IEEE Transactions on Nuclear Science, Vol. NS-29, No. 1, February 1982

A METHOD OF INTERPROCESSOR COMMUNICATION FOR A MULTIPROCESSOR ENVIRONMENT*

J. L. Alberi

Brookhaven National Laboratory
Upton, New York 11973

ABSTRACT

An efficient method of interprocessor communication between a Digital Equipment Corporation LSI-11 minicomputer and an Intel 8085 microprocessor has been developed for a multiprocessor environment. In this method, which optimizes the handling of transactions between the two processors, an LSI-11 minicomputer interrupts the execution of the 8085 microcomputer under hardware control to place data directly in 8085 memory. Details of hardware implementation and interprocessor communication blocks are presented.

INTRODUCTION

The low cost of digital processors and hardware makes it advantageous in many cases to partition the function of a computer system among several processors. These processors generally have different speeds, computational ability and word length tailored to the design goals of the system. One such partitioning incorporates an eight-bit microprocessor used as a low-level controller in a system containing a 16- or 32-bit minicomputer.

This paper discusses a method of interprocessor communication between a 16 bit minicomputer and a 8 bit microcomputer where the processors are physically close together, usually within the same logic enclosure, and one processor is defined as a master and the other as a slave. The master/slave relationship implies that the master processor initiates a function request through an interprocessor communication block by asserting a function code and request flag. Slave acceptance of the function request is signalled by the slave asserting a busy flag, and completion by negating the busy flag and asserting the acknowledge flag.

One convenient realization of the interprocessor communication block uses a segment of random access memory accessible on a single byte basis with ordinary memory reference instructions. This segment is contained within the logical address space of both master and slave processors. Described below is the hardware using a minimum number of components that implements the common memory segment.

GENERAL FUNCTION

This section details the method of interprocessor communication between a Digital Equipment Corporation (DEC) LSI-11 minicomputer and an Intel 8085 microprocessor. This method eliminates the necessity of having communication registers external to the processor memory. External registers are usually used in two ways to implement short-distance, parallel-data communication between processors: (1) The entire interprocessor communication block is implemented external to processor memory by using hardware multiplexed between the two processors, or (2) more commonly, the entire interprocessor communication block is transferred between processor memories through a single buffer register. The first alternative requires extensive hardware because the interprocessor communication blocks can be large (256 words) for complex functions. The second alternative, although requiring less hardware, complicates the interprocessor communication protocol. Not only must the communication protocol be implemented at the function level but a lower level protocol necessary for controlling word by word transfer must also be considered.

The solution to the above shortcomings is to allow the minicomputer (LSI-11) to access a memory segment in microprocessor physical address space from its own physical/logical address space by memory-cycle stealing. Thus both the LSI-11 and 8085 processors have byte access to this segment with ordinary memory reference instructions. Figure 1 shows a memory map of the two processors. The segment containing the interprocessor communication block resides physically in a read/write portion of the 8085 memory. Yet the LSI-11 also has access to the segment from a portion of its input/output page. The low order byte of each 16 bit word in the LSI-11 maps into sequential bytes within the 8085 address space. The upper byte in each LSI-11 word is unused within this segment. The reason for this mapping is explained below in the section on hardware implementation.

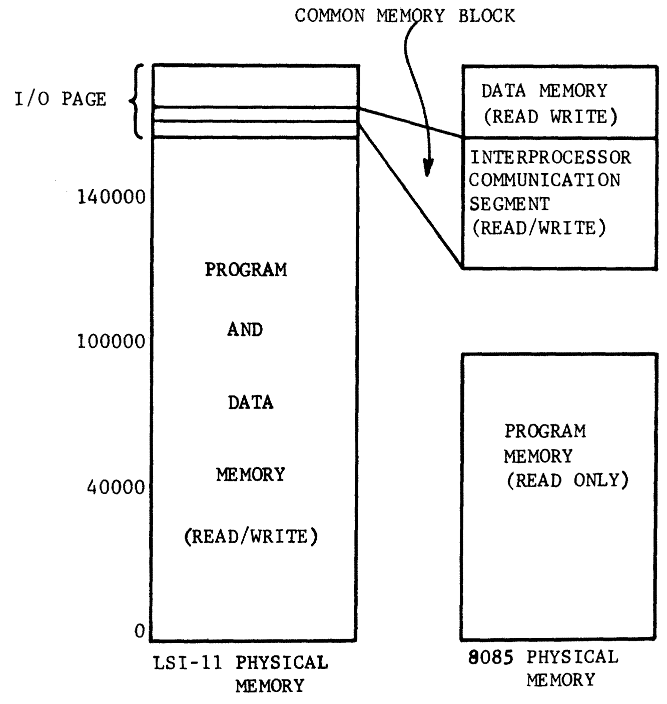
COMMON MEMORY BLOCK
I/O PAGE
140000
100000
40000
0
PROGRAM AND DATA MEMORY (READ/WRITE)
LSI-11 PHYSICAL MEMORY
DATA MEMORY (READ WRITE)
INTERPROCESSOR COMMUNICATION SEGMENT (READ/WRITE)
PROGRAM MEMORY (READ ONLY)
8085 PHYSICAL MEMORY

Fig. 1. Memory map of physical address space of the LSI-11 and 8085 processors showing the common memory segment used for interprocessor communication.

*This research was supported by the U. S. Department of Energy: Contract No. DE-AC02-76CH00016.

CIRCUITRY

A block diagram of the circuit necessary for communication between a master LSI-11 and a slave 8085 processor is shown in Fig. 2. Only the components necessary for interprocessor communication are shown; other components that are usually present in a useful device are omitted for clarity. Signals from the LSI-11 bus (QBUS) are connected to the circuit by the usual bus-interface integrated circuits. Among the QBUS signals used are: (1) DIN, data input, (2) DOUT, data output, (3) RPLY, reply, (4) SYNC, bus synchronize, (5) BS7, input/output page select, and (6) DAL, time multiplexed data and address lines. No interrupt signals are connected because interrupts are not necessary for this method of interprocessor communication. There is no reason to preclude interrupts, however, if the application warrants them. They must be implemented with auxiliary circuitry.

Whenever the LSI-11 accesses memory within the interprocessor communication segment of the input/output page, the ADDRESS COMPARE circuit asserts the SELECT signal. This signal, in coincidence with either the data input (DIN) or data output (DOUT) signal, sets the HOLD REQUEST bistable. The HOLD signal is intended by the manufacturer of the 8085 for use during direct memory access operations. It causes the 8085 to bring the current memory cycle to an orderly conclusion, to place the 8085 bus lines in a high impedance state, to idle the processor and to assert hold acknowledge (HLDA). The 8085 local read/write memory, in this case an Intel 8155, is now available for access from a second source. The HLDA signal initiates the memory cycle that transfers data to/from the LSI-11 bus from/to the 8085 memory. The memory-cycle signals are generated by a microsequencer comprised of a sequence counter, a fast read-only memory containing the pattern of the memory signals and a sequence latch to reclock these signals. Data and high-order address information are transmitted through bidirectional, tristate gates whose timing of opening and closing is controlled by the memory-cycle microsequencer. Low-order address information is taken from the low-order QBUS address and latched in a tristate latch at the beginning of the LSI-11 memory cycle. The end of the memory cycle in 8085 memory causes the HOLD signal to be negated. This in turn negates HLDA, and asserts the reply (RPLY) signal on the LSI-11 bus. The LSI-11 memory cycle now runs to completion.

As noted above, every other byte of the interprocessor communication segment in LSI-11 address space is unused; low order bytes in LSI-11 address space map to sequential bytes in 8085 address space. This mapping results from the difference in bus widths of the two processors. The LSI-11 has a 16-bit data bus while the 8085 data bus is 8 bits wide. Thus for the LSI-11

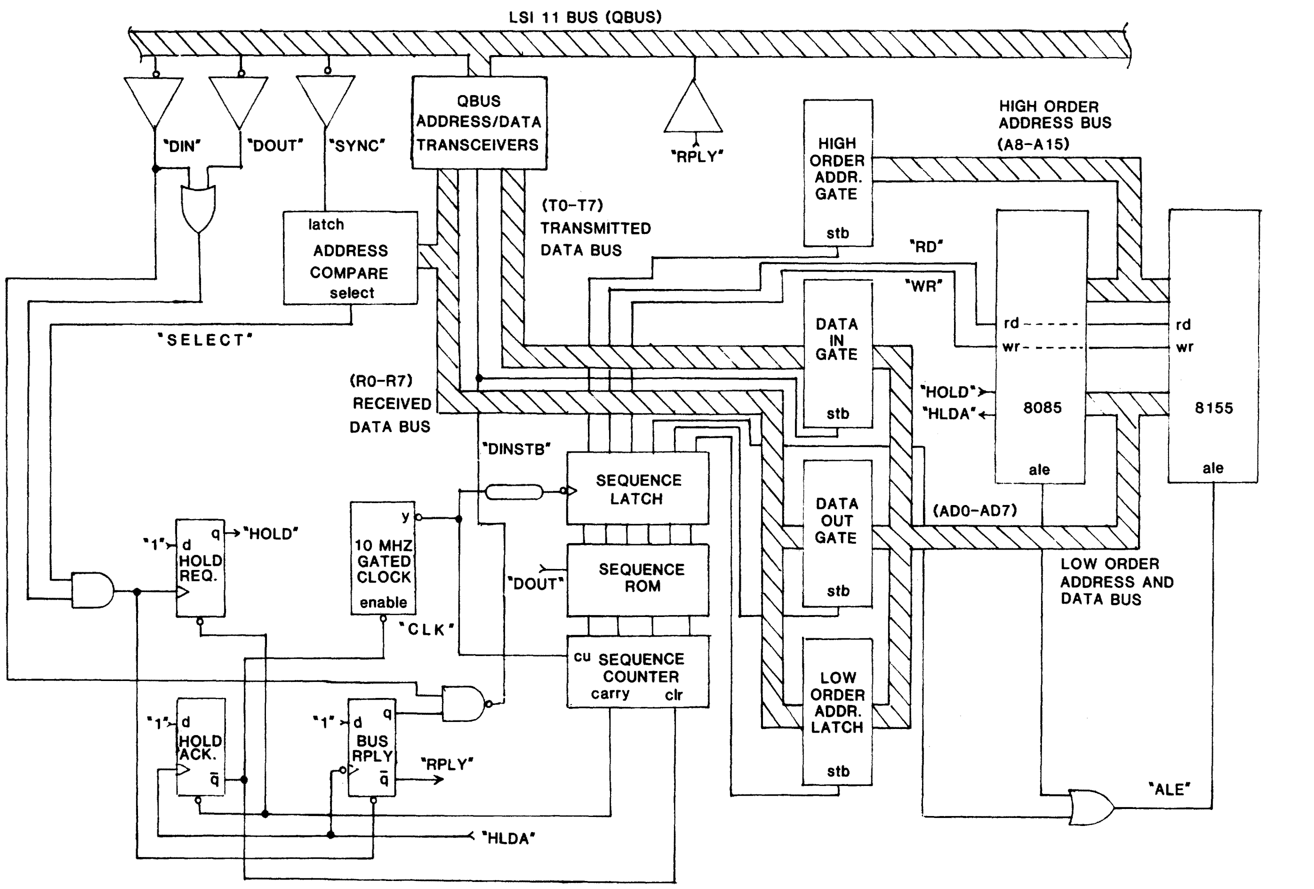

Fig. 2. Block diagram of the circuit necessary for communication between a master LSI-11 and a slave 8085 processor.

to read both bytes simultaneously, 16 bits of data has to be assembled, a byte at a time, within the LSI-11 memory cycle. The complication in hardware to accomplish 16 bit transfer was not deemed worthwhile, especially considering the powerful addressing modes available in the LSI-11 instruction set.

A timing diagram of an LSI-11 memory cycle reading a location in 8085 memory is shown in Fig. 3. The signal names correspond to those shown in Fig. 2. The signal traces are taken from a logic analyzer. The entire memory cycle typically takes ~ 4 μsec, depending on where the 8085 is in its execution cycle at the start of the read cycle from the LSI-11. The second 8085 read pulse (RD) from the left in Fig. 3 is the signal that transfers data to the LSI-11. The other RD pulses shown in the diagram correspond to 8085 program execution cycles.

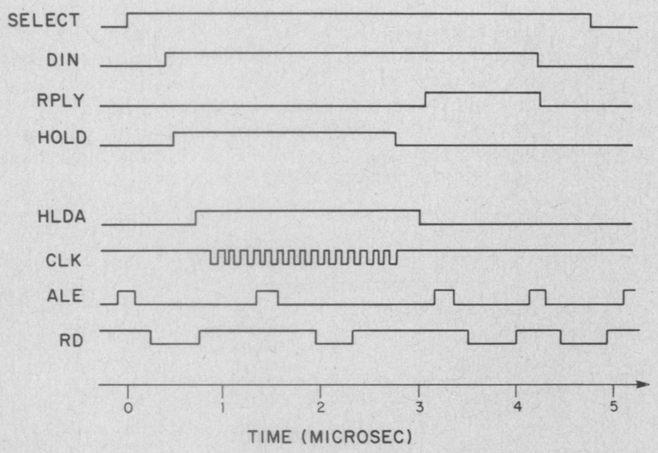

Fig. 3. Timing diagram of an LSI-11 memory cycle reading a location in 8085 memory.

APPLICATION

The technique of interprocessor communication described above has been used to implement a stepping-motor controller module, shown in Fig. 4, capable of controlling four motors. The functions of which the module is capable include for each motor: (1) moving a motor a predetermined number of steps, (2) moving a motor a predetermined number of steps with backlash takeup, (3) search for a mechanical zero position, (4) abort a function.

Each motor is independently controlled by one of four Intel 8085 microprocessors contained on the module. The microprocessor generates the pulse train necessary to move the motor. Many values of gradual acceleration and deceleration, which prevent loss of motor steps, and peak velocity are selectable under microprocessor program control. Limit switch protection is also provided.

Interprocessor communication as described above takes place between an LSI-11 and each of the four microprocessors. (There is no direct communication between microprocessors.) The method has been slightly extended so that one microsequence controller accesses four microprocessors. In this case the control signals and data buses are multiplexed among the four processors.

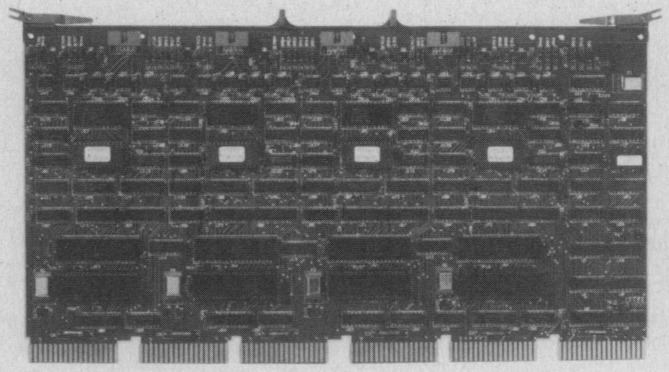

Fig. 4. Picture of the motor driver module, which plugs into the LSI-11 bus. The four microprocessors are clearly visible from left to right. The two large chips at the bottom of each microprocessor section are the 8085 microprocessor and 8155 memory. Code for the 8085 is contained in the PROM with the white label above the 8155. The far right hand side of the module contains the microsequencer and support chips necessary to implement the interprocessor communication.

Each microprocessor has a memory segment containing the interprocessor communication block mapped to consecutive segments of LSI-11 physical/logical address space. The microprocessors, which continuously monitor request flags in these blocks, accept requests for the four basic functions implemented and their associated parameters.

Although an application of this method using only one type of microprocessor has been presented, it is adaptable to any microprocessor chip set that has direct-memory-access capability over a tristate data bus. The Intel 8086, a more powerful 16-bit microprocessor, has these attributes. It has in fact a very similar signal nomenclature and timing, which make it useful for dedicated functions with more stringent requirements than the 8085.