Hardware configurations and I/O protocol of the WE32100 microprocessor chip set

by MICHAEL L. FUCCIO, LAKSHMI GOYAL, and BENJAMIN NG
AT&T Bell Laboratories
Holmdel, New Jersey

ABSTRACT

In this paper, the hardware design features of the WE32100 Microprocessor chip-set are briefly summarized. That is, the hardware protocols and chip-set configurations are discussed. The WE32100 chip family provides the VLSI core of general purpose computer systems providing virtual memory and high speed floating point arithmetic. The system design features of the chip family support the software architecture of the chip-set and allow easy system interface and integration. The WE32100 chip family includes the WE32100 Microprocessor (CPU), WE32101 Memory Management Unit (MMU),[1] WE32106 Math Accelerator Unit (MAU),[2,3] WE32103 Dynamic RAM Controller (DRC), WE32104 Direct Memory Access Controller, and the WE32105 System Interface Unit (SIU).

INTRODUCTION

The WE32100 chip-set performs a multitude of general purpose computer functions. The chip-set consists of 6 VLSI chips implemented in 1.75μm CMOS technology. Typical operating frequency of all chips is 14 MHz. All chips have a TTL compatible interface and are specified with a 130 pf loading on outputs except for the WE32103 Dynamic RAM Controller (DRC), which controls large banks of DRAMS, and hence has some outputs specified for greater than 130 pf load. Each chip dissipates less than 1.5W of power.

FEATURE LIST

The WE32100 chip-set provides super mini-computer features and performance in just a few VLSI chips. Some features are described below.

WE32100 CPU Features

1. WE32100 CPU is an upward compatible version of the WE32000 (formerly BELLMAC-32A) microprocessor. The WE32100 CPU has a rich instruction set with operating system support instructions included.[4]
2. A general purpose support processor interface consisting of 10 instructions and associated I/O protocol is provided to allow for support-processor elements such as WE32106 Math Accelerator Unit.
3. The WE32100 CPU has a rich interrupt structure including a "Quick-Interrupt," "process switch interrupt," and "automatic-vector interrupts."
4. The WE32100 CPU has an on-chip instruction cache for enhanced performance.
5. The WE32100 CPU has a dynamically selectable two-word block-fetch capability for filling the instruction cache. This feature allows fetching of a 2-word block while issuing only one address. This feature reduces the MMU translation overhead for instruction fetches.

WE32101 MMU Features

1. The WE32101 Memory Management Unit has the same software model as the WE32001 memory management unit.[5,6]
2. The WE32101 MMU can operate in concert with other WE32101 MMUs to provide a virtual-address environment for the WE32100 chip-set, or can operate alone to perform the same task.
3. The WE32101 MMU supports segmented and/or paged virtual memory systems.[2]
4. The WE32101 MMU provides a PAGED/CONTIGUOUS translation indication to support the "early RAS" feature in the WE32103 Dynamic RAM Controller. This "early RAS" feature eliminates a cycle in typical DRAM accesses.

WE32106 MAU Features

1. The WE32106 Math Accelerator Unit supports fully the IEEE draft 10 floating point standard[7] and can operate via the support processor interface or as a common peripheral chip, the former providing significant performance benefits.

WE32105 SIU Features

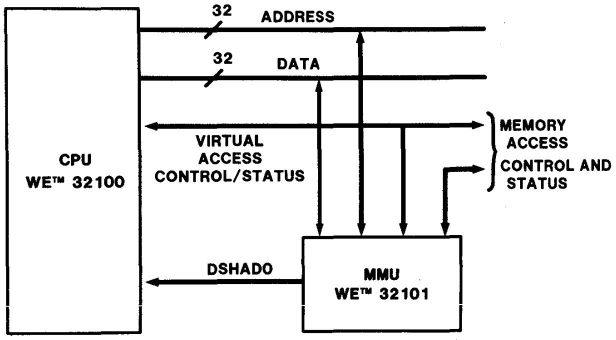
1. The WE32105 System Interface Controller provides a flexible system interface and is a general purpose bus interface chip. This chip is useful but not essential to the operation of the chip-set.

WE32103 DRC Features

1. The WE32103 Dynamic RAM Controller supports normal READ/WRITE operations to dynamic memory including AT&T Technologies' 256K dynamic RAMs.
2. The WE32103 DRC has six programmable refresh modes and is also programmable for different access time DRAM chips.
3. The WE32103 supports page or nibble mode on DRAMS and supports "early RAS" feature for paged virtual to physical address translation.
4. The WE32103-DRC supports double and quad word read/write operations.
5. Also supported on the DRC are handshake signals for interfacing to an error detection and correction chip.
6. Dual ported memory configurations are also supported by the DRC.

WE32104 DMAC Features

1. The WE32104 DMAC provides a separate 8 bit peripheral bus in addition to a fully demultiplexed 32-bit system bus.
2. The WE32104 DMAC supports up to 4 independent channels with programmable priority levels and internal data buffering for each channel.



H04ATC521.004

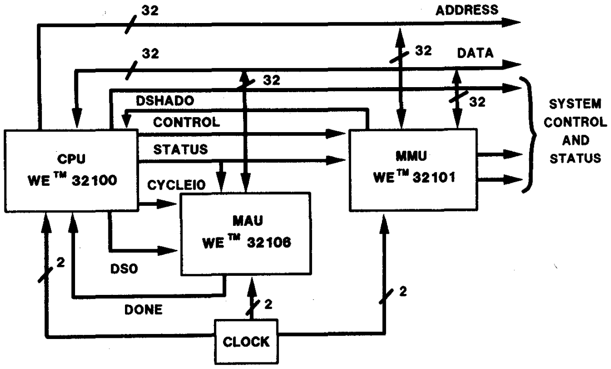
Figure 1—CPU with MMU

3. The DMAC provides the ability for command chaining.
4. The DMAC allows accesses to the peripheral bus from the system bus.

CHIP-SET CONFIGURABILITY

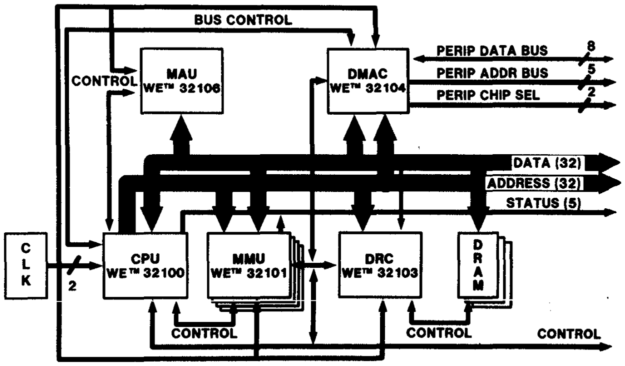
The WE32100 chip-set was architected for multiple configurations. The WE32100 CPU can be a stand-alone microprocessor or it can operate in a number of chip-set configurations depending upon system needs. Each chip set configuration requires no SSI "glue" to integrate the members of the chip family into a fully functional core of high performance computer systems. This "no-glue" solution provides very high functional density and saves precious board space for single-board-computer designers. Some configurations are shown in stylized form in the following figures.

Figure 1 shows a CPU with MMU. The salient features of the CPU-MMU configuration are shared address and data buses. The MMU becomes a bus master, in its own right, when it performs miss-processing to fill its internal translation caches. The MMU translation overhead is one cycle, hence a native three cycle access becomes a four cycle access with virtual to physical address translation. The MMU uses the DSHAD0 signal to preempt an on-going CPU access when miss processing is performed.



H04ATC521.002

Figure 2—CPU with MAU and MMU



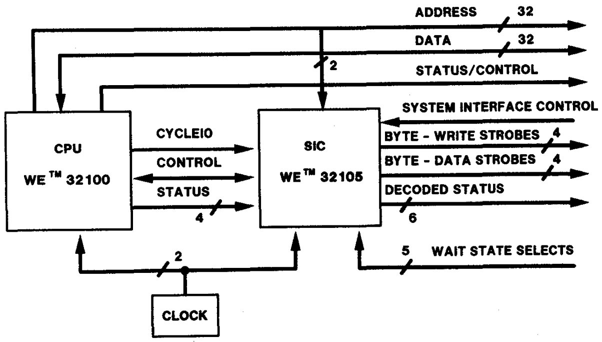
H04ATC521.003

Figure 3—CPU with MMU, MAU, DMAC, and DRC

Figure 2 shows a CPU with MMU and MAU. The notable feature of the CPU-MMU-MAU configuration is, once again, the shared, non-multiplexed address and data busses. The math accelerator unit, when configured as a support processor, takes data directly off of the data bus. Floating point operand addresses are generated by the CPU.

Figure 3 shows a CPU with MMU, MAU, DMAC and DRC. This configuration shows the shared address and data busses. This configuration also shows the separate peripheral bus on the DMA controller. This peripheral bus is ideally suited to character oriented I/O such as UARTs (Universal Asynchronous Receiver Transmitter) or a local area network interface. Figure 3 also shows that multiple MMUs can provide a larger translation buffer if necessary. Furthermore, the DRC chip is shown controlling multiple banks of dynamic RAM.

Figure 4 shows a CPU with System Interface Unit. The System Interface Unit (SIU) provides many common functions which are usually implemented with SSI logic such as byte-write strobes and WAIT-STATE generation circuitry. The WE32100 chip-set can be configured with or without the SIU chip. The SIU chip can be part of all the WE32100 chip-set configurations or none of them.



H04ATC521.001

Figure 4—CPU with SIC

INTERCHIP PROTOCOL

The WE32100 chip-set has a 3-cycle (zero wait-cycle) memory access transaction with the added overhead of one cycle for the MMU to perform address translation. The block fetch feature requires five cycles (zero wait-cycle) for the two-word transfer with only one additional cycle for the MMU to perform address translation. The block fetch feature reduces the MMU translation overhead on instruction fetches. This reduction in overhead is large when there are many wait-states required to access memory. The operands for the support processor require three cycles (zero wait-cycle) to complete the support processor transaction, hence no more time is spent loading operands to the MAU than are spent by the CPU to fetch the operands. There is no overhead in loading support processor operands.

The hardware interface signals presented to the rest of the system (e.g., memory) are not unlike common microprocessors. Though the interchip protocol is essentially a synchronous protocol, the system interface is asynchronous in the sense that a 2-direction handshake between the chip-set and the external system is implemented. All asynchronous signals are sampled and doubly latched to avoid metastability of input signals. The four-cycle memory access with MMU translation is described below:

Read
Transaction
(See Figure 5)
Cycle Activity

1 At the beginning of the first cycle, the address is issued. Status indicating the size of the datum to be read is issued. Also a read indication is generated.

In mid-cycle, the virtual address strobe and a data strobe are asserted indicating that a valid address is on the address bus and that data may be put on the data bus.

2 If the address is a virtual address, it is tri-stated in this cycle to allow the translated address to be put on the bus. If an asynchronous data transfer acknowledge (DTACK) signal is present at mid-cycle then the transaction will terminate in the following cycle. If the synchronous data transfer acknowledge (SRDY0) signal is present at the end of this cycle then the transaction will terminate in the next cycle. If no data transfer acknowledge is present then the next cycle will be a WAIT-CYCLE. Note that the MMU requires one WAIT-CYCLE to perform virtual to physical address translation.

3 Without an MMU, the CPU can sample data in this cycle and terminate the transaction. With an MMU present this cycle is a WAIT-CYCLE as seen by the CPU. In this cycle, the MMU issues the physical address. At mid-cycle, the MMU issues the physical address strobe (PAS0) and data strobe indicating that a valid physical address is on the address bus and that data may be put on the data bus.

If the DTACK or SRDY signals are asserted in this cycle, the CPU will sample the data and terminate the transaction in the following cycle, otherwise the next cycle will be another WAIT-CYCLE.

4 If the DTACK or SRDY signals were asserted in cycle 3 and no bus exceptions occurred, then the data is sampled by the CPU in mid-cycle and the transaction is terminated. The DRDY0 signal is issued indicating that the transaction has completed without exceptions.

Write
Transaction
(See Figure 6)
Cycle Activity

1 Address, Status and Read/Write signals are generated at the beginning of the cycle.

In mid-cycle, the virtual address strobe is asserted.

2 Data is driven at the beginning of the cycle and the data strobe is asserted in mid-cycle.

If the address is a virtual address, it is tri-stated during this cycle.

If DTACK0 is received mid-cycle or SRDY0 is

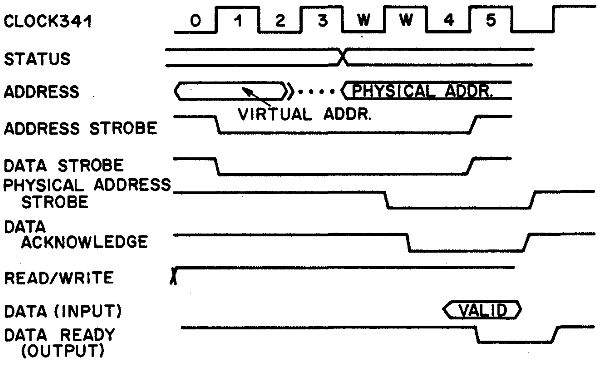


Figure 5—Read with address translation

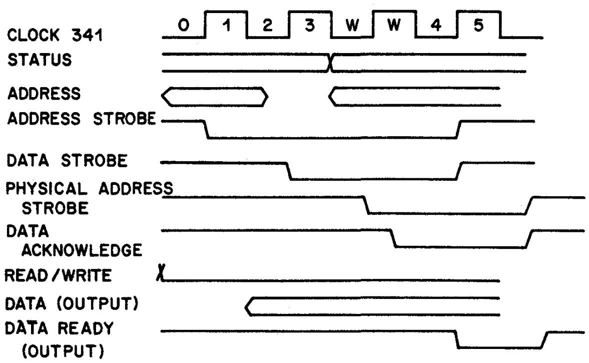


Figure 6—Write with address translation

TABLE I—Physical characteristics of the chip-set

Chip	No. of Transistors	Die Size	Total No.of I/Os (Active)
WE32100	180,000	1.0 cm^2	132/118
WE32101	92,000	0.7 cm^2	132/95
WE32103	13,000	0.4 cm^2	132/89
WE32104	113,000	1.0 cm^2	132/113
WE32105	4,500	0.4 cm^2	100/64
WE32106	160,000	1.0 cm^2	100/53

received at the end of the cycle, the access will be terminated during the next cycle. If no acknowledge is received, the next cycle will be a WAIT-CYCLE. Note that the MMU requires one WAIT-CYCLE to perform the virtual to physical translation.

3 Without an MMU, the transaction can be terminated if DTACK0 or SRDY0 was received during the previous cycle.
With an MMU, the physical address is issued, the physical address strobe is asserted at mid-cycle and the MMU's data strobe is asserted at the end of the cycle.
If DTACK0 or SRDY0 are received, the next cycle will terminate the transaction, otherwise, the next cycle will be another WAIT-CYCLE.

4 If the DTACK0 and SRDY0 signals were asserted the previous cycle, then the CPU will terminate the transactions by negating address strobe and data strobe. The DRDY0 signal will be issued if no bus exceptions were received.

Another salient feature of the chip-set is a preemptive method of obtaining the microprocessor's bus. Typical bus arbitration schemes do not allow a bus master to take control of the microprocessor's bus until the microprocessor has completed an ongoing transaction. The WE32100 CPU allows a non-interlocked transaction to be preempted by another bus master. After the new bus master returns control of the bus to the WE32100 CPU, the CPU will retry the preempted access. This preemptive bus arbitration, along with the common two-wire arbitration scheme allows easy implementation of multitiered system-bus structures. The preemptive arbitration allows for deadlock resolution in such multitiered systems.

PHYSICAL CHARACTERISTICS

Each VLSI chip in the WE32100 chip-set is state-of-the-art. Packaging is designed to maximize routability of chip I/Os as well as reduce $\delta i/\delta t$ noise. Table I summarizes the state-of-the-art characteristics of the chips in the chip-set.

SUMMARY

The WE32100 chip-set implements many minicomputer functions in six VLSI chips, with minicomputer performance.

ACKNOWLEDGMENTS

The synthesis of a VLSI chip-set such as the WE32100 chip-set requires contribution from many areas of expertise. The author wants to acknowledge the other chip-set architects: Thomas Lee, Peter Voldstad, William Dietrich, and Ulhas Gumaste. Also recognized are designers James Seery, Mark Kaplan, Frank LaRocca, Mark Thierbach, Jonathan Fields, Winston Pekrul, Tim Sippel, L. H. Blendinger, Barbara Tai and Lincoln Pierce.

REFERENCES

1. Goksel, A., J. Agraz-Guerena, H. Jacobs, J. Molinelli, P. Swartz, Y. Wo, and W. Troutman. "A Memory Management Unit For a Second Generation Microprocessor." Talk presented at COMPCON, Spring 1984.
2. Gumaste, U., P. Lu, and Z. Mao. "The Architectural Features of WE32106 Math Acceleration Unit (MAU)." Unpublished manuscript,
3. Goksel, A., J. Fields, U. Gumaste, and C. Kung. "An IEEE Standard Floating Point Chip." *Proceedings of the International Solid State Circuits Conference,* New York: IEEE, 1985.
4. Berenbaum, A., M. Condry, and P. Lu. "The Operating System and Language Support Features of the BELLMAC-32 Microprocessor." *Proceedings of Symposium on Architectural Support for Programming Languages and Operating Systems;* March 1–3, 1982; Palo Alto, CA.
5. Lu, P., W. Dietrich, M. Fuccio, L. Goyal, C. Chen, D. Blahut, J. Fields, A. Goksel, and F. LaRocca. "Architecture of a VLSI MAP for BELLMAC-32 Microprocessor." *Digest of Papers, COMPCON Spring 1983.* San Francisco, Calif.: IEEE, 1983.
6. Goksel, A., J. Fields, F. LaRocca, P. Lu, W. Troutman, and K. Wong. "A VLSI Memory Management Chip: Design Considerations and Experience." *IEEE Journal of Solid-State Circuits,* SC-19, No. 3 (1984).
7. *A Proposed Standard for Binary Floating-Point Arithmetic.* Draft 10.0 of IEEE Task, p. 754, December 2, 1982.