SESSION VII

MICROPROCESSORS IN PSYCHOLOGY: A SYMPOSIUM

MISHA PAVEL, *New York University, President*

What is a microprocessor?

MISHA PAVEL
New York University, New York, New York 10003

A general introduction to microcomputer terminology and concepts is provided.

The purpose of this introduction is to provide an overview of this session and to introduce the terminology and some of the concepts that will be discussed in greater detail by the rest of the session papers.

A block diagram of a typical small computer system is shown in Figure 1. Four distinct blocks can be distinguished: (1) the central processing unit (CPU); (2) external memory; (3) peripherals—mass storage, standard input/output, and man-machine interface; (4) special purpose (experimental) interface.

The different functional units in the system shown here are connected, more or less in parallel, by a number of lines commonly referred to as a bus. The bus mediates transfer of information among the units by carrying address information, data, and control signals. For example, when the CPU transfers data to memory, it activates appropriate control lines, asserts the desired destination memory address, and then outputs data on the bus.

Traditionally, the entire system was built from a multitude of relatively simple components mounted on interconnected printed circuit cards. With the advances of integrated circuit technology, large amounts of circuitry were squeezed into a single chip. At present, it is possible to implement the entire CPU on a single chip that is referred to as a microprocessor. Thus, a microprocessor system is a computer system that uses a microprocessor as its central processing unit.

Typically, the CPU is characterized by its word size and its architecture. Architecture refers to types of registers, their organization, instruction set, and timing. Figure 2 shows an example of a microprocessor CPU and its characteristic architecture. The heart of the CPU is the arithmetic-logic unit (ALU), which executes instructions held in the instruction register (IR) after decoding by the instruction decoder (ID). The program counter (PC) holds the address of the next instruction to be executed in a program stored in external memory.

Figure 1. Block diagram of a small computer system.

Figure 2. Characteristic architecture of a microprocessor CPU.

When the normal sequence of program execution is interrupted by jumps to subroutines or to external interrupts, the CPU saves the current address on a stack in a last-in/first-out manner. The stack is also known as push-down storage. The stack pointer register (SP) holds the location of the last entry on the stack so that the main program execution can be restored from the point of interruption upon completion of the intervening task. When the execution returns to the original sequence, the stack pointer will be set back by one location and will point to the previous entry on the stack. Stack locations may be in a section of external memory or a memory within the CPU.

For a given set of registers, the flexibility of data manipulation depends on characteristics of the instruction set. From a programmer's point of view, a microprocessor instruction set can be divided into four categories: data movement, data manipulation, decision and control, and input/output. Suitable address specification, and a large variety of data manipulation, decision control instructions and flexible input/output instructions are essential for efficient utilization of storage space and processing time.

Since speed of operation is often critical for real-time processing, it is important to note that microprocessors are in general three to eight times slower than larger, conventional computer systems. However, in order to evaluate speed of operation, it is not sufficient to compare single instruction cycle times because instructions may require more than a single machine cycle.

The second component of a computer system that was, until recently, quite bulky and expensive is external memory with information available regardless of location order. Functionally, there are two types of external memory: Read-only memory (ROM) is a permanent memory that can be accessed only to read out permanently stored information. ROM is used to store system programs, bootstraps, or fixed lookup tables. Random access memory (RAM) is a temporary storage capable of both storing and retrieving information. This type of memory is used for temporary storage of operating programs, user programs, and data. An example of a block diagram of a typical RAM unit is shown on Figure 3. In this example, a word stored in each location consists of 4 bits. The information is stored in a three-dimensional matrix: A column and a row identify a unique address (r,c) of a word; each bit of a word is stored at a different depth. Thus, the word

Figure 3. Block diagram of a RAM (random access memory) unit.

stored in Location 3,5 is 1001. When a memory location is to be accessed, the desired address together with appropriate control information is asserted on the bus. This address is decoded to a row and a column, and the transfer is mediated through the input/output register.

A CPU with an external memory alone is capable of data processing, provided that the data and the programs are already in the memory. Typically, this is not the case, and therefore some equipment must be coupled to the system, enabling an operator to enter programs and data into the memory. Such devices (peripherals) mediate standard types of interaction between the CPU, memory, and the rest of the world. It is common to find fewer and less sophisticated peripherals attached to a microprocessor than to other computer systems, because the peripheral equipment is often much more expensive than the microprocessor including memory. Nevertheless, technology has provided an adequate set of peripherals that directly interface with many microprocessor systems. Table 1 contains examples of peripheral equipment for microprocessor-based systems.

The amount and complexity of necessary peripheral equipment depends on a given system application. For example, a microprocessor system dedicated to a

Table 1
Peripheral Equipment

Mass Storage Equipment	Input/Output Equipment	
	Man-Machine	Machine-Machine
Floppy disks Reel-to-reel tape recorders Cassette recorders	Operator console Teletypewriters and CRT terminals Line printers, plotters	Paper tape punch and reader Serial line interface to modem

Table 2
Software Tools

Operating Software	Program Development Software	Diagnostic Software
Loaders	Editors	Hardware diagnostics
Operating systems	Assemblers	I/O device diagnostics
Utility programs	Compilers	Emulators
Application programs		Debugging programs

relatively narrow range of tasks can be operated satisfactorily with only a keyboard and a single display. With increasing requirements on versatility, speed, and ease of system operation (programming), one finds increasing needs for more sophisticated peripheral equipment.

On-line use of microprocessors for control of experiments requires, in addition to standard types of peripherals, a set of special-purpose interfaces connecting particular laboratory equipment to the microprocessor system. Such an interface is usually implemented by a circuit connected to the bus on one side and to the laboratory equipment on the other. Any transfer of data between the laboratory equipment and another device is similar to a transfer of data between the CPU and memory.

Thus far, our discussion has been limited to a description of the hardware aspects of microprocessor system. However, for efficient utilization of a given hardware system, it is necessary to have sufficient software tools so that programming of a microprocessor system becomes a reasonable task. Programming in a

machine language is a tedious, repetitive, and error-prone job. Fortunately, for many microprocessor-based systems, there are sufficient software packages available. Table 2 lists software tools that are desirable for a versatile microprocessor system. In order to operate these software development programs on a microprocessor system, it is necessary that such systems have sufficient external memory space and mass storage peripherals. Alternatively, it is possible to develop and debug a program on a conventional computer system prior to transferring it to the microprocessor. To this end, microprocessor emulators were developed to run on larger machines.

In conclusion, it is useful to point out major advantages and disadvantages of microprocessor systems as compared to conventional minicomputer systems. The major advantages of microprocessors are their low cost, ease of interfacing, and modular structure. Therefore, they are well suited for dedicated applications. Their disadvantages are slow speed of operation, smaller word size, and limited availability of peripherals, software, and technical support.