MOTOROLA

Performance Review of the MVME6100

William Dennen

1 GENERAL

This paper provides product performance benchmarking and analysis of the MVME6100 single board computer (SBC).

1.1 Introduction

MVME6100 is the first Motorola single-board computer developed as part of the VME Renaissance program. It has been designed to provide "flagship" computing performance in a balanced fashion for demanding VMEbus applications. This SBC is targeted to meet the needs of OEMs servicing the defense and aerospace, industrial automation and medical imaging market segments. Figure 1 shows a block diagram of the board.

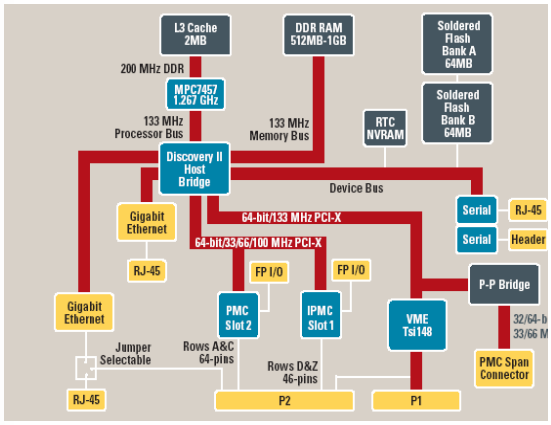

Figure 1 - Block Diagram of MVME6100

The heart of the MVME6100 single board computer is the Discovery II™ (Disco 2) host bridge from Marvell. The Disco 2 provides interconnectivity between the processor, DDR memory, PCI-X, and device bus circuitry. In addition it also provides two gigabit Ethernet interfaces, integrated SRAM memory, and an integrated DMA (IDMA) engine for moving blocks of data between PCI-X devices and memory.

The processing function of the MVME6100 is delivered by the Motorola MPC7457 PowerPC® processor. This is a 1.267GHz, Altivec™-enabled G4 core with integrated primary and secondary caches and 2MB of backside SRAM.

The MVME6100 also incorporates the Tundra Tsi148™ VMEbus interface. The Tsi148™ offers 2eSST protocol, allowing the VMEbus to run at bandwidths exceeding 250MB/s. This additional data movement capability, combined with processing and communications features, will fuel the deployment of the MVME6100 into applications previously unattainable by VME-based boards.

1.2 Note on Benchmarks

Whenever performance benchmarks are examined, the following disclaimer must be made:

BENCHMARK COMPARISONS ARE EASY TO DISTORT AND MISREAD

The reasons for this are manifold. Here are a few:

- **Benchmark version** – most benchmark programs have undergone several revisions over time. Comparison of results from differing versions may lead to erroneous conclusions.
- **Compiler** – compiler selection and optimization choices can account for as much as a 4x improvement on some benchmarks. In order to ensure fair comparisons, compilers most like those available to the customer should be used. For any given compiler, optimizations should remain consistent.
- **Library code** – differences in library code can account for as much as a 2x improvement in some benchmarks. The libraries used should be those immediately available to the customer. Exceptions should be clearly stated.
- **O/S** – operating system overhead, driver efficiency, and task priority can account for up to a 1.5X improvement in benchmark performance. Benchmark comparisons must be performed utilizing the same operating system in a manner consistent with customer use.

- **Hardware Optimization/Tuning** – hardware optimization and tuning can account for substantial performance differences between benchmark runs. Benchmarks should be performed with hardware tuned in a manner consistent with customer use. Tuned systems should not be compared against untuned systems.
- **Not Equivalent -** Benchmarks that are not performed in a manner that is consistent with end-use. This might include:
  - Overclocking – running hardware at speeds outside its intended operating range.
  - Unstable tuning – tuning hardware in a manner that provides favorable results but is unsuitable for stable application usage.
  - Unrealistic loading – removing operating system components that would normally be required.
- **Equivalent but not Identical** – a similar benchmark is run that differs from the norm. An example of this would be comparing the Linpack algorithms after converting and compiling them in C, to the original Linpack code written in FORTRAN. Another example of this is the commonly quoted Dhrystone benchmark provided by SandraSoft for Intel processors. At the time of the writing of this paper, SandraSoft's benchmark was similar but not identical to the standard Dhrystone.

This being said, every effort has been made to represent the data contained in this document in a fair and accurate fashion. Operating system and compiler selections are those likely to be used by the customer, and hardware tuning closely matches those of the product as it will be shipped (exceptions are noted). In addition, benchmarks for the previous generation product (MVME5500) are also provided for comparative purposes.

Best practice dictates that benchmarks used for comparison be provided by a single, unbiased party. The reader should use caution when making comparisons with results from other sources.

2 PROCESSOR PERFORMANCE

This section outlines the performance of the processing subsystem on the MVME6100 single board computer.

2.1 Overview

The MVME6100 SBC utilizes the Motorola MPC7457 PowerPC® embedded 32-bit processor. The MPC7457 features a high-frequency (1.267GHz) superscalar G4 core which is capable of issuing up to four instructions per clock cycle within its eleven independent execution units. Figure 2 shows a block diagram of the processor.

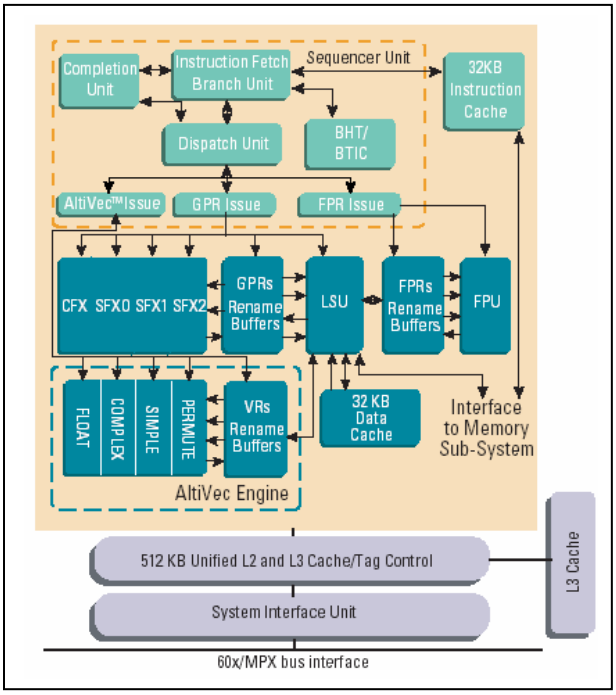

Figure 2 - MPC7457 Processor Block Diagram

Improvements that the MPC7457 offers over its predecessor (MPC7455) are lower power and higher clock frequencies. In addition the MPC7457 doubles the size of the integrated L2 from 256K bytes to 512K bytes. For many applications this will result in improved performance due to greater likelihood that program data will reside in the lower latency cache interface.

Although the 7457 allows for up to 2M bytes of SDRAM to be connected to the backside cache bus, the current revision of the processor only

allows 1MB of this memory to be utilized as L3 cache. This chip erratum effectively reduces the amount of level-three cache by half when compared to the 7455. In some applications performance degradation may result. Specific impacts of this change are dependent upon application memory footprint and memory usage patterns.

2.2 Results

Table 1, below, shows a side-by-side comparison of the processing capabilities of the Motorola MVME6100 and the MVME5500 single board computer products.

Table 1 - Processor Performance

Description	OS	6100	5500	Units	Ratio
General					
Processor Type		7457	7455		
CPU Clock Speed		1267	1000	MHz	1.27
CPU Bus Speed		133	133	MHz	1.00
Bus Mode		MPX	MPX		
Bus BW (theoretical)		1064	1064	MB/s	-
Benchmarks					
Dhrystone 2.1*	**V**	3003	2310	DMIPS	1.30
SpecCPU2000	**L**				
SpecInt_base2000		433	250		1.73
SpecFp_base2000		249	166		1.50
ByteMark	**L**				
Integer Index		7.257	5.882		1.23
Fp Index		8.227	6.506		1.26
FSB Bandwidth	**V**				
Read		946	946		1.00
Write		1064	1064		1.00
LMBench	**L**				
Context Sw 16p/16k		4.79	11.5	uS	2.40
Context Sw 16p/64k		68.5	84.8	uS	1.24
exec proc		807	1687	uS	2.09
sh proc		2797	5863	uS	2.10

* Dhrystone testing was performed consistent with Motorola testing methodologies.
** FSB bandwidth performance shown is for streaming access. Performance to non-streamed regions will be lower.

Definitions of the fields within this table are as follows:

Description: The description of the parameter or benchmark results being reported.

OS: The operating system the test was run under. "V" corresponds to Tornado 2.2.1 / VxWorks 5.5.1 from Wind River Systems. "L" stands for Linux version 2.4.20.

6100/5500: Performance characteristics or results for the corresponding SBC.

Units: The units of the reported result.

Ratio: the ratio of the performance of the MVME6100 as compared to the MVME5500. The ratio directly relates to performance improvement (e.g. 2.0 represents a two times performance improvement). Higher numbers are better. Numbers under 1.00 represent performance degradation.

2.3 Analysis

Since the processing clock ratio between the MVME6100 and the MVME5500 is 1.27, computation bound benchmarks will show at most a 27 percent improvement between the two products. Memory-bound applications would be expected to run at parity between the two products since the processor front-side bus is clocked at 133 MHz on both boards. Performance of applications that contain a mixture of memory and processing requirements will be influenced by the nuances of cache differences.

The performance benchmarking results largely reinforce this assessment. Dhrystone 2.1, which runs entirely out of L1 cache, reflects almost exactly the processor clock ratio. ByteMark integer and floating point ratios also show this trend.

SpecCPU utilizes large blocks of memory and heavy computation. Here, the MVME6100 shows an increased performance edge that can be explained by the larger L2 cache size. LMBench provided much better results than expected on all but the 16p/64k context switch test.

A special comment needs to be made regarding the MPC bus bandwidth. Although the Discovery II CPU bus interface is capable of delivering the results shown in Table 1, some pathological cases may perform much worse. The root of this degradation is due to the fact that MPX bus streaming does not occur.

Figure 3 shows two MPX bus traces for sequential memory reads taken from a Discovery II based platform with a pathological piece of code. The bottom trace results from sequential access of memory locations within the Discovery II SRAM region. Regular data beats result in efficient utilization of the MPX bus and near-theoretical operation. The top trace results from sequential access to addresses within the DDR subsystem. Irregular data beats

and stalling point to inefficient bus utilization. In fact, measured bus performance is approximately 55% of the theoretical performance.

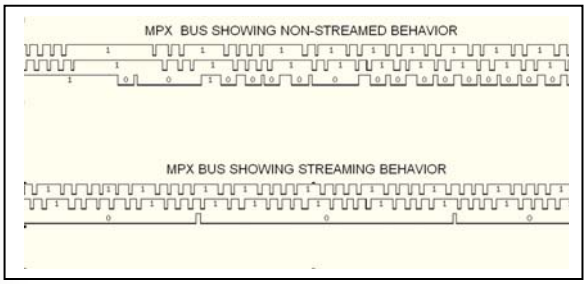

Figure 3- MPX bus traces showing streaming and non-streaming behavior

This performance degradation was unexpected. Since the DDR subsystem runs at twice the clock rate of the MPX bus, there should be more than enough bandwidth to keep up with the processor needs. Preliminary investigation of this issue points to interaction between Discovery II processor-to-memory latency and the size of the posted read queue within the G4 processor. The precise interaction between these parameters has not yet been fully characterized; however, this phenomenon is not expected to occur in user code.

3 MEMORY/CACHE PERFORMANCE

This section outlines the performance of the memory and cache subsystems on the MVME6100 single board computer.

3.1 Overview

Memory and cache subsystems when viewed as a whole are responsible for providing data to the processor's execution units and I/O subsystems in an on-demand fashion with little or no delay. In general, it is not cost effective to provide an entire memory subsystem with latencies and throughput to do this consistently; however, reasonable cache architectures greatly reduce this burden. Caching strategies take advantage of temporal and spatial data access patterns in order to "predict" what data is likely to be required and store it in a smaller amount of high performance memory.

The MPC7457 provides three levels of cache support: separate 32K byte primary instruction and data caches, a unified 512K byte second level cache, and an interface for another 1M byte of backside tertiary cache. Both the primary and secondary caches reside on the PowerPC® processor die for fast, low latency access. The level-3 cache tag RAMs are also included on the processor die while the SRAM devices are connected via a 200MHz dedicated cache bus.

A 266 MHz DDR main-memory subsystem is provided by the Marvel Discovery II host bridge chip. The main memory interface is connected via an internal fabric to the CPU and PCI-X buses as well as Discovery II integrated peripherals. The Disco 2 is also responsible for providing arbitration between the different memory requestors and enforcing coherency with the processor cache subsystem.

3.2 Results

Table 2, shows a side-by-side comparison of the memory performance of the MVME6100 and the MVME5500 single board computer products.

Table 2 - Memory/Cache Performance

Description	OS	6100	5500	Units	Ratio
General					
L1 Cache Size		32/32	32/32	kB	-
L2 Cache Size		512	256	kB	2.00
L3 Cache Size		2048*	2048	MB	0.50
Max Memory Size		1024	1024	MB	1.00
DRAM Technology		DDR	SDRAM		-
Main Memory Speed		133	133	MHz	1.00
BW (theoretical)		2128	1064	MB/s	2.00
Benchmarks					
Bytemark	**L**				
Memory Index		6.711	5.311		1.26
LMBench	**L**				
Bcopy (libc)		161	128	MB/S	1.26
Memory Read		232	210	MB/S	1.10
Memory Write		360	262	MB/S	1.37
Latency to RAM		188	186	nS	1.01
MemBench	**V**				
Stream from RAM		565	604	MB/S	0.94
Stream to RAM		193	173	MB/S	1.12
Stream from L3		2333	2146	MB/S	1.09
Stream to L3		800	590	MB/S	1.36
Stream from L2		3805	4334	MB/S	0.88
Stream to L2		1925	2744	MB/S	0.70
Splatter from RAM		113	130	MB/S	0.87
Splatter to RAM		118	113	MB/S	1.04

* Due to chip errata at the time of writing this article, the MPC7457 processor is limited to 1MB of L3 cache, thus the .5 ratio. The remainder may be used for private memory.

Definitions for the fields in the table match those of Table 1.

3.3 Analysis

Analysis of memory performance is inherently difficult due to the complexities of predicting cache performance. A corner-case approach

seeks to quantify the best and worst expected latency and throughput of each memory region and then make extrapolations based on the results. Unfortunately the output is difficult to tie back to expected application performance. Best-case scenarios are almost always overly optimistic, and worst-case situations are rarely seen.

Another approach to solving the memory performance analysis problem is to perform tests that generate "typical" memory usage. As long as the user's application accesses memory in a similar fashion, performance gains should be comparable. The problem with this, of course, is knowing the application's memory access patterns *a priori* in order to select a reasonable test pattern. Even small differences in test assumptions can lead to large differences in results due to the nature of cache structure and utilization.

The approach employed by this paper was to perform both types of testing. This section relates to tests that are geared directly at the memory/cache subsystems. However, since other subsystems utilize memory, their results will be colored by memory performance as well. A good example of this coloring can be seen in the SpecCPU2000 results shown in the previous section. Spec performance reflects both processing and memory capabilities.

Starting with the MemBench results, one would expect streaming data transactions to same memory regions to be related to the relevant clock ratio. L2 cache performance would be expected to have a ratio equal to the difference in the processor clock frequency. L3 would be related to the backside cache bus clock frequency, and main memory performance would be related to the ratio between the smaller of the memory throughput or the front-side bus throughput on each board. Comparing the MVME6100 to the MVME5500, these ratios are 1.267, 1.06, and 1.00 respectively. The measurements roughly correlate and the discrepancies are easily within the uncertainty of measurement.

The MemBench Splatter routine tests the system's ability to service random data accesses. This test will uncover degradation due to paging, arbitration and other latency issues. Since the cache and memory subsystems are designed with the assumption that some spatial/temporal correlation exists between data accesses, random data access constitutes a worst-case pattern for memory usage. Since streaming accesses are not occurring, DDR performance would be expected to be similar or slightly worse to an SDRAM memory subsystem at the same clock frequency. The test data bears this out.

ByteMark and LMBench ratios all fall within the realm of expected results, slightly favoring the MVME6100 single board computer over the MVME5500.

.

4 PCI/PCI-X PERFORMANCE

This section outlines the performance of the PCI/PCI-X on the MVME6100 single board computer and estimates the performance associated with the PMCspan PMC expansion module.

4.1 Overview

The MVME6100 utilizes two PCI-X buses to provide peripheral interconnect to expansion devices and backplane (through an additional bridge). PCI-X was developed in 1999 as a follow-on to the already pervasive PCI bus. Performance enhancements that PCI-X offers are:

- Higher clock frequencies than PCI
- Tighter wait state and disconnect rules for better bus utilization
- Delayed transactions in PCI replaced by split transactions in PCI-X for better bus utilization.

PCI-X functionality on the MVME6100 is supplied by the Discovery II host bridge chip. The Disco 2 provides two independent PCI-X bus interfaces each capable of operating up to a maximum frequency of 133 MHz. The MVME6100 utilizes one of these buses to provide PMC expansion module capability. Due to loading on this interface, frequency is restricted to 100MHz. The second PCI-X interface bridges to an optional PMCspan expansion module and the VME bus interface.

Throughput is of primary importance on the PCI-X interface. High-speed peripherals must be

able to transfer blocks of data from their internal buffers to main memory, other peripherals, and devices on the VME bus. In order to facilitate these transactions DMA engines are provided in most interface chips. DMA engines are attractive because they dramatically reduce the amount of processing overhead associated with data movement (thus freeing the processor for computation) and since they are tightly coupled to the PCI-X bus they are capable of generating efficient burst transfers to transmit the data.

The MVME6100 has three possible DMA engines available to the user. The first set of DMA engines resides in the Discovery II host bridge. These integrated DMAs (IDMAs) are programmed through memory-mapped registers and descriptor tables within the Disco 2. The second set of DMA engines reside in the Tsi148™ VME bridge chip. The final set of DMA engines available on the MVME6100 may reside on PMC add-in modules that the user has installed.

This section examines the performance of the IDMA and PMC DMA engines. A separate section on VMEbus performance examines the behavior of the engines in the Tsi148™.

4.2 Results

Table 3 shows a side-by-side comparison of the PCI/PCI-X performance of the MVME6100 and the MVME5500 single board computer products. Definitions for the fields in the table match those of Table 1.

Table 3 - PCI/PCI-X Performance

Description	OS	6100	5500	Units	Ratio
Bus 1					
Technology		PCI-X	PCI		
Speed		100	66	MHz	1.52
Width		64	64	bits	1.00
Theoretical BW		800	528	MB/S	1.52
Devices		PMC1, PMC2	GigE, PMC		
Bus 2					
Technology		PCI-X	PCI		
Speed		133	66	MHz	2.02
Width		64	64	bits	1.00
Theoretical BW		1064	528	MB/S	2.02
Devices		Bridge to PMC Span, Bridge to VME	IPMC, Bridge to PMC Span and VME		
Benchmarks					
Discovery II IDMA	**V**				
To PMC		328 (139)*	369***	MB/S	0.89
From PMC		213 (82)*	164***	MB/S	1.30
To PMCspan**		132 (132)*	132	MB/S	1.00
From PMCspan**		132 (82)*	132	MB/S	1.00
PMC DMA	**V**				
To Disco II RAM		780 (374)*	246***	MB/S	3.17
From Disco II RAM		514 (220)*	94***	MB/S	5.47
PMCspan DMA	**V**				
To Disco II RAM**		132	132	MB/S	1.00
From DiscoII Ram**		132	132	MB/S	1.00

* Numbers in parenthesis show performance when cache coherency is enforced by hardware. Chip errata require that PCI-X burst sizes be decreased when operating in this mode.
** PMCspan performance is estimated
*** 5500 DMA performance not tested with same methodology. Results expected to be higher..

4.3 Analysis

A quick glance at Table 3 would lead one to expect close to 800M bytes per second DMA transfers to and from PMC sites on the MVME6100 with an improvement of 1.5X over the MVME5500. Unfortunately, there are two significant factors that make this impracticable.

The first significant factor relates specifically to the Discovery II IDMA engines. During long block reads or writes, the data transfer must be broken up into smaller transactions. The maximum amount of data moved in each smaller burst is device dependent; however, PCI-X limits the maximum byte count to 4096. Assuming the DMA engine still has bus ownership at the end of a burst, it is free to immediately initiate another block read/write, thus minimizing idle cycles and maximizing bus utilization.

The IDMA engines within the Discovery II do not exhibit this behavior. Instead, there is a large delay, or “dead time” between completion of one burst and the start of the next. Figure 4 shows this behavior. Since the IDMA engine is not

using the bus when it otherwise could, significant performance loss is incurred.

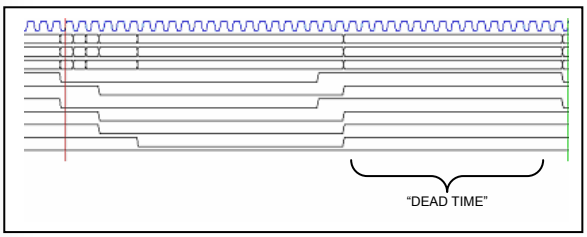

Figure 4 - IDMA waveform showing "dead time"

This phenomenon appears to be related to internal timing of the Discovery II DMA engine.

The second issue relates to all PCI block transactions to the Discovery II host bridge. Issues with cache coherency and burst size require that maximum burst size be decreased from 128 bytes to 32 bytes when hardware coherency is enabled. Shorter burst sizes change the ratio of PCI-X overhead cycles to data beats, lowering efficiency. Since maximum burst size with coherency enabled amounts to only four data beats, this degradation is significant. This is especially compounded when using the IDMA engine since the additional bursts are coupled with added "dead time".

Workarounds for this issue are non trivial. Disabling hardware coherency forces coherency maintenance on the system software. Besides burdening the developer with onerous driver modifications, valuable processing capability must be utilized. The alternative is to leave coherency enabled and utilize alternate DMA engines to provide the same function.

It should also be noted that DMA values shown for in the 6100 column were obtained using the method outlined in reference [12]. Values in the 5500 column were obtained through software measurement techniques using a "typical" PMC device. Consistent measurement techniques are expected to favor the MVME5500.

5 ETHERNET PERFORMANCE

This section outlines the performance of the 1000Mb Ethernet subsystems on the MVME6100 single board computer.

5.1 Overview

Ethernet communications interface on the MVME6100 is provided by the Discovery II host bridge chip. The Discovery II offers two 10/100/1000 MACs each with long frame support up to 9K bytes. Hardware performance features include dedicated DMA engines, hardware checksumming and eight each receive/transmit queues.

5.2 Results

Table 3 shows a side-by-side comparison of TCP Ethernet performance of the MVME6100 and the MVME5500 single board computer products. Definitions for the fields in the table match those of Table 1.

Table 4 - Ethernet Performance

Description	OS	6100	5500	Units	Ratio
General					
Number of devices		2	2		
Ethernet 1 Speed		10/100/1G	10/100		10.0
Ethernet 2 Speed		10/100/1G	10/100/1G		1.00
Aggregate BW		2000	1100	Mb/S	1.82
Benchmarks					
Netperf2.2p	**L**				
RX, 1kB packet		447	449	Mb/S	1.00
TX, 1kB packet		448	350	Mb/S	1.28
RX, 8kB packet		532	450	Mb/S	1.18
TX, 8kB packet		448	358	Mb/S	1.25
RX CPU Usage, 1kB		38%	57%	Mb/S	1.50
TX CPU Usage, 1kB		14%	24%	Mb/S	1.71
Blaster/Blastee	**V**	91	62	MB/S	1.47
Blaster CPU Usage		83%	--		
Blastee CPU Usage		98%	--		

CPU Utilization not captured by netperf
TCP protocol configured for both Netperf and Blaster/Blastee

5.3 Analysis

Ethernet performance hinges on the proper balance of I/O memory and processing capabilities. While each protocol requires a different mix of these components, a general feel can still be achieved by observing the results of a single protocol. Many more tests were run than shown in Table 4; however, the results shown are instructive for understanding overall performance.

Under Linux using the netperf benchmark, the maximum throughput of the two boards is similar with the MVME6100 performing slightly above the MVME5500. The real story can be seen however when processing performance is added to the analysis. The MVME6100 utilizes 33% less processing capability to transmit the same number of 1K byte packets as the MVME5500. It utilizes 42% less processing capability to

receive 28% more 1K byte packets as the MVME5500.

VxWorks blaster/blastee shows 91M bytes per second of transfer capability compared to only 62MB/sec on the MVME5500.

A quick glance at the throughput numbers for netperf is a bit alarming. Bandwidths are all well below 50% of the link theoretical performance. At the present time, the Linux kernel and/or drivers are suspected as being the source of this phenomenon. Efforts are presently scheduled to investigate this.

6 VMEBUS PERFORMANCE

This section outlines the performance of the VMEbus interface on the MVME6100 single board computer.

6.1 Overview

The VME Renaissance began with the announcement of the "Tempe" PCI/X to VMEbus bridge ASIC. This ASIC was developed by Motorola and brought to market by Tundra Semiconductor Corporation as the Tsi148™ bridge chip.

Most notably, the TSi148 implements a two-edge source synchronous transfer (2eSST) protocol, enabling the VMEbus to run at a sustained bandwidth over 250 MB/s. This operation is backward compatible, allowing existing VMEbus cards and new cards to coexist within the same system.

The MVME6100 is the first Motorola single board computer to include the Tsi148™ bridge ASIC and 2eSST technology.

6.2 Results

Table 3 shows a side-by-side comparison of the VMEbus performance of the MVME6100 and the MVME5500 single board computer products. Definitions for the fields in the table match those of Table 1.

Table 5 - VMEbus Performance

Description	OS	6100	5500	Units	Ratio
General					
Bridge Device		Tsi148	Universe II		
Upstream Speed		133	33	MHz	4.03
Upstream Width		64	32	bits	2.00
Upstream Bandwidth		1064	132	MB/S	8.06
Benchmarks					
VME DMA BW*	**V**				
VME Bridge Read		250	38	MB/S	6.58
VME Bridge Write		250	43	MB/S	5.81

* Comparisons shown use fastest transfer mode available on each SBC.

6.3 Analysis

VMEbus DMA analysis has been performed using the fastest available transfer mechanism on each board. Testing of legacy DMA modes for the MVME6100 has also been performed with results as described in Table 6. All transfers were done in a VME64x chassis with MVME6100s as both master and target.

Table 6 - VMEbus Legacy Performance

T ransfer Mode	OS	Read	Write	Units	Size
2eSST	V	255	256	MB/s	8MB
2eVME	V	100	99	MB/s	8MB
MBLT	V	29	67	MB/s	8MB
BLT	V	19	34	MB/s	8MB
SCT	V	6	28	MB/S	8MB

It can be seen from the data that 2eSST provides a significant performance improvement over previous generations of VMEbus technology. The MVME6100 can transfer well over 8x traditional VMEbus modes with data rates in the 250 MB/second range. This rate is when only one DMA engine is used. Running multiple DMA engines concurrently has been demonstrated to achieve 300 MB/second of sustained bandwidth.

7 SUMMARY

This paper has outlined various aspects of MVME6100 product performance. Computation, memory/cache, PCI-X, Ethernet and VMEbus performance were all examined. In each case analysis was provided along with explanation of ongoing efforts. In addition to MVME6100 performance numbers, results were also given for the MVME5500 in order to facilitate understanding of relative product positioning.

While some issues were identified surrounding the Discovery II host bridge chip, when viewed as a whole, the MVME6100 offers performance

advantages beyond previous generation product. Most notably, 2eSST VMEbus transfers establish the MVME6100 as a flagship among Motorola's VMEbus product offerings. Product improvements in computation capabilities and Ethernet performance (VxWorks) will round out the story.

As with all complex computing platforms, some performance questions and tuning opportunities remain. Motorola continues to investigate these tuning opportunities and will pass them on to the customer as they become available.

8 REFERENCES

[1] "Tuning the Performance of the MV64360", December 1, 2002, Marvell Corporation

[2] *MV64460, MV64461, MV64462 System Controller for PowerPC Processors*, Marvell Corporation.

[3] *MV64360, MV64361, MV64362 System Controller for PowerPC Processors*, October 2003, Marvell Corporation.

[4] *GT-64130 Host Bridge*, Galileo Technology

[5] SpecCPU2000 Benchmark, http://www.spec.org

[6] ByteMark Benchmark, http://www.byte.com/bmark/bmark.htm

[7] LMBench, http://www.bitmover.com/lmbench/

[8] PCI-X Specification, Revision 1.0, September 1999, PCI Special Interest Group

[9] "MPC7457 Fact Sheet", http://www.motorola.com

[10] "MPC7455 Fact Sheet", http://www.motorola.com

9 REVISION INFORMATION

Revision	Description	Date
1.00	Initial Revision	July 2004
1.1	Minor edits	July 2004
1.2	Table 4 Corrections/Notes	Aug 2004
1.3	Minor edit, section 4.3	Sept 2004