field used to indicate the coherence atributes in the page table entry for the access. It is used to determine whether the access is cacheable, and what type of protocol to follow in filling the streaming cache.

The **No Match** field is a two bit field which reflects then status of the MATCH pin which was returned to the IU by the Tag RAM. The IU samples the various control signals from the Tag RAM and encodes the status of the Tag RAM lookup in the No-Match field of the TBus. The No Match

field also contains the status of whether the virtual synonym returned from the Tag RAM matched the virtual address bits from the original address. The number of virtual synonym bits actually checked depends on the size of the Data Cache. If the No-Match field indicates both a Tag address match and a virtual synonym match, the CC simply fetches and adds a sector to the existing line. If the field indicates a Tag address match and a virtual synonym mismatch, the CC removes the entire line, invalidates the corresponding entry in the Data Cache, fetches the needed sector and places it in the streaming cache (in the same set). If there is no Tag address match, indicating a miss, the CC removes the entire line at a set which the CC chooses, invalidates the Data Cache entry for athat line, fetches the needed sector and places it in the streaming cache at that set.

The **Set Address** field is a two bit field which reflects the status of the Tag RAM lookup and distinguishes which of the four sets in the Tag RAM the address compared.

The **State** field is a two bit field which indicates the coherency status for the given access; Invalid, Shared, or Exclusive.

The **Virtual Synonym** field is a four bit field which represents the status of virtual address bits 15:12 of the access which caused the operation.

The **Physical Address** is a 40 bit field and is used directly in processing the operation.

3.1.3.2 CC TO IU TBUS PROTOCOL

When the CC is in control of the Tbus, the orientation and definition of the Tbus bits is as follows:

FIELD	WIDTH	TBUS BITS
Reserved	4	71:68
Function	4	67:64
Data RAM Write En.	4	63:60
Data Ram Ext. Set Adr.	2	59:58
Data RAM Addr<21:4>	18	57:40
Tag RAM Addr<39:5>	35	39:5
Reserved	1	4
Virtual Synonym	4	3:0

The **Function** Field is a four bit field and is the only field by which the CC changes the internal state of the IU instead of just the Tag RAM's or the streaming cache RAM's. There are a number of different operations which can be performed based on the state of the function field. The function field controls such operations as invalidation of the first level Data Cache, instructing the IU to empty the floating point store address and store data queues by briefly returning the TBus to the IU for this operation, interrupt status, and signal generation for reading the Tag RAM.

The **Data RAM Write Enable** field is a four bit field which contains the write enables for the streaming cache. Assertion of each write enable controls the write of a word (4 bytes).

The **Data RAM External Set Address** field is a two bit field which encodes which of the four sets of the streaming cache the data is to be written.

The **Data RAM Address** field is an 18 bit field containing address bit 21:4 and forms the address for the streaming cache RAM's. Address bits 3 and 2 determine which write enables are asserted on the TBus. Address bits <1:0> are not generated by the CC as 32 bits is the smallest writeable quantity.

The **Tag RAM Address** field is a 35 bit field used to read and write the even and odd Tag RAM's address. The state and virtual synonym information is also written using these bits. The size of the field is large enough to support a 16 MByte streaming cache configuration.

The **Virtual Synonym** field is a 4 bit field containing address bits 15:12. When the Data Cache of the IU is invalidated bits 15:12 reflect the virtual address.These bits are concatenated with bits 11:5 of the Tag Ram Address field to