Meiko

To Simon Brown
Company Meiko
From Andrew Spence
Company 031 650 6555
Tel No.
No. of Pages 5

FAX
Post-it Fax Note
Ref No: 7688

1

GENERAL DESCRIPTION

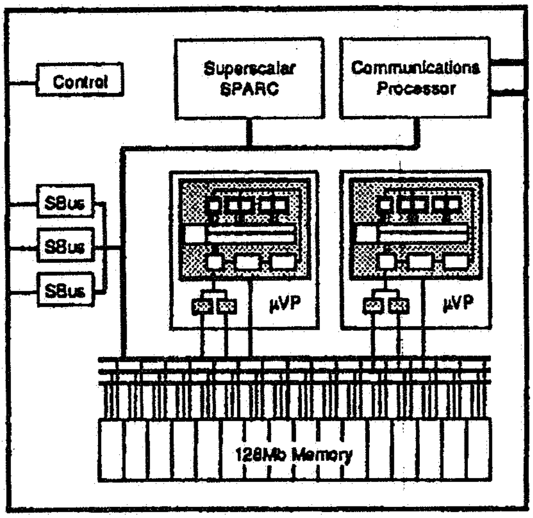
This document describes the architecture of the CS-2 vector element (MK403). It briefly describes the internal architecture of the Fujitsu μVP and the compilation strategy used to exploit the combined resources of the SPARC and multiple μVP processors.

For more details of the workings of the μVP see the "Programmers Reference Manual".

1.1 *MK403 Overview*

The CS-2 vector element incorporates a 40MHz Superscalar SPARC, a Meiko Elan Communications Processor and 2 Fujitsu μVP vector processors. All processors have access to the memory system via 3 memory ports, two of which are used by the vector processors and the third by the SPARC and Elan which share an MBus.

Fig. 1.1 CS-2 Vector Processing Element



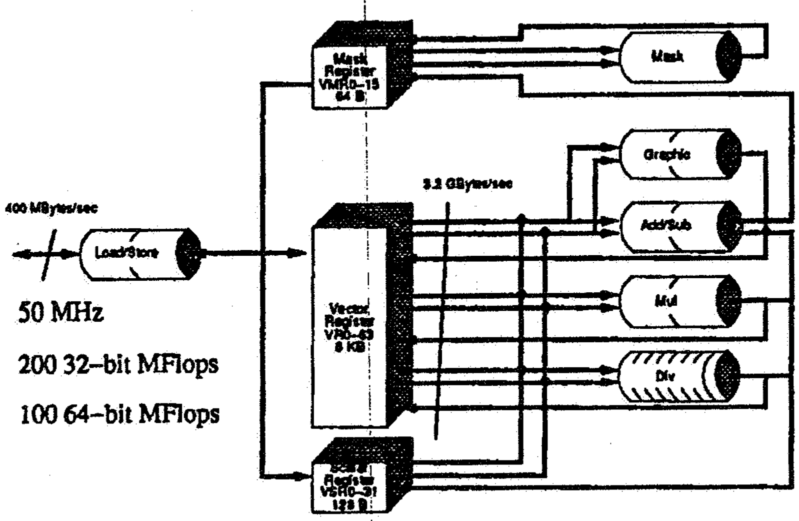
The memory system is implemented as 16 independent banks, with a (current) total capacity of 128 MBytes. Memory bandwidth for each of the 3 ports is 1.2 GBytes/s, with a total bandwidth of 3.2 GBytes/s.

External I/O support is provided through 3 SBus interface slots – primarily used for disk controllers, but capable of supporting network interfaces and graphics cards.

1.1.1 *μVP Vector Processor*

The μVP operates with a 50MHz (20ns) clock. It has a vector register architecture with 8 KBytes of vector registers, configurable as between 8 and 64 vectors each of 16-128 64-bit registers (see below). In addition there are 32 scalar registers and a set of vector mask registers whose format tracks that of the vector registers.

Fig. 1.2 μVP Vector Processor



Configuration of the μVP vector and mask registers:

Precision	Length	Number of registers
Single	32	64
Single	64	32
Single	128	16
Single	256	8
Double	16	64
Double	32	32
Double	64	16
Double	128	8

The μVP has seperate pipes for floating point multiply, floating point add, floating point divide, and integer operations. The floating multiply and add pipes can each deliver one double precision (64 bit) or two single precision (32 bit) IEEE format result(s) on every clock, giving a maximum theoretical perfomance of 100 MFLOPS/s double precision and 200 MFLOPS/s single precision; the divide pipe can simultaneously deliver an extra 6 MFLOPS/s in either single or double precision. Both the add and multiply pipes have the low latency (pipe depth) of two cycles (40ns), with one extra cycle being required to read and one to write the vector register file.

The vector register elements are scoreboarded, so that chaining between input and output operands occurs wherever possible without requiring explicit compiler or programmer intervention.

The μVP has a single load/store pipe which is used for accessing the memory system. This is a 64 bit interface which can generate four addresses on consecutive clock cycles before stalling for the returned data. Once the data is present a 64 bit word can be transferred on each clock cycle, giving a maximum bandwidth of 400 MBytes/s.

The instruction set includes masked vector operations, compressions (sum, maxval, maxindex, minval, minindex), vector compress under mask and expand under mask operations, as well as logical operations on integers and mask registers and conditional branches. Vector loads and stores can be perfomed with strides and under mask, as well as with an index vector ("indirect"). For further information about the μVP instruction set the *μVP Programmers reference Manual.*

1.1.2 *Superscalar SPARC Processor*

The MK403 uses SPARC MBus processor modules. It is generally populated with a 36 or 40HMz Viking SPARC, but other standard modules can be used.

The Superscalar SPARC has two independent integer ALUs which can execute separate arithmetic operations or can be cascaded so that the processor can execute two dependent instructions in the same cycle. It has instruction issue logic which can issue up to three instructions on the same cycle. Load and stores operations of all data types to the on chip 16 KBytes data cache occur in a single cycle. The floating point unit can execute multiply and add instructions simultaneously, though only one floating point instruction can be issued per cycle.

1.1.3 *Memory System*

The Superscalar SPARC processors and Elan communication processor are connected to a standard 40MHz MBus. The vector processors and MBus are connected to a 16 bank memory system, each bank providing 64 bits of user data (78 bits including error checking and correction, implemented using 20 by 4 bit DRAMs with two bits unused). Error detection and correction is implemented on each half word (32 bits), allowing write access to 32 bit (ANSI-IEEE 754-1985 `single`) values to be performed at full speed, without requiring a read modify write cycle.

Each bank of memory maintains a currently open DRAM page within which accesses may be performed at full speed. This corresponds to a size within the bank of 8 KBytes, giving 128 KBytes total for the 16 banks. When an access is required outside the currently open page a penalty of 6 cycles is incurred to close the previous page, and open the new one.

Refresh cycles are performed on all banks within a few clock cycles of each other, thus allowing the cost of re-opening the banks to be pipelined (since the μVP can issue four addresses before stalling for the data from the first), and reducing the overhead of refresh to a few percent of memory bandwidth.

The memory system is clocked at the same speed as the μVP processors (50 MHz), and accesses from the 40 MHz MBus are transferred into the higher speed clock domain. When accessing within an open page each memory bank can accept a new address every two cycles (40ns), and replies with the data four cycles (80ns) later, giving a bandwidth of 8 Bytes every two cycles (40ns), that is 200 MBytes/s. Since there are 16 banks, the total memory system bandwidth is thus 3.2 GBytes/s.

Each μVP can issue a memory request every cycle (20ns), and can issue 4 addresses before it requires data to be returned. In the absence of bank contention (which will be discussed below), after a start up latency of four cycles, these requests can be satisfied as fast as they are issued, giving each μVP a steady state bandwidth of 8 Bytes every 20ns, that is 400 MBytes/s.

Since each bank can accept a new address every two cycles (40ns), but the μVP can generate an address every cycle (20ns) there is the possibility of bank contention if the μVP generated repeated accesses to the same bank. With a simple linear mapping of addresses to banks, this would occur for all strides which are multiples of 16 (for 64 bit double precision accesses). Such an access pattern would then see only one half of the normal bandwidth, that is 200 MBytes/s. All other strides achieve full bandwidth.

To ameliorate this problem as well as allowing the straightforward linear mapping of addresses to banks, Meiko also provide the option (through the choice of the physical addresses which are used to map the memory into user space) of scrambling the allocation of addresses to memory banks. The mapping function has been chosen to guarantee that accesses on "important" strides (1, 2, 4, 8, 16, 32) achieve full performance. Access on other strides may see reduced performance, but there are no strides within the open pages which see the pathological reduction to one half of the available bandwidth.