Title

A CONTROL SCHEME FOR MICROCOMPUTERS BEING USED IN MULTIPROCESSOR ARRAYS

Authors

Meng, J.
Gin, F.

Publication Date

1984-06-01

LBL-17607 c.2

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

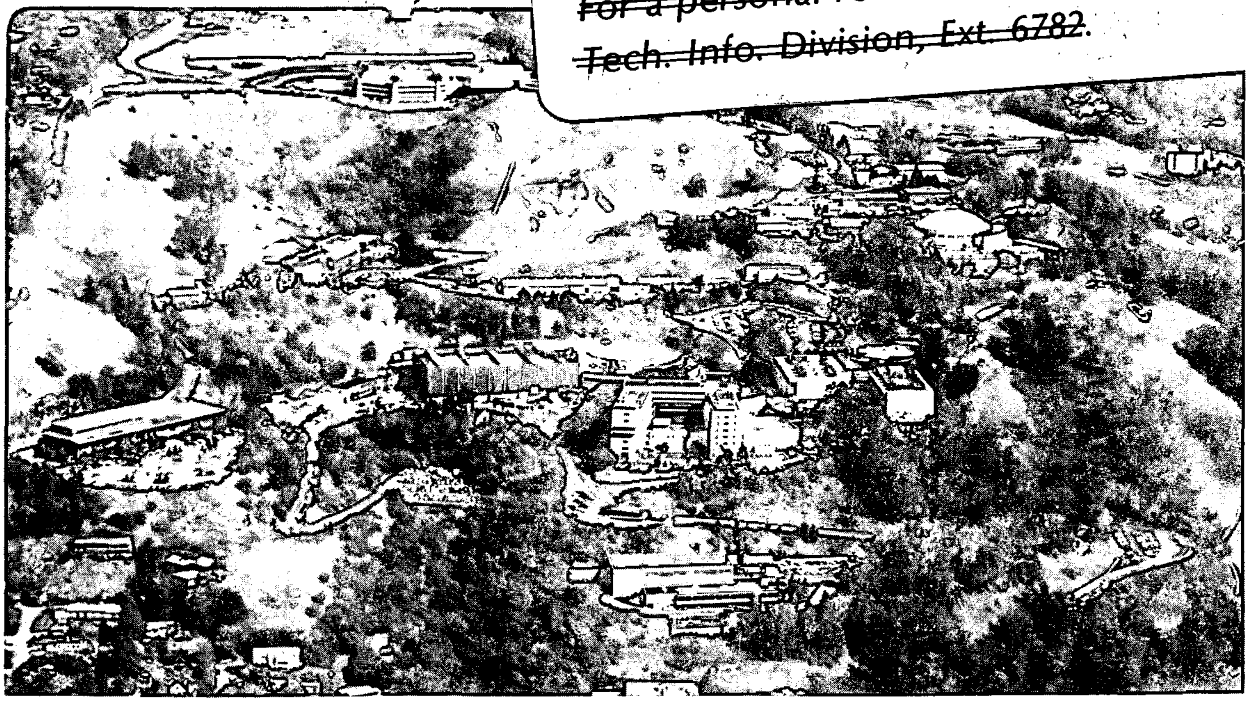
Engineering & Technical Services Division

Presented at the 25th International Symposium on
Mini and Microcomputers and Their Applications,
San Francisco, CA, June 5-6, 1984

A CONTROL SCHEME FOR MICROCOMPUTERS BEING USED IN MULTIPROCESSOR ARRAYS

J. Meng and F. Gin

June 1984



Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

LBL-17607

A CONTROL SCHEME FOR MICROCOMPUTERS BEING USED IN MULTIPROCESSOR ARRAYS

John Meng and Fong Gin

University of California, Berkeley
Lawrence Berkeley Laboratory
Berkeley, California U.S.A. 94720

ABSTRACT

In general, microcomputer central processor devices are completely controllable from memory and memory control lines. By interjecting a controlling processor between the central processor chip and its memory, and using the central processor "memory ready" signal for synchronization, data can be supplied to the microprocessor either from an attached memory or from the controlling processor. The controlling processor may also download codes into the microprocessor's memory to be used either as programs or as data. By manipulating restart, hold and interrupt signal lines in addition to the memory lines, total control is achieved. Such a scheme can be used to orchestrate the simultaneous application of arrays of microcomputers to single large problems or to many discrete smaller problems. We describe the details of such connections to three commercially available devices: A Motorola 68000, an Advanced Micro Devices 29116 and a National Semiconductor NS32032 and indicate how our scheme may be used to connect such devices into a cooperating parallel array.

INTRODUCTION

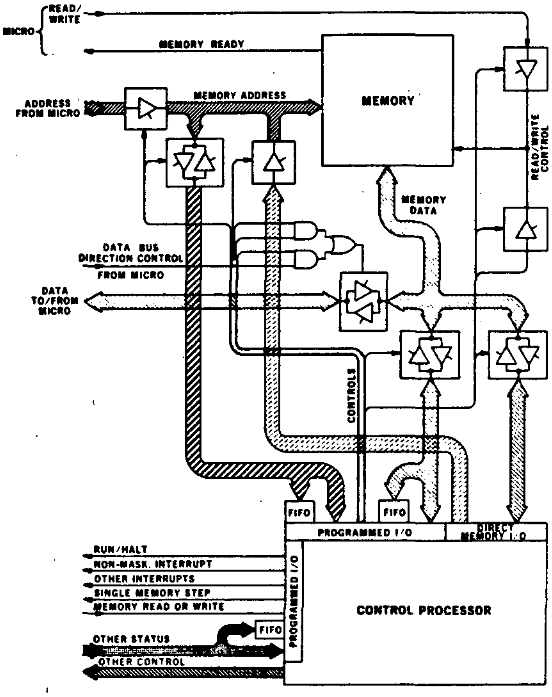
Microcomputers now functionally replace large chunks of what previously was minicomputer domain. This is happening because microcomputers are 1) capable, 2) physically smaller, 3) of equivalent or of greater memory addressing capacity, 4) less power consumptive, and 5) less expensive. These microprocessors are today prime candidates for becoming the computing elements in multiple processor computing engines.

Single path sequential machines, classic Von Neumann architecture, from the smallest microcomputer to the largest mainframe, are all facing an impenetrable barrier to performance. The speed of light being the upper limit to signal transit velocity, the time it takes for data to move within the physical structure of the microchip or within the physical structure of the mainframe will never be less than that determined by the distance it must move, divided by the speed of light. The smaller the computing device, the faster it can be. The only known route around the speed of light barrier is to use multiple processing devices all working on a problem simultaneously. We have successfully used minicomputer central processors in a parallel configuration to demonstrate speedups proportional to the number of processors.[2,3] It is important for us to be able to do similarly using microprocessors. Significant problems are initialization, downloading and functional control of the microprocessors. We achieve this in our minicomputer-based system by using the on-board control panel logic in each minicomputer central processor.[4]

A central authority--a controlling processor--must be able to completely orchestrate the performance of the micros in order to effectively apply their computing power in parallel to a user problem. Software-to-software communication is possible wherein programs are preplaced into each micro to request and execute tasks passed down from the central authority. This scheme has at least two major disadvantages. First, task and code passing is software driven and, hence, slow. Second, some significant parts of the micro's capacity must be dedicated to looking for commands, downloading programs, executing them and passing on results. This overhead not only interferes with execution of user programs, but competes with user jobs for memory space.

THE HARDWARE SOLUTION

By giving the central authority control over the microprocessor central processor connection to its memory, we can relieve these drawbacks (Fig. 1).



XBL 845-1750

Fig. 1. The above diagram illustrates the control processor connection into the system. Input/Output and gating are duplicated for each microprocessor central processor in the parallel array. FIFOs allow the control processor to take 'snapshots' of bus sequences.

When it is necessary to download programs, the controlling processor takes control of the micro's memory and performs a direct memory transfer. Having completed this, it returns the memory to the micro and uses its connection to the micro interrupt and control lines to start micro execution. To signal the controlling processor, as at the completion of a job, the micro simply halts or performs some other easily detectable stunt. The First In First Out (FIFO) stacks monitor data, address and status lines at micro bus speeds. The control processor can read these devices leisurely to interpret its 'snapshots' of microprocessor activity. Taking such snapshots does not interfere with the micro. They are useful in reconfigurable multi-processor systems when an optimum configuration of processors is being sought for a particular problem, and where this snapshot monitoring can tell the controlling processor where slowdowns are occurring.

Figure 1 illustrates a control processor's attachment to bus, control and status lines. Figures 2,3 and 4 complete this drawing with specific connections to each of three commercially available devices. In Fig. 1, gating is shown inserted into memory-data and address busses to allow their connection either to the micro or to the control processor, under control of the control processor.

The control processor has a direct memory link for downloading data or programs and for uploading results. It monitors status lines from the micro and drives control lines into the micro. Single-step control of the micro allows a debugging mode of operation where the control processor intercepts memory requests and itself acts as though it were the micro's memory. A programmer can then 'spoon feed' the micro and leisurely contemplate the results. The control processor can also be used to generate breakpoints during debugging based on micro bus addresses, data or status.

Figure 2 illustrates the connection of the National Semiconductor NS32032 into the controlling scheme illustrated in Fig. 1. A small amount of additional logic is required to insure synchronization of interrupt and control signals. In order to single step the micro, the READY signal, normally used to delay micro operations until external devices can perform, is held FALSE and synchronously released for a single micro bus cycle on the leading edge of a control signal from the controlling processor.

The NS32032 addresses memory in bytes, and uses four lines to select which byte or which group of bytes it is accessing. Gating is added for controlling this function and the memory of Fig. 1 is redrawn showing a four-independent-byte organization. The NS32032 is designed to work with a memory management chip and a floating point arithmetic chip. Either, both or neither may be included in the system.

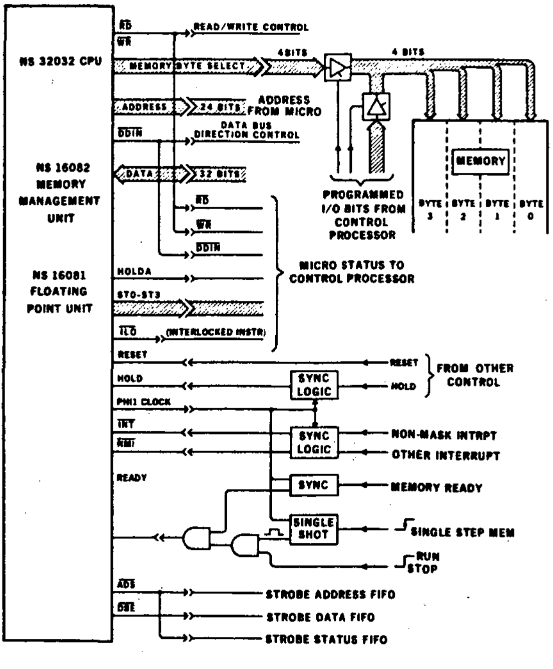


Fig. 2. With minimal perturbations in the general connection scheme shown in Fig. 1, the NS32032 is connected and controlled by the control processor. Labels on lines leading from this figure refer to the same labels on Fig. 1.

The Advanced Micro Devices AM29116 is unique among the three devices we are considering. Whereas the other two devices are designed to run programs with just the addition of memory to the central processor chip, the AM29116 is just the processing part of a central processor and needs a considerable amount of control logic in addition to memory to run as a stand-alone computer. The appropriateness of its consideration derives from its potential as a processing element in a special class of multiple-processor systems; namely pipeline processors, and from its potential as a high speed special purpose processor for quickly performing simple repetitive tasks on large blocks of data.

The AM29116 performs high speed 16-bit logic. The specific operation is determined by sixteen instruction line inputs into the chip. Data is passed into and out of the device on sixteen additional lines. In Fig. 3 we have shown the data input/output lines connecting into the memory in Fig. 1. An independent memory is shown in Fig. 3 to store instruction sequences. It has gating allowing it to be written and read by the control processor. Counters are included in the processor block to sequence memory accesses. Single stepping is a function handled by the AM2925 clock generator chip. Some additional control logic is shown to handle memory accesses and system initialization.

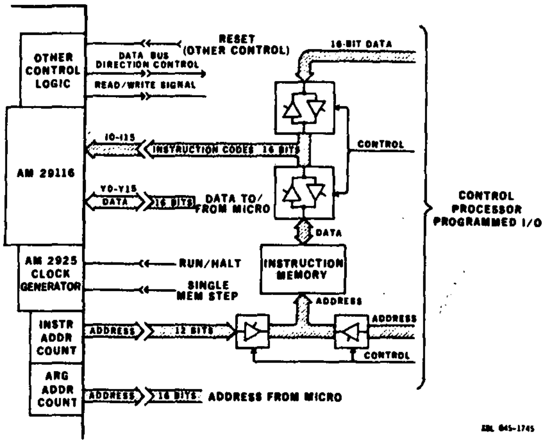


Fig. 3. Unlike the other two devices under consideration, the AM29116 is not functionally a fully integrated central processor device. Instruction select lines are separate from data input/output lines. This is uniquely useful for some applications. For example, repetitive operations on a block of arguments are very fast. The device makes an ideal computing element in pipeline processor stages. Connections must be made into the separate instruction memory as well as into the argument memory to effectively control the device.

Figure 4 illustrates connection of a Motorola MC68000 into the control scheme of Fig. 1.

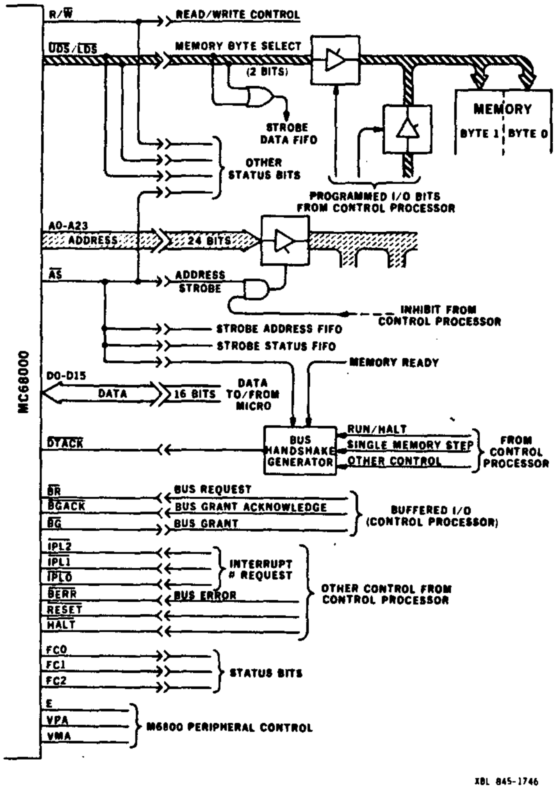


Fig. 4. The M68000 connection to the control processor is similar to that of the NS32032. However, the bus is asynchronous, simplifying the connection of some handshaking and control signals.

This connection is similar to the NS32032 connection. The differences are in the use of an asynchronous bus by the MC68000, and in the use of 16-bit instead of 32-bit data transfers. The MC68000 uses byte addressing and consequently the two byte-select lines are shown connecting to a byte-oriented memory through control gating.

The asynchronous bus eliminates the need for synchronizing logic shown in Fig. 2. This is replaced by some bus handshaking logic used to control the DTACK (Data Transfer ACKnowledge) line.

MULTIPLE-PROCESSOR CONFIGURATIONS

Using our control scheme, the obvious multiple processor configuration is that sketched in Fig. 5.

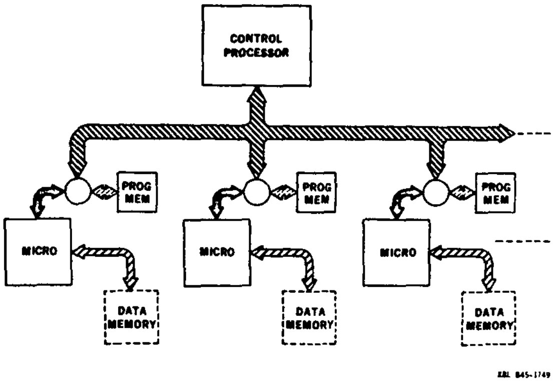


Fig. 5. Micros connected in the group above are all under the absolute control of the control processor. Independent problems may be downloaded and run or single problems may be downloaded and run in parallel. Processor intercommunication is handled via the control processor. With the added connection to data memories, this architecture is being used in the MIDAS^x multiprocessor system.

The micros are autonomous. After setup they run independently of the control processor and independently of all the other micros in the system. For problems not requiring processor intercommunication, this multiple processor configuration achieves processing speeds linearly proportional to the number of processors.[2,3]

Data memories in Fig. 5 are shown directly connected to each micro. In the MIDAS' multi-processing system these data memories are connected to the processors via a crossbar gating array, and used to transport data from processor to processor. The control processor has access to data memories via the micros. If the micro were the AM29116, this access route would not be available, since instruction memory and data memory are maintained separately throughout the chip. In Fig. 3 we have included gating in both memory-type connections, allowing the control processor access both to data memory and to instruction memory. With added logic, of course, the AM29116 can become a 'complete' processor just like the MC68000 or the NS32032, and can be used in equivalent fashion. Different microprocessor central processors can be mixed indiscriminately in arrays such as that of Fig. 5, and tasks passed to each can be selected to fit each's unique characteristics. The NS32032 features a 32-bit data path into and out of the chip, for example, and this might be useful for high precision integer arithmetic work. The AM29116, on the other hand, only handles sixteen-bit data, but executes arithmetic or shifting functions in 100 ns. It might be particularly valuable where a large block of data had to be transformed word-by-word.

Figure 6 is a pipeline structure. Data is passed through the pipeline at a fixed rate, and operations on the data must be completed within one time frame of the transfer clock.

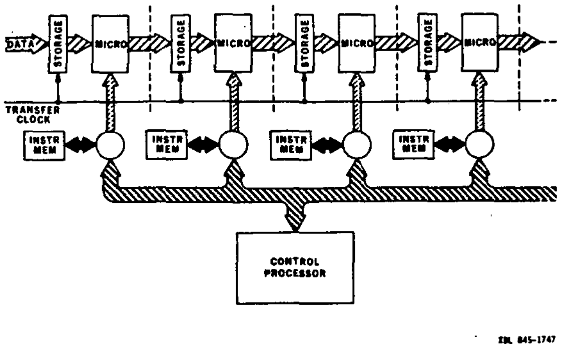


Fig. 6. In a pipelined configuration, data is stepped one stage per transfer clock. Parallel processing occurs when more than one stage simultaneously contains data. Processing per stage is limited to that which can be done in the time between transfer clocks.

Whereas full-blown microprocessor central processors would not normally be able to keep up with the clock in such a pipeline, the AM29116 can. The instruction lines on the AM29116 can be set by the controlling processor so that operations performed on the passing data are done in the proper sequence. One could even imagine a branched pipe with valves at branch points to send the data through the correct one of a choice of pipelined operations. In the conventional computer, instructions are sequentially presented to the processor. In the pipelined computer, data is sequentially passed to processors having fixed instructions. The pipelined processor has the advantage of an inherent parallelism as soon as more than one data word gets into the pipe.

CONCLUSION

Because there is an upper limit represented by the speed of light on the processing speed of sequential computing systems, moves are under way to produce multiple-processor systems. Microcomputers, because of small size and low power consumption, make large arrays for use in such systems practical. Control of our multiprocessor systems is done by interrupting the connection of the microprocessor central processor to its memory. Resulting parallel structures can accommodate a variety of microprocessors within one system making use of the unique advantages of each. A fast pipelined parallel processor is feasible where instruction memory and data memory connections are physically separated.

ACKNOWLEDGMENT

This work was supported by the Director's Office of Energy Research, Office of High Energy and Nuclear Physics, Division of Nuclear Physics, and by Nuclear Sciences of the Basic Energy Program of the U. S. Department of Energy under Contract Number DE-AC03-76SF00098.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U. S. Department of Energy to the exclusion of others that may be suitable.

REFERENCES

1. Meng J, D Weaver, C Maples, W Rathbun and D Logan, "An Interactive Parallel Processor for Data Analysis" IEEE Trans. Nucl. Sci., NS-31, No. 1, 162.

2. Meng J, "Multiplication of Processing Capacity with a Parallel Processor Array", Conf. Record of the 17th Asilomar Conf. on Circuitry, Systems and Computers, November 1983.

3. Maples C, D Weaver, J Meng, W Rathbun and D Logan, "Utilizing a Multiprocessor Architecture--The Performance of MIDAS", IEEE Trans. Nucl. Sci., NS-30, No. 5, 3827.

4. Meng J, "Controlling a Radially Connected Array of Minicomputers", Conf. Record of the 16th Asilomar Conf. on Circuits, Systems and Computers, November 1982, 280.

This report was done with support from the Department of Energy. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the Department of Energy.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

TECHNICAL INFORMATION DEPARTMENT
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720