

# Application of microprogrammed Emulation

Microprogramming witnessed an inordinate time delay between conception and fruition. the causes have been many available, technology dearth of experimentation and research tradition etc.

As these issues were overcome an aura of pragmatism surrounded. microprogramming with the development of writable control stores and user microprogrammable machines practitioners have facilities to explore new, applications yet in general applications of microprogramming are still in their infancy Interpretation of machine language instructions continues to be the most common application of microprogramming Here we consider some of the typical applications of .microprogramming

Historically emulation has connoted the facilities added to a computer system to enable it to execute with some efficiency programs developed for another system.

These facilities included hardware components for handling special features of the emulated machine. Adding such hardware resources essentially changes the architecture of a computer and generally lies outside the of most computer users. Microprogramming, however generally provides the facilities to execute with some efficiency programs developed for other systems and lies within the potential capabilities of many computer users. Accordingly microprogramming has become central to the modern concept of .emulation.

We use the term "emulator" to describe a complete set of microprograms which when embedded in a control store define a machine. We shall call a machine which is realized by an emulator a virtual [or target machine] and the machine which supports microprograms a host machine.

To define a machine microprograms must implement the architecture of the machine. Since the architecture of a Machine consists of a conceptual structure and a functional behavior the microprogram of the host machine must :

1. Map of the components of the virtual machine into those of the host machine
2. perform the “machine language” instructions of the virtual machine

performing the machine language instructions is accomplished by microprograms interpreting the machine language instructions.

As interpreting machine language instructions is also called simulation emulation is sometimes called microprogrammed emulator

Many applications of microprogramming are considered variations on the basic emulation concept.

In these schemes virtual machines which efficiently support programs for a particular application area are designed, the virtual machines are then emulated on some host machine.

Flexibility in defining virtual machines, their generality depends on the organization of the hardware resources of the machine, the extent to which a computer serves as a universal host machine measures its capability as a general emulation vehicle. A primary objective of a majority of the commercially available microprogrammed computers is the efficient emulation of a particular machine.

Machine language interpretation therefore remains a primary application of microprogramming and often is used as a definition for the hardware resources and microinstruction repertoire of such machines are often specialized for this emulation and thus may be difficult to adapt to general emulation problems.

The representation of data is of primary importance in this regard, for example; a machine whose registers and operations were designed to handle thirty-two bit integers in two's complement notation would probably not be a good host for emulating a machine whose basic data type is thirty-six bit integers in sign-magnitude representation.

A subject of current research is the design of microprogrammable computers that serve as efficient hosts for a variety of virtual machines.

The MATHILDA computer described in section 6.8 is an example of such research.

From book >>>

# Foundations of Microprogramming

Foundations of Microprogramming