C h a p t e r **16 MICROPROGRAMMED CONTROL**

The use of control signals, each micro-operation is described in symbolic notation. This notation looks suspiciously like a programming language. In fact it is a language, known as a **microprogramming language**.
Each line describes a set of micro-operations occurring at one time and is known as a **microinstruction**. A sequence of instructions is known as a **micro program**, or *firmware.*

Horizontal microinstruction

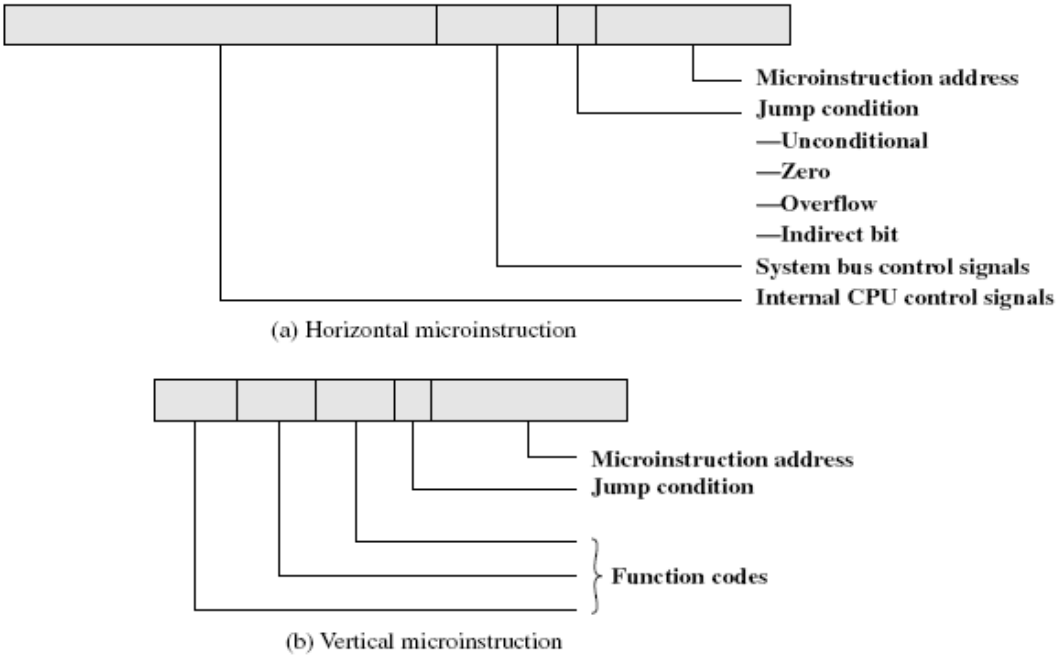

Figure 16.1 Typical Microinstruction Formats

This instruction is indicating the location of the next control word to be executed if a certain condition is true (e.g., the indirect bit in a memory-reference instruction is 1). Also, add a few bits to specify the condition.

The result is known as a **horizontal microinstruction**, an example of which is shown in Figure 16.1a.The format of the microinstruction or control word is as follows. There is one bit for each internal processor control line and one bit for each system bus control line.

- **Micro programmed Control Unit**

Figure 16.4,we see that the control unit still has the same inputs (IR,ALU flags, clock) and outputs (control signals).The control unit functions as follows:
1. To execute an instruction, the sequencing logic unit issues a READ command to the control memory.
2. The word whose address is specified in the control address register is read into the control buffer register.

3. The content of the control buffer register generates control signals and next address information for the sequencing logic unit.
4. The sequencing logic unit loads a new address into the control address register based on the next-address information from the control buffer register and the ALU flags.

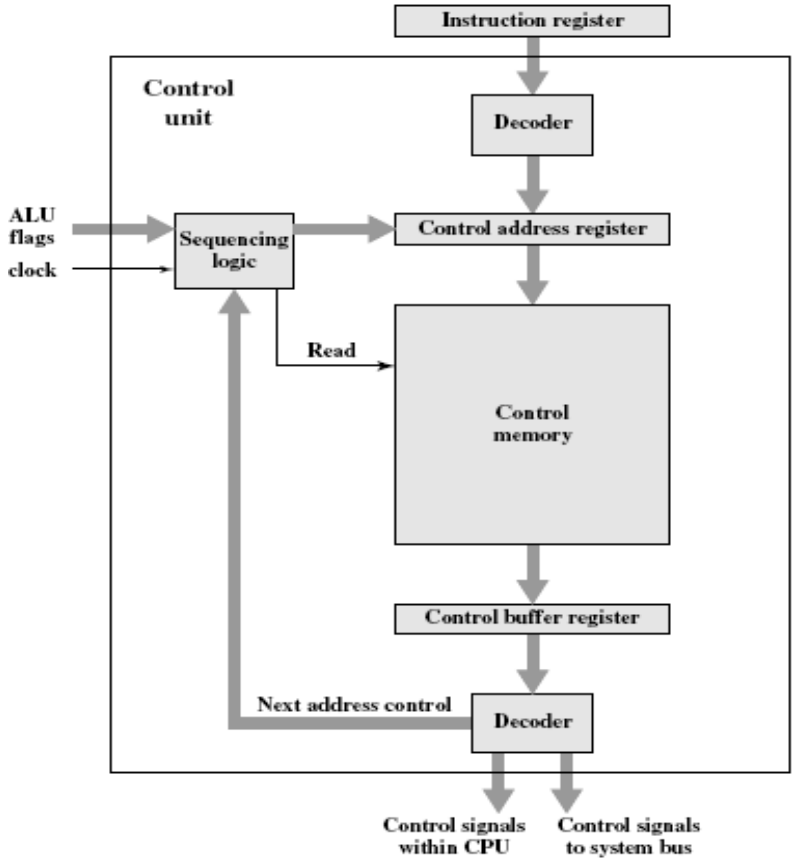

Figure 16.4 Functioning of Microprogrammed Control Unit

Figure 16.4 shows two modules labeled *decoder.* The upper decoder translates the op-code of the IR into a control memory address. The lower decoder is not used for horizontal microinstructions but is used for **vertical microinstructions** (Figure 16.1b).

- **Wilkes's Micro programmed Control Unit**

Wilkes first proposed the use of a micro programmed control unit in 1951 [WILK51]. This proposal was subsequently elaborated into a more detailed design [WILK53]. It is instructive to examine this seminal proposal.
The configuration proposed by Wilkes is depicted in Figure 16.5.The heart of the system is a matrix partially filled with diodes. During a machine cycle, one row of the

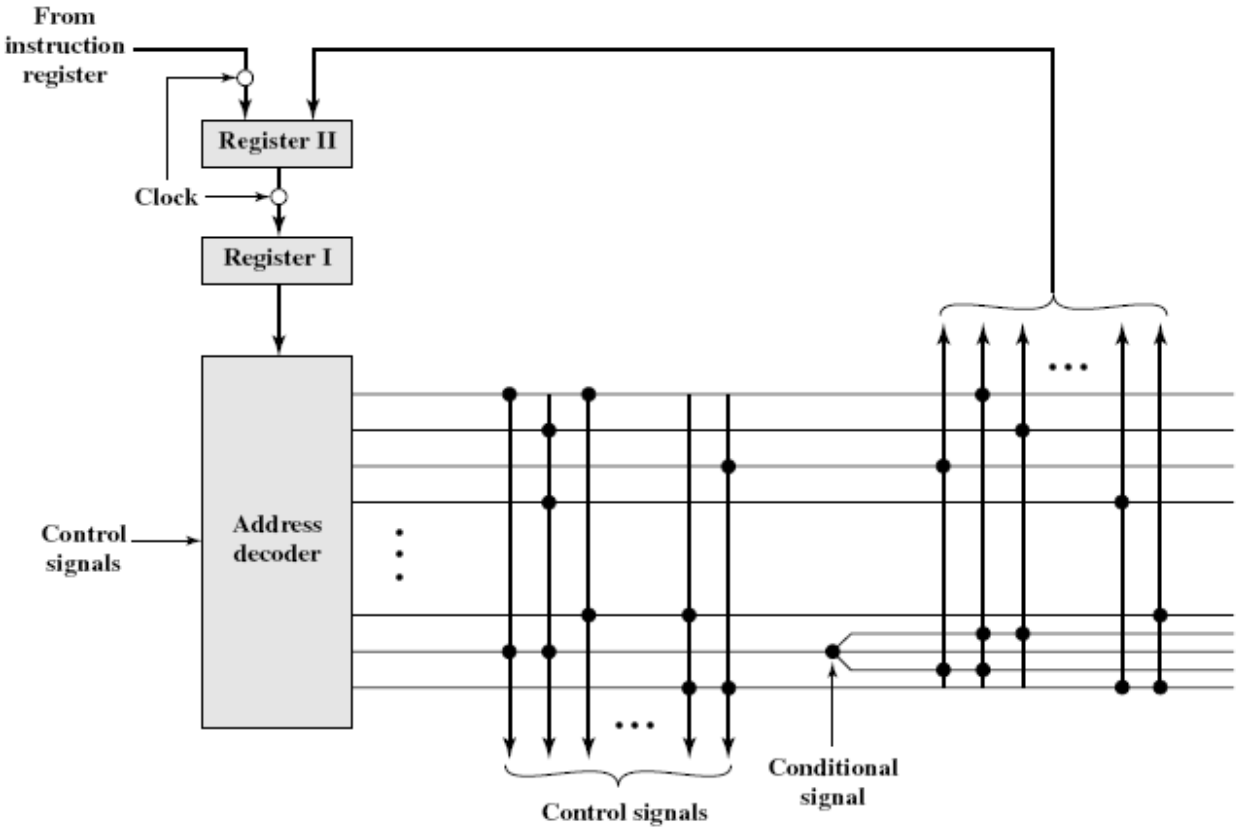

Figure 16.5 Wilkes's Microprogrammed Control Unit

Matrix is activated with a pulse. This generates signals at those points where a diode is present (indicated by a dot in the diagram).The first part of the row generates the control signals that control the operation of the processor. The second part generates the address of the row to be pulsed in the next machine cycle. Thus, each row of the matrix is one microinstruction, and the layout of the matrix is the control memory.

- **Types of Parallel Processor Systems**

A taxonomy first introduced by Flynn is still the most common way of categorizing systems with parallel processing capability. Flynn proposed the following categories of computer systems:

• **Single instruction, single data (SISD) stream:** A single processor executes a single instruction stream to operate on data stored in a single memory.

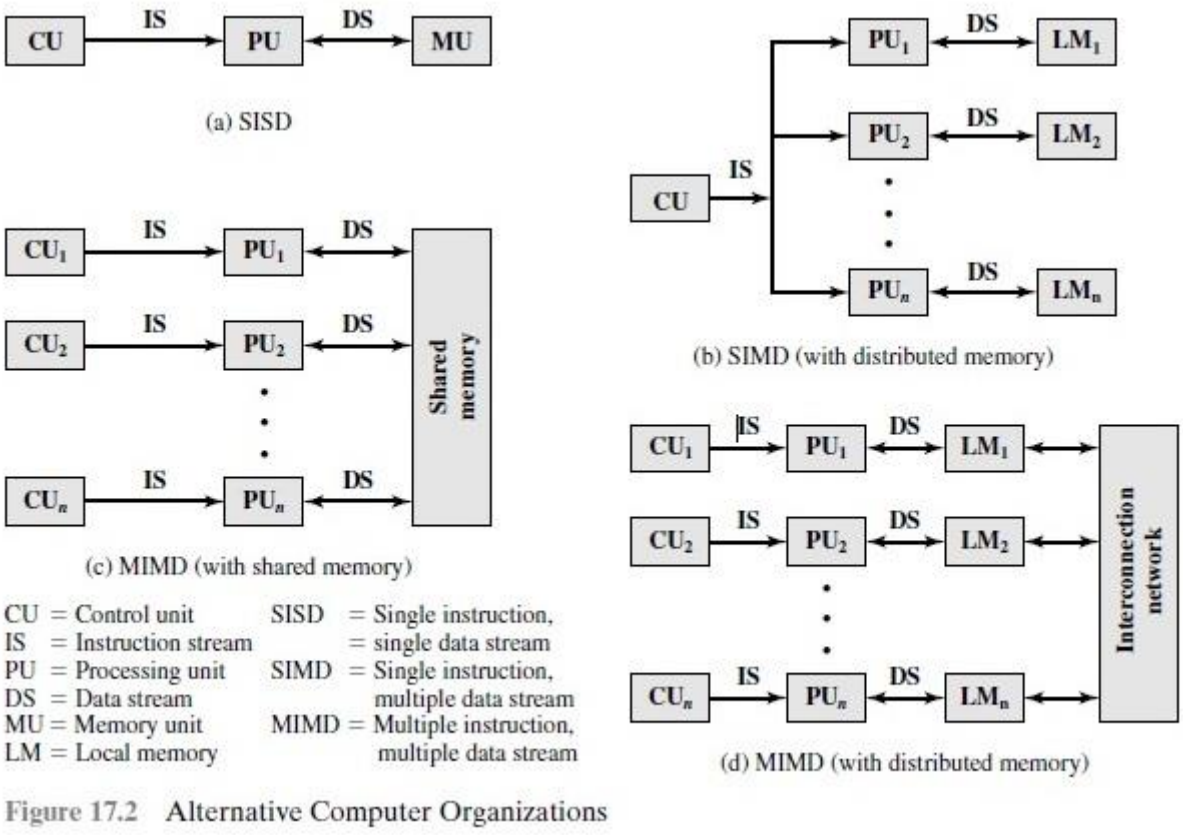

Figure 17.2 Alternative Computer Organizations

Uni-processors fall into this category.

• **Single instruction, multiple data (SIMD) stream:** A single machine instruction controls the simultaneous execution of a number of processing elements on a lockstep basis. Each processing element has an associated data memory, so that each instruction is executed on a different set of data by the different processors.

• **Multiple instruction, single data (MISD) stream:** A sequence of data is transmitted to a set of processors, each of which executes a different instruction sequence. This structure is not commercially implemented.

• **Multiple instructions, multiple data (MIMD) stream:** A set of processors simultaneously execute different instruction sequences on different data sets. SMPs, clusters, and NUMA systems fit into this category.