A direct processor o. data
The of main
Contiguous the chips.
Loth consisting the trans.fer point slave processor, Interrupt control unit, timers, and time-of-day clock. The 16.0 megabytes of random access memory are interleaved to allow windowed direct memory access on and off the FASTBUS at 80 megabytes per second.

Overview

The FB32000 computer is optimized for high speed block transfer of data over the FASTBUS. A large fraction of board space was devoted to an 80 megabyte per second DMA channel from main memory to the FASTBUS. The rest of the board is divided into four blocks: a 32-bit UNIX (TM) engine, the FASTBUS interface and control section, an ETHERNET interface; and memory, consisting of 128k bytes of ROM and 16 megabytes of DRAM.

The NS32032 microprocessor was chosen as the UNIX (TM) engine due to its availability, orthogonal instruction set, and the existence of a full complement of support chips. The CPU and its support chip set reside on a separate demultiplexed address and data bus and reserve the FASTBUS for inter-device block transfers via the DMA channel.

A direct memory access channel between the FASTBUS and the FB32000's memory supports multiple block transfers of data fields situated in contiguous or non-contiguous areas of memory. Block size is variable up to the entire 16 megabytes of main memory. A multiple windowing scheme in hardware allows stripping of header information into separate memory areas without processor intervention. Memory protection is provided both by windowing and by memory partitioning in the memory management unit's page table entries. High speed memory access is achieved by 4-way interleaving of main memory with read-ahead and write-behind of data within the memory banks.

The FASTBUS interface supports master/slave operation, service request detection, single or block mode transfers and parity checking. The computer can be addressed in either logical or geographical space.

An ETHERNET interface based on the Intel 82586 chip set has been built and tested, and an ETHERNET port will be included with a DMA interface to memory.

National Semiconductor's NS32000 chip set includes a powerful combination of functions supporting fast calculation and data reduction. The NS32032 CPU executes 1.0 million instructions per second when running on a 10 MHz clock, and addresses 16 megabytes of physical memory. Eight 32-bit general purpose registers and full 32-bit internal address and data paths allow efficient handling of 32-bit quantities. The 24-bit external address accesses 16 megabytes of physical address space. High level language support was designed into the very orthogonal instruction set replete with many addressing modes, and several data types.

The memory management function is performed by the NS32082 MMU. The NS32082 provides demand paged virtual memory translation from 16 megabytes of logical address space to 16 megabytes of physical address space. A program trace-back and hardware breakpoint register set within the MMU supports debugging.

Floating point number calculations are supported by the NS32081 FPU. The FPU performs single and double precision IEEE standard P754 binary floating point arithmetic. Eight 32-bit registers are accessible through a 16-bit data port. A double precision multiply takes 6.2 µs to complete on a 10.0 MHz NS32081.

The NS32202 Interrupt Control Unit contains 2 cascadable 16-bit counters and prioritized control of 16 maskable interrupt inputs.

An additional DMA controller, the NS32203, provides fast memory access for devices resident on the FB32000's internal execution bus. The NS32203 operates at up to 5 megabytes per second, packs 8-bit peripheral data into 16-bit words for faster data transfer, includes 4 16-bit I/O channels, and can generate interrupt vectors for simple slave devices. The NS32203 is functionally independent of the FASTBUS DMA channel, and competes at a lower priority for bus resources.

The MM58167 real time clock provides time-stamping of data and an alarm function interrupt. Battery backup prevents loss of date at power failure.

Memory is divided into 128k bytes of ROM consisting of 4 32k byte by 8-bit EPROM chips, and 16 megabytes of DRAM. The RAM memory consists of 4 banks of 4 megabytes each, interleaved on 32-bit word boundaries. Four finite state machines control memory operation. At the lowest level page mode cycle timing and read-ahead/write behind operation is controlled within each bank. Another state machine handles refresh cycles across all four banks, and a third state machine
controls interleaver operation during block transfers. The fourth state machine orchestrates the register to register pointer movement in the DMA controller. The state machine designs differed in speed, complexity, and ease of design. The interleaver machine, being asynchronous and very fast, proved to be a difficult design. The other machines are slower, and have been made synchronous. For the slower state machines an automated approach was possible, and an algorithmic description of their operation was entered in PEG, a Programmed Array Logic Equation Generator, which is a portion of the Berkeley VLSI Design Tools. PEG outputs truth tables and test vectors for a Mealy model state machine. Members of our group wrote a filter program which translates the PEG output into PALASM format. The PALASM format was translated to CUPL format, and CUPL was then used to compile PAL and FPLA implementations of the state machines.

Each bank is organized as 4 9-bit bytes arranged in 4 tiers of Texas Instruments TMS4164EL9 or TMS4256EL9 single-in-line packages. Depending on whether 64k-byte or 256k-byte SIs are used, and how many tiers are populated, the banks are configurable from 1/4 megabyte, or 1.0 megabytes, or 16 megabytes for the entire memory. Each bank has independent cycle control, and a 22-bit address counter used during block transfer to provide addresses during read-ahead and write-behind. A 36-bit latch holds data and parity at the bank-to-interleaver interface, and, in conjunction with the DMA control, allows clear-and-allocate operations to execute at the block transfer rate (80 megabytes/second). The bank control operates the DRAMS in page mode at all times. Page mode operation eliminates time and power penalties associated with the row address strobe. Row addresses are latched in, and need not be changed until the 512-bit page boundary is crossed. With 120ns memory parts the memory cycle times average 160ns, leaving a good margin for the 200ns bank timing required by the interleaver.

The interleaver uses bits 2 and 3 of the address word to multiplex the bank data registers into a 33-bit data latch accessible either by the FASTBUS or by the CPU. Interleaver input section also generates or checks parity going into or out of memory, assembles or disassembles parity for bank storage and word parity for FASTBUS transmission. Interleaver operation is completely asynchronous, with 4-cycle handshaking.

DMA transfers of single or multiple blocks of data on 32-bit word boundaries of heterogeneous size and possibly random locality are accomplished by a hardware stack of pointer pairs stored in the DMA control section. Prior to starting a DMA transfer, the CPU writes a list of address pointer pairs into the stack via a random access port. The CPU then initializes the DMA stack pointer to point at the first pointer pair, indicating the first block to be transferred. Two inter-device protocols are available for use in coordinating transfers across the FASTBUS. In the first, a device wishing a block transfer transmits a list of block sizes which are used to compute destination addresses in the FB32000. The second technique allows a 'mailbox' model by having the FB32000 open an area in memory by preloading the DMA pointer stack. Another FASTBUS device could then gain FASTBUS mastership, and request a block transfer. The FB32000 DMA channel and FASTBUS interface state machines would respond by gaining internal FB32000 bus mastership, and accept any size message up to the 'mailbox' size, which could be any size from one 32-bit word to the entire 16 megabytes of memory. After the transfer is complete, the CPU would be interrupted and could examine the 'mailbox' contents to determine what, if any, action to take. Two FB32000 computers could use the latter protocol to implement a shared memory.

Once an inter-device protocol has been agreed upon, and the DMA pointer stack and control registers loaded, the DMA controller loads the first pointer pair as the starting and stopping address for the first block. A fast state machine then moves data starting at the lower address and incrementing up to the stopping, or higher, address. After each block is transmitted, a new pair of pointers is popped from the stack, and the next block is moved between the new pointers. By alternating data blocks with header information, headers may be stripped off and stored in separate lists. The block-to-block switching time is on the order of 1.0 μSec.

The FASTBUS interface is composed of four modules: master logic; slave logic; CSR's; and the FASTBUS to local bus interface. The FASTBUS interface consists of four ASKOM FMA601 ECL macrolever arrays, 10124 and 10125 ECL-TTL level translators, and an NTA register. Each FMA601 is an 8-bit slice which implements FASTBUS drivers and receivers, output latches, parity logic, logical address register and compare, and local bus drivers and receivers. Worst case propagation delay through the FMA601 is 8.2 ns. Parity generation and checking is supported for the FASTBUS data.

CSR's 0, 3, 5, 7 and 8 are implemented. CSR0 contains the device ID and 16 control and status bits. CSR3 is the logical address register and CSR8 is the master arbitration level register. CSR5 is used to inform other masters of the size of the DMA window which has been opened. CSR7 is used for storing the broadcast case 2 "class N" values in slave mode.

The slave logic supports geographical, logical and broadcast case 1--primary address cycles. Random read/write, block transfer, and secondary address cycles are supported. The slave responds with SS-6 for data transfer errors.

The master logic contains the bus mastership arbitration circuitry and a finite state machine which translates between FASTBUS cycles and on-board CPU/DMA cycles. The state machine coordinates:

1. obtaining and releasing bus mastership;
2. obtaining and releasing AS/AK lock;
3. issuing MS <2:0> and monitoring/responding to a SS <2:0>;
4. translating between FASTBUS and on-board data cycles; and
5. issuing a DONE signal at the end of DMA transfers.
Status

The FASTBUS computer is in the final prototyping stage, with a full function prototype expected to be completed in early summer, 1985. Several sections have been built separately and are running in a breadboard system. The CPU, Memory, and internal bus have been tested, and data transfers at 80 megabytes per second have been demonstrated over the FASTBUS. We are waiting for delivery of National Semiconductor's GENIX operating system, a Berkeley 4.2 derivative. Members of our group have completed a Digital Equipment Corporation VAX assembly language to 32032 assembly language translator. Cross development software which runs under UNIX (TM) is being used to develop test software for the FB3200 on a Digital Equipment Corporation Vax 11/780.

Current plans call for the completion of 5 FASTBUS computers during fiscal year 1985. Two of these computers will be kept at E-10 and operated as a dual CPU system. LANL is investigating transferring the design to one of several private firms for production.

Acknowledgment

This work was supported by U.S. Department of Energy.

References