

The IBM System/360 Model 91: Some Remarks on System Development

Introduction

• System objectives

The primary goal of the System/360 Model 91 development program was to produce the highest performance capability that advanced design philosophy and extensions of System/360 circuit technology could achieve, within a balanced development schedule. "Performance," as used here, means general computer availability and high-speed execution of general problem programs.

A system, of course, is composed of many parts other than the processor, and the design was implemented to incorporate a number of existing models and modules of peripheral equipment. For example, the system was to be optimized for a $3/4\text{-}\mu\text{sec}$ memory, the same basic unit used in System/360 Models 65 through 75.

Another consideration in system development was manufacturability. The system was designed so that timing considerations, "de-skewing" and tuning of the clock did not become inordinately complex. Similarly, component design for any part which is used extensively was carefully matched to manufacturing requirements.

A major aspect of the Model 91 development effort was the checking and serviceability philosophy. It was recognized that all arithmetic operations and all data transfers should be fully checked. Some of this checking, such as parity checking to the byte level on data transfers, was made necessary by the compatibility requirement. Experience has shown, however, that the size and complexity of a large system demand an active concern for checking features on the execution of as many operations as possible. Similarly, with respect to serviceability, it was felt at the outset that display provisions for every storage element in the system should be provided. Such provisions, together with the System/360 "log out" feature which provides for a complete readout of the machine state when an error is incurred, served as the initial basis for serviceability design.

• Technology objectives

The obvious objective of the technology[†] group was to develop a high-speed, very efficient technology and place it in production within the time provided by the development schedule. It was recognized at the outset that if this goal were to be attained, tooling compatibility would have to be maintained between the Model 91 technology—or ASLT (Advanced Solid Logic Technology), as it has been named—and SLT (Solid Logic Technology).

One could find references in the literature to kilomega-cycle transistors as early as 1962 and references reporting work on 2-nsec circuits in 1963. Ultra-fast circuits, however, do not necessarily make an ultra-fast computer. For example, 2 nsec is approximately the time required for the electromagnetic wave representing the state of a circuit to travel along 12 in. of printed circuit conduction path; thus, if the particular implementation of the ultra-fast circuits requires that, on the average, the circuits be spaced 12 in. apart, one will have a delay of not 2 nsec, but 4 nsec per circuit. Furthermore, circuit delay quoted in the literature sometimes fails to consider the effect of interactions among circuits connected to the input and output of the device under test. These so-called "loading effects" typically add from 2 to 4 nsec to the basic circuit delay. Thus, when all factors have been considered, one finds that the circuit delay in the systems environment may not be 2 nsec, but may in fact be much closer to 6 nsec. As a result of this confusion, the performance objectives for the ASLT technology were always stated in terms of delay "in the environment." It was estimated that a 5-nsec circuit measured as described above would be required if the

* Present address: Northwestern University, Evanston, Illinois.

† "Technology" as used here will be taken to mean those aspects of components, circuitry, physical structure, etc. which are considered separately from system architecture, programming, organization and gross-function algorithms. The latter have been identified as "systems."

machine were to meet its performance objectives. To achieve such an objective would require a balanced effort that would simultaneously develop circuits, devices, modules, and second-level packaging.

The basic circuit, of course, had to be as fast as possible. Equally important, however, were the rules by which the circuits could be interconnected, so that the system's designer was not unduly constrained during the logic design and card layout phase of the program. In high-speed computer design, predictable circuit delays can be as important as short delays. Thus, a rather extensive effort was undertaken to be certain that the various combinations of circuits and printed conduction paths would behave in a predictable manner. In certain sections of the machine, pulse-width preservation (and thus the ability to operate at a high repetition rate) was extremely important. Hence it was necessary to guarantee that the circuit rise and fall times would be as short as possible. It would have been unacceptable, for example, to have a circuit that had a 1 nsec delay but a 3–5 nsec rise or fall time.

The basic packaging density had to be significantly increased to assure that the performance of the circuit would not be wasted by excessively long printed circuit paths. At the module level, a packaging density increase of at least a factor of three over SLT was required. An increase in the circuit packaging density, however, would put additional loads on the cooling and power signal distribution systems; significant modifications both to those areas and to the printed circuit card and back panel supporting structure were therefore required.

Results

• System

After consideration of all requirements, the prime specification was established as the execution of an average of one instruction per machine cycle. The basic machine cycle was defined as that time required to decode a System/360 instruction, which for Model 91 is 60 nsec.

The concept of processing one instruction per cycle has now become the basis for the entire system and forms the interface between major areas of the system. Thus, the storage unit and its communication system must be able to provide the processor with data and instructions at a rate sufficient to permit execution of one instruction per cycle. Similarly, the instruction unit must be able to distribute, queue, and organize its resources at a rate consistent with this same (one instruction per cycle) target. Finally, the execution unit itself must be designed to provide an average execution rate of one instruction per cycle. Clearly, all the processing required for any one instruction cannot be completed in one machine cycle. In-

stead, as many as nine or ten instructions are in various stages of processing at the same time. The instruction unit coordinates the acquisition, distribution, and execution of the instructions by buffering both instructions and data. The design of this unit is described in the paper by D. W. Anderson, F. J. Sparacio and R. M. Tomasulo, entitled "The IBM System/360 Model 91: Machine Philosophy and Instruction Handling."

The interface from the instruction unit to the storage communications unit and thence to the storage unit itself, and the subsequent achievement of a regular supply of instructions and data from the storage unit are discussed in the paper, "The IBM System/360 Model 91: Storage System," by L. J. Boland, G. D. Granito, A. U. Marcotte, B. U. Messina and J. W. Smith. An interface also exists between the main storage unit and the hierarchy of storage and input/output equipment. The important point here is that, despite the delay that accrues when the desired words must be accessed, the flow of information is maintained at a rate of one instruction per cycle, even across this hierarchy of boundaries.

Even when the information is available however, its execution, especially in important floating-point programs, is quite difficult at the target rate of one instruction per cycle. Two major developments represented in the Model 91 accomplish this rate. The first development is an algorithm that permits the decomposition of apparently dependent instruction sequences into independent elements; execution may proceed regardless of the order of these elements. The algorithm is described by R. M. Tomasulo in the paper entitled "An Efficient Algorithm for Exploiting Multiple Arithmetic Units."

In addition to the Tomasulo algorithm, the actual execution of floating-point instructions must be considerably enhanced over prior art in order to meet the goal. The second development, described in the paper by S. F. Anderson, J. Earle, R. E. Goldschmidt and D. M. Powers, achieves this with a three-cycle multiplication using a highly efficient iterative design approach. Sharing the same iterative hardware as the multiply function is an iterative divide scheme based on the Newton-Raphson algorithm. This is the first known hardware implementation of such a divide scheme.

The Model 91 system has a substantial performance improvement compared with previous IBM systems. (Table 1 makes this comparison.) In particular, note that floating-point programs can be executed more than a hundred times faster than on a machine such as the IBM 7090. Programs with many branch instructions, however, suffer somewhat by comparison because of the delay that results from the access to storage.

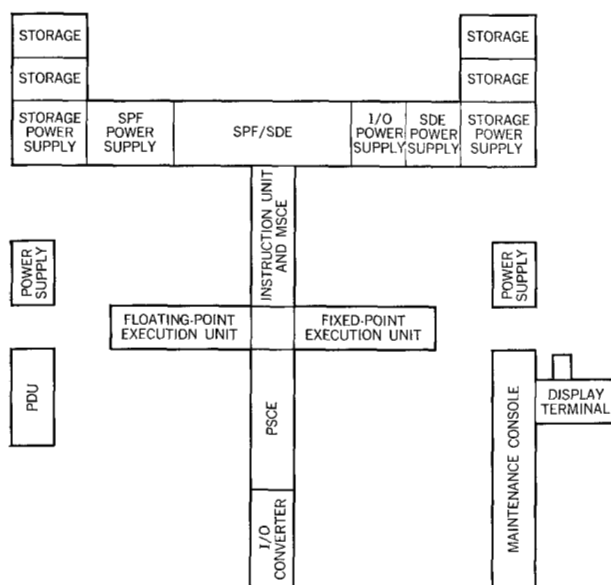
The system is also compatible with other models of System/360. This compatibility, however, is qualified by three specific exceptions:

1. Decimal option instructions are not provided in the hardware because the system was optimized for the scientific user, and thus it was felt that there would not be substantial interest in these instructions. However, programming simulations can be provided to effect the execution of these instructions.
2. Data- and address-dependent interrupts are not discretely noted, since instruction processing is concurrent; instead, these cause an "imprecise interrupt."
3. The quotient of a floating-point divide instruction may differ somewhat from that obtained from other models of

Table 1 Comparison of machine performance on various problem kernels.

Kernel Type	7090	Relative Internal Machine Speed	
		System/360 Model 75	System/360 Model 91
FORTRAN 1F (Branch instructions predominate)	1	6.6	14.5
Matrix Multiply (Floating-point instructions predominate)	1	6.85	102
Polynomial Evaluation (Floating-point instructions predominate)	1	7.6	95.7

Figure 1 Schematic of system configuration (I/O and peripheral storage not shown).



the System/360 family. Although integrity of integers is preserved and the resulting precision is at least as good as that in other machines, the nature of the algorithm is such that somewhat different fractions may result.

The Model 91 can assume a number of configurations, but these vary only in the size and location of storage and peripheral equipment. The processor itself is organized to accommodate the instruction handling, storage control and arithmetic functions that are common to all variations, and its appearance in the same form is a characteristic of every Model 91 system.

One possible configuration is shown in Fig. 1. The processor, or CPU, is assembled in the form of a cross. One arm of the cross contains the floating-point execution unit, and another contains the fixed-point execution unit. A third arm contains the instruction unit and the main storage control element (MSCE), and the fourth arm contains an I/O converter and the peripheral storage control element (PSCE). Assembled around the CPU are free-standing frames such as the power distribution unit (PDU) and two power frames which feed regulated dc power directly to the CPU itself. Also adjacent to the CPU are the maintenance console and its associated display unit. The main storage element is shown adjacent to the main storage control element and contains the power supplies used to operate storage. In addition, the storage frame contains a storage distribution element (SDE) and the storage protect feature (SPF).

The Model 91 processor can communicate with a great variety of I/O equipment and peripheral storage, via the PSCE and the I/O converter. Two types of channels may be used. A possible channel layout is shown in Fig. 2, in which two Selector channels are connected directly to the processor and the processor storage. Each of these Selector channels controls high-speed I/O equipment as indicated. In addition, a Multiplexor channel may also be provided to control communication with slow-speed I/O equipment and to provide additional Selector sub-channels for such devices as magnetic tape units.

As indicated in the initial remarks, much attention has been devoted to checking and serviceability. The resulting system has all its arithmetic operations and data transfers checked. Also, the contents of all register positions (more than 6000 in all) are displayed either on the maintenance CRT console or with indicator lights, and features such as log-out are implemented via the maintenance console.

Careful attention has been paid to the detection of failures; indeed, for some portions of the machine, failures can be detected, the error pattern stored, and the failure later diagnosed to the level of the failing card, all from the maintenance console. It is possible to enter information to any buffer register or any level of a queue or to inject an error (to check the parity circuitry) from the console.

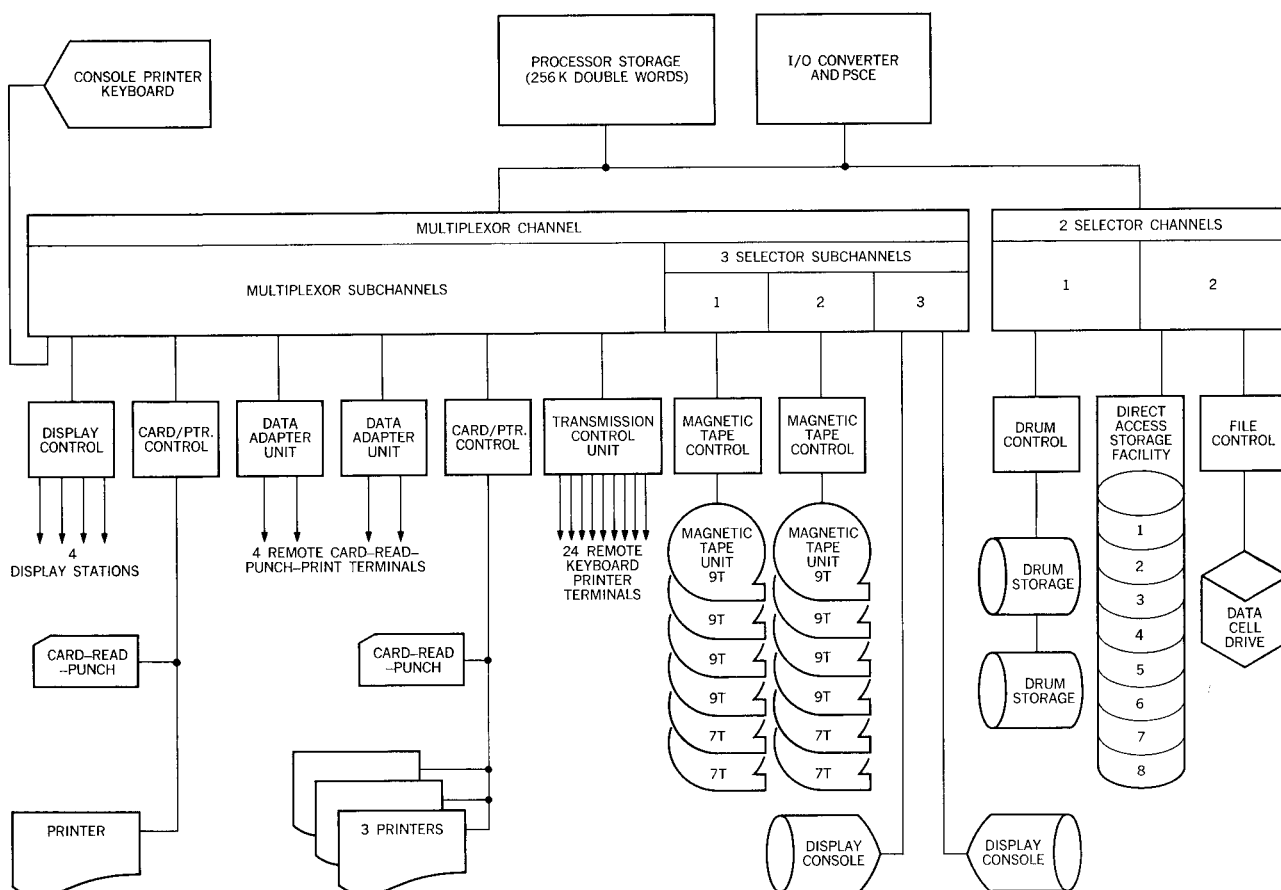


Figure 2 Example of I/O configuration.

Also, the timing of the machine may be varied so that only one instruction is being executed at a time, and the rate of processing may be slowed from a multiple-cycle rate to single-cycle (manual or stepped). Complete memory testing may be performed by searching or comparing all of storage with respect to a reference pattern. Memory contents may be displayed sixteen 72-bit words at a time. Re-try is also possible on a machine check interrupt; i.e., the program will be re-run from the last checkpoint.

Attention was given to timing and the attendant logical design to minimize the potential manufacturing problems. All the latches in the machine are of the fail-safe variety; that is, they are not sensitive to skewing between data and the control lines. (This particular method of timing is described in the paper by S. F. Anderson et al.)

• Technology

The results of the technology development program were quite successful in meeting the design objectives. Circuit

delay (without wire) ranged from 1.6 to 2.5 nsec, depending upon the environment in which the circuit was placed. The output rise time varied from 1 to 1.5 nsec. The circuit fan power (that is, the number of inputs that a given circuit may have and the number of similar circuits that a given circuit may drive) was significantly higher than had been attained in any previous high-speed circuit family. The effective number of inputs for a given circuit was 16, and each of the standard logic blocks could drive 10 others. A technique was developed that allowed the systems designer to predict the circuit delay within 10-12%, and when prediction errors in that range did occur, the measured delay was in general smaller than predicted. A general comparison of the technology with high-speed SLT is shown in Table 2.

An important checkpoint was established early in the development phase of the engineering program. A standard, or test model, was jointly developed by the technology and systems development groups and was implemented using the ASLT technology. The logic implemented represented

Table 2 General comparison of high-speed SLT and ASLT.

	<i>High-speed SLT</i>	<i>ASLT</i>
Circuit speed	5.5 nsec	1.8 nsec
Circuits/back panel board	800 (max.)	5000 (max.)
Transmission line impedance	100 Ω	50 Ω and 90 Ω

one version of a floating-point adder that was under consideration by the systems group during development. The logic design and card layout were done by the systems group; the construction and evaluation were done by the technology group.

The logic path consisted of 23 levels of logic, packaged on ASLT cards. One hundred and forty-five circuits, representing the 23 logic stages, plus the additional fan-out and fan-in loading dictated by the systems design, were used. A total of 247.6 in. of printed circuit, wire, and cable interconnected the 23 levels of logic. For the systems group to meet their machine performance objectives, it was essential that the logic delay through this path, including the delay introduced by the printed circuit wire, be less than 115 nsec (or 5 nsec per logic level). The actual delay measured after the model was constructed was 103 nsec, representing 4.48 nsec per logic decision, measured in the system environment. It was also found that a 6 nsec pulse width could be used.

As a check, the delay was calculated using delay prediction techniques. The prediction program indicated that the delay through this network would be approximately 108 nsec and that the minimum pulse width that could be propagated through this network would be approximately 5.5 nsec.

Once the basic performance objectives of the development program had been achieved, the technology was released to the manufacturing plants concerned. At this writing, the modules, cards and boards are all in production. The technology has been improved somewhat during its manufacturing phase. This improvement derives primarily from refinements in transistor fabrication and from the more positive control of process variables provided by automatic manufacturing tools. The result is that the delay through the test model, using current production modules, has been reduced from 103 to just under 100 nsec.

A more detailed discussion of the various aspects of the technology development programs is contained in three papers in this issue. "ASLT Circuit Design and Engineering," by A. R. Strube, R. F. Sechler and J. R. Turnbull, describes the basic circuit design philosophy, the per-

formance objectives, and the delay prediction techniques. Another paper, by J. Langdon and E. J. Van Derveer, describes the design and development of the high-speed switching transistor used in the circuit described by Strube et al. A final paper, by R. H. F. Lloyd, outlines the techniques which have extended SLT to the packaging of high-speed circuits. Particular attention is paid to the increase in density achieved at the module level.

Some considerations for the future

It has become evident that the target design goal of the execution of one instruction per cycle was a prudent one; targets much in excess of this figure would require exponentially more hardware. We infer from this that, in the future, order-of-magnitude improvements in computer performance due to systems organization alone will be unlikely, at least where emphasis is placed on simplex instruction processing (single-line instruction processing as against multi-processing). The converse of the last statement is, of course, that, if substantially greater performance improvements must be achieved by means of systems organization, probably the most fruitful areas would be based on the intimate connection of many independent processors.

Acknowledgments

Adequate acknowledgment for a development effort of this size is impossible. We make a sincere attempt below, knowing beforehand that it will be inadequate.

Primary direction was given by Mr. B. O. Evans.

Mr. L. E. Kanter provided the overall engineering management, and Mr. D. J. Galage was manager of the development group.

Drs. G. Amdahl and T. C. Chen worked tirelessly to insure both market and management acceptance for the system.

Although not discussed in detail in the papers mentioned above, an important feature of the Model 91 is its means for accommodating variable field-length instructions. Messrs. R. J. Litwiller and J. G. Adler were responsible for its development.

Mr. R. M. Meade provided early management guidance. Mr. C. J. Conti was an indispensable contributor to both the over-all system and the processor organization.

Messrs. W. Baskin, R. W. Emery and E. Morris provided the direction of the system technology, while Messrs. G. Halgas, R. Lehtonen and J. Laschenski developed the maintenance features.

Dr. E. M. Davis was responsible for the over-all direction of the technology program.

Mr. R. Rinne and Mr. R. Zurowski were responsible for and made important contributions to the printed circuit packaging phase of the effort.

Mr. D. D. Metzger had the responsibility for the module development effort and Mr. D. DeWitt was responsible for the semiconductor device development program.

Important contributions to the circuit designs and specifications were made by Mr. E. J. Rymaszewski.

Mr. R. A. Henle and Mr. J. L. Walsh provided early technical direction and established some of the basic principles on which the circuit design was based.

Received January 11, 1966.