

7030 Data Processing System Bulletin

IBM 7030 Systems Program Package

A data processing system is not complete without the systems programs specifically designed for use with the computer. The IBM 7030 Systems Programs have been defined and are presently being written and tested.

These programs will assist IBM 704, 709 or 7090 users in their transition to the 7030, will permit applications to be programmed easily and brought quickly to the computer, and will provide efficient and productive usage of the computer system.

The programs that have been designed are a simulation and assembly program in 704, 709, and 7090 versions, additional simulation programs, a master control program, and a set of processors that operate on languages at several levels of sophistication.

704-709-7090 Programming Package for the 7030

The transition from one computer system to another is always a tedious matter. In order that the transition may be as smooth as possible, an assembly program, Strap I, and a 7030 Simulator that accepts the output of the assembly program have been prepared in versions that run on the 704 and 709. The 7030 Simulator also runs on the 7090.

This Package permits programming to be done and partially debugged in advance of installation of the 7030.

704 and 709 Simulators

Two programs, a 704 Simulator and a 709 Simulator, are part of the Systems Program Package. These routines will permit 704 and 709 programs to be run on the 7030 with a minimum of reprogramming. They should be used when it is uneconomical to rewrite a program completely because its remaining useful life is short.

Master Control Programs

The Master Control Program provides an automatic operating system, a flexible input-output system, debugging aids, automatic interruption handling, and a first level of multiprogramming.

Operation of the 7030 Data Processing System is automatic in two respects:

1. A queue of programs may be loaded and executed sequentially without the intervention of a human operator;
2. Tape mounting and dismounting instructions are issued by the Master Control Program to the operator through the console printer. This is made possible by virtue of the fact that programs address all input-output units symbolically. Input-output units are assigned values corresponding to absolute physical addresses prior to program loading.

In the case of tape, the reel number which is marked on a reel for identification purposes is also carried as a tape label (i. e. , the number is contained in a record written on the tape). The Master Control Program can then verify whether or not the operator has carried out mounting instructions correctly.

Another advantage of addressing input-output units symbolically is that setup time for the next program of a queue can generally be overlapped with the execution of the program that currently occupies the computer. The Master Control Program plots the overlap logic, keeping as far ahead as possible.

Symbolic input-output unit addressing is also essential to any form of multiprogramming to insure that two programs to be run concurrently will not conflict by requiring the same physical input-output unit. Although the programs might use the same type of unit, the Master Control Program will assign different physical units to each program.

Input and output can occur simultaneously with the execution of a problem program. This multiprogrammed operation of the 7030 Data Processing System permits program and data files to be transferred from card to tape and output data to be printed, punched, or recorded on tape while an independent problem program is being executed.

The Master Control Program runs the system as a whole. It allocates space for systems programs, calls them in from storage for execution under the control of various options (e. g. , the "compile and go" option, or certain debugging options), and is the agent by which other service routines (such as an installation logging routine) may be added conveniently to the system as new components.

Strap II

The 7030 Assembly Program, Strap II, accepts programs written in a symbolic language that is essentially equivalent to absolute machine language. Mnemonic operation codes and symbolic addressing eliminate much of the routine bookkeeping that confronts the machine language programmer. Strap II translates symbolic language programs to machine language.

SMAC

The 7030 System Macro Language and Processor, SMAC provides a more sophisticated programming medium that is designed as a tool for systems program extension and expansion, and as a link between the Assembly Program and the highly sophisticated compiler, Fortran, that completes the Systems Program Package.

FORTRAN

The language of Fortran was originally designed as a mode of expression closely modelled after the language of mathematics, that could be directly translated by a computer into machine language. The 7030 Fortran processor will translate Fortran statements to SMAC statements, thereby materially reducing duplication of effort in the preparation of the Systems Program Package and making the components of the Package more accessible to the system programmer.

Fortran language is intended to be capable of expressing any problem of numerical computation. In particular, it deals easily with problems containing large sets of formulae and many variables, and its notational features include subscripting abilities which facilitate the manipulation of arrays.

A considerable body of practical experience with the use of Fortran in putting problems on machines has now been accumulated. Three exceptional advantages which have been observed in connection with this extensive usage are:

1. The language is essentially machine independent, and is available for use in all major IBM programming systems. The transition period potentially involved in installing an advanced system is considerably eased as a result, since the changes required to run old Fortran-coded problems on a new machine are generally minor.
2. The amount of specialized training required to program problems in Fortran language is very much less than that which is needed for machine language coding. As machines themselves become more complex and intricate in order to gain in power, the significance of this advantage increases.
3. The amount of time required in an individual case to program a problem is ordinarily very much less when Fortran is used than when the problem is "hand-coded."