Chapter 52

The IBM System/360, System/370, 3030, and 4300: A Series of Planned Machines That Span a Wide Performance Range

C. G. Bell / A. Newell / M. Reich / D. Siewiorek

Introduction

In this section, besides making some general comments on the IBM System/360 and System/370 series and System/370 follow-ons, we will attempt an analysis of the performance and costs of the series. Performance is notoriously difficult to measure, as we noted in Chap. 5, and costs are even more so. With respect to the latter, what is publicly available is price data, not manufacturing cost data.

These prices reflect not only marketing policies but also accounting policies within the organization for the attribution of cost to product lines. Nevertheless, the 360 and 370 series provide two things which make a comparative analysis worthwhile. First, the common ISP makes simple performance measures more comparable; second, the common manufacturer makes relative prices more a reflection of relative costs than would otherwise be the case. Neither of these aspects is perfect, as we will note at several points in the discussion. Nevertheless, the 360 and 370 series provide as good an opportunity to attempt cost/performance analysis as we know.

Analyses of the type we attempt here produce only rather crude pictures and are subject to question if all the input data are not very carefully checked. We have not done this, depending instead on published sources. For the purpose of this book, illustration of the style of analysis seems sufficient. In addition, using a performance measure based only on Pc power measurements leaves many questions unanswered because it does not address the soft areas of analysis relating to throughput, task environment, and the operating-system software.

Figure 1 depicts the family tree of IBM computers as a function of introduction date and relative processing power. It can be used as a concise summary and reference for the following discussions.The reader is encouraged to follow the procession of this chapter on Fig. 1.

The IBM System/360 architecture was introduced in Chap. 40. The series has been superseded by the IBM System/370, 3030, and 4300 series. Each series is upward-compatible with the System/360 so far as the user problem state is concerned. The series also share an upward-compatible ISP, as outlined in Chap. 51. The various models differ in interpreter speeds and PMS structure. Many PMS elements are used in common, particularly K's, Ms's, and T's. The 3030 and 4300 series constitutes the currant primary IBM product line.

The System/360, System/370, 3030, and 4300 series are presented both because IBM's market dominance makes it the most prevalent mainframe computer and because its implementations span the largest performance and price range of any series. The various models are compared in Table 1.

This chapter will open with a discussion of the various 360, 370, 3030, and 4300 series models. Finally, the System/360-System/370 series will be evaluated in terms of cost and performance.

The IBM System/360 Family

Figure 2 illustrates the introduction dates of the various System/360 models. Chapters 40, 41, and 12 discuss the logical structure of the System/360, the implementations,[1] and the microprogrammed Model 30, respectively.

A succinct description of the design goals and innovations is given in the abstract of Amdahl, Blaauw, and Brooks [1964]. The architecture[2] of the newly accounted IBM System/360 featured four innovations:

1 An approach to storage which permits and exploits very large capacities, hierarchies of speeds, read-only storage for microprogram control, flexible storage protection, and simple program relocation.

2 An input/output system offering new degrees of concurrent operation; compatible channel operation; data rates approaching 5 million characters per second; integrated design of hardware and software; a new, low-cost, multiple-channel package sharing mainframe hardware; new provisions for device status information; and a standard channel interface between central processing unit and input/output devices.

3 A truly general-purpose machine organization offering new supervisory facilities, powerful logical processing operations, and a wide variety of data formats.

4 Strict upward and downward machine language compatibil-

[1] Chapters 40 and 41 are from *IBM Systems Journal*, vol. 3, no. 2, 1964, which was devoted exclusively to the System/360. Other articles, listed in the bibliography at the end of this chapter, are recommended for additional details.

[2] The term *architecture* is used here to describe the attributes of a system as seen by the programmer, i.e., the conceptual structure and functional behavior, as distinct from the organization of the data flow and controls, the logical design, and the physical implementation.

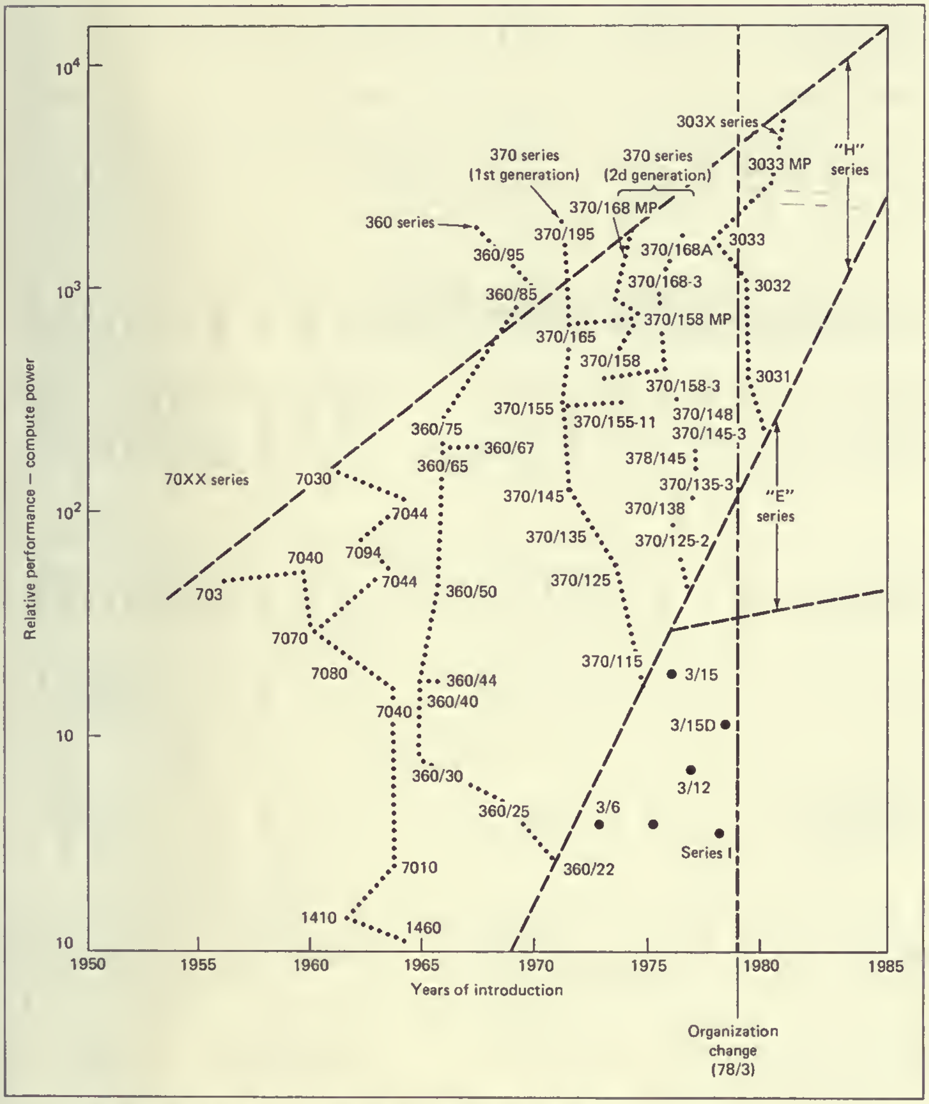


Fig. 1. Family tree of IBM computers. (*Courtesy of Gnostic Concepts, Inc.*)

ity over a line of six models having a performance range factor of 50.

The above four featured innovations are all stated as IBM Corporation design results. It seems better to analyze them in terms of design constraints and implementation results. It appears that the design constraints, from marketing and management directions, were compatibility (item 4 above) and the use of common peripheral equipment (item 2 above). Thus we can measure the 360 design in terms of how well it meets these constraints. With some minor exceptions, all the peripheral components existed at the time of the design and had been used with other IBM computers; thus a goal was already realized. A difficult and important constraint, though not mentioned above, is the necessity of program compatibility with almost all earlier IBM computers.

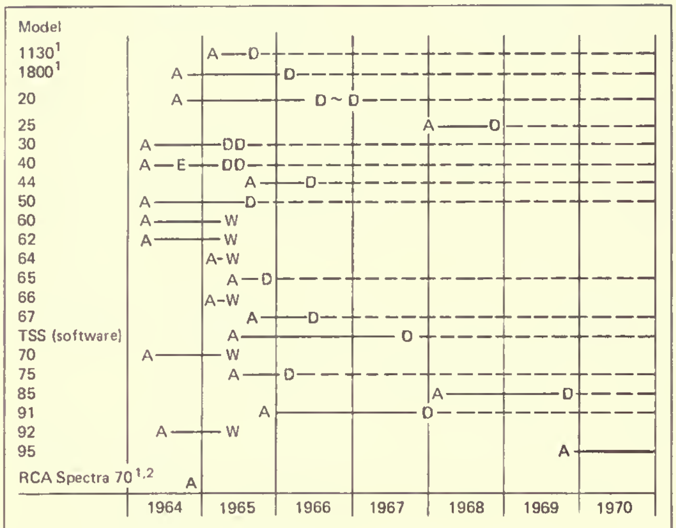
It should be noted that, at the outset of the IBM System/360 announcement, another company, RCA, adopted most of the 360 ISP as a design constraint for its own future computer development. Although some price/performance characteristics appear to be better in the RCA series, the implementation scheme is similar. The lower RCA prices do not reflect entirely implementa-

Table 1 Summary of IBM Processor Characteristics

	CPU		Control store				Transaction-lookaside buffer (TLB)	Main memory				
Model	T. cycle (ns)	Width (bytes)	Size (K)	Word size (bits)	Type (RW/RO)	T. cycle (ns)	Number of entries	Technology	Size (Kbytes) min. to max.	T. cycle (ns)	Number bytes fetched	Interleaving
22	750	1	4	50 + 5	RO	750	None	Core	24–32	1500	1	None
25	900	1	8	16 + 2	RW	900	None	Core	16–48	1800	2	None
30	750	1	4	50 + 5	RO	750	None	Core	16–64	1500	1	None
40	625	2	4	52 + 2	RO	625	None	Core	32–256	2500	2	None
44	250	4	None	. . . .	. . . .	. . . .	None	Core	32–256	1000	4	None
50	500	4	2.75	85 + 3	RO	500	None	Core	128–256	2000	4	None
65	200	8	2.75	87 + 4	RO	200	None	Core	256–1024	750	8	2
67	200	8	2.75	87 + 4	RO	200	8	Core	256–1024	750	8	2
75	195	8	None	. . . .	. . . .	. . . .	None	Core	256–1024	750	8	4
85	80	8	0.5 2	105 + 3 105 + 3	RW RO	80 80	None	Core	512–4096	960	16	4
91	60	8	None	. . . .	. . . .	. . . .	None	Core	2048–6144	750	8	16
95	54	8	None	. . . .	. . . .	. . . .	None	Core	1024–4096	756	8	16
115	480	1	20–28	20 + 2	RW	480	8	MOS	64–192	480	2	None
115-2	480	2	12–20	19 + 3	RW	480	16	MOS	64–384	480	2	None
125	480	2	12–20	19 + 3	RW	480	16	MOS	96–256	480	2	None
125-2	320	2	16–24	19 + 3	RW	320	16	MOS	96–512	320–480	2	None
135	275–1485	2	12–24	16 + 2	RW	275	8	BIP	96–512	770 R 935 W	2 or 4	None
135-3	275-1485	2	64	16 + 2	RW	275	8	BIP	256–512	880 R 935 W	2 or 4	None
138	275–1430	2	64	16 + 2	RW	275	8	MOS	512–1024	880 R 935 W	2	None
145	203–315	4	8–16	32 + 4	RW	203	8	BIP	160–2048	540 R 608 W	4 or 8	None
145-3	180–270	4	32	32 + 4	RW	180	8	BIP	192–1984	405 R 540 W	4 or 8	None
148	180–270	4	32	32 + 4	RW	180	8	MOS	1024–2048	405 R 540 W	4	None
155	115	4	6	69 + 3	RO	115	None	Core	256–2048	2070	8	None
155-II	115	4	8	69 + 3	RO	115	128	Core	256–2048	2070	8	None
158	115	4	8	69 + 3	RW	115	128	MOS	512–6144	1035 R 920 W	8 or 16	None
158-3	115	4	8	69 + 3	RW	115	128	MOS	512–6144	1035R 920 W	8 or 16	None
158 MP	115							MOS		1035 R 920 W	8 or 16	None
158-3 MP	115							MOS		1035 R 920 W	8 or 16	None
165	80	8	2 2	105 + 3 105 + 3	RO RW	80	None	Core	512–3072	2000	8	4
165-II	80	8	4 1	105 + 3 105 + 3	RO RW	80	128	Core	512–3072	2000	8	4
168	80	8	4 1	105 + 3 105 + 3	RO RW	80	128	MOS	1024–8192	320	8	4
168-3	80	8	4	105 + 3	RO	80	128	MOS	1024–8192	320	8	4
168 MP	80							MOS	2048–16384		8	4
168-3 MP	80							MOS	2048–16384		8	4
195	54	8	None	. . . .	. . . .	. . . .	None	Core	1024–4096	756	8	16
3031	115	4	8	69 + 3	RW	115	128	MOS	2048–6144	345	8	4
3032	80	8	4	105 + 3	RW	80	128	MOS	2048–6144	320	8	4
3033	58	8	4	105 + 3	RW	58	128	MOS	4096–8192	290	8	8
4331	900	4	64		RW	900–1300	None	MOS	512–1024	1300	4	
4341	150–300	8			RW		Yes	MOS	2048–4096	600	8	

	Cache								
Model	*Size (Kbytes)*	*T. cycle (ns)*	*Line width (bytes)*	*assoc.*	*ISPs implemented in microprogram*	*Relative performance (COBOL benchmarks)*	*Relative performance [Phister, 1979]*	*Relative performance (Data Pro)*	*Maximum I/O rate (Mbytes/s)*
22	None	....	....	....					
25	None	....	....	....	1401[a]				
30	None	....	....	....	1401[a]|1620	1.0	1.0		
40	None	....	....	....	1401[a]|1410[b]	2.0	1.8		
44	None	....	....	....					
50	None	....	....	....	1410[b]|7070[c]	4.5	3.9		
65	None	....	....	....	7070[c]|7090[d]	10.0	13.7		
67	None	....	....	....	7070[c]|7090[d]				
75	None	....	....	....		19.3			
85	16–32	80–160	256 × 4	16	7090[d]				
91	None	....	....	....					
95	32	54–162	8 × 8	4					
115	None	....	....	....	1401[a]		1.0	1.0	0.9
115-2	None	....	....	....	1401[a]		1.7	1.4	0.9
125	None	....	....	....	1401[a]	1.4	1.95	1.45	0.9
125-2	None	....	....	....	1401[a]		2.7	1.8	0.9
135	None	....	....	....	1401[a]	1.8	3.9	2.9	2.4
135-3	None	....	....	....	1401[a]		5.1	3.8	2.6
138	None	....	....	....	1401[a]		5.1	3.9	2.6
145	None	....	....	....	1401[a]|1410[b]	4.0	7.8	5.45	5.3
145-3	None	....	....	....	1401[a]|1410[b]			7.1	5.3
148	None	....	....	....	1401[a]|1410[b]		10.4	7.7	5.0
155	8	115–230	16 × 2	2	1401[a]|1410[b]|7070[c]	10.0	13.5	10.0	5.4
155-II	8	115–230	16 × 2	2	1401[a]|1410[b]|7070[c]			10.0	5.4
158	8	115–230	16 × 2	2	1401[a]|1410[b]|7070[c]	24.2	17.8	15.0	
158-3	16	115–230	16 × 2	4	1401[a]|1410[b]|7070[c]	24.1	19.6	16.4	
158 MP					1401[a]|1410[b]|7070[c]	21.8		27.0	7.5
158-3 MP					1401[a]|1410[b]|7070[c]			29.4	7.5
165	8–16	80–160	8 × 4	4	7070[c]|7080|7090[d]	31.7	38.7	34.5	8.0
165-II	8–16	80–160	8 × 4	4	7070[c]|7090[d]			34.5	8.0
168	8–16	80–160	8 × 4	4–8	7070[c]|7090[d]		47.4	41.8	16.0
168-3	32	80–160	8 × 4	8	7070[c]|7090[d]	51.4	53.9	45.5	16.0
168 MP					7070[c]|7090[d]			75.3	28.0
168-3 MP					7070[c]|7090[d]	57.6		81.8	28.0
195	32	54–162	8 × 8	4		39.9	73.0	86.4	3.0
3031	32	115–230	8 × 4	8			23.5	19.7	
3032	32	80–160	8 × 4	8			53.9	45.1	
3033	64	58–116	8 × 8	16			97.0	77.4	
4331	None	....	....	....	1401[a]		4.8	4.0	2.3
4341	8	50–225	8		1401[a]		16.1	17.0	9.0

[a] C[IBM 1401|1440|1460]. [b] C[IBM MP|7010]. [c] C[IBM 7070|7074]. [d] IBM [709|7040|7090|7094].



A—announced; D—delivery; E—exhibited; W—withdrawn

[1] Not part of System/360

[2] Uses same ISP

Fig. 2. IBM System/360 model introduction dates.

tion and technology but include RCA marketing and profit strategy. In addition, of course, there should have been lower development costs. RCA's exit from the computer business when it sold its user base to UNIVAC also may be indicative of the fact that the 360 costs were not out of line with the product and service costs.

An interesting aspect of the design is the method used to implement the individual computer models (of the range) and their associated costs. From the standpoint of innovation, the 360 was the first computer series to cover a wide range. The more basic P's (Models 20 ~ 65) were implemented via a microprogrammed processor. This is based on a computer program within an M[read only], i.e., a read-only storage (ROS), to interpret the common ISP. A payoff from this implementation strategy is a solution to the "compatibility design constraint," which is the ability to provide compatibility with the customer's previous (IBM) machine, which, of course, was not a member of the 360 series. This is undoubtedly the most difficult constraint to meet in the P designs, and probably the most significant real innovation. From the marketing viewpoint, it provided the user with a crutch to go from a former IBM computer to the System/360. This is accomplished through "emulation," which (as defined by IBM) means the ability of one C to interpret another's programs *at a reasonable performance level.* These emulations are realized by various microprogrammed P's designed to interpret both the 360 ISP and one or more of IBM 704, 709, 1401, 1410, 1440, 1460, 1620, 7010, 7040, 7044, 7070, 7074, 7090, and 7094.

Most of the above ISPs have a different structure from the 360 ISP. For example, the 1401 series instructions and data [Bell and Newell, 1971] are variable-length character strings; the 1620 has variable-length data strings; the 704 series process fixed- and floating-point data with single-address instructions; and the 7070 is a fixed-word decimal computer. Thus the 360 C's represent the first machines to be two logical processors in the same physical implementation.

The emulated speeds are often better than that of the original hardwired computer. This is not surprising, considering the change in technology; it is a very attractive feature. The 360 Mp performance is often a factor of 5 to 10 times the "emulated" computers; and the M[ROS] data rates are a factor of 25 times the Mp's. For example, the Model 65 emulating a 7090 runs faster than a hardwired 7090. Note, by way of contrast, that the hardwired models 91 and 44 have the lowest cost/performance ratios in the series.

There are minor deviations in the particular models, but all implementations belong to a common ISP subset. The Model 22 and the Model 91, the extremes of the series, deviate most from the standard 360 ISP. The range of models shows the comparative effects of implementation on the actual processing times. For example, the designers of the various C's were constrained by memory bandwidths. Since the core memories have about the same cycle time (0.75 ~ 2.5 μs), variation in bandwidth is obtained by increasing the data-path width from 8 to 64 bits and by increasing the number of independent Mp's. By looking at just Mp bandwidth, for models 30 ~ 65, we obtain a range of 5.3 to 133.5 Mbit/s, corresponding to a performance range of about 1 to 25. By doubling the number of independent memories, this factor can be increased to 50. These models correspond to a Pc performance range of 1 to 50. Although we might expect a narrower range (based on Mp speed), the range can be increased by performance suppression (at the low end). Power range can be increased by lowering the absolute performance of Model 30. This is accomplished by making performance tradeoffs to lower cost.

Logic Technology

The logic of the 360 series is realized in a hybrid technology, composed partly of integrated-circuit techniques and partly of the solid-state techniques standard in second-generation machines. It is a "thick-film" technology, which deposits the circuitry on a ceramic substrate. This is called Solid Logic Technology (SLT) and is used solely by IBM. This production technique allows only for the fabrication of passive circuit elements on the substrate. The semiconductor elements (diodes and transistors) are produced independently, using standard semiconductor production techniques on a wafer. The semiconductors are then cut and bonded to the substrate, and the complete SLT logic unit is encapsulated.

The substrates correspond roughly to logic elements (gates, inverters, flip-flops, etc.). The SLT units are placed on larger printed-circuit boards.

Although SLT differs fundamentally from integrated-circuit technology, the overall size of the final printed-circuit boards is about the same. At the time the decision was made to develop the technology, it was unclear that integrated-circuit technology would reach mass-production state. Thus the SLT program was an intermediate design prior to integrated-circuit technology. The two approaches are about the same from the standpoint of reliability, especially when one considers the soldered printed-circuit mounting. The number of connections to the printed-circuit board are about the same. The production technology of the 360 series is outstanding, perhaps surpassed only by the 360 marketing plan.

PMS Structures and Implementations of the Computer

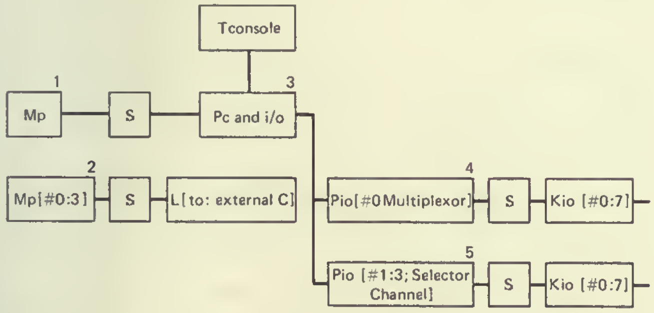
The PMS structures of the various models in System/360 are basically similar, except for the upper end of the series and for the Model 44 (complete compatibility can be purchased as an option). We take up the main group first and then discuss the others individually.

Models 30, 40, 50, and 65. The PMS structures of Models 30, 40, and 50 are all very similar. Figure 3 shows the tree-structured Mp-Pc of the Model 50.[1] They all use a P.microprogram, although with different ISPs. Some gross characteristics are given in Table 1. The Pc of Model 65 is also microprogrammed, but it has hardwired Pio's. A PMS diagram of Model 65 is given in Fig. 4.

The C structures with M[ROS] use a single physical P.microprogram to realize the Pc, the Pio [Multiplexor Channel], and the Pio [Selector Channel]. This technique of using a single shared physical P.state is the same one that Pio['Multiplexor] uses. The Pio['Multiplexor] is equivalent to multiple Pio's. Within the physical P, interrupts are used to switch among the P's.

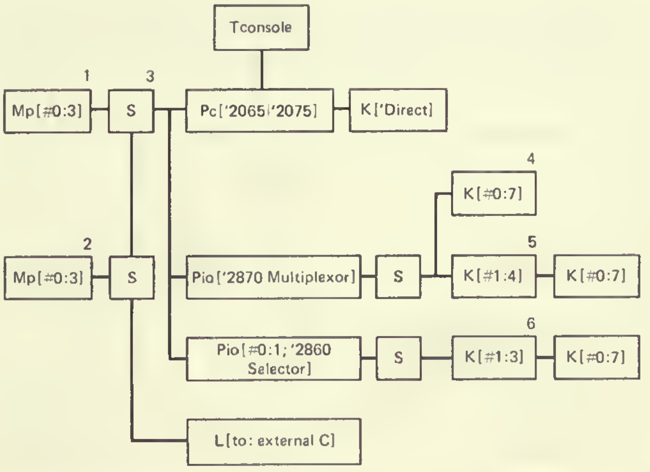
The interpretation cycle for the 360 ISP starts by fetching the instruction, proceeds to fetch the operands, executes the instruction, and then returns results to Mp. The instruction-interpretation process takes only a few Mp references for most instructions. The P tests for interrupt requests during instruction fetch. Pending interrupts are serviced by processor microcode. Pio hardware handles data transmission by breaking in (interrupting) for memory service and updating channel information.

A few instructions require a long (or indefinite) interpretation time—e.g., character translate and edit—since the operations are on character strings. Here, the iterative program loop which operates on each character of the string must test the attached K's to detect when the Pio interpreter is to be run for data transfers. The long instructions can take several hundred microseconds and



Notes
1. Mp[128-256 Kbyte; core; 2 μs/4 consecutive byte]
2. Mp['2361-2 Large Capacity Store/LCS; 8 μs/w; t.access: 3.2 μs; 262 Kw; 8 byte/word; 8,1 parity bit/byte]
3. Pc[!Time multiplexed between central processing and I/O!]
4. Pio[8yte Multiplexor Channel; to: 8 Kio max; address capability: 192 I/O Devices]
5. Pio[Selector Channel]

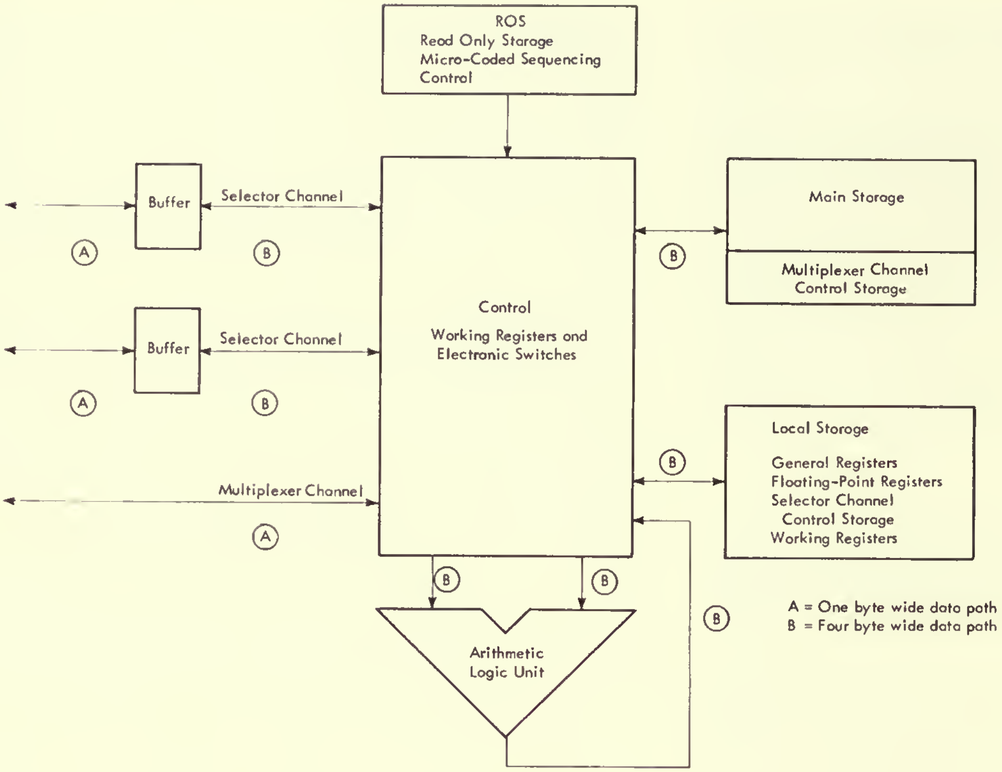
Fig. 3. IBM System/360 Model 50 PMS diagram.



Notes
1. Mp['2365-3; 256-1024 Kbyte; core; .75 μs/8 consecutive byte]
2. Mp['2361-2 Large Capacity Store/LCS; 8 μs/w; t.access: 3.2 μs; 262 Kw; 8 byte/w; 8,1 parity bit/byte]
3. S[8 M; 4 P; time multiplexed; concurrency : 1]
4. K[Address capability: 192]
5. K['2870 IO Selector Subchannel]
6. K['2860 IO Selector Subchannel]

Fig. 4. PMS structure for IBM System/360 Models 65 and 75.

[1] The structure of the Mp's does not include the local M's used for access control, i.e., the storage protect key mechanism, which it is hoped the student will forget about (forever).



	Capacity/Number	Data Width	Access/Speed/Rate
General registers	16	4 bytes	0.5 microsecond R/W cycle/4 bytes
Floating-point registers	4	8 bytes	0.5 microsecond R/W cycle/4 bytes'
Adder		4 bytes	0.5 microsecond
Local storage			0.5 microsecond R/W cycle/4 bytes
Read only storage			0.5 microsecond Rd cycle
Basic machine cycle			0.5 microsecond
Multiplexer channel			
Burst mode		1 byte	
Multiplex mode		1 byte	
Selector channel		4 bytes	
Data transfers			
Processor to storage		4 bytes	
Storage to storage		4 bytes	
Selector channel to processor		4 bytes	
Multiplexer channel to processor		1 byte	
Control unit to channel		1 byte	

Fig. 5. IBM System/360 Model 50 data-flow diagram and system characteristics. (*Courtesy of International Business Machines Corporation.*)

cannot be interrupted; thus the response time for an interrupt can be very poor. Figure 5 gives a simplified picture of the register organization of a Model 50.

The actual System/360 ISP interpretation program in each of the models is different. In addition, each model has microprograms for interpreting other ISPs through emulation. Tucker [1967] discusses how the models were changed as the emulation constraint was added. Table 1 gives the computers which each of the models can emulate. A register structure of the C['30] and the operation for the P.microprogram ISP are given in Chap. 12. Tables 2 and 3 in Chap. 41 give the additional parameters which influence the instruction interpretation rate of the P.microprogram. The significant parameters for a P.microprogram are the M[ROS] hardware characteristics (speed, size, and information width); the number of fields in the M[ROS] instructions, which gives an indication of the number of control functions performed in parallel; the M[general register] rates and their location in the structure; the Mp data rate; and the characteristics of M[temporary] within P. The activity of transferring data from a K, via the Pio['Selector], is done concurrently with normal instruction interpretation in Models 30, 40, and 50. A program in M[ROS] sets up the data transmission with Mp, and transmission is controlled by an independent hardware control.

Model 25. The Model 25 is an interesting C. Perhaps some of the interest of the authors is caused by the mystery (to the authors) as to what its ISP is. Its ISP is no doubt described in maintenance manuals. We can make the following observations based on its characteristics taken from its manual of Functional Characteristics. The observations are:

1 It has a very high-performance Mp, namely, Mp[core; .9 μs/w; 16 |24|32|48 kby; 2 by/w]; the Mp power is almost that of a Model 50.

2 There is a relatively straightforward Pc which is microprogrammed. The Pc uses Mp for its memory. The System/360 ISP is defined in conventional M[read, write]. Of the Mp[48Kby] 16 kilobytes are reserved for a microprogram.

3 Its performance is between that of Models 20 and 30, performing a 360 ISP instruction in about 80 μs.

4 The penalty paid (slowdown factor) to interpret the 360 ISP is therefore 80/1.8 ≃ 45.

5 A small 180-ns local store is used for operands.

6 The Pc cost appears to be about the lowest in the series.

We should ask ourselves:

1 Why do we want an intermediate-level P.microprogram with its own M.read-only, as in the other processors? These P's just seem to waste power.

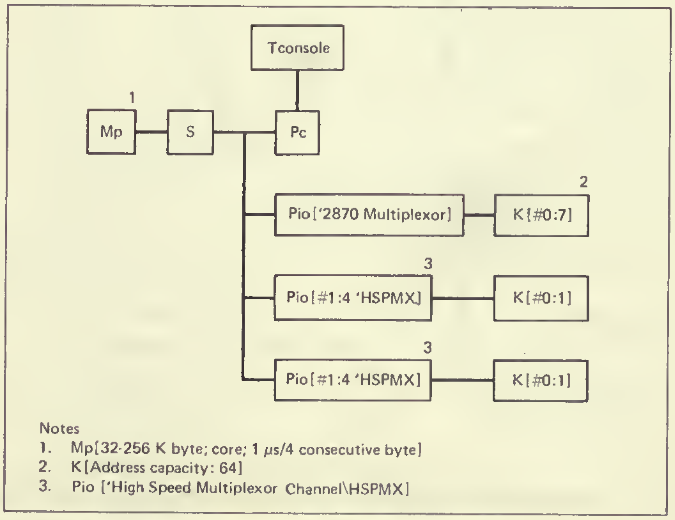
2 Why should we bother to implement an intermediate-level 360 ISP? We know the final user will write programs in a much higher-level language. Thus two levels of interpretation are required instead of one. It is assumed that to program a given task will take, say, x μs if we are using the 360 ISP. We assume the same task programmed directly in the Pc could take as short a time as $x/45$ μs if the Pc were used directly.

We assume that if the P.microprogram, which is used to define the System/360 ISP, were used to interpret a FORTRAN ISP, the speed for a Model 25 FORTRAN ISP might easily approach that of the Model 50.

Model 44. Model 44 does not use M[ROS], but its Pc and Pio are hardwired (Models 75 and 91 are also hardwired). The PMS structure of the Model 44 is given in Fig. 6. Model 44 (and Model 91) stand out as having better performance per unit of cost than their nearest neighbors, which are implemented with M[ROS].

It must be noted that Models 44 and 91 are not strictly compatible with the 360 ISP, since they do not process variable-string and variable-decimal-data formats, although Model 44 options can make it completely compatible. (Subroutines will probably perform satisfactorily for most applications.)

The PMS structure of the Model 44 (Fig. 6) is a tree. The C['44] structure indicates 2-Pio['High Speed Multiplexor Channels/HSMPX], which is between a P['Selector] and P['Multiplexor] in power, since a single physical P['HSMPX] with four subchannels



Notes
1. Mp[32-256 K byte; core; 1 μs/4 consecutive byte]
2. K[Address capacity: 64]
3. Pio ['High Speed Multiplexor Channel\HSPMX]

Fig. 6. IBM System/360 Model 44 PMS diagram.

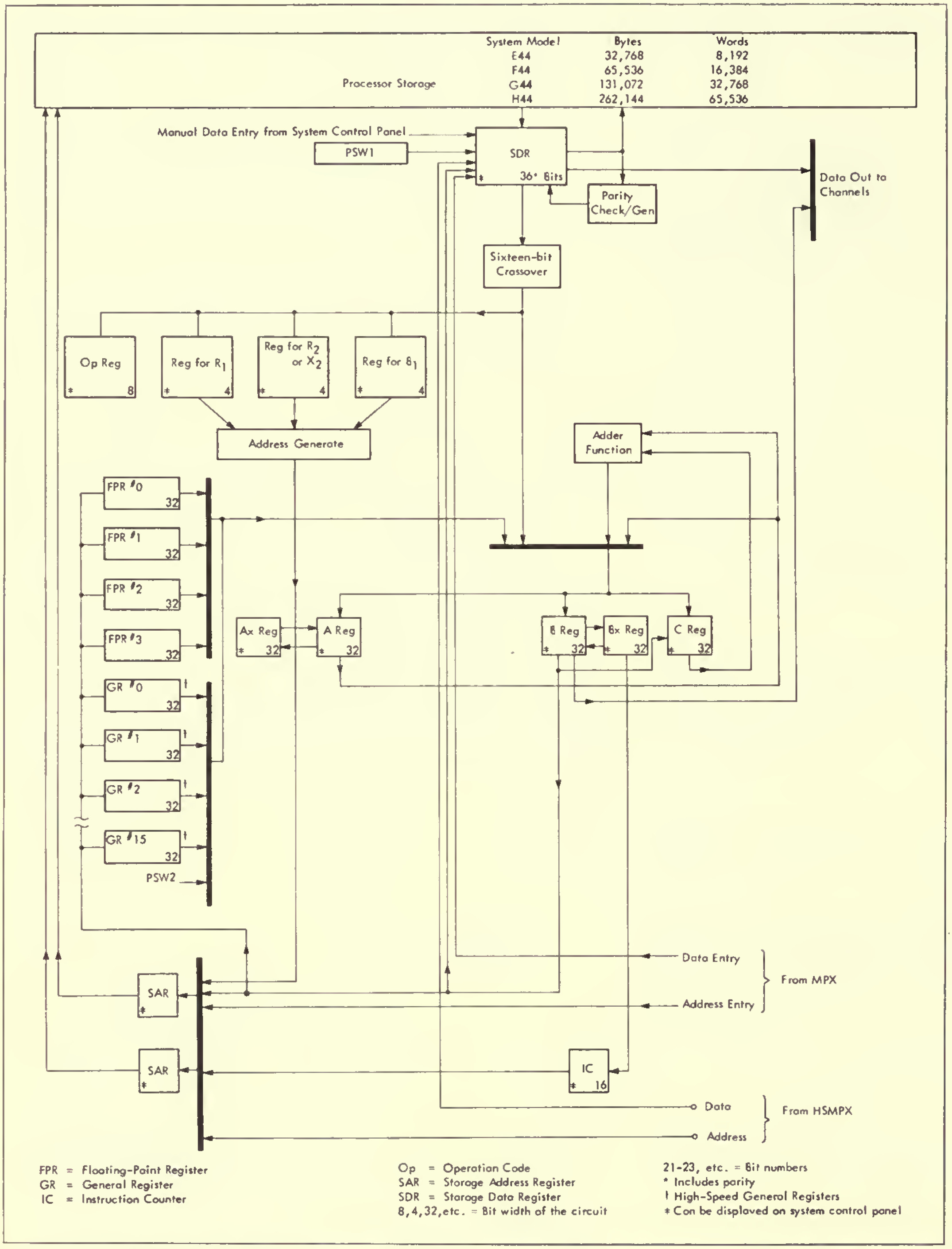


Fig. 7. IBM System/360 data flow in Model 44 CPU. *(Courtesy of International Business Machines Corporation.)*

can behave as four independent Pio's. The organization of the Model 44 Pc registers is given in Fig. 7, which reveals a straightforward implementation. The heavy lines in Fig. 7 indicated an ORing of register outputs to form a single data bus (usually 16 or 32 bits wide). The 16-bit crossover function box allows the right and left halves (16 bits) of the input to be exchanged when output. Almost all the units are registers (except the adders, parity generators, and ORers). The A, Ax, B, and Bx registers are used as the M.working for performing instructions, where the *x* indicates an extension register used in the 64-bit floating-point operations. The C register is a second operand register used for arithmetic and logical operations.

Model 75. The PMS structure of Model 75 is given in Fig. 4. Models 65, 67, 75, and 91 all use the same basic Mp['2365; core]. The S[n Mp; mP], which switches between the n Mp modules and the m Pc and Pio's, varies with model, however. C['65] and C['75] use a simple time-multiplexed S in Pc, called the S['Bus Control Unit/BCU]. This S makes decisions about which P is to use which Mp, rather than having each Mp arbitrate the P requesting service locally. When the memories are all about the same speed, such an S is all right; however, it has severe limitations when slow-speed (8 μs for the large core store) and high-speed memories (0.75 μs) are intermixed. The principal difference between Models 65 and 75 is that C['75] is hardwired and, depending on the size of the configuration, may have lower cost/performance.

The simplified functional unit diagram of C['75] (Fig. 8) is more abstract than the register interconnection diagram of a C['44](Fig. 7). From this description (Fig. 8) of the logic design, one is able to conjecture what is necessarily within the instruction, execution, variable-field length, and decimal functional units. The diagram is presented at a nonuniform level at both the PMS and register-transfer levels. There is somewhat more detail than in the PMS structure (Fig. 4). The Model 75 is possibly the first System/360 to require an intermedite-level diagram between a PMS structure and a register-transfer diagram. The instruction unit contains the instruction location counter (part of the ISP) and is responsible for obtaining the next instruction and the operands. Since there can be overlap in the instruction fetching process, this unit is responsible for holding a number of instructions and stores up to 128 bits (2 doublewords) of instructions at a time. The execution unit and the variable-field and decimal units carry out operations on data. The execution unit processes floating-point and fixed-point data.

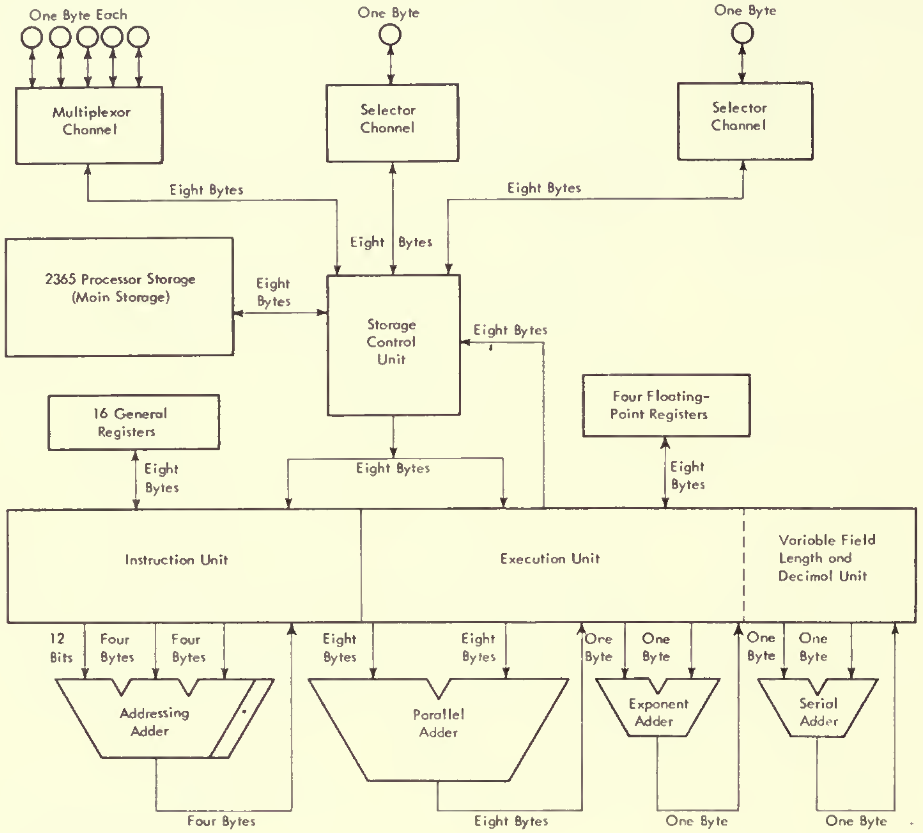
Model 67. The Model 67 was introduced in April 1965 for the purpose of timesharing. The entry was prompted by MIT's project MULTICS. MIT had ordered a GE 645 for experimental research in timesharing. IBM formed a group for the development of a time-shared computer and responded with the Model 67. The Model 67 is essentially a Pc['65] with adequate S's for multiprocessing and a K between Mp and Pc for multiprogramming and memory mapping. Because of the software uncertainties, the Model 67 ran as a Model 65 in most installations (in 1968). The University of Michigan and MIT's Lincoln Laboratory, the first two customers having considered the MULTICS proposal, were instrumental in outlining the specifications [Arden et al, 1966]. The hardware (Fig. 9) is interesting from several aspects. First, there are adequate facilities for memory mapping and program segmentation. This general scheme is outlined in Fig. 10. In the Model 67 a user's segment and page maps are in Mp, and these maps point to physical Mp blocks of the program. Each time a reference is made, the map is checked for the actual reference. In order to avoid the accesses to Mp for each Mp reference, a K, with an M[content address], is located between Pc and Mp to transform a 24- or 32-bit virtual address in Pc into an actual 19- to 22-bit physical address in Mp. This K is not shown in Fig. 10 because it is not logically necessary. The scheme suggested in Fig. 10 uses control bits in the map to determine legal Mp accesses. In the Model 67 the storage key mechanism holds the information whether a given page can be accessed by a given numbered user (instead of associating the control with the mapping as shown in Fig. 10).

Second, the Model 67 is the first acknowledgment by IBM of multiprocessor computers, since it provides adequate switching to allow multiple Pc's. The C['65] multiprocessing configuration has been introduced based on Model 67 structure. Multiprocessors are necessary for reliability, not solely for performance reasons.

The PMS structure of C['67] in Fig. 9 does not have to use the S['Bus Control Unit/BCU],[1] as in the C['65]. The C['67] can have an S in each Mp, so that four P's can communicate with an Mp, as shown in Fig. 9. Each Mp makes the decision about the P request to be honored next. Thus the problem of having an "all-knowing" S['BCU] is solved by allowing each Mp to do local scheduling, rather than having a dialogue with another component (with time delays). The S['BCU] in a duplex C['67] is still present, but with less power, in the form of the S['2846 Channel Controller]. It is used to arbitrate the Pio accesses to Mp.

Without multiprocessing, the Pc seems very badly mismatched with respect to Mp. Consider, for instance, the data rates on the C['67]. From Fig. 9 its maximum possible Mp data rates are:

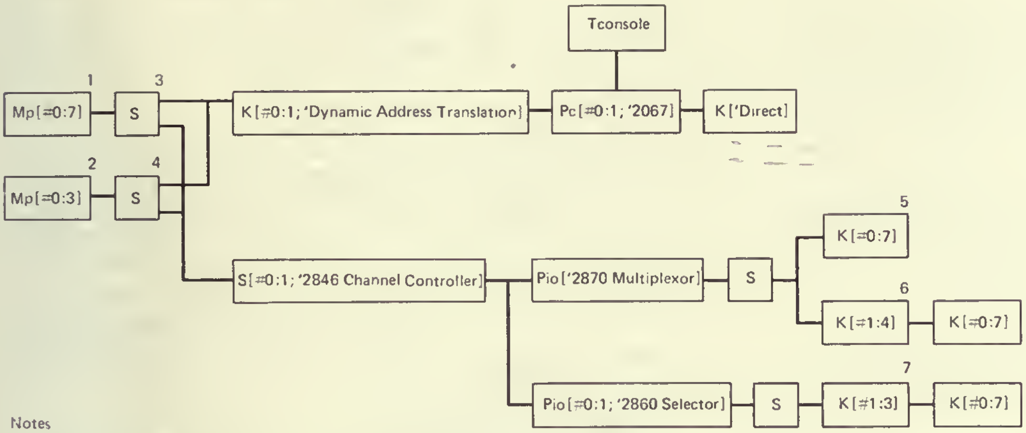
[1] A system with only one port at Mp, controlled by BCU, is called a *simplex*. A system with multiport Mp is called a *duplex*.



* One byte address bypass

	Data Width	Access/Speed/Rate	Comment
2365 Processor Storage	8 bytes	.75 microsecond storage cycle	All models
2361 Core Storage	8 bytes	8 microsecond storage cycle	All models
General registers	1 word	200 nanoseconds	16 General registers
Floating-point registers	2 words	200 nanoseconds word	4 Floating-point registers
Addressing adder	3 bytes	200 nanoseconds	
Parallel adder	8 bytes	200 nanoseconds	
Exponent adder	1 byte	200 nanoseconds	
Serial adder	1 byte	200 nanoseconds	
Basic machine cycle		200 nanoseconds	
2860 selector channel	1 byte	1.3 million bytes per second	8 bytes to storage
2870 Multiplexor channel	1 byte	110 kb to 450 kb	8 bytes to storage
Burst mode	1 byte	50-110 kb	
Multiplex mode	1 byte	50-110 kb	
Selector subchannel	1 byte	100 kb, each	

Fig. 8. IBM System/360 Model 75 data-flow diagram and system statistics. *(Courtesy of International Business Machines Corporation.)*



Notes

1. Mp['2365-12; 256-1024 K byte; core; .75 μs/8 consecutive byte]
2. Mp['2361-2 Large Capacity Store/LCS; 8 μs/w; t.access: 3.2 μs; 262 Kw; 8 byte/w; 8+1 parity bit/byte]
3. S[8 M; (4 ~ 6) P; crosspoint; concurrency: 8; t.delay: .1 μs]
4. S[4 M; 2 P; crosspoint; concurrency: 2; t.delay: 1 μs]
5. K[Address capacity: 192]
6. K['2870 IO Selector Subchannel]
7. K['2860IO Selector Subchannel]

Fig. 9. IBM System/360 Model 67 PMS diagram.

For 1 Mp['2365-12]:

$$\frac{2 \times 64 \text{ bits}}{0.75\ \mu\text{s}} = 171 \text{ Mbit/s}$$

and for 1 Mp['2361 Large Core Store]:

$$\frac{64 \text{ bits}}{8\ \mu\text{s}} = 8 \text{ Mbit/s}$$

Thus the total data rate is

$$\begin{aligned} 171 \times 8 + 8 \times 4 &= 1{,}368 + 32 \text{ Mbit/s} \\ &= \sim 1{,}400 \text{ Mbit/s} \end{aligned}$$

The processing rate is approximately

$$\frac{64 \text{ bits}}{2.2\ \mu\text{s}} = 29 \text{ Mbit/s}$$

An Ms.drum rate is approximately

$$\frac{8\text{b} \times 1.2}{\mu\text{s}} = 10 \text{ Mbit/s}$$

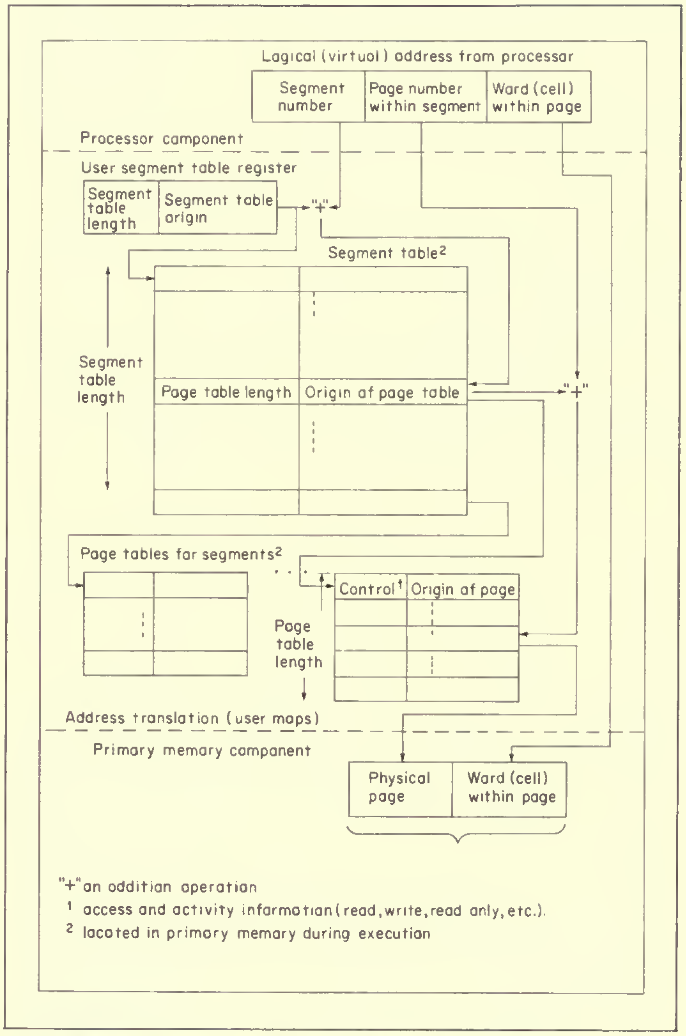
Thus, for the several P's, an effective Mp request rate of 100 Mbit/s might be needed. The data-flow mismatch (between Mp and the P's) occurs because of the P's, the S (the L's connecting P and Mp), the lack of P's, and the fact that t.access = ~ ½ t.cycle.

The Pio['2870], used in Model 65 and above, is described at two structural levels in Fig. 4. The Pio includes a large M.working to store the state of each of the logical Pio's. This Pio state includes the instruction location counter, the control state bits (active, running, interpreting an instruction, processing data, etc.), and buffering (one 8-byte word). By having an M.buffer, the demands on Mp from the Pio's are reduced by a factor of 8. Although the expected data rate from many K's does not require the extra M, there are possible times when the uncertainty of the access times for Mp might cause data loss. Since the M.working is necessary to store the Pio state, the additional space for buffering is not expensive. An alternative design might use Mp for this buffering.

The four Pio['2860 Selector Channel]'s are implemented as independent Pio's, using conventional hardwired logic and buffering. However, they are packaged as one unit.

Model 85. The model 85 was announced in February 1968, with the goal of being the highest-performance Model 360 in production. The performance is about 3 to 5 times that of the Model 65 and in some cases outperforms a Model 91 [Conti, Gibson, and Pitkowsky, 1968].

The PMS diagram of the Model 85 is shown in Fig. 11. The Pio, T, Ms structure is identical to that of Models 65 and 75 (Fig. 4). The two interesting aspects of the structure in Fig. 11 are the M[content addressable; 'Buffer Storage; 16|32 page; 1024 by/page] and the Pc. The pages are filled in groups of 64 bytes, as



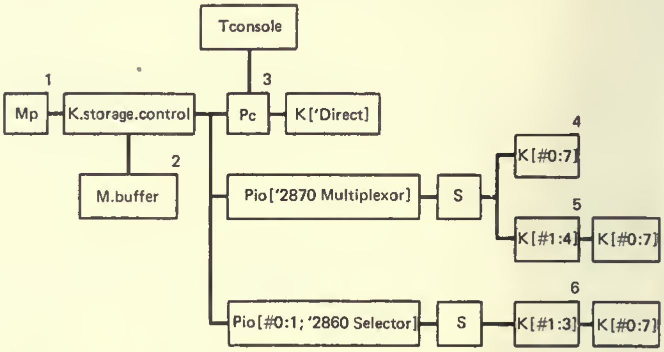
"+" an addition operation

[1] access and activity information (read, write, read only, etc.).

[2] located in primary memory during execution

Fig. 10. Memory allocation using pages and segments.

references to a particular physical block in Mp.core are made. Conti, Gibson, and Pitowsky [1968] give running times for various programs as a function of buffer memory size. Multiprogramming may degrade the performance more than any other case. This process, which has been referred to as *look-aside*, or a *slave memory*, was suggested by Wilkes [1965]. It is completely analogous to the Model 67 M[content.addressable; 8w] which is used to hold the segment-page map for a multiprogrammed timesharing system. It is also analogous to a one-level storage system (Atlas; see Chap. 10) formed from two physical M's whose performance differs significantly. Here, the effect is to try to approximate a computer with a large Mp[80 ns/w] by using a large Mp[1 µs/w] and a small Mp[80 ns/w]. The CDC 7600 has a similar structure, but the Mp-Ms migration is under programmed control.



Notes

1. Mp[512 ~ 4096 K byte; core: .96 µs/16 consecutive byte; single error correction/double error detection]
2. M. buffer ['cache; 16 ~ 32 K byte; 80 ns/word; content addressible]
3. Pc[t.cycle: 80 ns; instruction prefetch: [2w; 16 byte/w]]
4. K[Address capability: 192]
5. K['2870 IO Selector Subchannel]
6. K['2860 IO Selector Subchannel]

Fig. 11. IBM System/360 Model 85 PMS diagram.

The P.microprogram used for controlling the Pc[K['Execution Unit]] allows for great flexibility in the definition of ISPs. An Mp[500 w] is available for the user; this may be loaded by a program, and it specifies an ISP. One standard option is to emulate the 704-7094 series.

The Model 85 removes the restriction of aligning words at particular boundaries. Thus any logical word, independently of its length, can be located at any physical location addressed in bytes.

The Pc's data operation performance is impressive. A fixed-point multiply is done in 0.4 µs, and a floating-point multiply takes 0.56 µs (not including accesses).

The data-type *extended floating-point number* is used in Model 85. Thus a 24-, 56-, or 112-bit fraction part can be used.

Model 91. This model has a very low cost/performance ratio. Only about 20 Model 91's were produced before it was withdrawn from the market. It has the highest performance of the series. The Mp is 0.75 µs, but 16 are overlapped to provide a theoretical maximum bandwidth of 16 × 64/0.75 = 1,370 Mbit/s. About 2.5 mega-instructions per second are executed; thus, a total of 70 Mbit/s of Mp is absorbed by Pc.

There are other interesting models in the '90 series; the Model 92 was a paper machine, and the Model 95 was unannounced but produced, a version of the Model 91 with an Mp[integrated circuit; 60 ns/w; 8 by/w]. The Model 91 is covered in detail in Chaps.

18 and 19. It is similar to other very large computers in that many techniques are employed to obtain parallelism. The January 1967 *IBM Journal of Research* is devoted to design issues of the Model 91.

Models 1130 and 1800. These computers are presented as reference points and have nothing to do with the C['360]. They are implemented outside the System/360 framework but use its technology, and so cost comparisons are still somewhat meaningful. These computers are straightforward, and for a given task which does not use floating-point arithmetic, they should perform as well as any System/360 model. The arguments we use for the intermediate Pc for the Model 25 apply equally well here, too, namely, Why have such a complex ISP when simple ones will do just as well?

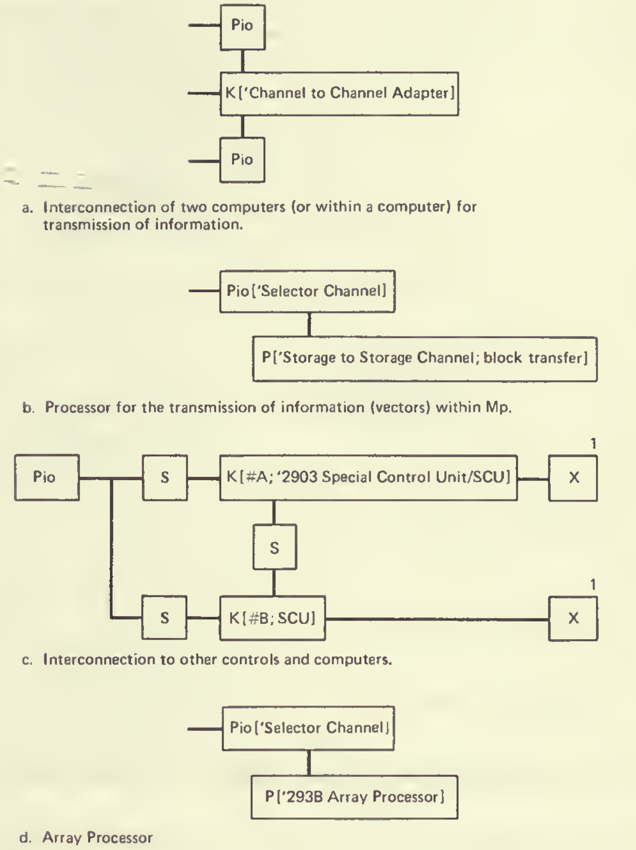
The programmed floating-point arithmetic times for 4-µs 1800 and the "hardwired" (microprogrammed) System/360 Model 30 are compared in Table 2. We would expect the 2-µs 1800 to be better by a factor of 2. Note that the times are about the same for Model 30 and the slower 1800. The cost/performance is especially low with the 1130. It is interesting to speculate why the 1130 and 1800 cannot be implemented within the System/360 framework. Are they "loss leaders"? Are they in response to more sophisticated, performance-oriented users?

The PMS Structure of the Controls, Terminals, Secondary Memories, and Special Processors

There are many common components which attach to the C's (Figs. 12 to 17). Most of the components which attach to a Pio are not especially interesting, but they give an idea of the behavior and parameters. For example, the expression T['1403 Model 3; line printer; 1100 line/min; 132 char/line; 8 bits/character; 64 ~ 240 character set] pretty well describes a typical line printer. From the above description one can deduce the data rate of a T[line printer]. It is 132 char/line × 1100 line/min × $\frac{1}{60}$ min/s × 8 bits/char = 19.4 Kbit/s.

Table 2 IBM 1800 (4 µs) and IBM System/360 Model 30 Floating-Point Arithmetic Timing

	Operation times (µs)	
Operation	*1800 (4 µs)*	*System/360 Model 30*
+ {sf}; + {df}	460; 440	75; 115
× {sf}; {df}	560; 790	320; 1060
÷ {sf}	766	600
√ {f}	4500	2965
sin {f}	3000	3876
exponential {f}	2000	4173


Pio
K['Channel to Channel Adapter]
Pio
Pio['Selector Channel]
P['Storage to Storage Channel; block transfer]
Pio
S
K[#A; '2903 Special Control Unit/SCU]
1
X
S
S
K[#B; SCU]
1
X
Pio['Selector Channel]
P['2938 Array Processor]

a. Interconnection of two computers (or within a computer) for transmission of information.

b. Processor for the transmission of information (vectors) within Mp.

c. Interconnection to other controls and computers.

d. Array Processor

Notes

1. X := (C|K|T|Ms)
2. P[operations: (vector move, vector multiplication, vector inner product, sum of vector elements, sum of squares, convolution, difference equation, fixed-floating conversion; data types: fixed|floating]

Fig. 12. IBM System/360 Special P's and K's diagram.

The Channel-to-Channel Adapter Control. The most interesting group of components (outside the C structures) are the special components shown in Fig. 12. The K['Channel to Channel Adapter] allows two P's, on either the same or a different C, to communicate with one another. This K is used in the construction of a dual C system or the N['Attached Support Processor/ASP]. A C['40|'50] is attached to a C['65|'75]. The C['40|50] is used as a Cio with file processing capabilities. The K has M.buffer. Data can flow in only one direction at a time.

The Special Control Unit. The K['2903 Special Control Unit/SCU] consists of two independent K's which are physically

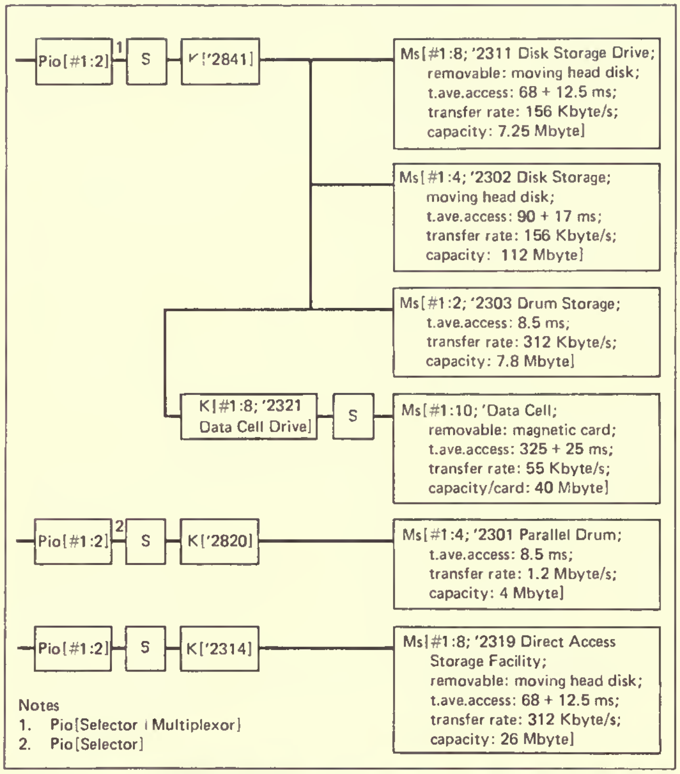

Pio[#1:2]
1
S
K['2841]
Ms[#1:8; '2311 Disk Storage Drive;
removable: moving head disk;
t.ave.access: 68 + 12.5 ms;
transfer rate: 156 Kbyte/s;
capacity: 7.25 Mbyte]
Ms[#1:4; '2302 Disk Storage;
moving head disk;
t.ave.access: 90 + 17 ms;
transfer rate: 156 Kbyte/s;
capacity: 112 Mbyte]
Ms[#1:2; '2303 Drum Storage;
t.ave.access: 8.5 ms;
transfer rate: 312 Kbyte/s;
capacity: 7.8 Mbyte]
K[#1:8; '2321
Data Cell Drive]
S
Ms[#1:10; 'Data Cell;
removable: magnetic card;
t.ave.access: 325 + 25 ms;
transfer rate: 55 Kbyte/s;
capacity/card: 40 Mbyte]
Pio[#1:2]
2
S
K['2820]
Ms[#1:4; '2301 Parallel Drum;
t.ave.access: 8.5 ms;
transfer rate: 1.2 Mbyte/s;
capacity: 4 Mbyte]
Pio[#1:2]
S
K['2314]
Ms[#1:8; '2319 Direct Access
Storage Facility;
removable: moving head disk;
t.ave.access: 68 + 12.5 ms;
transfer rate: 312 Kbyte/s;
capacity: 26 Mbyte]
Notes
1. Pio[Selector | Multiplexor]
2. Pio[Selector]

Fig. 13. IBM System/360 Ms [drum; disk; data cell] PMS diagrams.

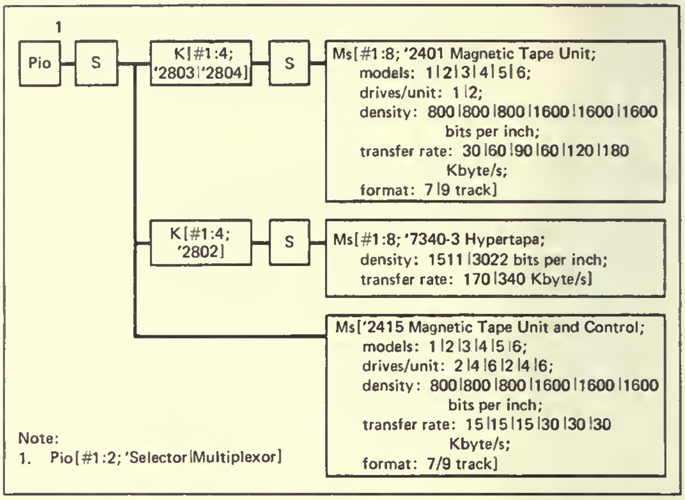

1
Pio
S
K[#1:4;
'2803|'2804]
S
Ms[#1:8; '2401 Magnetic Tape Unit;
models: 1|2|3|4|5|6;
drives/unit: 1|2;
density: 800|800|800|1600|1600|1600
bits per inch;
transfer rate: 30|60|90|60|120|180
Kbyte/s;
format: 7|9 track]
K[#1:4;
'2802]
S
Ms[#1:8; '7340-3 Hypertapa;
density: 1511|3022 bits per inch;
transfer rate: 170|340 Kbyte/s]
Ms['2415 Magnetic Tape Unit and Control;
models: 1|2|3|4|5|6;
drives/unit: 2|4|6|2|4|6;
density: 800|800|800|1600|1600|1600
bits per inch;
transfer rate: 15|15|15|30|30|30
Kbyte/s;
format: 7/9 track]
Note:
1. Pio[#1:2; 'Selector|Multiplexor]

Fig. 14. IBM System/360 Ms [magnetic tape] PMS diagram.

packaged together and allow users to interface with the Pio's. Although it has not been discussed, the actual interconnection with a Pio, via the S[Pio; K]; is via a physical I/O bus which is arranged in a bus (or chained) fashion. Such a single interface to handle a wide range of needs (high and low response and data rates) via a single set of electrical conductors requires a great deal of control information to be passed along the link. Therefore a K must have a great deal of knowledge of the dialogue in order to communicate. The hardware to attach to the I/O bus at a K is costly and must be designed carefully. The K['SCU] provides a rather simplified interface to the Pio. All I/O bus synchronization control, communication protocol control, buffering, and electrical isolation are within K['SCU]. The K['SCU] is fairly flexible, in that devices connected to it can communicate with one another without Pio (see Fig. 12).

Storage-to-Storage-Channel Processor. The P['Storage to Storage Channel] is a special processor which performs the sole function of transferring data blocks (a word vector) between one location in Mp to another in Mp. It qualifies as a P, since it takes an instruction from Mp containing the location and length, and once the instruction is executed, another is fetched and executed (if it exists). Thus the component has a well-defined interpretation cycle and set of operations. This P is useful in a multiprogrammed environment requiring programs to be moved.

The 2938 Array Processor. The P.array['2938] is an extremely interesting special P (Fig. 12). It can be connected to Model 44, 65, or 75. It has a limited instruction repertoire, but the instructions it interprets are more complex than those in the ISP of the Pc. The instructions are algorithms for operating on an array (a vector or a matrix). These instructions include:

1 Vector move, similar to the P['Storage to Storage] described above, with conversion either way between fixed and floating point
2 An element-by-element vector sum
3 An element-by-element vector multiplication
4 A row-by-column vector inner product
5 A convolution multiply
6 The solution to a step in a difference equation

The P.array is microprogrammed, using an M[ROS], which makes it possible to construct complex algorithms in a flexible manner. The hardware logic is capable of doing a combined floating-point multiplication and addition in 200 ns. The impressive results this P achieves in the interpretation of the algorithms are principally

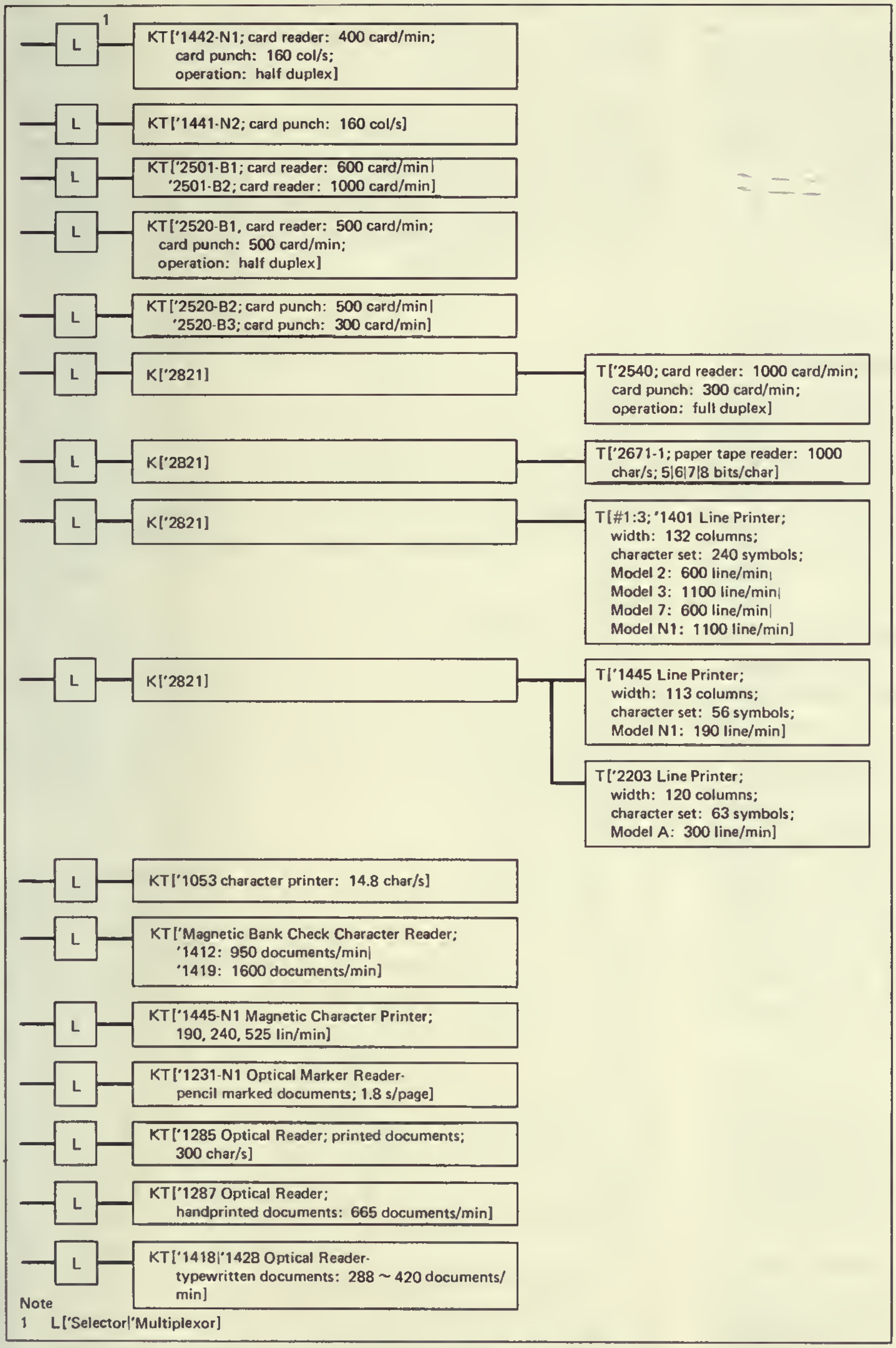
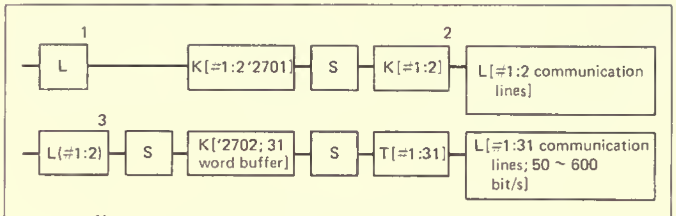

1
L
KT['1442-N1; card reader: 400 card/min;
card punch: 160 col/s;
operation: half duplex]
L
KT['1441-N2; card punch: 160 col/s]
L
KT['2501-B1; card reader: 600 card/min|
'2501-B2; card reader: 1000 card/min]
L
KT['2520-B1, card reader: 500 card/min;
card punch: 500 card/min;
operation: half duplex]
L
KT['2520-B2; card punch: 500 card/min|
'2520-B3; card punch: 300 card/min]
L
K['2821]
T['2540; card reader: 1000 card/min;
card punch: 300 card/min;
operation: full duplex]
L
K['2821]
T['2671-1; paper tape reader: 1000
char/s; 5|6|7|8 bits/char]
L
K['2821]
T[#1:3; '1401 Line Printer;
width: 132 columns;
character set: 240 symbols;
Model 2: 600 line/min|
Model 3: 1100 line/min|
Model 7: 600 line/min|
Model N1: 1100 line/min]
L
K['2821]
T['1445 Line Printer;
width: 113 columns;
character set: 56 symbols;
Model N1: 190 line/min]
T['2203 Line Printer;
width: 120 columns;
character set: 63 symbols;
Model A: 300 line/min]
L
KT['1053 character printer: 14.8 char/s]
L
KT['Magnetic Bank Check Character Reader;
'1412: 950 documents/min|
'1419: 1600 documents/min]
L
KT['1445-N1 Magnetic Character Printer;
190, 240, 525 lin/min]
L
KT['1231-N1 Optical Marker Reader-
pencil marked documents; 1.8 s/page]
L
KT['1285 Optical Reader; printed documents;
300 char/s]
L
KT['1287 Optical Reader;
handprinted documents: 665 documents/min]
L
KT['1418|'1428 Optical Reader-
typewritten documents: 288 ~ 420 documents/
min]
Note
1 L['Selector|'Multiplexor]

Fig. 15. IBM System/360 T [reader; punch; printer] PMS diagram.


1
L
K[#1:2'2701]
S
K[#1:2]
2
L[#1:2 communication lines]
3
L(#1:2)
S
K['2702; 31 word buffer]
S
T[#1:31]
L[#1:31 communication lines; 50 ~ 600 bit/s]

Notes
1. L['Selector|'Multiplexor]
2. K := (KT['Bit Synchronous Data Adapter; 1.2 ~ 40.8 Kbit/s] |
KT['Telephone Line Adapter; 0 ~ 600 bit/s] |
KT['Parallel Data Adapter; 16 ~ 48 b/word]
3. L['Multiplexor]

Fig. 16. IBM System/360 communications PMS diagrams.

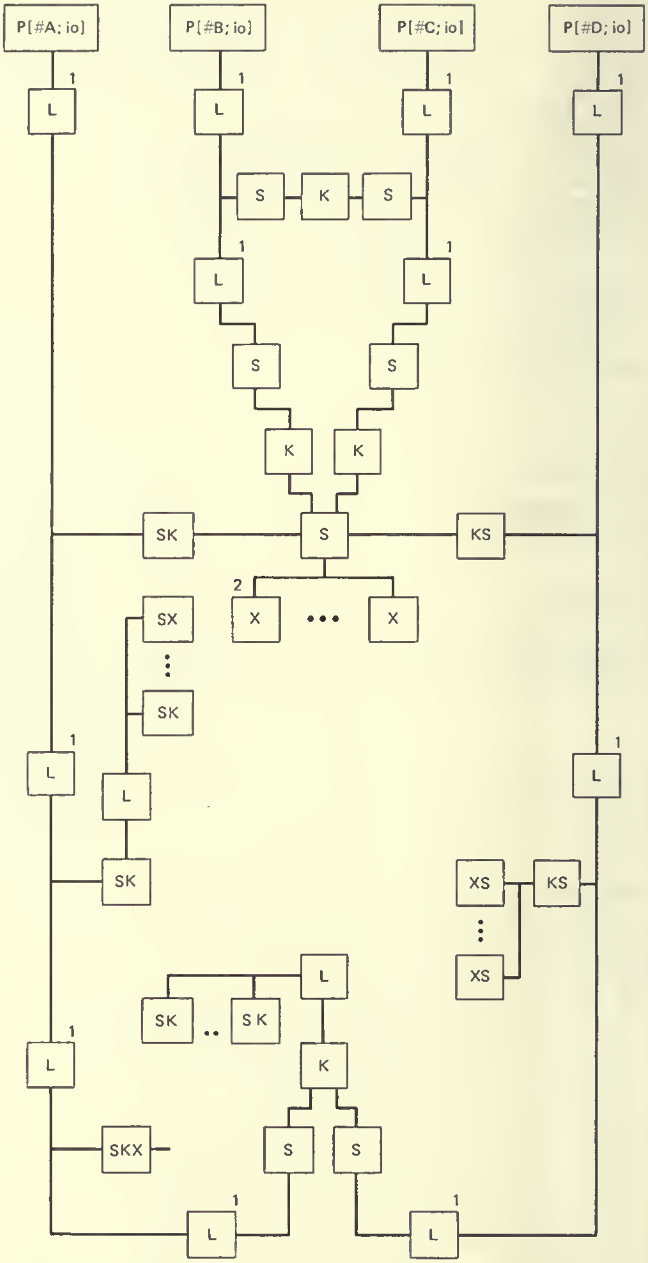
because the time to access the algorithm has gone to zero. A measure we might apply to a P is the ratio of the time it spends fetching the algorithm's data to the total time it spends executing the algorithm. In a conventional computer Pc we suggest that a ratio of nearly ½ is very good. Two fetches are usually required—one for data, one for the instruction. This P has a ratio near 1, as it is always accessing data (and rarely instructions).

Secondary-Memory Structure. Figures 13 and 14 present the Ms PMS structures. All the K's have an optional S, which can be placed between the K and the S[P;K] to allow two Pio's to access a common K (from either of two C's or two Pio's of the same C). The K['2841 Storage Control] is interesting only in being able to control a series of quite disparate devices, on a one-at-a-time basis.

Figure 14 presents all the Ms[magnetic tape]'s. The switch is interesting, as it can be used for up to four K's to access simultaneously any of 16 M.tapes. (The vast array of very similar devices is due undoubtedly to marketing rather than production or engineering reasons.) It should be noted that there are two distinct M.tapes: conventional magnetic tape and Hypertape. Hypertape is explicitly addressed and has built-in error-correction coding.

Terminal Structure. The structure of the vast array of printing devices that can attach to the C['360] is shown in Fig. 15. Some of the devices are interesting, such as the one that reads pencil-marked or typewritten paper. The main parameters of significance to PMS are the rate at which the device reads paper and the kind of paper it reads.

The T and the K's that connect to external processes are given in Fig. 16. The K['2701] and K['2702] are built to transform unsynchronized parallel data from the C into the synchronized serial form required by the telephone line. The K['2701] controls a small number of lines of high data rates; the K['2702] controls a



Notes
1. System/360 I/O Interface Bus
2. X := (T|Ms)

Fig. 17. IBM System/360 peripheral-switching PMS diagram.

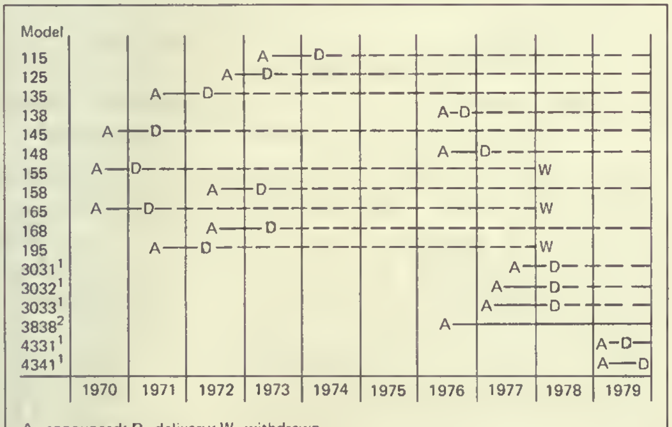
large number of lines at low data rates. The K['2702] is actually an array of up to 31 K's that are time-multiplexed, using an M.core to hold the state of each K.

Peripheral Switching. For performance, communications, and reliability reasons it is necessary to provide access to K's, M's, or T's from several C's or Pio's. A sample structure of a possible configuration, using the above components, is given in Fig. 17. The PMS diagram also shows the physical structure of S[from:Pc; to:K].

The IBM System/370 Family

The first System/370 model was introduced in the summer of 1970. During a period of 7 years, a total of 23 different processor models were realized. Chapter 51 explains why IBM expanded the System/360 architecture into the System/370 and it also highlights the main differences between the two architectures. Figure 18 illustrates the introduction dates of the various System/370 models.

As in the System/360 series, microprogrammed processors were used extensively for processor implementation. In fact, only the System/370 Model 195 is a hardwired implementation. Control store words varied from 16 to 105 bits wide with a capacity of 1 to 64 kilowords. It is interesting to note that the total number of control store bits is relatively constant in all models of the System/370 series. The total number of bits ranges from 380K to 1,024K. If the models with support for OS in microcode are not considered, the range is only 380 to 560 Kbit, with an average of 476 Kbit. This might indicate a basic complexity for representing the semantics of the System/370 ISP. The basic semantic complexity also holds for the System/360 ISP,[1] where the range of control store size is 128 to 263 Kbit. If the Model 25 and the high-performance Model 85 are ignored, the range is 200 to 240 Kbit, with an average of 220 Kbit. The System/370 implementations require generally twice the number of control store bits required by the System/360. Does this suggest that the semantic content of the System/370 ISP is twice that of the System/360 ISP?



A–announced; D–delivery; W–withdrawn

¹Not 370 but same ISP

²Array processor as attached processor. Has own memory.

Fig. 18. IBM System/370 model introduction dates.

Emulation of prior-generation ISPs is also a major feature of the 370 implementations. Emulators exist in one or more 370 models for the following: IBM 709, 1401, 1410, 1440, 1460, 7010, 7070, 7074, 7080, 7090, and 7094.

CPU cycle time varies from 54 to 480 ns, a 9:1 range. Memory cycle time ranges from 320 to 2,070 ns, a 6:1 range. Performance variations between models are also provided by varying the width of Pc data paths (1 to 8 bytes), the width of Pc-Mp data paths (2 to 16 bytes), the memory interleaving factor (1 to 16), the size of address translation buffers (0 to 128), the size of main memory cache (0 to 32 Kbyte), and the number of Pc functional units. These variations produce a performance range of 40:1 from the System/370 Model 115 (70 K instructions per second) to the System/370 Model 168-3 (2 to 7 MIPS). Inclusion of the 3030 series and multiprocessors pushes the performance range to over 100:1.

Logic Technology

Whereas the System/360 models are implemented in Solid Logic Technology (SLT) as described previously in this chapter, many 370 models are implemented in Monolithic System Technology (MST). An SLT chip usually contains one type of component (e.g., transistors or diodes), with multiple chips and resistors assembled in one package (i.e., a half-inch ceramic substrate with interconnections). MST is similar to SLT but integrates many elementary components (e.g., transistors, diodes, and resistors) on single chips which in turn are mounted several to a package. MST circuits in the System/370 Model 145 are about twice as fast as SLT circuits in the System/360 Model 40. MST logic takes up less space than SLT because of a higher packing density per chip. In the System/370 Model 145, an MST logic chip is about one-sixteenth of an inch square and contains over 100 components. An SLT chip contains only one component. Also, MST logic requires fewer off-chip interconnections than SLT logic and thus is more reliable.

[1] The PDP-11 series implementations ranged from 9,960 bits to 23,424 bits of control store, with an average of 15,942 bits. The VAX-11/780 requires about 390 Kbit.

PMS Structures and Implementations of the Computer

The PMS structures of the System/370 family members fall into three main classes. The Models 115 through 125 are low-performance distributed systems. The Models 135 through 158 are medium-performance machines that have the CPU and the channels sharing certain data paths and contending for microprogram control. The Models 165 through 195 are high-performance machines that have a great amount of overlap in CPU functions. They also have separately packaged channels. See Fig. 19 for a genealogical System/370 family tree. Each main trunk of the tree will be discussed individually.

Mid-Range Machines: Models 135, 138, 145, 148, 155, and 158. The medium-performance machines are the 135, 138, 145, 148, 155, and 158. These models have similar PMS structures. (See Fig. 20.)

The Models 155 and 165 were introduced in June 1970. These were the first 370 models to appear on the market. The 155 was marketed as a general growth system for System/360 Model 50 and large System/360 Model 40 users. Thus users could upgrade to the better price/performance ratio of the 155 with minimal software changes.

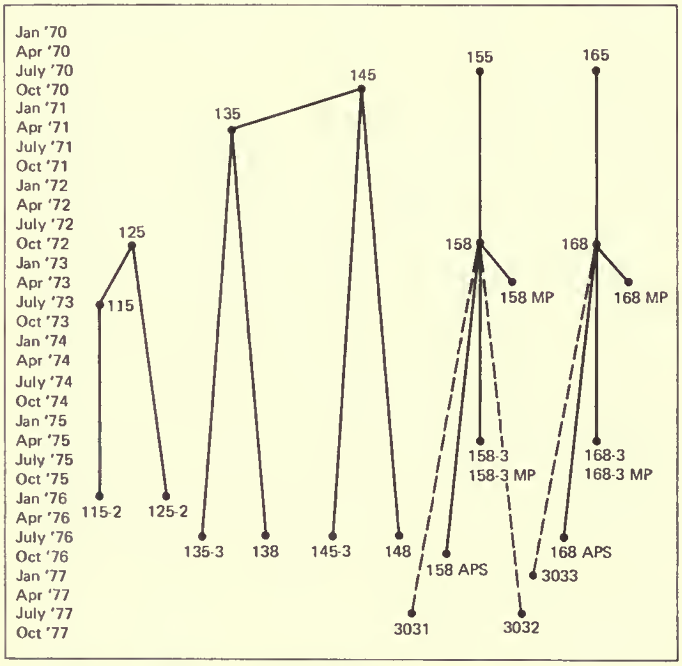
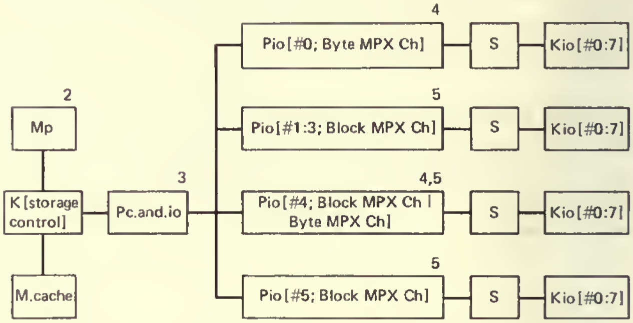


Fig. 19. IBM System/370 family tree.



Notes

1. M[write through cache; setsize: 128; no. of sets: 2; block size: 32 byte; CPU access rate: 4 byte/230 ns; channel accesses: [bypass cache; read/write Mp only; invalidate cache when appropriate]]
2. Mp[256 Kw ~ 2048 Kw; core; ECC (SEC/DED); 2.1 μs/B consecutive byte]
3. Pc[!time multiplexed between central processing and I/O!]
4. Pio[Integrated Byte Multiplexor Channel; to: 8 Kio max; Address Capability: 256 I/O devices]
5. Pio[Integrated Block Multiplexor Channel]

Fig. 20. IBM System/370 Model 155 PMS diagram.

Although the 155 does not incorporate any technological breakthroughs, it is a relatively high-performance machine (see Fig. 21) on account of the following features:

- As do the System/360 Model 85 and the Model 195, the 155 has a cache memory of 8 Kbyte with a cycle time of 115 ns. This 8-Kbyte write-through cache has a set size of 128, an associativity (number of sets) of 2, and a block size of 32 bytes.[1] The channels never access the cache. When a channel writes to Mp in a location which is valid in the cache, the cache data are invalidated.
- The processor-memory data path is 8 bytes wide. Mp is core, with a 2-μs access time.
- When possible, instruction fetch and instruction execution are overlapped. However, operands are not prefetched.

The Pc and the channels share the Pc data paths and contend for microprogram control. The data paths are shared (time-multiplexed) by switching control between the Pc and the channels at well-defined points in the microprograms. When switching (called *break-in*) occurs, the current microprogram is stopped and the new one is given control. For example, once the channel is started, Pc and I/O operations can run concurrently until an I/O storage request occurs. Since the data path from storage is shared, the Pc microprogram stops and gives control to the channel microprogram.

[1] Since the basic Mp fetch is 16 bytes, blocks are loaded 16 bytes at a time. The second 16-byte half block is loaded upon demand, hence the 16×2 notation in Table 1.

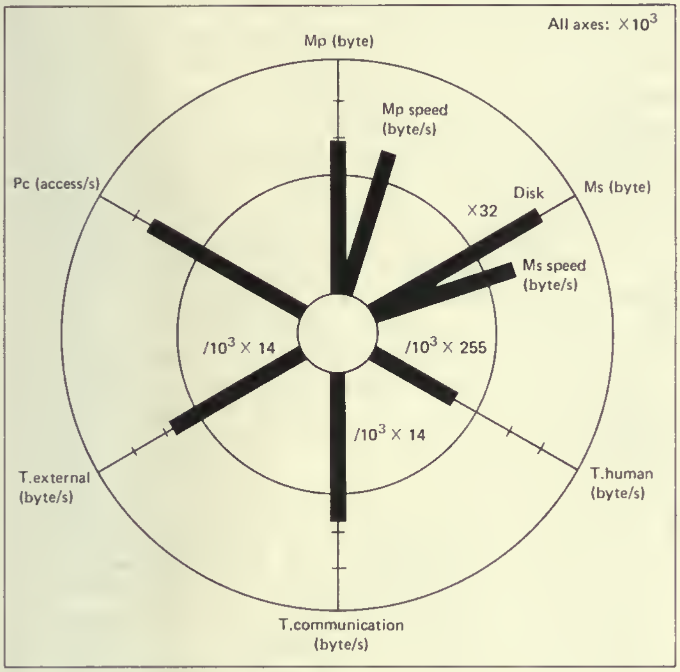


Fig. 21. IBM System/370 Model 155.

As shown in Fig. 22, the three basic data paths in the Pc consist of the following:

- A 4-byte path. This path includes adders, shifters, scratch-pad registers, and Pc and I/O local storage. Fixed-point, floating-point, and enhancement instructions (i.e., those peculiar to the 370 architecture) use this data path.
- A 1-byte data path. This path is used by variable–field-length instructions and single-byte operations. It is also used by certain emulations. Note that the 155 has the capability of emulating the 1401/1440/1460, 1410/7010, and 7070/7074 ISPs.
- An i-fetch path. This path is used for instruction prefetching. It includes buffers, counters, incrementers, and backup registers. Every data path in the Pc has byte parity.

The System/370 Model 145 was introduced in September 1970. It was marketed as a growth system for System/360 Model 40 and large System/360 Model 30 users. The medium-performance 145 was the first 370 to use monolithic (bipolar-semiconductor) main memory. This monolithic storage is similar to the monolithic logic previously described. Both 128-bit and 1-Kbit MST chips are used in the 145 memory.

It is interesting to note that a single, writable storage is used for both Mp and Mmicroprogram, the two memories being differentiated by their addresses. Eight bytes can be accessed at a time from Mp (540 ns for read, 608 ns for write) with no interleaving. The basic Pc data paths are 4 bytes wide, although an 8-byte path is used for instruction fetch. The Pc has a variable-length cycle time (203 to 315 ns).

As in the 155, the Pc and the channels share control storage, Pc-Mp data paths, and the Pc's ALU.

The microprogram can emulate the 1401/1440/1460 and 1410/7010 ISPs. The 145 provides for substantial flexibility in channel configuration. The 145-0 and 145-2 differ in their main storage capabilities (2,048K maximum for the Model 2, as opposed to 512K maximum for the Model 0). The Model 3 is an accelerated Model 2, the microcoding of many high-level operating-system functions being a standard feature. The Model 3 has 128 Kbyte of control store as standard, while the Models 0 and 2 have only 32 Kbyte.

The System/370 Model 135, a scaled-down version of the 145, was introduced in 1971. It was marketed as a growth system for 360 Models 20, 22, 25, and 30. Like the 145, the 135 uses monolithic (i.e., bipolar-semiconductor) Mp and Mmicroprogram.

Four bytes (as opposed to 8 bytes for the 145) can be accessed at a time from Mp (770 ns for read) with no interleaving. The basic CPU data paths are 2 bytes wide, although a 4-byte path is used for instruction fetch (no prefetching) and data access for certain instructions. The Pc has a variable-length microcycle time (275 to 1,485 ns). As in the 145 and 155, the Pc and the channels share control storage and some data paths.

The microprogram can emulate the 1401/1440/1460 ISPs. The 135 allows for a flexible I/O configuration, with direct attachment of various disk storage drives via an integrated file adapter, and direct attachment of up to eight communication lines.

The 135 Model 3 is an accelerated Model 0. The acceleration is basically due to larger Mp, larger Mmicroprogram, and certain high-level operating-system functions implemented in microcode.

The 158 and 168 were introduced in August 1972. Although the 158 is similar to the 155, it provides more processing power in smaller cabinets. It has a larger and faster Mp, and has an Integrated Control Storage (ICS) as an option. The ICS provides two data and control paths, each of which can attach to thirty-two IBM 3330, 3340, or 3350 series disk drives. Also, the 158 and 168 were the first System/370 machines to use MOS main memory (the 155 and 165 use core). It is interesting to note that the older 145 and 135 processor used bipolar Mp, the only two models in the 370 family to do so.

However, the main claim to fame and the real *raison d'être* of the 158 (and 168) is "virtual storage." The concept of virtual

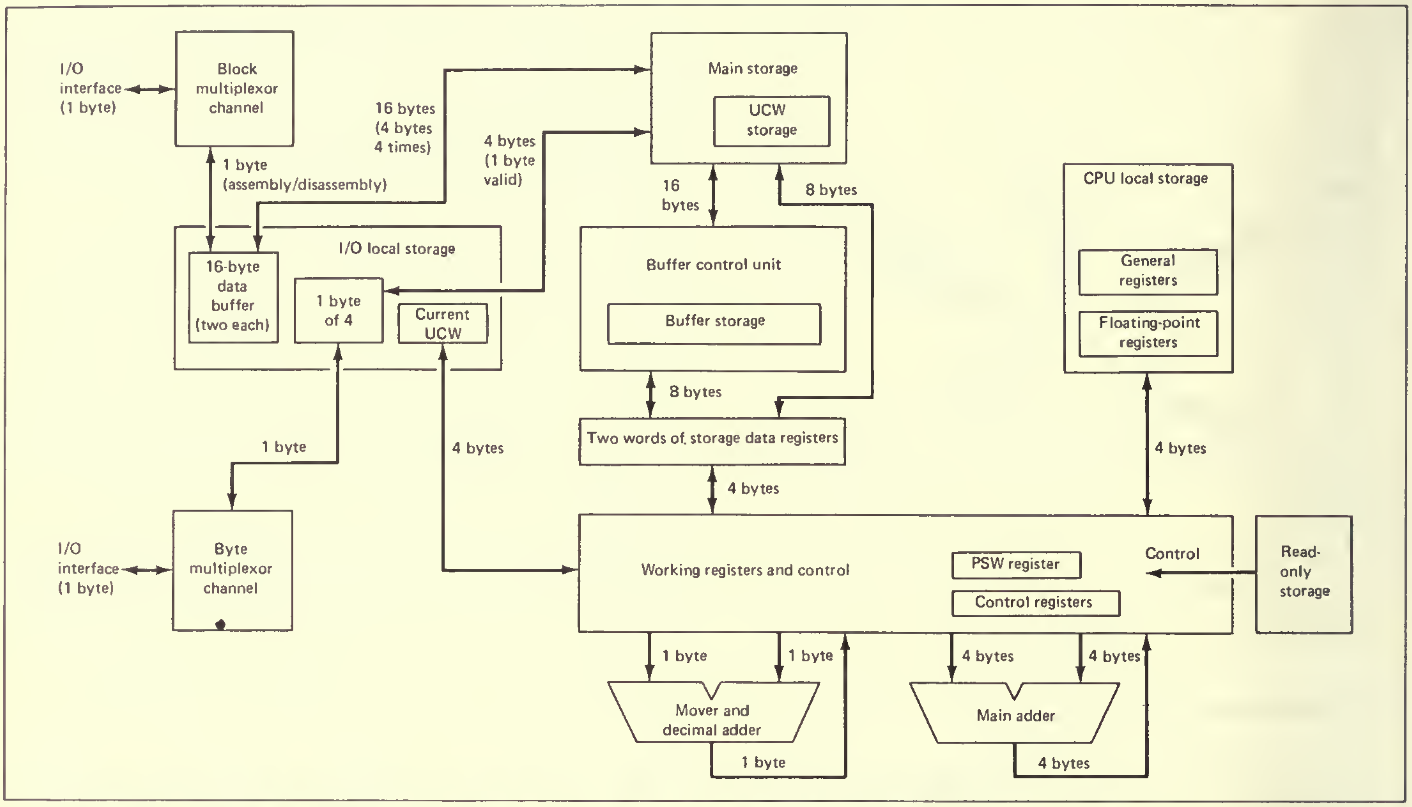


Fig. 22. IBM System/370 Model 155 data-flow diagram. ***(Courtesy of International Business Machines Corporation.)***

storage was not implemented in prior processors, including the System/360 Model 67. The 158 and 168 were the first 370 models to incorporate this feature.

At the time of the 158-168 announcement, Dynamic Address Translation (DAT) hardware became available at no charge for the Models 135 and 145. It also became available as an expensive option for purchased Models 155 and 165.

IBM's multiple-processor philosophy is primarily oriented toward increased system throughput (as opposed to decreased time per job) and can be divided into three general categories: tightly coupled, loosely coupled, and peer-coupled. Tightly coupled systems consist of two processors sharing main storage and executing a single operating system. Loosely coupled systems consist of a number of processors coupled via channel-to-channel adapters. Each processor executes its own operating system and shares job queues. One processor is designated the controlling processor. Peer-coupled systems consist of multiple loosely coupled systems.

IBM's tightly coupled systems employ only high-performance processors, presumably to extend the System/370 performance range until the next set of technological advances provides a higher-performance uniprocessor. One of two general forms is used: (1) multiprocessing (MP) with two Pc's, each with a complete set of I/O options, and (2) attached processors (AP) with two Pc's, one with and one without I/O options. Special instructions have been added for synchronization (e.g., COMPARE AND SWAP, and COMPARE AND DOUBLE SWAP). Also added were serialization (i.e., provisions to allow programs that execute on MP's to have the same storage-reference sequencing they would have if they were executed on a uniprocessor), interprocessor communication (SIGNAL PROCESSOR instruction), redundant time-of-day clocks, and cross-cache invalidation.

February 1973 brought forth the 158 MP. The 158 MP system consists of two 158's or two 158-3's, coupled by a 3058 Multisystem Unit. Channels belong to the Pc to which they are attached, although devices may be shared by using a special "two channel switch" feature. Memory can be configured so that some address space is dedicated to a particular Pc while other space is shared between both processors. Interprocessor communication is via an interrupt capability.

March 1975 saw the announcement of the 158-3. The 158-3 is a 158 with an expanded cache (16 Kbyte as opposed to 8 Kbyte).

Both the 8K and the 16K caches have a set size of 128. However, the 16K cache has an associativity of 4 while the 8K cache has an associativity of 2. Block size is 32 bytes.

In June 1976 the 138 and 148 were introduced. The 138 was announced to have a 29 to 36 percent internal performance increase over the 135, while the 148 was announced to have a 28 to 43 percent internal performance increase over the 145. Data Pro Research Corporation reports that the 138 and 148 cost about 45 percent less for purchase and approximately 22 percent less for rental than the 135 and 145. These machines use MOS Mp, whereas the 135 and 145 use bipolar. However, the MOS memories, created in 1976, of the 138 and 148 are faster than the bipolar memories created in 1971, of the 135 and 145. Also, during June 1976 the 135-3 and 145-3 were announced. They are upgraded versions of the 5-year-old (at that time) 135 and 145 models, and have internal performance equal to the newer 138 and 148, respectively.

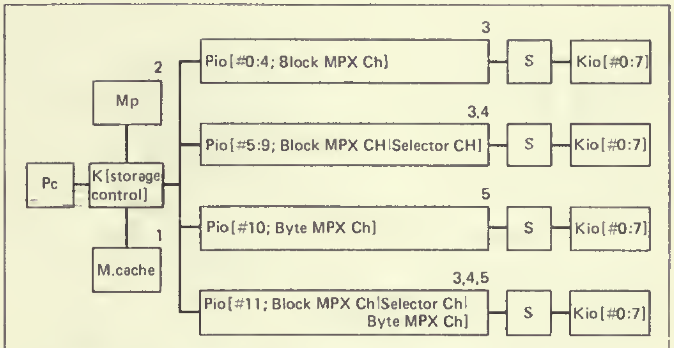
Later, in October 1976, the 158 Attached Processor System (APS) was announced. Whereas the 158 MP of 1973 is a tightly coupled system, the 158 AP is even more tightly coupled. The IBM 3052 APU (Attached Processing Unit) and a 158 (or 158-3) are connected to form a dual-processor system, with shared memory and shared I/O. The 3052 APU (with a 115-ns cycle time) is a bare-bones instruction processor with no Mp and no channels. However, it does have a cache. The APU's writable control store (WCS) can execute all of the System/370 instructions, plus the 1401/1440/1460 and 1410/7010 ISPs. Although an AP system is less expensive than an MP system (because it has less hardware), it has lower performance. Contention for memory and I/O results in decreased throughput. The 158 AP yields only from 1.5 to 1.8 times the performance of one 158-3.

High-Range Machines: Models 165, 168, and 195. The high-performance machines consist of the 165, 168, and 195. See Fig. 23 for a PMS diagram of the 165.

The System/370 Model 165, introduced at the same time as the 155 (June 1970), was originally marketed as a high-speed growth system for System/360 Model 65 and 75 users. The 165 is a higher-performance machine than the 155 because of its wider data paths, greater concurrency of operations, and larger and faster memory.

Whereas the 155 has an 8-Kbyte cache, the 165 cache can be extended to 16 Kbyte. The 8K cache of the 165 is set-associative, with a set size of 64 and an associativity of 4. The 16K cache is simply an 8K cache with the set size increased from 64 to 128. Block size is 32 bytes. The channels write into the cache (i.e., cache write-through to Mp) but do not read from it.

An 8-byte data path between Pc and Mp with four-way interleaving yields 32 bytes fetched per Mp reference cycle (as opposed to 16 bytes for the 155).



Notes
1. M.cache[Setsize: 128; no. of sets: 4; Block size: 32 byte]
2. Mp[512 Kw ~ 3072 Kw; core; 2 μs/8 consecutive byte]
3. Pio['2880 Block Multiplexor Channel; 1.3|2.6 Mbyte/s]
4. Pio['2860 Selector Channel; 1.3|2.6 Mbyte/s]
5. Pio['2870 Byte Multiplexer Channel; Interleaved Mode: 41 Kbyte/s; Burst Mode: 149 Kbyte/s]

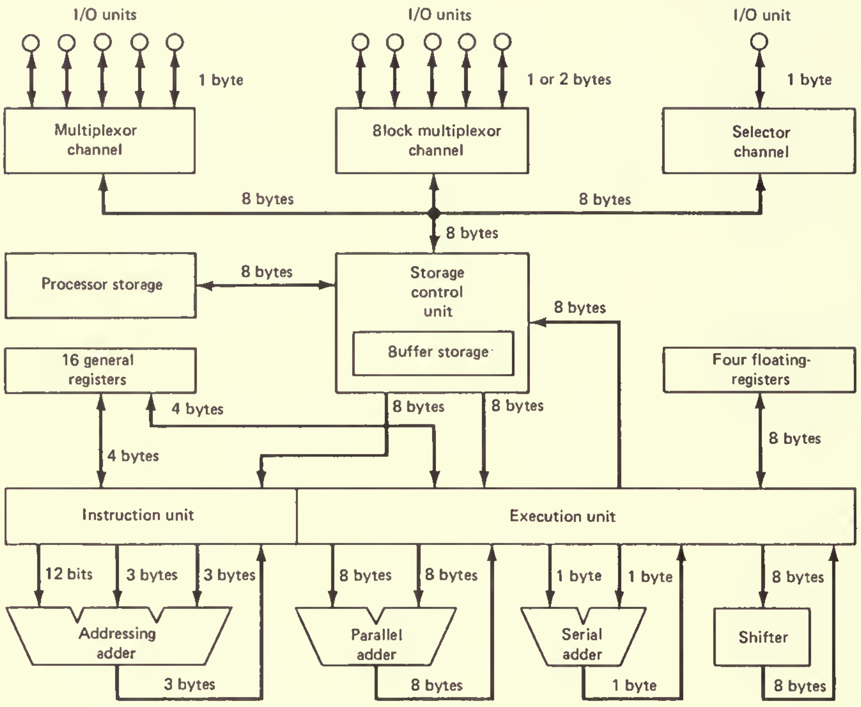
Fig. 23. IBM System/370 Model 165 PMS diagram.

The 165 has a greater amount of instruction fetch–execute overlap than does the 155 because it uses larger buffers, operand prefetching, and more logic.

The 165 uses separate IBM 2860 Selector Channels, IBM 2870 Byte Multiplexor Channels, and/or IBM 2880 Block Multiplexor Channels (as opposed to having the channels share CPU logic, as in the 155). These channels contain the hardware to control their I/O operations. The 165 extended-channel feature provides a maximum of 12 channels.

The main elements of the Pc are the "Instruction Unit" and the "Execution Unit." (See Fig. 24.) The hardwired Instruction Unit performs fetching, decoding, and buffering of instructions, address calculation, and operand fetching and has partial control of the execution unit. The Instruction Unit contains two 16-byte instruction buffers, a 4-byte instruction register, three instruction-queue registers, a 24-bit, three-input adder, four 24-bit address registers, an incrementer, and a decoder. Two 16-byte instruction buffers are used, so that when a branch instruction is encountered, one buffer continues to prefetch sequential instructions (i.e., assumes the branch will be unsuccessful) while the other buffer prefetches instructions from the branch target location (i.e., assumes the branch will be successful).

The Execution Unit is capable of executing a new instruction each microcycle. It contains two 8-byte buffers for prefetched operands, four 8-byte data registers, and an 8-byte result register. There is a 64-bit parallel adder used for binary and floating-point arithmetic, a 32-bit ALU, a 64-bit shifter, and a 1-byte serial adder, which is used for SS-format instructions, floating-point exponent calculations, and packed decimal arithmetic. The basic data-path width in the execution unit is 8 bytes, with byte parity. While the microprogram controls the Execution Unit most of the



Element	Data width (bytes)	Performance	Comments
Processor storage	8	Two-microsecond cycle (Note 1)	....
Buffer storage	8	80-nanosecond cycle	8,192 byte capacity (Note 3)
Basic machine cycle	-	80 nanoseconds	....
General registers	4	Once per machine cycle	16 general registers
Floating-point registers	8	Once per machine cycle	4 floating-point registers
Addressing adder	3	Once per machine cycle	....
Parallel adder	8	Once per machine cyle	....
Serial adder	1	Once per machine cyle	....
2860 Selector Channel	1	1.3 million bytes per second	8 bytes to storage
2870 Multiplexor Channel	1	110 kb to 670 kb (aggregate)	8 bytes to storage
Burst mode	1	110 kb (Note 2)	....
Multiplex mode	1	110 kb (Note 2)	....
Selector subchannels 1-3	1	120 kb each	....
Selector subchannel 4	1	100 kb (kb = 1000 bytes per second)	....
2880 Block Multiplexor Channel	1	1.5 million bytes/second (Note 4)	8 bytes to storage

Notes

1. Processor storage effective cycle time is greatly reduced by the buffer storage function.
2. Aggregate 192-subchannel rate for first or second 2870; reduced by concurrent selector subchannel operation.
3. May be expanded to 16,384 bytes.
4. Three million bytes per second when optional two-byte interface is used.

Fig. 24. IBM System/370 Model 165 data-flow diagram and system characteristics. ***(Courtesy of International Business Machines Corporation.)***

time, hardwired control is used when data results determine the execution sequence (e.g., conditional branches).

The Pc can read four general registers and write into a fifth all in one microcycle. This, coupled with Instruction Unit and Execution Unit overlap, yields a great amount of concurrency in CPU operations.

The 80-ns control store contains both read only (ROS) and read/write memory. The WCS is well suited for microdiagnostics and various emulations. The 165 can emulate 7070/7074, 7080, and 709/7090/7094 ISPs.

The year 1971 also saw the introduction of the System/370 Model 195, a slightly upgraded version of the powerful System/360 Model 195. (See Fig. 25.) The 195 contains a 32-Kbyte cache memory with a set size of 128, a block size of 8 doublewords (64 bytes), and an associativity of 4.

A storage control unit (SCU) controls all fetching and storing from Mp, M.cache, and the channels. Sixteen-way memory interleaving, coupled with an 8-byte-wide data path from Mp to SCU, results in 128 bytes accessed per reference cycle. The SCU also contains much of the circuitry for cache control. A processor store always updates Mp, and updates M.cache if the location has previously been valid in the cache. However, I/O fetches and stores completely bypass the cache, invalidating cache data if necessary.

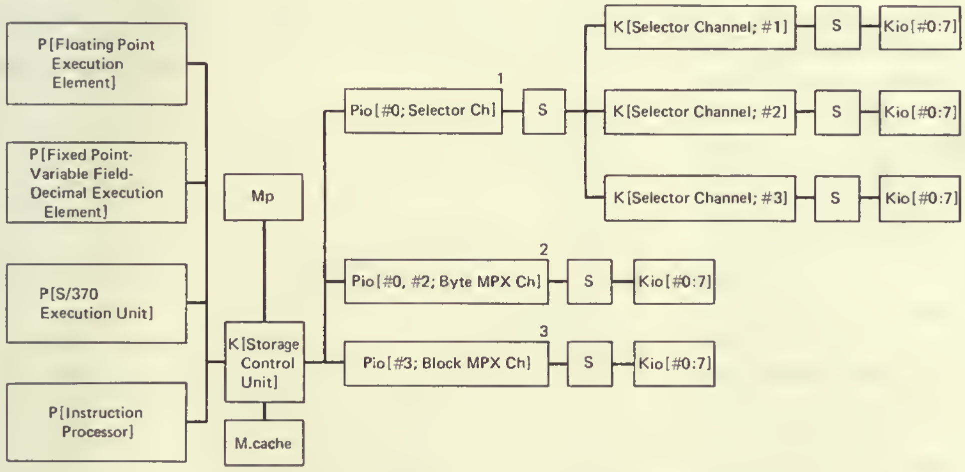
This hardwired machine (which has a 54-ns CPU cycle time) realizes a high degree of concurrency of operations. The central processing element (CPE) consists of the Instruction Processor, the fixed-point/variable–field-length (VFL)/decimal execution element, the floating-point execution element, and the System/370 execution unit.

The Instruction Processor (IP) fetches both instructions and operands, controls the other execution units, handles interrupts, and executes all branch, status-switching, and I/O instructions. (See Fig. 26.) The IP has an 8-doubleword instruction stack (4 times larger than that of the 165), three instruction-control registers, two doublewords of temporary instruction buffer, a decoder, and a three-input adder for effective-address calculation. The fixed-point/VFL/decimal execution element contains the general registers, which the IP uses.

The 8-doubleword instruction stack normally contains the current instruction to be decoded, a few doublewords of instructions already decoded (i.e., instruction history), and a few doublewords of instructions yet to be decoded. Three stack pointers, called the instruction-control registers, keep track of the stack. The instruction register (IR) points to the instruction being decoded. The upperbound register (UB) points to the most recent doubleword brought into the stack, and the lowerbound register (LB) points to the least recent doubleword in the stack.

The 8-doubleword stack allows tight loops to be executed totally within the stack (in what is called the *loop mode*). This is much more efficient than the 2-doubleword buffer of the 165, whose hardware does not provide for the execution of tight loops totally within the buffer.

Conditional mode is entered when a conditional branch is decoded for which the condition code has not yet been evaluated. When this occurs, the IP continues to fetch sequential instructions into the 8-doubleword stack. The sequential instructions prefetched still result in orders to the fixed-point and floating-point execution elements. However, these orders are specially tagged as being conditional, so that they cannot be decoded or



Notes
1. Pio['2860 Selector Channel, Model 3: 1.3|2.6 byte/s; !Models 1,2 and 3 have 1,2 and 3 Selector Channels respectively!]
2. Pio['2870 Byte Multiplexor Channel; interleaved mode: 41 Kbyte/s; brust mode: 149 Kbyte/s]
3. Pio['2880 Block Multiplexor Channel; 1.5|3 Mbyte/s]

Fig. 25. IBM System/370/Model 195 PMS diagram.

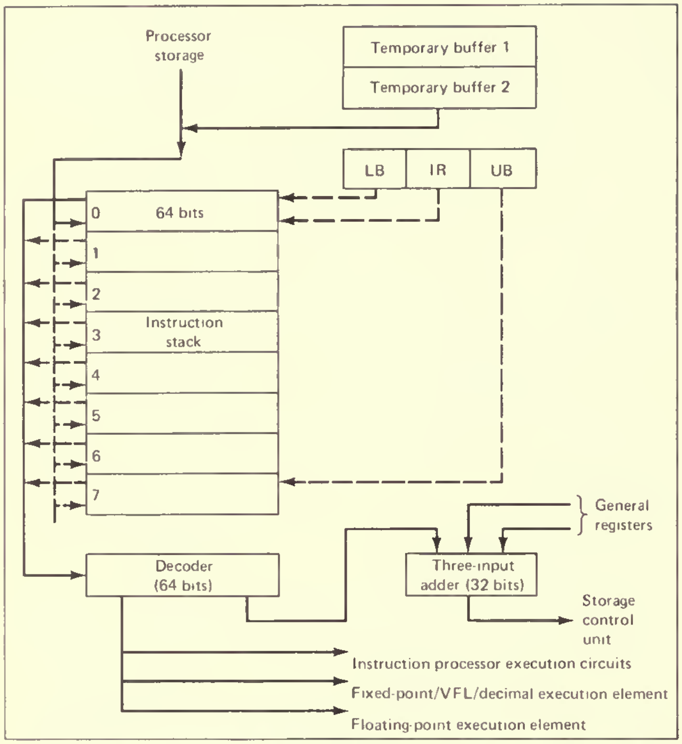


Fig. 26. IBM System/370 Model 195 instruction processor data-flow diagram. ***(Courtesy of International Business Machines Corporation.)***

executed until the IP signals to do so. Two doublewords from the branch target location are also prefetched and stored in buffers.

In the highly concurrent 195, the various execution units can simultaneously execute different instructions. Therefore an exceptional condition cannot always be identified with the specific instruction causing it, since the PSW may be pointing to another instruction. This kind of program interruption is called *imprecise.* Imprecise interrupts are identified as such by setting certain bits in the PSW to a known state. Since all instructions decoded prior to the signaling of an imprecise interrupt are executed, more than one exception can occur for a given imprecise interrupt. Also, because instructions may be executing concurrently and out of sequence, the exceptional condition that causes the interrupt may not be the one that should be logically recognized first. In any case, each type of exception that takes place is identified in bits 16 through 27 of the old PSW, with bits 28 to 31 set to zero. There are certain instructions that are executed only after all previously decoded instructions are fully executed. Among these instructions are the six I/O instructions, LOAD PSW, SUPERVISOR CALL, SET STORAGE KEY, DIAGNOSE, STORE CHANNEL ID, LOAD CONTROL, and STORE CONTROL. Also, a special branch on condition (BCR) instruction (usually implemented as a NOP on other 360 and 370 models) causes all previously decoded instructions in the stack to be executed before the decoding of the next instruction.

See Fig. 27 for a diagram of the fixed-point/VFL/decimal execution element. As its name implies, it executes fixed-point, logical, variable–field-length, and decimal instructions. Operations from the six-position fixed-point operation stack (FXOS) are decoded serially and issued to the appropriate execution units. If the data are available and the execution hardware is free, then the operation can be performed. After completion, the IP is notified that the FXOS position and associated operand buffers are free.

The floating-point execution element is highly efficient because it uses operand and instruction buffering and because it contains multiple execution units linked via a common data bus similar to that of the System/360 Model 91 (Chap. 19). (See Fig. 28.) The floating-point element contains an eight-position operation stack (FLOS), four floating-point registers (FLR), six operand buffers (FLB), and three execution units. The add unit, preceded by three reservation stations, can perform two add operations

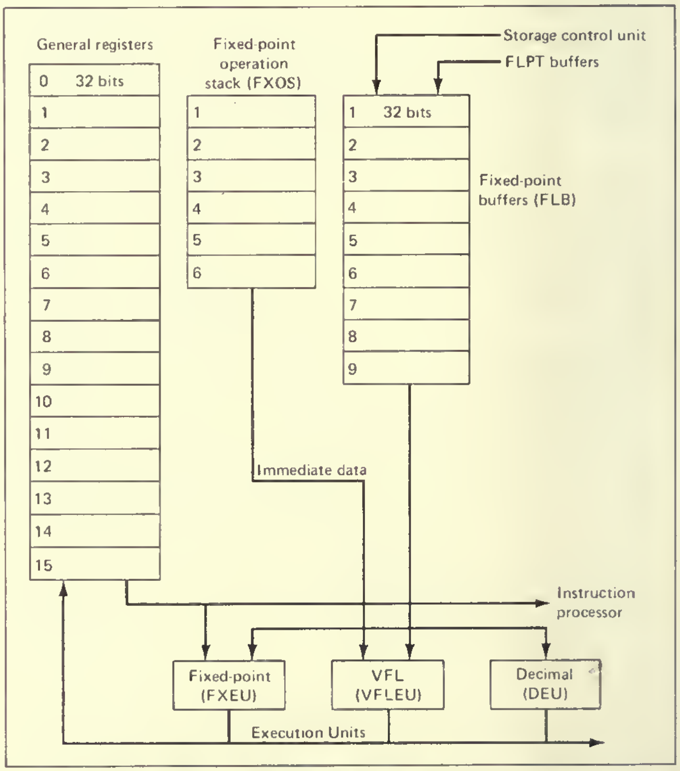


Fig. 27. IBM System/370 Model 195 fixed-point/VFL/decimal execution element data-flow diagram. ***(Courtesy of International Business Machines Corporation.)***

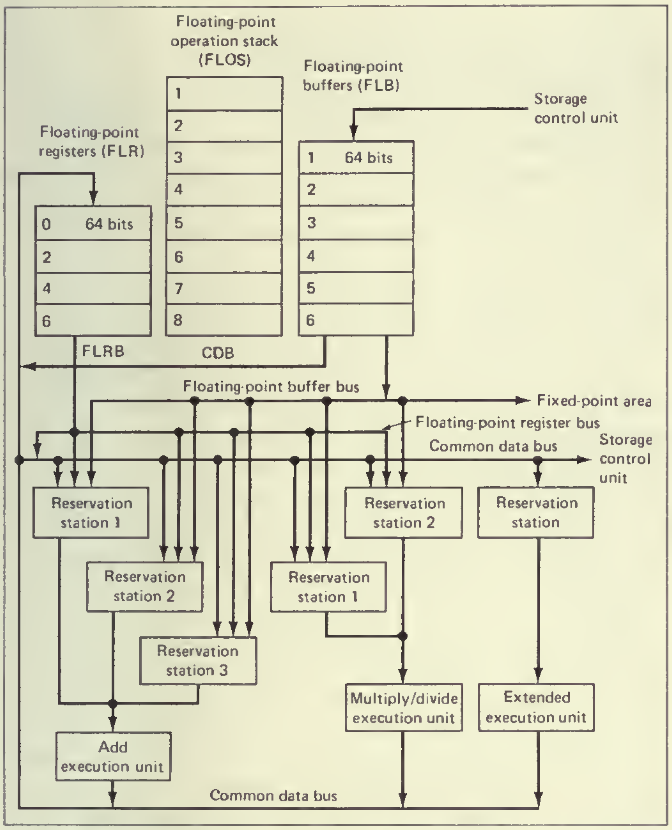


Fig. 28. IBM System/370 Model 195 floating-point execution element data-flow diagram. ***(Courtesy of International Business Machines Corporation.)***

concurrently by offsetting them by one cycle. The third reservation station can acquire data while the other two operations are executing. The multiply/divide execution unit has two reservation stations. Only one multiply or divide operation can be executed at a time. The extended execution unit, with one reservation station, handles extended-precision floating-point operands.

The FLOS has its instructions decoded serially. The FLOS issues operations provided that an appropriate reservation station is available. Since several operations may be executing concurrently, dependent operations are sequenced through the use of tagging on the common data bus.

The 168 was introduced in August 1972. It has the same advantages over the 165 that the 158 has over the 155, i.e., more processing power in smaller cabinets, larger and faster Mp (MOS), and an Integrated Storage Controller as an option. At the same time that the virtual-storage 168 was announced, Dynamic Address Translation (DAT) hardware became available as an expensive option for the 165.

In March 1975 the 168-3 was announced. The 168-3 has twice as many channels as the Model 168 (22, as opposed to 11) and has a cache twice as large (32K, as opposed to 16K).

February 1973 brought forth the 168 MP. The 168 MP is similar to the 158 MP previously described. Two 168's or 168-3's are connected via a 3068 Multisystem Communication Unit (MCU).

In February 1976, the 168 Attached Processor System (APS) was introduced. Similar to the Model 158's 3052 AP in configuration, the 3062 AP (with an 80-ns cycle time) can execute all System/370 instructions except those involved with the "direct control" facility. The 168 AP yields only from 1.5 to 1.8 times the performance of one 168-3. It is possible that the 168, being a high-performance machine with a great amount of prefetching and buffering, is not well suited for use in a very tightly coupled system.

Low-Range Machines: Models 115 and 125. The low-performance machines consist of the Models 115 and 125.

The 125 was announced in October 1972. It was marketed as a growth system for System/360 Model 20 users. The low-performance 125 is implemented in a distributed fashion. (See the discussion of the Model 115.) Two bytes are fetched at a time from the MOS Mp, with no memory interleaving. Data paths are 2 bytes wide. There is no cache. The 480-ns WCS can emulate the 1401/1440/1460 ISPs.

The System/370 Model 115 was introduced in the spring of 1973. It was marketed as a growth system for users of System/360 Models 20, 22, and 25. It bridged the gap between the IBM System/360 and the higher-performance System/370 machines.

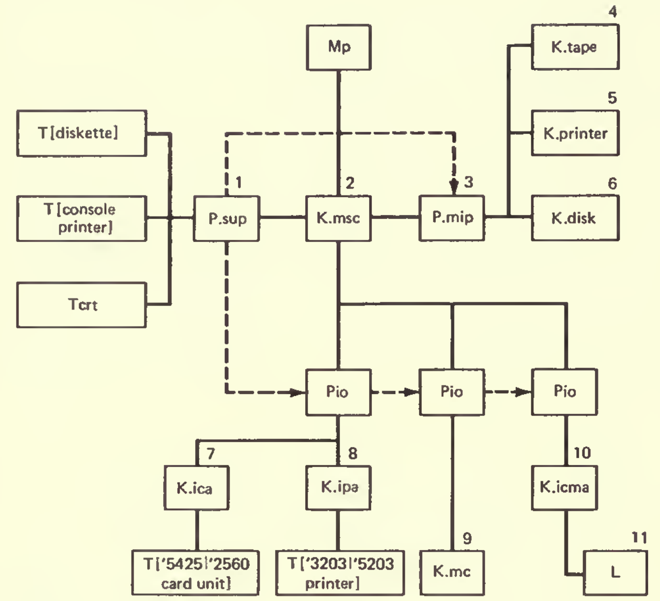
The virtual-storage Model 115, a distributed system, is very similar to the 125. The 115, being the lowest-performance 370, has 1-byte data paths (the 125 has 2-byte paths).

As shown in Fig. 29, the system contains three independent subprocessors. They are the Machine Instruction Processor (MIP), the Input/Output Processor (IOP), and the Service Processor (SVP). Each subprocessor has its own storage, working registers, ALU, and microprogram (MOSFET WCS).

Mp (which is MOS, and requires 480 ns for read/write, and from which 2 bytes are fetched per access, with no interleaving) is controlled by a hardwired Main Storage Controller (MSC). The MSC regularly checks requests for Mp access by the subprocessors, and grants the request with the highest priority. The MSC and the subprocessors communicate via a data bus, a control bus, and direct control lines.

The MIP basically fetches and executes program instructions. The IOP executes I/O commands and supervises the data transfer between the MSC and I/O devices. The interface between the operator and the rest of the system is the SVP. It loads microcode into the subprocessors and boots itself from the console file.

The 115 does not support any block multiplexor or selector channels. However, a byte multiplexor channel can be imple-



Notes

1. P.sup['Supervisor]
2. K.msc['Main Storage Controller]
3. P.mip['Machine Instruction Processor]
4. K.tape[I/O; '3411 Magtape Control|'3803 Tape Control]
5. K.printer['5213 Printer Control]
6. K. disk[I/O; '3344 Disk Controller]
7. K.ica]'Integrated Card I/O Attachment]
8. K.ipa['Integrated Printer Attachment]
9. K.mc['Multiplexor Channel; 32 Subchannels max; 8 shared channels]
10. K.icma['Integrated Communications Adpater]
11. L[5 synchronous lines| [4 synchronous; 8 asynchronous lines]]

Fig. 29. IBM System/370 Model 115 PMS diagram.

mented by using an IOP. Also, some I/O devices can be connected to the MIP with direct attachment features.

The 115-2 and 125-2 were announced in November 1975. The 115-2 split the 115's Machine Instruction Processor into two separate units: a dedicated I/O processor for 3340 disks, and an Instruction Processing Unit.

The 115-2 and 125-2 have higher performance, greater I/O capabilities, and bigger Mp's than the 115 and 125, respectively.

3030 Series Machines

In March 1977, the Model 3033 processor was introduced, and in October of that same year, the Models 3031 and 3032 were announced. The 3030 processors have enhanced price/performance characteristics over their predecessors. High-level operating-system functions are supported in microcode. The 3031 can be configured as an AP system, while the 3033 can be configured as either an MP or an AP system.

The 3031 has a 115-ns Pc microcycle time, a 32-Kbyte cache, and six integrated channels (one byte multiplexor and five block multiplexors). The 3031 yields approximately 1.2 times the performance of a System/370 Model 158-3 [Data Pro, 1978].

The 3032 has an 80-ns Pc microcycle time, a 32-Kbyte cache, and six integrated channels. The system can be extended to include 12 integrated channels. The 3032 yields approximately 2.75 times the performance of a System/370 Model 158-3 [Data Pro, 1978].

The 3033 is a performance-enhanced System/370 Model 168-3 [Connors, Florkowski, and Patton, 1979]. Technology improvements reduced gate delays from 1.7 ns to 1 ns, cache access time from 32 ns to 27 ns, Mp access time from 210 ns to 185 ns, and Pc cycle time from 80 to 58 ns. Cache size, block size, and associativity were all doubled, yielding a 92 percent hit ratio. Memory interleaving was increased from four-way to eight-way. Instruction prefetch buffer (branch and no branch cases) size was increased from two to four doublewords, operand address and data buffer were increased from two to six doublewords, and a third buffer was added for the situation where a second branch is encountered prior to the resolution of a previously encountered branch. Instruction decoding and address generation was decreased from two cycles to one. The 3033 provides 12 channels as standard, divided into two groups of six, and four optional channels. Unlike the 168's stand-alone channels, those of the 3033 are accessible by the service processor, and the group concept allows maintenance, including microdiagnostics, to be performed on one group of channels while the other group is being used for customer work. Each byte multiplexor channel is capable of a data rate of from 40 Kbyte to 75 Kbyte per second, while a block multiplexor channel is capable of up to 1.5 Mbyte. The 3033 system requires one-half of the space and 30 percent less power than a similarly configured 168. Performance is 1.8 to 1.9 times that of a 168. Both the 3033 MP and 3033 AP are rated at 1.6 to 1.8 times the performance of a uniprocessor 3033.

4300 Series Machines

In January 1979 the models 4331 and 4341 were announced, thus launching the 4300 series. The series is implemented with IBM's bipolar gate arrays, with up to 704 gates/chip at 3-ns switching speeds and 64-Kbit MOS memory chips. The 4300 series added 12 new instructions to the System/370 ISP.

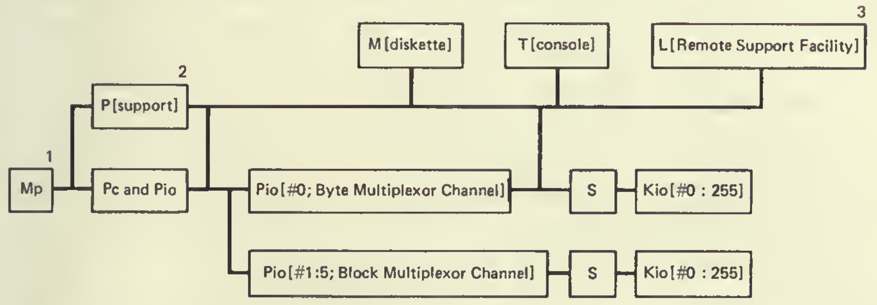
The 4331 is rated at 0.88 to 0.99 times the performance of, requires 70 percent less power than, and costs approximately 28 percent as much as the System/370 Model 138. The 4331 features a 64K writable control store, dynamic address translation, storage protection, a time-of-day clock, support for remote diagnostics, and a support processor for monitoring and recording environmentally caused problems such as power variances. The Pc also requires from 16 to 53 Kbyte of main memory for extra control store capacity. An integrated DASD (Direct Access Storage Device) adapter allows the direct attachment (without controllers or channels) of four strings of disk storage. One byte multiplexor channel and one block multiplexor channel are provided for attachment of a variety of System/360 and System/370 peripherals. The channel data rate for the byte multiplexor channel is 18 Kbytes in byte mode and 500 Kbyte in burst mode. The block multiplexor channel has a 500 Kbyte/s maximum data transfer rate. A Communications Adapter allows for attachment of up to eight communications lines operating from 75 bits/s to 56 Kbit/s. Two of the following line protocols can be supported simultaneously: Synchronous Data Link Control (SDLC), Binary Synchronous Control (BSC), and asynchronous line.

The 4341 has two modes of operation (corresponding to whatever microcode is loaded): (1) System/370-compatible mode with Extended Control Program Support for VM/VS software and (2) Extended Control Program Support: Virtual Storage Extended. The latter mode is unique to the 4341, yet is basically compatible with the System/370. The 4341 is rated at 3.2 times the performance of the System/370 Model 138 while costing 6 percent less. Figure 30 depicts the overall organization of the 4341. A portion of Mp, ranging from 14 to 108 Kbyte, is required for dynamic table storage. A separate support processor controls initialization (initial microcode loading), error analysis and logging, and the display console. The Remote Support Facility (RSF) provides the capability of remotely controlling the 4341 from an IBM service center. Diagnostics can be executed remotely and error information sent back to the service center. RSF, ECC (SEC/DED) Mp, Pc parity checks, instruction retry, channel command retry, and internal-state logout provide a comprehensive reliability/maintainability environment.

The 4341 I/O channels, with a few variations, are identical to those of the System/360 and System/370. One byte multiplexor and up to five block multiplexor channels can be configured. The byte multiplexor channel operating in byte-interleaved mode permits several low-speed devices to operate concurrently at up to 32 Kbyte/s if no block multiplexor channels are operating. It can also operate in burst mode, allowing one high-speed device at a time to function at a maximum rate of 1 Mbyte/s. Two block multiplexor channels are standard and three more are optional. Each block multiplexor is capable of a maximum data rate of 1 Mbyte/s. The total system data rate is limited to 9 Mbyte/s.

The PMS Structure of the System 370 Pio, K, Ms, and T

The System/370 peripheral structure has resulted from a slow evolution of the System/360 peripheral structure. The PMS diagrams for the System/370 have been laid out to allow a quick comparison with the System/360 PMS diagrams. In many cases, the System/370 PMS diagrams are a superset of the System/360 diagrams.[1] Thus while some peripherals have been discontinued, IBM continues to support other System/360 peripherals.


M[diskette]
T[console]
L[Remote Support Facility]
3
2
P[support]
1
Mp
Pc and Pio
Pio[#0; Byte Multiplexor Channel]
S
Kio[#0 : 255]
Pio[#1:5; Block Multiplexor Channel]
S
Kio[#0 : 255]

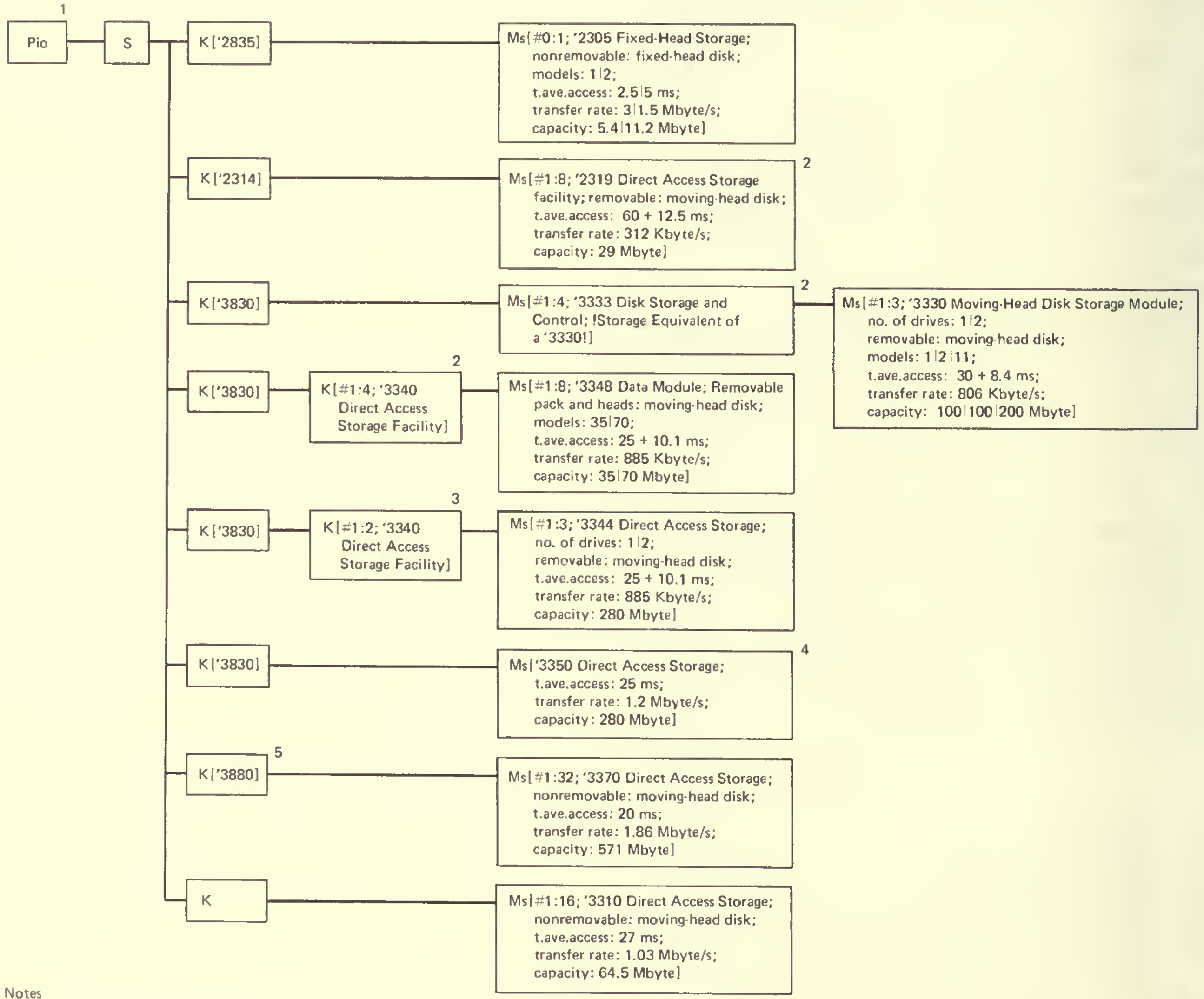
Notes
1. Mp[MOS; 2-4 Mbyte; ECC (SEC/DED)]
2. P[Support Processor; error analysis, display console, initial microcode loading]
3. L[!Remote Support Facility for remote diagnostics]

Fig. 30. IBM 4341 PMS diagram.

[1] The rate of change of the PMS diagrams is directly related to the rate of change of the major technology composing the PMS structure. Thus, as indicated in Part 1, Sec. 2, the Pc, based on electronic technology, changes fastest. Ms, composed of electronic, magnetic, and mechanical technologies, changes at the next highest rate. Indeed, there is only one carryover between the System/360 and System/370 Ms PMS diagrams. The rate of change of magnetic tape, which has a large component of mechanical technology, is slower still than Ms technology. The slowest rate of change is exhibited by paper-handling peripherals (e.g., paper-tape devices, card reader/punches, and line printers), which are essentially mechanical technologies. Thus technologists seek higher performance by seeking ways to replace mechanical technology by electronic technology (e.g., electrostatic units for printing, charge-coupled devices or magnetic bubbles for Ms).

The major I/O port for the System/360 processors above the Model 50 was either a 2860 Selector Channel (for high-speed devices) or a 2870 Multiplexor Channel (for low-speed devices). The System/370 added the 2880 Block Multiplexor Channel to support even higher-speed, block-data–oriented peripherals. A major departure from the System/360 I/O architecture is the integration of controller/channels at the high end and adapters at the low end into the Pc cabinet, which eliminates cost and performance penalties of stand-alone cabinets.

Figure 31 depicts secondary storage for System/370 processors. The multipurpose 2841 Storage Control has given way to the 3830. The System/370 utilizes high-speed fixed-head disks for


1
Pio
S
K['2835]
Ms[#0:1; '2305 Fixed-Head Storage; nonremovable: fixed-head disk; models: 1|2; t.ave.access: 2.5|5 ms; transfer rate: 3|1.5 Mbyte/s; capacity: 5.4|11.2 Mbyte]
K['2314]
2
Ms[#1:8; '2319 Direct Access Storage facility; removable: moving-head disk; t.ave.access: 60 + 12.5 ms; transfer rate: 312 Kbyte/s; capacity: 29 Mbyte]
K['3830]
2
Ms[#1:4; '3333 Disk Storage and Control; !Storage Equivalent of a '3330!]
Ms[#1:3; '3330 Moving-Head Disk Storage Module; no. of drives: 1|2; removable: moving-head disk; models: 1|2|11; t.ave.access: 30 + 8.4 ms; transfer rate: 806 Kbyte/s; capacity: 100|100|200 Mbyte]
K['3830]
2
K[#1:4; '3340 Direct Access Storage Facility]
Ms[#1:8; '3348 Data Module; Removable pack and heads: moving-head disk; models: 35|70; t.ave.access: 25 + 10.1 ms; transfer rate: 885 Kbyte/s; capacity: 35|70 Mbyte]
K['3830]
3
K[#1:2; '3340 Direct Access Storage Facility]
Ms[#1:3; '3344 Direct Access Storage; no. of drives: 1|2; removable: moving-head disk; t.ave.access: 25 + 10.1 ms; transfer rate: 885 Kbyte/s; capacity: 280 Mbyte]
K['3830]
4
Ms['3350 Direct Access Storage; t.ave.access: 25 ms; transfer rate: 1.2 Mbyte/s; capacity: 280 Mbyte]
K['3880]
5
Ms[#1:32; '3370 Direct Access Storage; nonremovable: moving-head disk; t.ave.access: 20 ms; transfer rate: 1.86 Mbyte/s; capacity: 571 Mbyte]
K
Ms[#1:16; '3310 Direct Access Storage; nonremovable: moving-head disk; t.ave.access: 27 ms; transfer rate: 1.03 Mbyte/s; capacity: 64.5 Mbyte]

Notes

1. Pio[Block Multiplexor Channel | Selector Ch]
2. Ms | K[!Can attach directly to Integrated File Adapters without need for controller]
3. K[!Can attach directly to Integrated File Adapters or '3145 or '3345 Integrated Storage Control]
4. Ms[!Can attach directly to a Byte or Block Multiplexor Channel]
5. K[!Programmable controller for use with 4341 processor!]

Fig. 31. IBM System/370 Ms [disk] PMS diagrams.

swapping store rather than the high-speed drums used in the System/360. Hence drum storage has been discontinued. The increased storage capacity of disks partially fulfills the requirement for online archival memory as represented by the discontinued System/360 Data Cell magnetic card technology. Disks thus evolved as the dominant MS technology, as graphically indicated by comparing Figs. 31 and 13. New technologies, such as charge-coupled devices or magnetic bubbles, may eventually replace disks as the dominant Ms technology, just as the disks replaced drums. However, disks will remain the dominant Ms technology for at least the next several years.

Figure 32 depicts the growth in magnetic tape peripherals over those of the System/360. The System/360 Hypertape was discontinued.

Figure 33 shows the two card punches, the card reader, and the three card reader/punch units of the System/360, which were augmented for the System/370 by the addition of three reader/punch units. A lower-performance paper-tape reader has been added to the 1000 character per second 2671. A wider range of line printer models (six instead of one) is offered with the System/370, including an electrostatic printer that is 8 times faster than the highest-performance System/360 mechanical line printer. The System/370, like the System/360, offers a range of commercial-document readers, including magnetic character readers, optical readers (for printed, marked, and hand-printed documents), and document-processing units (for semiautomatic document-information input).

Finally, Fig. 34 illustrates that front-end processors have replaced the System/360's simple controllers for communication lines.

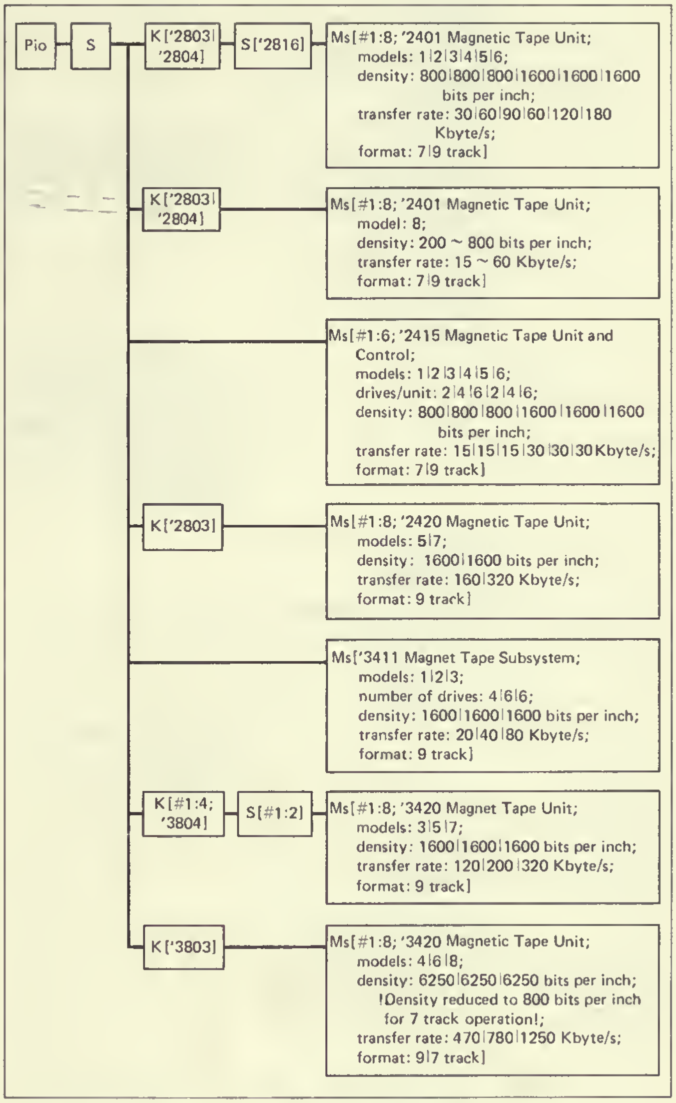

Pio
S
K['2803|'2804]
S['2816]
Ms[#1:8; '2401 Magnetic Tape Unit; models: 1|2|3|4|5|6; density: 800|800|800|1600|1600|1600 bits per inch; transfer rate: 30|60|90|60|120|180 Kbyte/s; format: 7|9 track]
K['2803|'2804]
Ms[#1:8; '2401 Magnetic Tape Unit; model: 8; density: 200 ~ 800 bits per inch; transfer rate: 15 ~ 60 Kbyte/s; format: 7|9 track]
Ms[#1:6; '2415 Magnetic Tape Unit and Control; models: 1|2|3|4|5|6; drives/unit: 2|4|6|2|4|6; density: 800|800|800|1600|1600|1600 bits per inch; transfer rate: 15|15|15|30|30|30 Kbyte/s; format: 7|9 track]
K['2803]
Ms[#1:8; '2420 Magnetic Tape Unit; models: 5|7; density: 1600|1600 bits per inch; transfer rate: 160|320 Kbyte/s; format: 9 track]
Ms['3411 Magnet Tape Subsystem; models: 1|2|3; number of drives: 4|6|6; density: 1600|1600|1600 bits per inch; transfer rate: 20|40|80 Kbyte/s; format: 9 track]
K[#1:4; '3804]
S[#1:2]
Ms[#1:8; '3420 Magnet Tape Unit; models: 3|5|7; density: 1600|1600|1600 bits per inch; transfer rate: 120|200|320 Kbyte/s; format: 9 track]
K['3803]
Ms[#1:8; '3420 Magnetic Tape Unit; models: 4|6|8; density: 6250|6250|6250 bits per inch; !Density reduced to 800 bits per inch for 7 track operation!; transfer rate: 470|780|1250 Kbyte/s; format: 9|7 track]

Fig. 32. IBM System/370 Ms [magnetic tape] PMS diagram.

Performance and Cost

The IBM System/360 and System/370 series provide a number of data points in the implementation space for a common ISP. Furthermore, being marketed by a single organization, they are probably constrained by a common corporate profit goal. In this section, we will focus on Pc-Mp cost, performance, and cost/performance. Costs vary over time as technology and marketing competition change. However, we can plot cost as a function of time. Performance is more difficult to determine, since it depends on system configuration, software quality, and work load. IBM will announce relative performance ratios for Pc's but will not divulge absolute measures such as mega-instructions per second. Finally, cost/performance is extremely difficult to calibrate because of the many variables impacting both cost and performance.

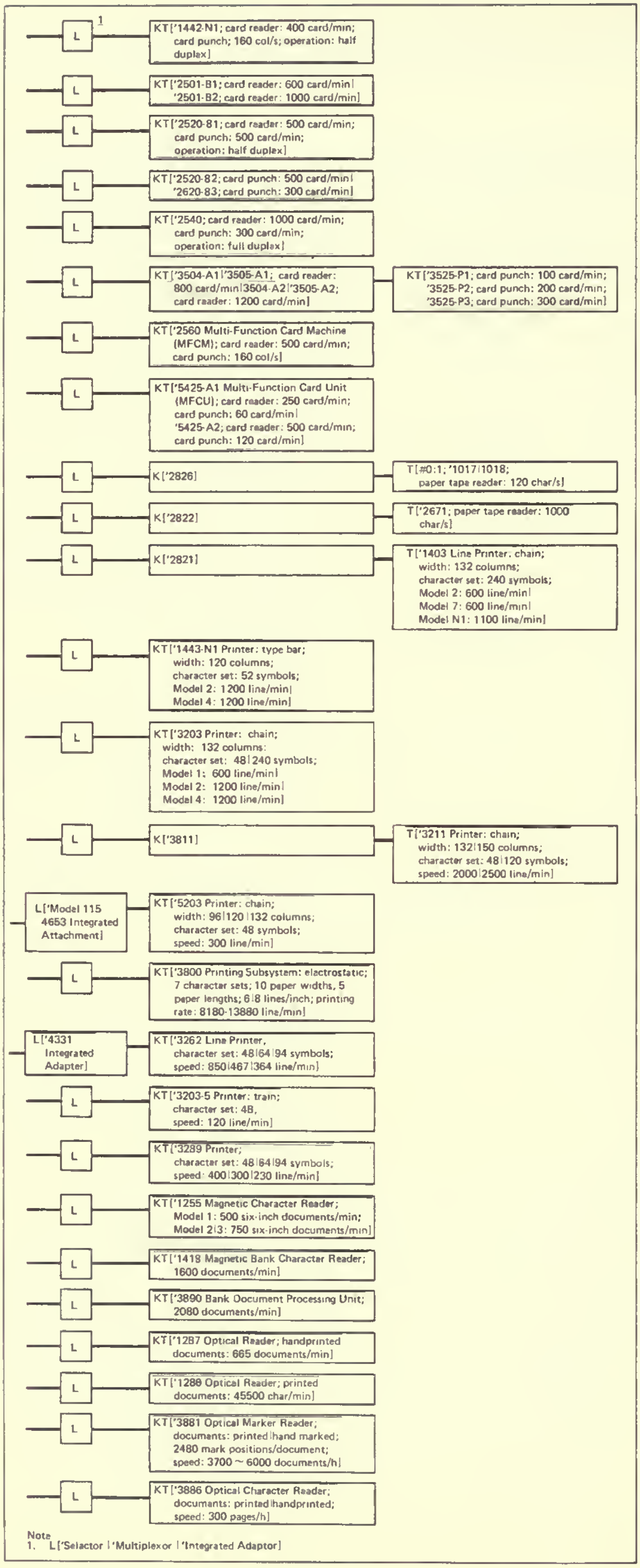
Price

Figure 35 depicts the price of IBM memory per byte as a function of year. The slope of the line indicates an average decrease in the cost of memory per byte of 19 percent per year from the introduction date of the first System/360. Figure 36 plots the price of disk storage, indicating a 21 percent decrease per year in cost.

Unlike memory, where there is essentially one dominant technology at any given time, a plot of processor cost is not very meaningful, because of the wide range in processor performance. Processor evaluation must wait until we discuss performance.

Performance

As indicated, determining the performance of a processor is very difficult because of all the variables that determine performance.


1
L
KT['1442-N1; card reader: 400 card/min; card punch; 160 col/s; operation: half duplex]
KT['2501-B1; card reader: 600 card/min | '2501-B2; card reader: 1000 card/min]
KT['2520-81; card reader: 500 card/min; card punch: 500 card/min; operation: half duplex]
KT['2520-82; card punch: 500 card/min | '2620-83; card punch: 300 card/min]
KT['2540; card reader: 1000 card/min; card punch: 300 card/min; operation: full duplex]
KT['3504-A1 | '3505-A1; card reader: 800 card/min | 3504-A2 | '3505-A2; card reader: 1200 card/min]
KT['3525-P1; card punch: 100 card/min; '3525-P2; card punch: 200 card/min; '3525-P3; card punch: 300 card/min]
KT['2560 Multi-Function Card Machine (MFCM); card reader: 500 card/min; card punch: 160 col/s]
KT['5425-A1 Multi-Function Card Unit (MFCU); card reader: 250 card/min; card punch; 60 card/min | '5425-A2; card reader: 500 card/min; card punch: 120 card/min]
K['2826]
T[#0:1; '1017 | 1018; paper tape reader: 120 char/s]
K['2822]
T['2671; paper tape reader: 1000 char/s]
K['2821]
T['1403 Line Printer: chain; width: 132 columns; character set: 240 symbols; Model 2: 600 line/min | Model 7: 600 line/min | Model N1: 1100 line/min]
KT['1443-N1 Printer: type bar; width: 120 columns; character set: 52 symbols; Model 2: 1200 line/min | Model 4: 1200 line/min]
KT['3203 Printer: chain; width: 132 columns: character set: 48 | 240 symbols; Model 1: 600 line/min | Model 2: 1200 line/min | Model 4: 1200 line/min]
K['3811]
T['3211 Printer: chain; width: 132 | 150 columns; character set: 48 | 120 symbols; speed: 2000 | 2500 line/min]
L['Model 115 4653 Integrated Attachment]
KT['5203 Printer: chain; width: 96 | 120 | 132 columns; character set: 48 symbols; speed: 300 line/min]
KT['3800 Printing Subsystem: electrostatic; 7 character sets; 10 paper widths, 5 paper lengths; 6 | 8 lines/inch; printing rate: 8180-13880 line/min]
L['4331 Integrated Adapter]
KT['3262 Line Printer, character set: 48 | 64 | 94 symbols; speed: 850 | 467 | 364 line/min]
KT['3203-5 Printer: train; character set: 48, speed: 120 line/min]
KT['3289 Printer; character set: 48 | 64 | 94 symbols; speed: 400 | 300 | 230 line/min]
KT['1255 Magnetic Character Reader; Model 1: 500 six-inch documents/min; Model 2 | 3: 750 six-inch documents/min]
KT['1418 Magnetic Bank Character Reader; 1600 documents/min]
KT['3890 Bank Document Processing Unit; 2080 documents/min]
KT['1287 Optical Reader; handprinted documents: 665 documents/min]
KT['1288 Optical Reader; printed documents: 45500 char/min]
KT['3881 Optical Marker Reader; documents: printed | hand marked; 2480 mark positions/document; speed: 3700 ~ 6000 documents/h]
KT['3886 Optical Character Reader; documents: printed | handprinted; speed: 300 pages/h]
Note
1. L['Selector | 'Multiplexor | 'Integrated Adaptor]

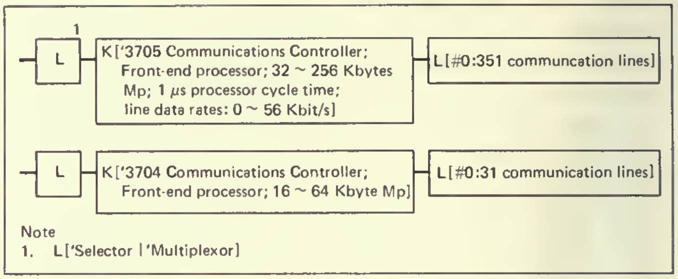

1
L
K['3705 Communications Controller; Front-end processor; 32 ~ 256 Kbytes Mp; 1 μs processor cycle time; line data rates: 0 ~ 56 Kbit/s]
L[#0:351 communcation lines]
L
K['3704 Communications Controller; Front-end processor; 16 ~ 64 Kbyte Mp]
L[#0:31 communication lines]
Note
1. L['Selector | 'Multiplexor]

Fig. 34. IBM System/370 communications PMS diagrams.

This is especially true if only one performance number is sought. Table 3 indicates the different results of measuring performance that are quoted in the literature. Both *Electronics* and Dean Witter Reynolds present raw performance ranges in unspecified environments and work loads. Data Pro presents relative processor model performances, again in unspecified situations. *Computer World* (February 5, 1979) provides both raw and relative performance. The COBOL benchmarks represent a synthetic benchmark composed of 11 operations typically found in COBOL application programs. Each test is executed 100,000 times. The COBOL Analysis System (CAS) has been run in 125 different hardware/software environments representing 13 major manufacturers. The first test was run in 1965. Distributed by U.S. Steel, the CAS results allow relative comparisons between various hardware/compiler/operating-system environments on a uniform task. Figure 37 is a compilation of the relative performances of most of the IBM models.

We have attempted to model System/360 and System/370 performance by a simple model relating performance to microcycle time and memory pause time per byte:

$$1/p = k_1 t_1 + k_2 t_2$$

where t_1 is the time for one microcycle and t_2 is the memory pause time per byte. The time for t_2 is the memory access time minus the microcycle time, increased to the next higher multiple of a microcycle and then divided by the number of bytes fetched per memory reference cycle. The same model has been applied to the PDP-11 and resulted in a good fit (see Chap. 39).

Three sources of relatively complete performance data were available: the average instruction time, as used in Bell and Newell [1971] (add up all instruction times and divide by the number of instructions); the relative performance from Data Pro; and the COBOL benchmarks. The average instruction time was not used, since it varied significantly from the more detailed sources (see Table 4).

The data for each machine are given in Tables 5 and 6. The performance is relative to the System/360 Model 30. Where a

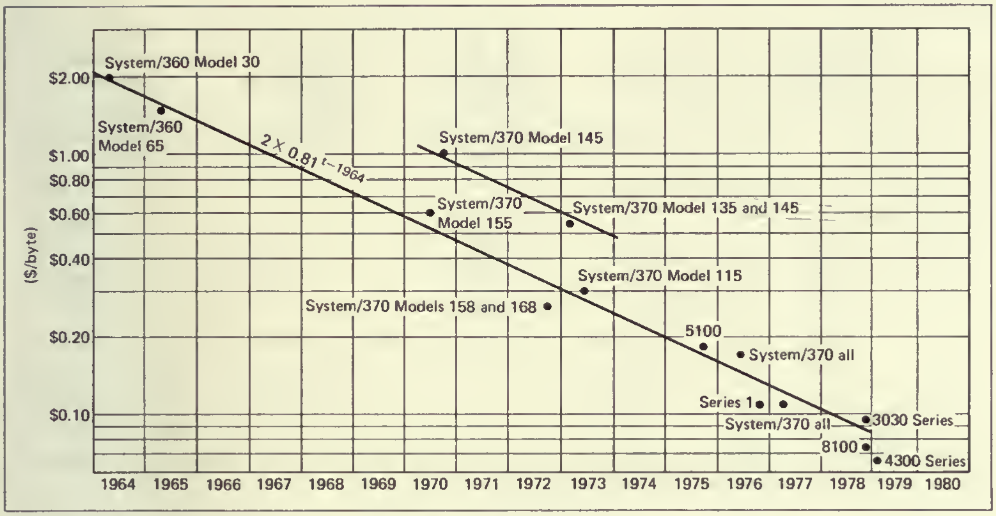


Fig. 35. IBM memory price history. (11 years = factor of 10.)

model is found in both tables, the relative performances agree within a factor of 1.5. A linear regression was applied to both sets of data to determine the coefficients k_1 and k_2 and to find out how much of the variance was explained by the model.

The first question to answer is, Do the System/360 and System/370 constitute one or two families? Figure 37 plots the relative average instruction time versus the microcycle time for both the COBOL benchmark and Data Pro data. It can be seen that the curve through the System/360 processors is substantially displaced with respect to the System/370 curve. The regression analysis upheld this observation.

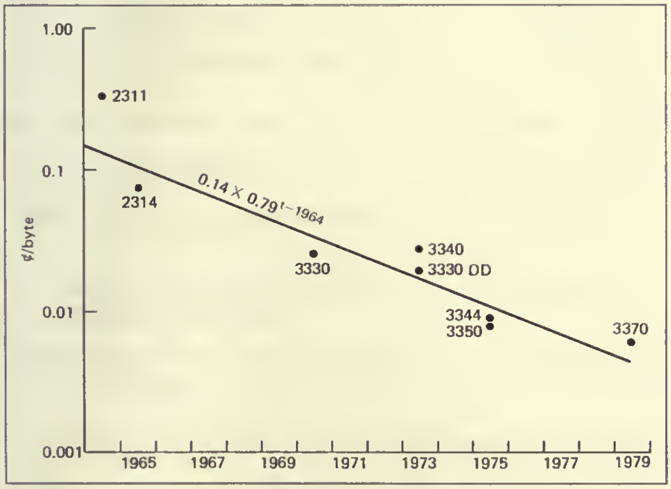


Fig. 36. IBM disk storage price history.

Applying the regression model to the four System/360 processors in Table 5 yielded $k_1 = 8.32 \times 10^{-4}$, $k_2 = 1.19 \times 10^{-4}$, $k_1/k_2 = 7$, and $R^2 = .88$. R^2 is the amount of variance accounted for by the model, or 88 percent. The large ratio of k_1/k_2 indicates that the Pc microcycle time is the dominating factor in determining System/360 performance, almost to the exclusion of memory system performance. Thus System/360 models are processor-bound, or memory subsystems are overdesigned with respect to the processing engine. This mismatch was observed in a different way in the prior discussions on the System/360 Model 67 and the System/360 Model 91.

The regression model was applied to the six System/370 models in Table 5 (the Model 125 was excluded to give a better fit), yielding $k_1 = 4.8 \times 10^{-4}$, $k_2 = 2.1 \times 10^{-3}$, $k_1/k_2 = 0.2285$, and $R^2 = .99$. In the System/370 series the processor is no longer the system bottleneck.

The regression model was also applied to the fifteen System/370 models in Table 6, yielding $k_1 = 7.34 \times 10^{-4}$, $k_2 = 6.51 \times 10^{-4}$, $k_1/k_2 = 1.13$, and $R^2 = .90$. The differences between the two System/370 models represent variation in System/370 performance in different task environments. Indeed, the smaller k_1 (less processor-bound) and larger k_2 (more memory-bound) for the COBOL benchmarks seem to indicate that the System/370 is tuned to the COBOL task. This observation corresponds to the consistently higher relative performance of the COBOL benchmarks in Table 3.

Price/Performance

The System/360 cost is based on dollars per second to rent the equipment. The figures were derived at one point in time from

Table 3 Measures of Performance of System/370 Models

Model number	Electronics *performance (mips)*[a]	Computer World *performance (mips)*[b]	*Dean Witter Reynolds performance (mips)*[c]	Electronics *performance relative to Model 135*	*Data Pro performance relative to Model 135*[d]	Computer World *performance relative to Model 135*	*COBOL benchmark performance relative to Model 135*[e]
135	0.12–0.16	.18	....	1.0	1.0	1.0	1.0
138	0.15–0.20	.2	....	1.28	1.34	1.11	....
145	0.23–0.3	.3	0.4	2.0	1.87	....	2.29
148	0.33–0.43	.4	0.5	2.86	2.66	2.28	....
158-3	0.7–0.9	....	0.9	5.72	5.65	4.27	13.38
168-3	2.5–2.7	....	2.5	18.57	15.68	....	28.53
3031	....	....	1.1	....	....	5.13	....
3032	....	....	2.5	....	....	....	....
3033	4.5–4.9	....	5.0	33.57	....	21.18	....
4331	....	.2	0.2	....	....	....	....
4341	....	.5	....	....	....	....	....

[a] From A. Durniak, "Soviet Computers: Better than Expected," *Electronics,* Sept. 28, 1978, pp. 85–86.

[b] *Computer World,* Feb. 5, 1979.

[c] Dean Witter Reynolds, Inc., *Random Access Monthly,* June 1978, p. 4.; March 1980, p. 5.

[d] Data Pro [1978].

[e] U.S. Steel [1978].

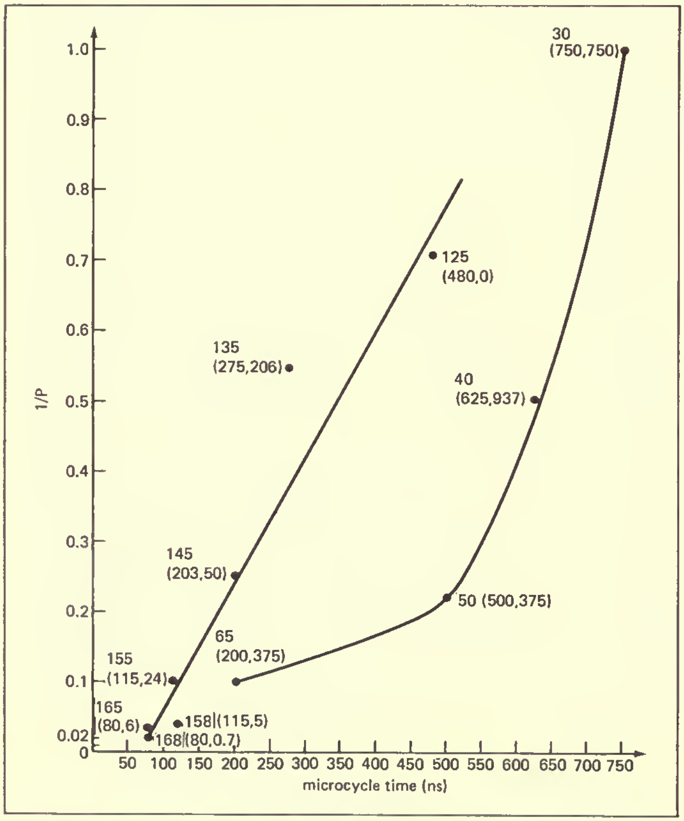


Fig. 37. Relative performance as a function of microcycle time for the IBM System/360 and IBM System/370 series.

the IBM monthly rental prices. The computer prices are based on estimates of minimum, average, and maximum configurations in the *Adams Computer Characteristics Quarterly* [Adams Associates, 1960]. The conversion factors are

$$\begin{aligned} 1 \text{ month} &= 1 \text{ month} \times 173.3 \text{ h/month} \times 3{,}600 \text{ s/h} \\ &= 0.625 \times 10^6 \text{s} \end{aligned}$$

The price to buy, in dollars, is approximately

$$\begin{aligned} \text{Purchase price (\$)} &= 45 \times \text{price(\$/month)}^{\dagger} \\ &= 45 \times 0.625 \times 10^6 \times \text{price(\$/s)} \\ &= 2.82 \times 10^7 \times \text{price(\$/s)} \end{aligned}$$

Three costs are calculated:

Pc(price: ($/s|$)): = c.Pc: = price of Pc alone
Mp(price.avg): = c.Mp.avg: = price of average-size Mp for a model
C(price.min:): = c.C.min: = price of minimum-size computer configuration
C(price.avg:): = c.C.avg: = price of average-size computer configuration

The relative System/360 Pc powers (in instructions per second)

†As of 1977 the purchase/lease ratio on some System/370 systems was as low as 35:1.

Table 4 Relative Performance as Computed from Average Instruction Time and COBOL Benchmarks

	Performance relative to the Model 30	
Model	*Bell and Newell [1971]*	*COBOL benchmarks*
30	1	1
40	3	2
50	7.5	4.5
65	31.5	11.5

and prices are given in the graph of Fig. 38. The most significant fact from the graph is that the price/power ratio is roughly constant for each of the Pc's (especially if we ignore Model 44 and Model 50). Figure 38 gives the relative computing power versus price for various configurations.

The performance of C['20] is inaccurately high, since it is a limited subset of the 360 ISP. (C['20] does not have floating-point or fixed-point multiply and divide instructions, and it has only eight 16-bit general registers.) The hardwired Model 44 has a better price/power characteristic than any of the other C's, by any measured criteria (see Fig. 38). In the case of the Model 44, the Pc price also includes Ms.disk. Perhaps the Model 44, designed initially for real time scientific problem solving, is priced more competitively with similar machines (DEC PDP-10 and SDS Sigma 5, 7), whereas the other models compete in a performance-insensitive, competition-free market for general-purpose business data processing. Thus its anomalous position may be due to external market pressures and not manufacturing cost.

Table 5 COBOL Benchmarks

Model	*p*	*1/p*	$t_{\mu c}$*(ns)*		t_p	
30	1	1	750	[2.4]	750	[3.46]
40	2	0.5	625	[2.0]	937	[9.32]
50	4.5	0.22	500	[1.6]	375	[1.73]
65	10	0.1	200	[0.64]	375	[0.173]
125	1.4	0.71	480	[1.5]	0	
135	1.8	0.55	275	[0.88]	206	[0.949]
145	4	0.25	203	[0.65]	50	[0.23]
155	10	0.1	115	[0.37]	24	[0.11]
158-3	23.7	0.04	115	[0.37]	5	[0.023]
165	31.7	0.03	80	[0.26]	6	[0.028]
168-3	50.5	0.02	80	[0.26]	0.7	[0.003]

$1/p = c_1 t_{\mu c} + c_2 t_p$

Bracketed quantities are normalized wrt average

For $t_{\mu c}, t_p$ normalized:

$$\left.\begin{aligned} c_1 &= .40 \\ c_2 &= .05 \end{aligned}\right\} \text{360 \& 370 } R^2 = .87$$

$$\left.\begin{aligned} c_1 &= .31 \\ c_2 &= .0038 \end{aligned}\right\} \begin{aligned} &\text{360 \& 370 } R^2 = .85 \\ &\text{without 125} \end{aligned}$$

$$\left.\begin{aligned} c_1 &= .15 \\ c_2 &= .46 \end{aligned}\right\} \begin{aligned} &\text{370 } R^2 = .99 \\ &\text{without 125} \end{aligned}$$

$$\left.\begin{aligned} c_1 &= .26 \\ c_2 &= .026 \end{aligned}\right\} \text{360 } R^2 = .88$$

$t_{\mu c}$ = time for one micro cycle

tp = memory pause time per byte

Table 6 Data Pro Evaluation of the System/370

Model	*p*	*1/p*	$t_{\mu c}$	t_p
125-2	1.8	0.55	320	160
135	2.9	0.34	275	206
135-3	3.8	0.26	275	206
138	3.9	0.26	275	206
145	5.45	0.18	203	50
145-3	7.1	0.14	180	45
148	7.7	0.13	180	45
155	10	0.1	115	24
155-2	10	0.1	115	24
158	15	0.07	115	5
158-3	16.4	0.06	115	5
165	34.5	0.03	80	6
165-2	34.5	0.03	80	6
168	41.8	0.02	80	0.7
168-3	45.5	0.02	80	0.7

The design of the IBM System/360 models is undoubtedly predicated on the basis that performance or computing power is proportional to the price raised to some power g:power = $k \times \text{price}^g$, where $g > 1$.[1] Almost all models follow the above relationship with $g > 1$. When $g > 1$, there is an advantage to having large configurations, since the price/computation will decrease. If $g \leq 1$, then an alternative implementation for the 360 C's would simply use multiple C's or Pc's to obtain the same power. Unfortunately, such an approach does not provide for the interconnection of the components to function as a single unit. In many cases, a single task cannot be broken into a number of parallel and independent subtasks. If the performance for the system varied by a factor of 100, then 100 Pc's or C's would be placed together.

The following discussion takes computing power to be measured by instructions per second and Mp [size; t.cycle]. Prices are measured in dollars per second of rental time. The graph (Fig. 39) shows the relationship to computing power p and prices. The power (actually p.Pc) is taken from the measures of instruction

[1] Herb Grosch [Grosch, 1953] first noted this relationship and estimated g to be 2; thus we use g for this exponent. Adams suggested $g = ½$ [Adams, 1962]. See also Sharpe [1969].

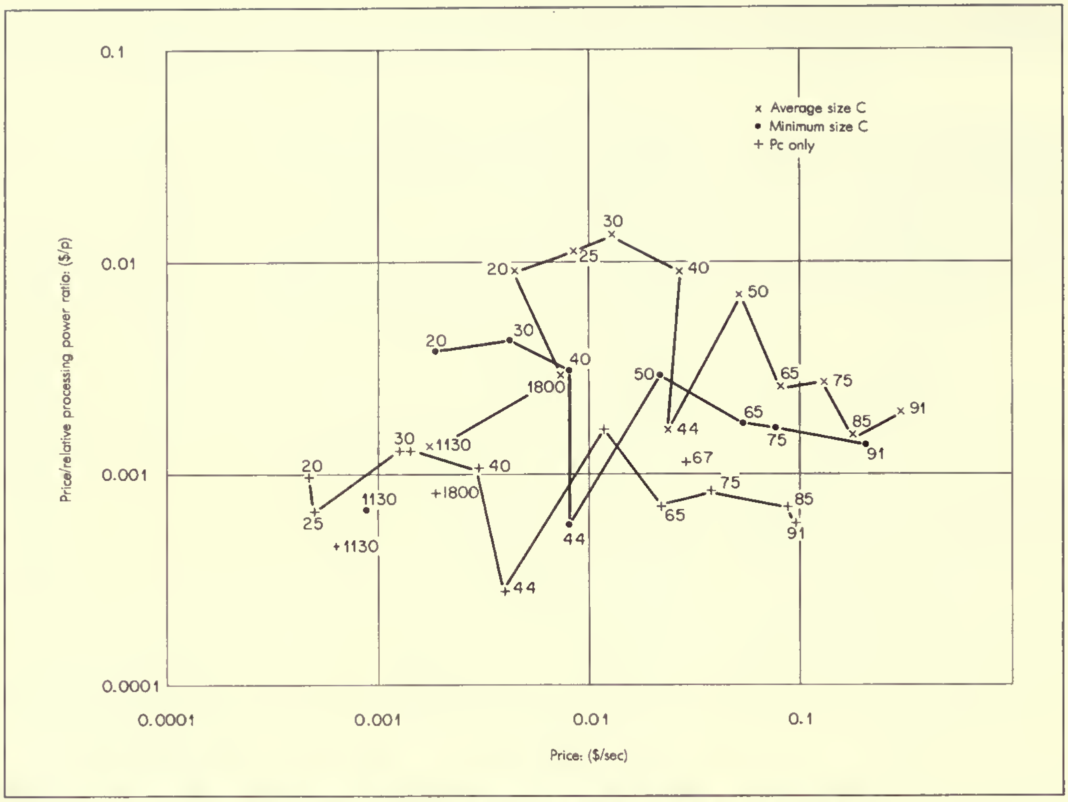


Fig. 38. Graph of IBM System/360 cost/processing power ratio versus rental price.

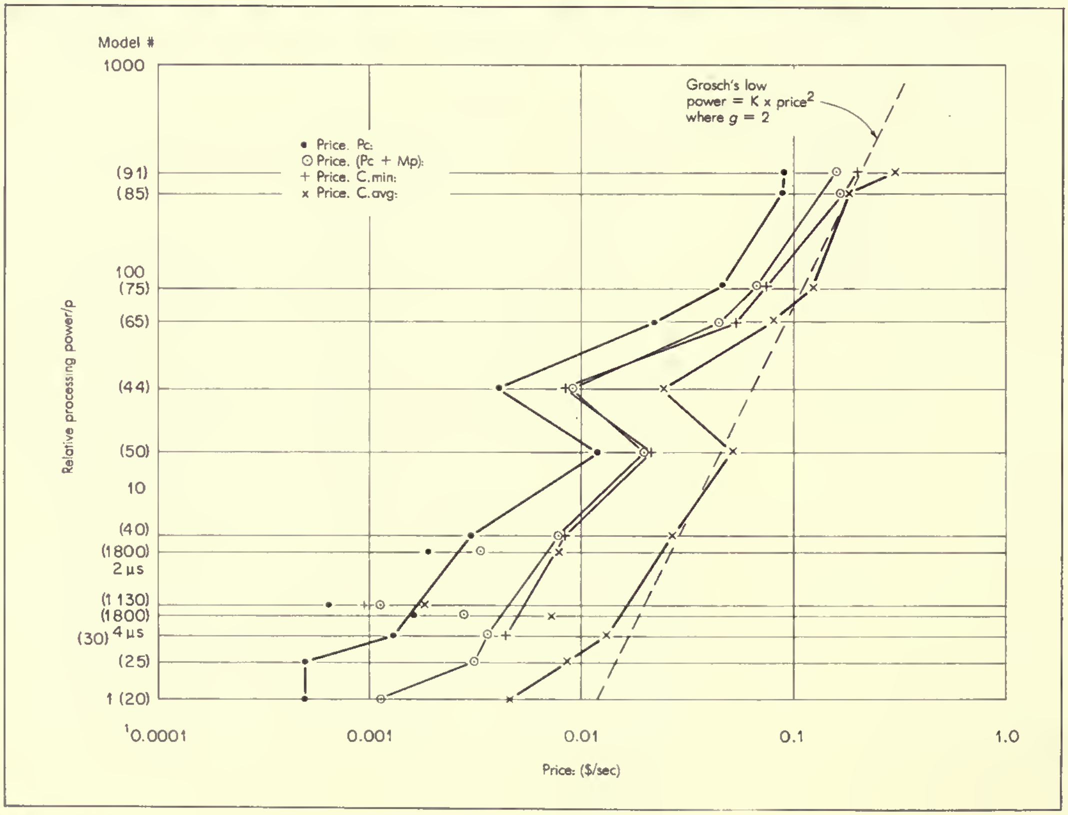


Fig. 39. Graph of IBM System/360

times for certain fixed work. Solomon [1966] observed Grosch's law to hold for Models 30, 40, 50, 65, and 75. This line is drawn in Fig. 39 for C(price.average). Considering Models 20, 25, 44, 85, and 91, a line with a less steep slope might fit the points better. If we consider C(price.minimum), $g > 2$; considering only Pc, a $g = 1$ might be appropriate (see Fig. 39) for which the Pc power/price is essentially constant with cost.

> Pc(price)/Mp(price.avg): = c.Pc/avg.Mp = ~ 1.1, the ratio of processor to memory price
>
> C(price.min)/C(price.avg): = c.min.C/avg.C = ~ 0.47, the ratio of the smallest computer configuration to an average configuration
>
> Pc(price)/C(price.avg): = c.Pc/c.avg.C = ~ 0.23, the ratio of processor to computer price

These are averages over all the series and can be rather misleading. For example, in higher-numbered models the C(price.min)/C(price.avg): = c.min.C/c.avg.C is about 0.6, whereas in lower-numbered models the ratio is 0.3. We might have expected this, since it indicates that a higher proportion of system cost is in Ms and T on lower-number models.

The price for the System/370 series is based on purchase price. Figure 40 gives the relative computing power versus price for a Pc with average Mp size. Again, the price/power ratio is almost constant with at most a 3:1 variation. The best-performing models seem to be the replacement models (i.e., the 138 for the 135, the 148 for the 145). Presumably, newer technology and packaging yielded an increase in performance. Models 165 and 168 are clear price/performance leaders.

Figure 41 plots the relative performance to purchase price for various processor models and minimum to maximum memory size. Grosch's law is also plotted. It appears that the System/370 series follows a power law with g ~ 1.6.

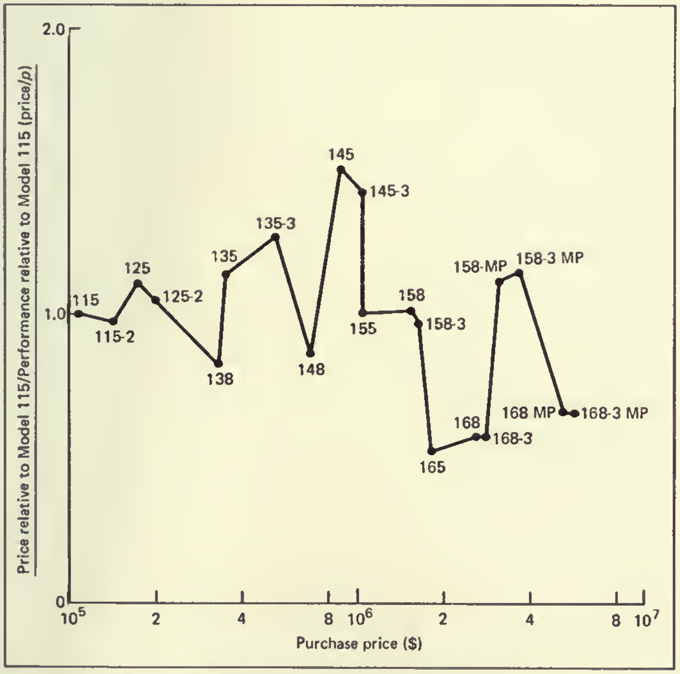


Fig. 40. Graph of IBM System/370 cost/processing power ratio versus purchase price.

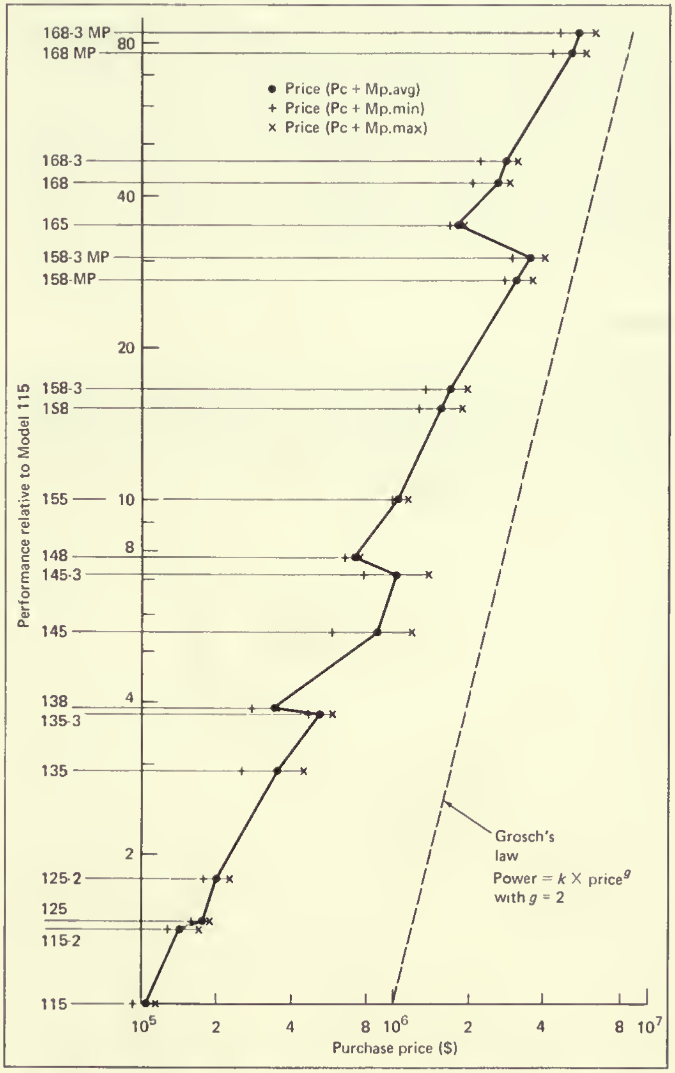


Fig. 41. Graph of IBM System/370 relative processing power versus price.

Conclusions

The IBM System/360 and System/370, by achieving a production record, have fulfilled this principal design objective. The technical goals, however, are of interest to us here. The most interesting aspect of the design is achieving a performance range and a primary memory size range each in excess of 100:1 for both series. Thus a user is given a very large set of configuration alternatives.

There is a vast array of secondary memory and terminal devices to couple with almost any other system. The System/360 is the first computer to make extensive use of microprogramming. Microprogramming is used for the definition of the System/360 instruction-set processor, but more important, microprograms define previous IBM computers so that a user can operate satisfactorily during the interim period when older programs are being updated to use the System/360. Microprogramming also plays a major role in the System/370. There are provisions for multicomputer structures. Within a single computer structure there is adequate means of peripheral switching so that reliable and high-performance structures can be assembled.

References

Adams [1962]; Amdahl, Blaauw, and Brooks [1964]; Arden et al. [1966]; Bell and Newell [1971]; Connors, Florkowski, and Patton [1979]; Conti, Gibson, and Pitkowsky [1968]; Data Pro [1978]; Grosch [1953]; Sharpe [1969]; Solomon [1966]; Tucker [1967]; U.S. Steel [1978]; Wilkes [1965].

Selected Bibliography

Architecture and logical structure: Amdahl, Blaauw, and Brooks [1964]; Blaauw [1964]; Blaauw and Brooks [1964]. General implementations: Amdahl [1964]; Carter et al. [1964]; Padegs [1964]; Stevens [1964]. Microprogramming: Greene, Dean, and Updike [1964]; Tucker [1967]; Weber [1967]. Formal description of Pc: Falkoff, Iverson, and Sussenguth [1964]. Performance and reviews: Hillegass [1966]; Solomon [1966]. Model 40 modifications for multiprogramming: Lindquist, Seeber, and Comeau [1966]. Model 67: Arden et al. [1966]; Fikes, Lauer, and Vareha [1968]; Gibson [1966]; Lauer [1967]. Model 85: Conti [1964]; Liptay [1968]; Padegs [1968]. Model 91 architecture and technology: Anderson, Sparacio, and Tomasulo [1967]; Anderson et al. [1967]; Boland et al. [1967]; Flynn and Low [1967]; Langdon and Van Derveer [1967]; Lloyd [1967]; Sechler, Strube, and Turnbull [1967]. Servicability: Carter et al. [1964]. IBM reference manuals: System/360 and System/370 functional characteristics manuals for each model; System/360 and System/370 configurator (diagram) for each model; System/360 Principles of Operation; System/360 System Summary; processor facts folders for the 4331 and 4341; *Guide to the System/370 Model 135; Guide to the System/370 Model 145*, 6th ed.; *Guide to the System/370 Model 155,* 1st ed.; product announcements for the System/370 Models 135 and 145, System/370 Model 138, System/370 Model 148, and System/370 Model 168; *4341 Processor Functional Characteristics and Processor Complex Configurator.*