w3 COMP IBM.

#01 zOS Tid-Bits@CPO (microcode/multiprocessing concepts)

Microcode concepts

To better explain how z/Architecture instructions are implemented on a mainframe, let's introduce the concept of *microcode*. The vast majority of the z/Architecture instruction set is implemented through microcode.

Microcode is a design option (not an architecture option) that is used to implement Central Processing (CP) logic. To make it simple, we can divide the CP into two pieces: *data contro*l and *data flow* (also called ALU).

Instructions are really executed in the data flow (where data is transformed) however, the sequence and timing of each of the multiple operations done in the data flow is ordered from the data control.

> *It is similar to an orchestra: musicians (like pieces of data flow) know how to play their instrument, but they need guidance and tempo from the maestro (the data control) in order to execute.*

In a microcoded CP, for each possible instruction of the instruction set, there is one micro program that tells data control what to do in order to execute the instruction in the data flow. The micro program has, in a special language, the sequence of orders to be sent by the data flow. These micro programs are loaded in a special internal memory in the CP, called *control storage*, at power-on reset time (machine activation). Decoding an instruction consists of finding the address in the control storage of its micro program. Note: The superscalar design of the mainframe microprocessor allows for the decoding of up to two instructions per cycle and the execution of three instructions per cycle. Execution takes place in order, but storage accesses for instruction and operand fetching may occur out of sequence, (*more on this is a subsequent Tid-Bit).*

The opposite of microcoding is *hardwiring*, in which

the logic of data control for each instruction is determined by Boolean hardware components. The advantage of microcoding is flexibility, where any correction or new function can be implemented by just changing or adding to the existent microcode. It is also possible that the same CP may switch instantaneously from one architecture (such as from ESA/390 -31bit to z/Architecture -64bit) to another by using another set of microcode to be loaded in another piece of its control storage.

Multiprocessing concepts

Allowing many processes at the same time in a system can cause a single CP to be heavily utilized. In the 1970s, IBM introduced a *tightly coupled multiprocessing complex*, allowing more than one CP to execute more than one process (task) simultaneously. All these CPs share the same real storage (main storage, central storage, and real storage are different terms for the same kind of memory), which is controlled by a single z/OS copy in such
storage.

To implement tightly coupled systems, the following items are needed in the architecture:
* Shared main storage, which allows several CPs to share the same main storage
* CP-to-CP interconnection and signalling
* Time-of-Day Clock (TOD) synchronization, to guarantee that all TODs in the same CEC are synchronized

A *system* is made up of hardware products, including a central processor (CP), and software products, with the primary software being an operating system such asz/OS. Other types of software (system application programs, end-user application programmatic tools) also run on the system. The CP is the functional hardware unit that interprets and processes program instructions. The CP and other system hardware, such as channels and storage, make up a central processor complex (CPC).

z/Architecture defines that a single CP processes one—and only one—instruction from a program at a time. The MVS operating system (z/OS) manages the instructions to be processed and the resources required to process them. When a single copy of the z/OS operating system (MVS image) manages the processing of a CPC that has a single CP, the system configuration is called a *uniprocessor*.

When you add more CPs to the central processor complex (CPC), you add the capability of processing program instructions simultaneously. When all the CPs share central storage and a single MVS image manages the processing, work is assigned to a CP that is available to do the work. If a CP fails, work can be routed to another CP. This hardware and software organization is called a tightly coupled multiprocessor.

A tightly coupled multiprocessor has more than one CP, and a single z/OS image, sharing central storage. The CPUs are managed by the single z/OS image, which assigns each of them work.

Need to be added to the z Distribution List?

Document details

Date: Nov. 16, 2007

Content owner:
John H Kettner

For IBMers and IBM business partners only. Not intended for customers.

Report handling guidelines

Content provider:
IBM SWG Competitive Project Office

Feedback

Questions? Comments? Contact the Competitive Project Office at **swgcpo@us.ibm.com**

Return to the main **Competitive Project Office** page:
https://w3

03.ibm.com/sales/competition/compdlib.nsf/pages/swgcpo

Tag and save this page to your dogear favorites.

What is dogear ?