5/12/78

CHAPTER 2: TECHNOLOGY, PACKAGING AND MANUFACTURING

It is customary when reviewing the history of various industries to ascribe the events therein to either market pull or technology push. The history of the auto industry contains many good examples of market pull, such as the trends toward large cars, small cars, tail fins, and hood ornaments. The history of the computer industry, on the other hand, is almost solely one of technology push. Whereas the rest of this book gives examples of the effects of technology push, this chapter explores the individual elements that have made up the technology push.

Semiconductors have been the dominant factor in technology push within the computer industry, but magnetic recording density on disks and tapes has evolved rapidly too, and must be understood as a component of cost and a limit of system performance. Communications links have been left out of the discussion because during the past twenty years they have not evolved as rapidly as other technologies, nor have they been a major influencing factor on computer system design. The packaging (cabinets and boxes), interconnection, and power aspects are discussed, principally because these do not represent pushes but rather are large, high inertia objects that have to be leaned ~~pushed~~ against in the hope that they will eventually yield.

The semiconductor portion of the discussion begins by presenting a family tree of the possible technologies arranged according to the function they carry out and showing how these have evolved over the last two or three generations to affect computer engineering. The cost, density, performance, and reliability parameters are briefly reviewed, and then the application of semiconductors, using various logical design methods, is discussed with particular emphasis on how the semiconductor technology has pushed the design methods.

The discussion of ICs is from the user viewpoint because, until recently, DEC has always bought its ICs from the semiconductor manufacturers, and its engineers (users of ICs) have viewed the IC as a black box with a carefully defined set of electrical and functional parameters. Most design engineers will probably continue to hold that view (and be encouraged to do so), even though some ICs will be supplied by an in-house design and manufacturing facility. The advantages and disadvantages of intra-IC design will be discussed later in the chapter.

The discussion of the use of semiconductors in logic applications is followed by a section on memories for primary, secondary, and tertiary storage. The memory section is followed by a section containing some general observations about technology evolution: how technology is measured, why it evolves (or doesn't), cases of it being overthrown, and a general model for how its use in computers operates and is managed.

Packaging and interconnection are the physical structure by which computers are actually formed. The discussion in the packaging section builds on two of the models from Chapter 1: view 1 (structural levels) and view 3 (packaging levels of integration). The chapter concludes with a section that briefly describes the manufacturing process.

Semiconductor Logic and Memory

A single transistor circuit performing a primitive logic function within an integrated circuit (IC) is among the smallest, most complex of man-made objects. Alone, such a circuit is intrinsically trivial, but the fabrication process ~~for a set of structures~~ to form a complete integrated circuit is intricate and complex. To the digital IC users there are several relevant parameters:

1. The function an individual circuit performs within the IC, the aggregate function of the IC, and the functions that a complete IC family (such as the 7400-series) performs.

2. The number of primitive digital switching circuit functions per IC. This density is a measure of the capability of the IC process and the ingenuity of the designers.

3. Cost.

4. The speed of each circuit and the speed of the aggregate IC and set of ICs within a family. The semiconductor device family (Transistor-Transistor Logic = TTL, Schottky TTL = TTL/S, Emitter Coupled Logic = ECL, Metal Oxide Semiconductor = MOS) usually determines this performance.

5. The number of interconnections (pins) to communicate outside the IC.

6. The reliability. This parameter is a function of the ~~circuit~~ device technology,

the density, the number of pins, the operating temperature, the use (or misuse), and the maturity (experience) of the manufacturing process.

7. Power consumption and speed-power product ~~power/speed index~~. ~~The power/speed index is the power per switching element divided by the switching speed. Another~~ A frequently used metric is the "speed·power product", where the delay through a typical gate is multiplied by the power consumption of the gate. For a particular technology, the speed power product tends to be constant because ~~technologies that offer~~ short gate delays usually feature high power consumption. A technical advance that lowers the speed power product is considered noteworthy.

Figure ICtree shows a family tree (taxonomy) of the most common digital IC's. The tree is arranged such that the least complex functions are in the upper portion of the figure and the most complex are at the bottom of the figure. In addition, the tree is arranged to order the circuits by generation, starting with the second generation at ~~along~~ the left hand edge and progressing to the fifth generation. ~~along the right hand edge.~~ The circuits are clustered roughly by the regularity of the function being implemented, and whether there is memory associated with the function. The concept of circuit regularity is important in large scale integrated circuits because it is desirable to implement regular structures to minimize area-consuming interconnections and thus ~~to~~ simplify layout, simplify understanding, ~~and~~ aid testing, and reduce power.

As indicated in Figure ICtree, the branching of the IC family tree began in earnest at the beginning of the third generation. At that time, integrated circuit technology advances permitted collections of basic logic primitives (AND, NAND, etc.) and sequential circuit components (flip flops, registers, etc.) to occupy a single IC rather than an entire module. This had the benefit of providing a drastic reduction in size between the second and third generation computer designs, as shown most vividly by comparing the PDP-9 and PDP-15 (page 00), but had the drawback that modules now contained a wide variety of functions and were thus specialized.

As the densities began to improve to a hundred gates, the construction of complete arithmetic units on a single chip became possible. The earliest and most famous function, the 74181 arithmetic logic unit (ALU) (Fig. ALU, and Fig. ALUFCN) provided up to 32 functions of two 4-bit variables. By the fourth generation, it became possible to construct very large combinational circuits, such as a complete 16 x 16-bit multiplication circuit (the TRW multiplier) requiring about 5000 gates, on a single chip.

Progress during the fourth and fifth generations has not been without its problems, however. Without well-defined functions such as addition and multiplication, semiconductor suppliers cannot provide high density products in high volume because there are few large scale, general purpose universal functions. The alternative for the users is to interconnect simple logic circuits (AND gates, flip flops), but this does not permit efficient use of the technology and the cost per function remains high (about that of the third generation) because the printed circuit board and IC packaging costs (pins) limit the attendant cost reduction.

To address these problems, two methods of effectively customizing LSI logic are included in Fig ICtree and discussed in greater detail later in the chapter. These are the PLA and the gate array. The PLA (Programmable Logic Array) is an array of AND-OR gates that can be interconnected to form the sum of products terms in a combinat~~ori~~ional design. Gate arrays are simply a large number of gates placed on the chip in fixed locations where they can be interconnected during the final metalization stages of semiconductor manufacture. Gate arrays have been used extensively for several years by IBM, and Amdahl. ~~and others for their biggest machines, but may soon appear in smaller computers.~~

There is a special branch of the tree shown in Figure ICtree for the purely memory functions. Memory is used in the processor as conventional memory, but can also be used as an alternative to conventional logic for performing combinational logic~~al~~ functions. For example, the inputs to a combinat~~or~~ionial function can be used as an address, and the output obtained by reading the contents of that address. Memory can also be used to implement sequential logic functions. For example, the memory can be used to hold state information for a microprogram. Since memories have so many uses, this branch is discussed separately in the memory section.

The remainder of the interesting ~~logical~~ functions include combinations both ~~of~~ logic and memory. There are various special functions such as ~~LPC (~~Linear Predictive Coding~~)~~ ~~encoding~~ algorithms for use in real time applications and data encryption algorithms for use in communication systems. One of the most useful communications functions, and the first one to use LSI, was the UART

(Universal Asynchronous Receiver Transmitter).

There is a special branch for bit-slice components that are concatenated ~~can be combined together~~ to form datapaths of arbitrary width. These are being used to construct most of today's high-speed digital systems, mid-range computers, and computer peripheral~~s~~ controllers. Although there have been several bit-slice families, the AMD 2900 series, whose register-transfer diagram~~s~~ is ~~are~~ shown in Figure AMD, has become the most widely used. Note that all the primitives of this series were present in the RTM family (Chapter 19), including the microprogrammed control unit referred to as the K(PCS) (Programmed Control Sequencer).

The final branch of the tree is the most complex, and is used to mark the fourth (microprocessor-on-a-chip) generation of technology and the beginning of the fifth (computer-on-a-chip) generation. The fourth generation is marked by a complete processor being packaged on a single silicon die, and the fifth generation, by this measure, has already begun since a complete computer (processor with memory) now occupies a single die. The evolution ~~in~~ along this branch results in either longer ~~complexity during each generation simply permits larger~~ word length processors or richer computers. ~~to be placed on one chip.~~ At the beginning of the fourth generation, a 4-bit processor was the benchmark, whereas toward the end of the fourth generation, either a complete 16-bit processor, ~~such as the PDP-11~~ or an 8-bit processor with RAM and ROM, could be placed on a chip.

Gates per Chip

The function performed by a chip is clearly dependent on the number of gates that can be placed on a chip. Thus, density in gates/chip is the single most important parameter determining chip functionality. From this measure, one can predict the functions likely to be implemented by just following the tree. It should be noted that the whole tree is relatively alive and has dense areas of new branches everywhere except at the top, where unconnected gate and register structures have been relatively static. In the growing areas, as density increases sufficiently, a new branch grows. For example, the processor-on-a-chip started out as a 4-bit processor ~~(or rather as 2 chips for a single processor)~~ and then progressed to have 8-bit and 16-bit processors on a single chip. Similar effects can be observed with the arithmetic logic unit and with memories.

The number of gate circuits per chip not only determines chip functionality, it also is the measure of density as seen by a user (see Fig. Semiden). This metric is actually the product of the circuit area and the number of circuits per unit area. Progress in lithography has le~~a~~d to reduced conductor linewidths ~~and~~ with a corresponding reduction ~~of~~ in circuit size ~~to yield~~ and hence higher speeds and higher densities. Linewidths have decreased from 10 microns in early LSI chips to 6 microns in the LSI11 chips, and more recently to ~~3 or~~ 4 microns in Intel's 8086. Linewidths of less than a micron have been achieved at the research level, but require electron beam techniques instead of present photographic methods. The processing techniques to create the semiconductor materials have also been improved to have better manufacturing yields (and lower costs). Circuit and device innovation (such as reducing the number of transistors per memory cell) have also contributed to density and yield increases.

The model given in Fig. Semidens is exponential and correlates with previous observations that the number of bits per chip for a MOS technology memory doubles every two years according to the relationship:

$$\text{number of bits per chip} = 2^{t-1962}$$

There are separate curves, each following this relationship, for ROMs in prototype quantities, ROMs in production quantities, RAMs in prototype quantities, and RAMs in production quantities. Thus, depending upon the product and the maturity of its production process, one must scale the state-of-the-art line appropriately by one to three years according to the following rules:

bipolar read-write memories lag by two to three years

bipolar read-only memories lag by about one year

MOS read-only memories lead by one year

This gives the availability of various sizes of semiconductor memories as shown in Fig. LMtimeline.

The significance of these values is that they determine (technology push) ~~where~~ when certain architectures and implementations can occur. The chapter critiquing the PDP-11 uses this model to show how semiconductors accomplish this push.

Cost

The most important characteristic of ICs after density is cost. The cost of

No. of elements per function (transistors for MOS, transistors + resistors for bipolar)

function	MOS			BIPOLAR		
	N	P	C MOS	ECL	TTL	I^2L
INVERTER	2	2	2	7	3	1
2 INPUT GATE	3	3	3 or 4	8	3	1
8 INPUTS GATE	9	9	9 or 16	14	3	2
R/S LATCH	6	6	6 or 8	12	6	2
MEMORY CELL (DYNAMIC)	2	2	2	-	-	2
MEMORY CELL (STATIC)	6	6	6	4-6	4-6	4
D F.F. Flip Flop	20	20	20 or 28	26	20	9
JK " "	20	20	20 or 36	-	26	11

Table Gatecomp The number of elements to implement various functions in different technologies

~~(from R.E.S. memo 3/22/78)~~

ICs is probably the hardest of all the parameters to identify and predict because it is set by a complex marketplace. For circuits that have been in production for some time and for memory arrays, the price is essentially that of a commodity like eggs or bacon, and users generally consider these ICs as very similar to commodities, with the attendant benefits (cost) and problems (having a sufficient source of supply). In low volumes, IC prices are proportional to the die cost (which is proportional to the die area), but at higher volumes, assembly, testing, packaging, and distribution become the dominant cost factors. Furthermore, for those low volume circuits that have not yet reached commodity status, the prices also depend on the strategy of the supplier, ~~-~~ e.g., whether he is willing to encourage competition.

Two curves are presented to reflect the price of various components (transistors) implemented in integrated circuits. Figure Compprice shows the price per gate for MOS and TTL circuits as a function of time and scale of integration (SSI, MSI, LSI). Table GateComp gives some idea of how circuit density (in elements) relates to actual function.

Insert α_{10}

The cost history of ICs is reflected very dramatically in the cost history of a special class of ICs, semiconductor memory. The semiconductor memory cost curves, given in Figure Memprice, are also interesting because of the important role of memory in past and future computer structures. As shown in the figure, the 1978 cost per bit was roughly .08 and .07 cents per bit for the 4 Kbit and 16 Kbit IC chips respectively, giving IC costs of $3.30 and $11.50 respectively. Two factors are at work to create the cost figures for an IC; these are density in bits per IC and cost per bit. The two factors have not had equal influence in reducing costs because while the chip density has improved by a factor of two each (Fig. Semiden) year according to Moore [Noyce, 1977], the cost per bit (at the IC level) has declined by only a factor of two every two years. The line drawn in Noyce's [1977] Fig. Memprice has the equation:

$$\text{cost/bit (in cents)} = .3 \times .72^{t-1974}$$

It is also interesting that the cost compares favorably with the price decline observed in core memory over the period since 1960-1970 for the 18-bit computers, (page 00) and for the cost declines in both the PDP-11 and the PDP-8 (pages 00 and 00).

Performance

The performance for each semiconductor technology evolves at different rates depending on the cumulative learning associated with design and manufacturing process together with the marketplace pressure to have higher performance for the particular technology. One may hypothesize that each technology can be looked at as being relatively appealing or relevant to the particular design(er) styles associated with the computer market levels (view 4, page 00). One would expect the evolution to continue along the lines shown in the following table for the period around 1980.

Table SemChar: Characteristics of Dominant (1978) Semiconductor Technologies

Type	**Evolution**	**Use**

TTL (Transistor- Transistor Logic)	TTL TTL/Schottky TTL/LS	logic, bus interfacing higher speed than plain TTL same speed as TTL, but low power
ECL (Emitter Coupled Logic	MECL II, III MECL 10K, 100K	very high performance easier to work with evolving to larger gate arrays
MOS (Metal-Oxide- Semiconductor	p-channel n-channel	low cost greater densities, cost evolving to performance (memory) evolving to shorter channels (HMOS, DMOS, VMOS)
CMOS (Complementary MOS)		low power, higher speed, better noise immunity

DEC's use of the various IC technologies shown in Table SemChar is probably typical of most of the computer industry: TTL for ~~mid- and high-sized~~ minicomputers; ECL for the larger scale machines ~~(DECsystem 10)~~; MOS for memories, microprocessors, and specialized high density circuits; and CMOS for special microcomputers, especially those intended for battery operation.

Some of the lesser used technologies such as I^2L (Integrated-Injection Logic) and SOS (Silicon on Saphire) have been omitted. I^2L features high density and very low power consumption, but is slow. SOS MOS enhances CMOS speed by removing stray capacitance, making it comparable with TTL/LS speed, while retaining MOS ~~gate~~ density ~~complexity~~ capabilities. Both I^2L and SOS have been touted as replacements for various technologies shown in the table. But, if an entrenched technology has evolved for some time and continues to evolve, it is difficult for alternative technologies to displace it because of the cummulative investment in process technology and understanding. Semiconductors appear to be characteristic of other technologies in that usually only a single technology is applied to ~~used for~~ a given problem.

The early technologies, RTL (Resistor-Transistor Logic), TRL (Transistor-Resistor Logic), and DTL (Diode-Transistor Logic) have also been omitted. These technologies are important historically, because they were used in the first ICs. However, many manufacturers, including DEC, did not use them in computers (RTL was used in DEC K-Series modules) because they did not represent a sufficient advance over the discrete transistor circuits already being used. In addition, early circuits were packaged in flat packages and metal cans rather then in the dual in-line package used today, and automated manufacture using the components was thus not economically feasible.

Table Gatedelay gives the speed-power product and gate delay which are the two most useful measures of performance for the various technologies as they have evolved with time. The speed-power product metric ~~for a technology at a given time indicates~~ suggests that the user may tradeoff performance against power. There are limits to this tradeoff. Only about one watt can be dissipated by a traditional off-the-shelf IC package, (and tradition in IC package design has been strong). The table was formulated by Jerry Luecke of Texas Instruments (TI) at a time when I^2L technology had just been introduced (Oct. 1975) by TI:

Table Gatedelay: Gate Delay of Various Semiconductor Technologies [Luecke, 1976]

Year	Type of Logic	tp (nsec)	P D W	Speed-Power Product (Picojoules)
1963	DTL	-	-	200
1964	RTL			180

1965	TTL	10	10	100
1967	TTL/H-series	5	20	100
1968	TTL	30	1	30
1970	TTL(Schottky)	3	20	60
1972	TTL(low power Schottky)	10	2	20
1967	ECL	2	30	60
1974	ECL	0.7	43	30
1970	PMOS	200	0.1	20
1973	NMOS	100	0.1	10
1973	CMOS	30	1.0	30
1974	SOS	15	0.05	7.5
1978	HMOS	x	x	x
1975	I^2L	35	0.085	3.0
1976	I^2L	20	0.05	1.0

Reliability

Over the past 15 years, the failure rate for standard ICs has been reduced by two orders of magnitude to the neighborhood of .01% per 1000 hours. This corresponds to 10^7 hours (about a millenium) mean time to failure (MTTF) per component. Figure Reliab, from a recent survey article by Hodges [1977] shows the trend. The lower curves show the higher reliability obtained when more extensive testing and screening are employed. The improved MTTF of between 10^8 and 10^9 are obtained at a cost increase of 4 to 100 times per component.

The LSI Dilemma

As indicated in the beginning of the discussion of Figure ICtree, a dilemma ~~involving a search of universal circuits~~ has developed in the manufacture of LSI circuits. The economics of the LSI circuit industry make it essential that IC suppliers produce circuits with a high degree of universality. This is because the learning curve of a manufacturing process causes cost to be inversely proportional to volume, and for a design to be sold in high volume, it must be usable in a large number of applications. However, the trend in circuit complexity, which allows semiconductor manufacturers to put more transistors on a constant die area each year, tends to increase specialization of function, lowering the volume and hence raising the price.

The LSI-product designer is therefore continually in search of universal primitives or building blocks. For a certain class of applications, such as controller applications, the microprocessor is a fine primitive and has been so exploited [Noyce, 1977]. For other applications, circuit complexity can embrace even higher functionality at the PMS level. The Intel 827X is an interesting example here: two processors, a 1.25 microsecond byte-processor and a 250 nanosecond bit-processor are combined in one LSI circuit.

Moore [1976] discusses the LSI dilemma in a paper on the role of the microprocessor in the evolution of microelectronic technology. He points out that a similar situation existed when ICs were first introduced. Users were reluctant to relinquish the design prerogative they had when they built circuits from discrete components. It was not until substantial price reductions were made that the impasse was broken. Then the cost advantages were sufficient to force users to adopt circuits that fit the technology.

~~The first high functionality, high universality circuit that comes to mind is the microprocessor on a chip~~. For many applications, including most computer systems, the microprocessor on a chip is not a cost-effective building block, and other solutions to the dilemma are found ~~used~~. For example, microprogramming is a systematic ~~highly general~~ way of generating control signals for data path elements, and table lookup is a highly general technique. Both methods are attractive because they use memory, an inherently low cost LSI circuit. Microprogramming, however, does have limitations. The extra level of interpretation extracts a performance penalty and some potential datapath parallelism is often given up to reduce cost. A more subtle, but practical, limitation is the development cost of microcode. Assuming the writing rate to be 700 microwords per man year for wide-word, unencoded (horizonal) micro machines, a desire to limit the effort to 20-24 manyears would limit the maximum control store size to about 16 Kwords. This maximum will tend to increase in the future, when the use of better microprogramming tools increases the microcode writing rate beyond the 700 microwords per man year figure given here.

At the RT level, the standard microprogramming design method is (conservatively) twice as expensive per instruction as conventional programming. Moreover, because microinstructions are usually not as powerful as conventional instructions, more microinstructions than conventional instructions are ~~will be~~ required to solve a given problem. These two factors, more expense per instruction and more code ~~instructions~~, ~~will~~ cause a microprogram to be 5 to 10 times as expensive as a conventional program to solve the same problem. ~~The payoff is that the internal speeds of a~~ ~~micro-level machine~~ ~~microprogrammed~~ ~~contro~~ ~~controller are at least a factor of 10 faster than a conventional mini.~~ However, microinstructions execute at least ten times faster.

The characteristics of microprocessor and ROM methods of tailoring ~~creating customized results from~~ universal LSI circuits are summarized, along with the characteristics of a number of other methods, in Table FunVar.

Table funvar

Building block	**Technique for varying function**	**Degree of generality**	**Permanence of change**
Computer module	program	v. high	none
microprocessor	program	high	low to medium
bit slice	microprogram	medium	medium
ROM	factory mask change	v. high	irreversible
PROM	field change	v. high	irreversible
EAROM	field change	v. high	low
PLA	factory mask change	medium	irreversible
FPLA	field change	medium	irreversible
gate array	factory mask change	medium	irreversible
RAM	write	v. high	none

The increased basic circuit functionality available at each new generation has not only been an important part of semiconductor design, but has also caused design methods to change with the generations. This book provides an examples, as summarized in the following table.

Table: Design Method versus Generation

Design Method:\Generation:	**First**	**Second**	**Third**	**Fourth**	**Fifth**	**Examples in this book**

Combinational and sequential; use of "standard" modules, IC's.	s	s	s			18-bit;PDP-8
Read only memory and PLA; microprogramming			s	m		PDP-9;PDP-11
Microprogramming with standard RT elements (high perf.) ; minor logical design				s	m	CMU-11
Programming using micros and logic for interfaces		p	p	s	x	LSI-11
PMS-level ~~PMS~~ design using completely specified and pre-designed microcomputer components					s	Cm*
Customized chip design and standard logical design (high performance)			m	m	m	LSI-11

s - the standard method for most digital systems
m - done by manufacturers of ~~basic equipment~~ integrated circuit
x - also used
p - prelude to micros, also done using minis

The design of most relatively high speed digital systems (including ~~low- to~~ stet mid-range minicomputers) is carried out using standard register transfer ICs complete with data path and memory.

For higher performance computers, there is no alternative to using either tightly packed standard ICs or building a unique set of ICs using some form of customization. The high performance IBM and Amdahl machines, for example, use custom ECL circuits or gate arrays to improve packaging. Although Seymour Cray continues to build his high speed computers (the CDC 6600, 7600 and Cray 1) with no custom logic, he does so by using impressively dense modules with high density interconnection and freon cooling.

The current spectrum of IC's and their use is summarized in Table ICspectrum.

Table ICspectrum: A Spectrum of IC application ~~Organization and Use~~ in Various Computers

Application ~~Organization~~ ←	Technology ←	Unique Chips ←	Performance (MIPS) ←	Cost	Examples
Microcomputer	MOS, (VLSI)	1	0.1	Lowest	Intel 8048, MOSTEK 3870
Microprocessor	MOS	1			Intel 8080, Zilog Z80, Motorola 6800
Microprocessor	MOS	2-4			DEC LSI-11, Fairchild F-8
Microprocessor	MOS	> 4			Burroughs B80, National IMP 16
Bit slice (micro-programmed)	TTL	few			DEC 11/34 Floating Point Unit
Gate array	TLL	most			Raytheon RP16, IBM Series 1
MSI	TTL	few			DEC VAX 11/780, 11/70, HP 3000
Gate array	ECL	all			IBM 370/168, Amdahl 470/v6
SSI	ECL (SSI)	std.	80	highest	CDC 7600, CRAY 1

The Changing Nature of System Design

With the advent of the processor on a chip, digital system design has been, or soon will be, converted completely to computer system design (PMS-level design). Problems such as controlling a CRT, controlling a lathe, building a billing machine, or implementing a word processing system become computer system design problems similar to those attacked over the first three generations. The hardware part of the design, the interface to the particular equipment, is straightforward. The major part of the design is the programming. Since the late 40's three complete generations have learned about computer design, especially programming. The first generation discovered and wrote about it. Then it was rediscovered and applied to minicomputer systems. This time, it is being learned by everyone who must use ~~and program the~~ a microcomputer. Each time, for each individual or organization, the story is about the same: people start off by programming (using binary, octal or hexadecimal codes) small tasks, using no structure or method of synchronizing the various multiple processes; the interrupt mechanism is learned, and the symbolic assembler is employed; and finally some more structured system--possibly an operating system is employed. Occasionally users move to higher level languages or macro assemblers. *

In view of this cyclical history, it seems likely that current digital systems design practice, which consists of building simple hardware interfaces to relatively poorly defined busses together with programming the application, will be relatively short lived. The design method of the future will ~~be at~~ be with ~~the~~ PMS-level components, although at the moment it is still too difficult to be done reliably and cheaply by large numbers of engineers. The components from which the microcomputer systems will be formed will be significantly more advanced; they will use much better packaging, clearly defined busses, more general interfaces, and ~~base level~~ rudimentary operating systems. ~~The opera~~ ~~that are embedded in hardware~~ The latter, by being placed in read only memory, will ~~to~~ give a ~~the~~ feeling of permanency so that users are less likely to embark on the expensive, unreliable rediscovery path. Standard components will be built which can be interfaced to a wide range of external systems using parameters that are specified by a field programming method instead of using logical design and building with interconnection on modules. In this way, the complexity of individual ICs can be increased and having a standard method for interconnection, higher volume and lower costs will result.

Design Costs Versus Unit Costs

Before discussing the alternatives associated with IC design, it is important to characterize the various costs. Figure Design.cost shows, at a crude level, what one might expect the relative design costs to be for various inter- and intra-IC design methods. Even the design cost is highly variable depending on the project size, its goals, the manufacturing volumes expected, and more importantly, on the computer aided design programs.

The lowest design cost is achieved by ~~staying~~ completely avoiding ~~away from~~ modifying the ICs(except for programming read only memories). There are two elements to the cost of read only memories,: programming cost and parts cost. The programming cost has already been discussed on page 00, so this discussion will be limited to parts cost. There are two kinds of read only memories, the field programmable read only memory (PROM) and the factory programmable (masked) read only memory (ROM). PROM chips have a higher initial cost than ROMs but provide some inventory advantages in a manufacturing environment because a common stock of unprogrammed parts can be divided up into various programmed parts rather than stocking a full supply of each required part. In many high volume applications, however, the cost of the extra testing steps involved in the common stock approach, plus the extra piece part costs for PROMs, cause masked ROMs to be preferable.

The design costs discussed in the preceeding paragraphs are summarized in Figure Semiuse, which shows the costs for conventional programming, costs for microprogramming and the design costs for ROM/PLA designs using combinat~~or~~ionial techniques rather than programming techniques. The most costly approach of all shown in Figure Semiuse is design using standard circuits and associated design techniques.

Design of ICs (Intra-IC Design)

Despite the prospects of higher design cost with custom ICs than standard ICs, and, in some cases, higher manufacturing cost, there are numerous reasons why a designer is often forced to design ICs. These are summarized in Tablereasons.

Tablereasons: Reasons to do Custom IC Design

1. Performance bells and whistles can be added that give the product a selling advantage over a less-LSI competitor. The manufacturing "mix problem" can be simplified by building and stocking a single design, and then shipping features that are inhibited unless the customer ordered them.

2. Monthly charges for maintenance might be reduced if diagnosability or reliability is improved. Since the customer pays these charges over and over, a small reliability improvement at the system level might justify an entire custom LSI program for large-volume applications.

3. Diagnostic labor can be a high percentage of printed circuit board manufacturing cost. Diagnosis to the chip level can be speeded up by

features within the chip, and by a lower chip count.

4. Data busses can be absorbed entirely within a chip to avoid bus interface costs. Even shortening a data bus from multi-board to single-board length may reduce cost and/or improve performance by reducing stored energy and its attendant drive/speed penalties.

5. Innovations concealed within a chip are difficult for the competition to study and duplicate.

6. Performance barriers may be breakable only through custom LSI. In CPU design especially, and perhaps for certain RAM interfacing jobs, a custom LSI approach is the only practical way to get around conflicting issues of size, power, capacitance, etc.

7. In some engineering environments there are extremely small amounts of space or very little power.

There are some drawbacks to custom LSI design. These are listed in Table CusLsiDra.

Table CusLsiDra: Reasons Not to Do Custom LSI Design

1. For designs in the 100-500 equivalent gate complexity range, it may take up to a year to do the design with primative design tools.

2. For designs in the 100-500 equivalent gate complexity range, it may take up to $100,000 to do the design.

3. Unless substantial product volumes are obtained, the chip cost will be high relative to off-the-shelf chips.

4. A decision will have to be made whether to have the design done by an outside vendor or within the company. This can be a very complicated and expensive decision.

5. The logic design and logic partitioning for LSI design is different from that of conventional logic design and designers used to dealing with conventional design will have to assimilate new knowledge to design LSI themselves or even to talk with LSI designers.

The use of custom ICs to reduce the number of discrete components or to reduce the total number of ICs in a machine improves the reliability because the reliability of a system is mostly a function of the number of explicit physical connections, including the bond to the semiconductor die. Thus the likely reliability of two equal functionality designs can be compared by counting discrete circuit pins, IC pins, module pins, and connector pins.

Assuming that one decides to do IC design, the various design methods that might be used for various objects and densities are given in the following table.

The most straightforward and extensively used intra-IC design method is to modify an existing design. If this approach cannot be used, the next most straightforward method is to use arrays of gates and interconnect them to form the desired function. Design with gate arrays occurs in a completely defined environment because there is only one circuit from which the gate is formed and the gate can be completely parameterized and defined. The manufacture of gate arrays is fairly simple because the fabrication of all but the last few semiconductor processing steps is identical for all designs. The customization, accomplished by interconnection of the gates by metal, is carried out last. Interconnection is a well understood aspect of logic design and is used to form the more complex macro structures (various flip flop types, adders) and then to form the higher levels of design by using arrays of gate arrays. There is a disadantage to gate arrays, which is that gate array design methods do not permit the high density possible with the more custom methods because device placement is fixed.

It should be noted that gate array design is not a new idea brought about by the need for a simple method of customizing LSI. Rather, it was one of the design philosophies advocated in the first few generations. The concept then was to have a single module containing a set of gates, and all subsequent logical design would be done in terms of that module. For example, flip flops would be constructed by interconnecting the gates. A design predicated on a single module type simplifies the spare stocking and servicing aspects immensely, and it is possible to troubleshoot a problem by simply replacing circuits according to a pattern. Designers did not find it important enough to get these advantages at that time, however, so the gate array concept was set aside until it was rediscovered by LSI designers.

A representative gate array is a Raytheon RA-116. It has 300 Schottky TTL gates, of two cluster configurations, each repeated 12 times within the 160 mil x 160 mil chip:

Type 1:	3 external driver gates	(4-input NAND)
	5 internal driver gates	(3-input NAND)
	5 internal expansion gates	(3-input NAND)
Type 2:	2 external driver gates	(4-input NAND)
	5 internal driver gates	(3-input NAND)
	5 internal expansion gates	(3-input NAND)

Within each cluster, the expansion gates may be combined with the driver gates to form 7- or 8-input NAND gates and AND-OR-INVERT circuits with up to six product terms.

The gates have a typical propagation delay of 5-6 nanoseconds and dissipate

5.5-6 milliwatts per driver and 1 milliwatt per OR expander. Two metal layers are used for interconnect, and the resulting circuitry can be connected to the outside world by means of 56 external pins, including power and ground.

Since the designer can arbitrarily interconnect, he constructs flip flops, adders, decoders, etc. Because the use of IC gate arrays is recent, data on package count reduction is scarce, but one informal study for the Raytheon RP-16 aerospace computer measured a 9 to 1 replacement ratio and an overall factor-of-three improvement over a system constructed with standard components [Parke, 1978].

A 920 gate MOS array of 3-input NORs has been reported by [Nakano, et al 1978]. Its 3 nanosecond gate delay illustrates the performance potential as MOS continues to be scaled down. For higher speed applications, an ECL gate array can be used. These devices, with sub-nanosecond speeds, exploit the inherent properties of current mode logic to obtain a particularly flexible element [Gaskill et al., 1976]. Examples are the TI and Fairchild ECL 168.

Standard cell design is identical to the logical design of the first few generations (e.g., PDP-1) since there is a well-defined set of components (AND gates, flip flops) in which the design is carried out. The advantage of the standard cell design methods is that special functions can be mixed on the chip in greater variety. There may also be a density advantage over gate arrays. However, in some schemes each cell occupies a different space and has a fixed shape. Careful planning of the cell arrangements is necessary to minimize loss of space. Hence, the improvement in packing density is not as substantial as direct comparisons between standard cell technology and gate array technology might at first indicate. In addition, if there are a large number of circuit types, their interconnection rules may not be characterized well enough to achieve a quick, cheap design that works the first time.

Custom design is in some ways a variant of the standard cell, since designers typically have a set of favorite circuits which they interconnect to create designs for specified applications. With custom design the designer can (theoretically) specify a circuit for each use within a particular logical design. For example, upon observing that a particular gate or flip flop only drives a certain load, the designer can modify that gate or flip-flop to provide only the appropriate driving capability. Therefore, with custom design, the whole IC can theoretically be an optimum, since each part is no larger than it need be. The advantages are clearly size, cost, and speed. The design costs are high because each part can, in principle, be customized. The quality of the circuit design is totally dependent on the designer who must analyze each circuit geometry in terms of his expectation of performance, operating margins, etc. To the extent that this analysis is carried out, the circuit is clearly optimal.

Also on the graph is a hypothetical line for universal logic arrays. For at least 15 years, academicians have studied the possibility of designing a single array of logical design elements, or a collection of such arrays, that could be interconnected on a custom basis to carry out a given function. The gate array can be looked at as the simplest example of this type of design. While many are skeptical that such a device exists, a line representing it is placed on the graph as a target for those who search for the one truly universal logic array.

Both Read Only Memory (ROM) and Programmable Read Only Memory (PROM) are commonly used, but trivial, forms of the truly universal arrays, because they can be used in a table look up fashion to create several functions of a number of input variables. For example, a 1,024 word ROM arranged in a 256 x 4-bit fashion can generate 4 independent functions of 8 variables. This is a distinct alternative for using a conventional gate structure to carry out combinational functions. A disadvantage of this method is that the required ROM size doubles for each additional input variable.

The PLA is a combinational circuit which remedies the disadvantages of the ROM implementation of combinatorial functions by allowing the use of product terms rather than completely decoding the input variables. Fig. PLA shows a typical circuit, which consists of separate AND and OR arrays. Inputs are connected to the AND array, and outputs are drawn from the OR array. Each row in the PLA can implement an AND function of selected inputs or their complements, thus forming a boolean product term, and the OR array can combine the product terms to implement any boolean function.

A simple application is operation-code decoding. For the PDP-11, the 16-bit Instruction Register could be directly connected to a PLA and the output thereof used to specify the address of the microprogram that executed that instruction. Three different types of operation-code decoding are customarily applied to PDP-11 instructions: source mode decoding, destination mode decoding, and instruction decoding. With a PLA implementation, a PLA could be used for each of these decoding operations, and only three chips would be required. A ROM implementation, on the other hand, would require 128x8 for address mode decoding and 64Kx8 for instruction decoding. Using 2Kx8 ROM's, 33 chips would be required. For this reason modern minicomputers, such as the PDP-11/34, use PLAs rather than ROMs or combinatorial logic for instruction decoding. The technique is also extended downward into microcomputers such as the LSI-11, where PLAs are used to conserve the die area used by their control units.

The PLA becomes an even more useful building block when it is made field programmable -- the FPLA. The programmable connectors shown in Figure PLA are fusible nichrome links that are burned out when the unit is programmed.

When a register is added to the outputs of the PLA and incorporated in the same integrated circuit, a simple sequential machine is obtained in one package. Since register circuit packages are pin intensive, adding registers to PLAs (or ROMs) permits about a factor-of-two package count reduction in typical applications.

The first PLAs had propagation times of the order of 150 nanosec and were thus suitable building blocks for slow, low-cost computers. Propagation times of 45 nanoseconds are quite common today, and the PLA is more widely used. An attractive application with these higher speed components is the replacement of the SSI and MSI packages used to implement the control logic for Unibus arbitration. Alternatively, such a structure may be more cost-effective when implemented in gate arrays.

A more complex application than instruction decoding has been documented in [Logue et al., 1975]. An IBM 7441 Buffered Terminal Control Unit was implemented using PLAs and compared with an SSI/MSI version. The PLA design included two sets of registers fed by the OR array (PLA outputs): one set fed back to the AND array (PLA inputs), and the other set held the PLA outputs. A factor-of-two reduction in printed circuit board count was obtained with the PLA version. The seven PLA's used in the design replaced 85% of the circuits in the SSI/MSI version. Of these circuits 48% were combinational logic, and 52% were sequential logic.

MEMORY TECHNOLOGY (AND SEMICONDUCTORS USED AS MEMORIES)

The previous section discussed the use of memory for microprogramming and table look up in logical design, but that is not the principal use of memory in the computer industry. Rather, the more typical use of memory components is to form a hierarchy of storage levels which hold information on a short term basis while a program runs and on a longer term basis as permanent files. This section will present the various parameters and discussion relevant to this use. While the principal focus will be on core and semiconductor memories, slower speed electromechanical memories (drums, disks, and tapes) will be considered superficially, as their performance and price improvements have pushed the computer evolution. Since the typical uses for memory usually require read and write capabilities, write once or read only memory such as video disks will be excluded from the discussion.

Because memory is the simplest of components, it should be possible to discuss memory using a minimal number of measurement parameters. Many of the parameters vary with time; this is particularly true of semiconductor memory price, which has declined at a rate of 28% per year compound, (which amounts to about 50% in two years). The price is expressed only as price/bit, but it is important to know the price (or size) of the total memory system for which that price applies because of the economy of scale. In order to get the lowest price per bit, a user may be forced to a large system.

Performance for cyclical memories, both the electromechanical types such as disks and the electronic types such as bubbles, is expressed in two parameters: the time to access the start of a block of memory, and the number of bits that can be accessed per second after the transfer begins. The operational environmental parameters of power consumption, temperature, space and weight effect the utility of memories in various applications, and the reliability measures are needed in order to see how much redundancy must be placed in the memory to operate at a given level of availability and data integrity.

In summary, the relevant parameters for a given memory are:

1. state of development of the technology at the time the measurements are taken relative to the likely life span of the technology

2. price per bit

3. total memory size or total memory price

4. performance

   a. the access time to the first word of the block

   b. the time to transfer each word (data-rate) in the block

5. operational power, temperature, space, weight

6. volatility

7. reliability and repairability

A good example of a technology that is young relative to its expected total lifetime is semiconductor memory. Figure Memprice gives past prices and expected future prices of semiconductor memory. These memories have declined in price every two years by a factor of two, and that rate of decline is expected to continue well into the 80's because of continued increases in semiconductor densities. Figure Memsizeperf, a graph by Dean Toome, Vice President of Engineering for Texas Instruments, shows memory size and performance improvements with time and includes Charge Coupled Devices (CCDs) and magnetic bubbles. These latter devices show slower performance figures than the other semiconductor memories because they are cyclically accessed in a fashion similar to disks.

Core and Semiconductor Memory Technology for Primary Memory

The core memory was developed early in the first generation for Whirlwind (1953) and remained the dominant primary memory component for computers until it began to be superseded by semiconductor technology. The advent of the 1 Kbit memory chip in 1972 started the demise, and the crossover point occurred for most memory designs with the availabiilty of the 4 Kbit semiconductor chip in 1974.

The description and operation of the core memory is given briefly in Chapter 0 on the PDP-8 (page 0), and several of the significant circuit and production innovations are given in Chapter 0 on the 18-bit computers (page 0).

Over the period since the early 60s, the price of core memory declined roughly at a rate of 19% per year. This decline can be seen in the 12-bit (page 0), 18-bit (page 0) and IBM 360/370 memory prices (since 1964). The price of DECsystem 10 memory has declined at 30% per year (page 0), although it is unclear why. A possible reason is that the modular memory structure had a high overhead, and that with subsequent implementations the memory module size was increased, thereby giving an effective decrease in overhead electronics cost and a greater decrease in the cost per bit.

The cost of various memories was projected by several technology marketing groups in the period 1972-1974. Each study attempted to analyze and determine the core/semiconductor memory crossover point.

Three such studies are plotted in Fig. Core.cost along with Turn's [1974] memory price data and Noyce's [1977] semiconductor memory cost (less overhead electronics) projection. Note most crossover points were projected to be in 1974, whereas one study showed a 1977 crossover. Even though all studies were done at about the same time, the variation in the studies shows the problem of getting consistent data from technology forecasts.

Figure Memprice shows the cost of semiconductor memory decreasing at a rate of 28% per year.

While these graphs of core and semiconductor prices and performance permit an understanding of trends in the principal use areas for these devices, processors, primary memory, cache, and small paging memories, additional information is needed for disk and tape memory in order to complete the memory hierarchy.

Disk Memories

Disk memories are a significant part of most systems costs in the middle-range minicomputer systems and for larger systems, dominate the costs.

Although access time is determined by the rotational delays and the moving head arm speed, the single metric that is most often used is simply memory capacity and the resultant cost/bit. In the subsequent section on memory hierarchies, it will be shown that performance parameters are less important because more higher speed memory can be traded off to gain the same system level performance.

Memory capacity is measured in disk surface areal density (i.e., the number of bits per square inch) and is the product of the number of bits recorded along a track and the number of tracks of the disk. Figure Areal.Digital gives the areal recording densities using digital and analog recording methods. Figure Large.disk.price shows the price of the state-of-the-art line for large, multiple platter, moving head disks. Note that the price decline is a factor of 10 in 9 years, for a price of decline of x% per year.

Figure Memtrends shows the performance plotted atainst the price per bit for the technology in 1975 and 1980.

Magnetic Tape Units

Figure Tapechar shows the performance characteristics that are relevant for magnetic tape units. The data is for several IBM tape drives between 1952 and 1973. It shows that the first tape units started out at 75 inches per second and achieved a speed of 200 inches per second by 1973. While this amounts to only a 5% improvement per year in speed over a 21 year period, this is a rather impressive gain considering the physical mass movement problems involved. It is akin to an improvement in automobile speed of a factor of three.

The bit density (in bits per linear inch) has improved from 100 to 6250 in the same period for a factor of 62.5, or 23% per year. Since both speed and density have improved, the tape data rate has improved by a factor of 167, or 29% per year.

Tape unit prices (see Fig. x) are based on the various design styles. Slow tape units (minitapes) are built for lowest cost. The most cost effective seem to be around 75 inches per second (the initial design), if one considers only the tape. High performance units, though disproportionately expensive, provide the best system cost effectiveness.

Memory Hierarchies

A memory hierarchy, according to Strecker [1978], "is a memory system built of a number of different memory technologies: relatively small amounts of fast, expensive technologies and relatively large amounts of slow, inexpensive technologies. Most programs possess the property of locality[1]: the tendency to access a small, slowly varying subset of the memory locations they can potentially access. By exploiting locality, a properly designed memory hierarchy results in most processor references being satisfied by the faster levels of the hierarchy and most memory locations residing in the inexpensive levels. Thus, in the limit a memory hierarchy approaches the performance of the fastest technology and the per bit cost of the least expensive technology."

The key to achieving maximum performance per dollar from a memory hierarchy is to develop algorithms for moving information back and forth between the various types of storage in a fashion which exploits locality as much as possible. Two examples of hierarchies which depend upon program locality for their effectiveness are the one-level store (demand paging) first seen on the Atlas computer [Kilburn, et al, 1962] and the cache first seen on the IBM

360/85 [Liptay, 1968]. Because both of these are automatically managed (exploiting locality), they are transparent to the programmer. This is in contrast to the case where a programmer uses secondary memory for file storage, because in that case he explicitly references the medium, and its use is therefore no longer transparent.

The following table, in order of memory speed, lists the memories used in current day hierarchies. There is a continuum based on need together with memory technology size, cost, and performance parameters.

Table: Computer System Memory Component and Technology

Part	**Transparency (to machine language programs)**	**Based-on**
Microprogram memory	yes[1]	very fast
Processor-state	no	very small, very fast register set (e.g., 16 words)
Alternative processor-state context	yes	same (so speed up processor context swaps)
Cache memory	yes	fast. used in larger machines for speed
Program mapping and	yes	small associative store
Primary (program) memory	no	relatively fast, and large depending on Pc speed
Paging memory	yes	can be electromechanical, e.g.,drum, fixed head disk, or moving head disk. Can be CCD or

		bubbles.
Local file memory	no	usually moving head disk, relatively slow, low cost
Archival files memory	yes (preferably)	very slow, very cheap to permit information to be kept forever

Microprogram Memories

Nearly every part of the hierarchy can be observed in the computers in this book. Chapter 0, 0, and 0 describe PDP-11 implementations than use microprogramming. These memories are transparent to the user, except in machines such as the PDP-11/60 which provide user microprogramming via a writeable control store. Mudge (chapter 0) describes the writeable control storage user aspects associated with the 11/60 and the user microprogramming. Chapter 0 describes similar possibilities in the LSI-11, although the writeable control store option was not available at the time the article was written.

In retrospect, DEC should have built upon the experience gained from the small read only memory used for the PDP-9 (1967) and exploited the idea earlier. In particular, a ROM implementation might have produced a lower cost PDP11/20 and might have been used to implement lower cost PDP-10s earlier.

In principle, it is possible to have a cache to hold microprograms; hence, there could be another level to the hierarchy. At the moment, this would probably be used only in high cost, high performance machines, because of the overhead cost of the loading mechanism and the cache control. However, like so many other technical advances, it will probably migrate down to lower cost machines.

Processor State Registers

To the machine language program, the number of registers in the processor state is a very non-transparent part of the architecture. This number is solely dictated by the availability of fast access, low cost registers. It is also occasionally the means of classifying architectures (e.g., single accumulator based, general register based and stack based).

In 1964, even though registers were not available in single IC packages, the PDP-6 adopted the general register structure because the cost of registers was only a small part of the system cost (see chapter 00). In the chapter on the DECsystem 10 there is a discussion of whether an architecture should be implemented with general registers in an explicit (non-transparent) fashion, or whether the stack architecture should be used. Altough a stack architecture does not provide registers for the programmer to manage, most implementations do incur the cost of registers for the top few elements of the stack. The general register structure was adopted to give better program control of a small number of local variables and hence gain performance advantages. The change in register use from accumulator-based design to general-register-based design and the associated increase in the number of registers from one to eight or sixteen can be observed between the 12-bit and 18-bit designs and the later DECsystem 10 and PDP-11 designs.

Alternative Processor State Context Registers

As the technology improved, the number of registers increased, and the processor state storage was increased to provide multiple sets of registers to improve process context switching time.

Cache Memory

In the late 60s, the cache memory was introduced for large scale computers. This concept was applied to the KL10 and 11/70 in 1975 when the relatively large (1 Kbit), relatively fast (factor of 5 faster than previously available), memory chip was introduced. The cache is described and discussed extensively in chapters 00, 00 and 00, so the reader may want to peruse page 0 if not familiar with the concept. It derives much power by the fact that it is an automatic mechanism and hence transparent to the user. It is the best example of the use of the principle of memory locality. For example, a well designed cache of 4 Kbytes can hold enough local computational memory so that, independent of program size, 90% of the accesses to memory are via the cache.

Program Mapping and Segmentation

A similar memory circuit is required to manage (map) multiprogrammed systems by providing relocation and protection among various user programs. The requirements are similar to the cache and may be incorporated in the caching structure. The KI 10 used an associative memory for this mapping function, and the VAX 11/780 uses a 64 entry two-way associative memory.

Paging Memory

The Atlas computer [Kilburn, et al, 1962] was designed to have a single, one level, large memory. This structure ultimately evolved so that multiple users could each have a large virtual address and virtual machine. However, the concept of paging mechanism works because there is not equally random access to each page, but rather only local access to various parts of a program by the processor at a given time. Denning pointed out the clustering of pages for a given program at a given time and introduced the notion of the working set [1968]. For most programs the number of pages accessed locally is small compared with the total program size. Initially a magnetic drum was used to implement the paging memory, but as disk technology began to dominate the drum, both fixed head and moving head disks (backed up with larger primary memories) were used as the paging memories. Denning's tutorial article [1970] is an excellent discussion of this section of the memory hierarchy. In the next few years, the relatively faster and cheaper CCD semiconductor memories and bubble memories are clearly the candidates for paging memmories.

Local File Memory and Archival File Memory

For medium sized to large scale systems there is no alternative to disks, with archival files on magnetic tapes. These permit files to be stored cheaply on an indefinite basis. For smaller systems there are usually fewer memory technologies used than in larger systems, because the smaller systems cannot afford the overhead costs (disk drives, tape drives, etc.) associated with the various technologies. At most, two levels of storage would probably exist as separate entities.

Alternatively, one might expect a combination of floppy disk, low cost tape, and magnetic bubbles to be used to reduce the primary memory size and at the same time provide file and archival memory. Currently the floppy disk operates as a single level memory. Here one can see two alternatives for technology tradeoff using the hierarchy: a tape or floppy disk can be used to provide removability, and archivability, whereas bubbles or CCD provide performance. The Strecker paper [1978] quoted at the beginning of this section on memory hierarchies elaborates on these concepts.

MEASURING (AND CREATING) TECHNOLOGY PROGRESS

The previous sections have presented technology in terms of exponentially decreasing prices and/or exponentially increasing performance. This section presents a basis for this constant change. The progress of a particular technology as a function of time, T(t) has been classically observed to be:

$$T(t) = K \text{ x } e^{ct}$$

This can be converted to a yearly improvement rate, r, by changing the base of the exponential to:

$$T(t) = T \quad \text{x } r^{t-to}$$

where T = the base technology at t_o

and r = yearly increase (or decrease) in the technology metric

This is the same form used for declining (or increasing) cost from base c

$$C = c \quad \text{x } r^{t-to}$$

Clearly there are manufactured goods that neither improve nor decrease in price exponentially, although many presumably could with the proper design and manufacturing tooling investments. The notion of price decline is completely tied to the cumulative learning curves of a) people building a product for a long time, b) process improvement based on learning to build it better, and c) design improvement by engineers based on learning from the history of design. Production learning per se is inadequate to drive cost and prices down because after an extremely long time in production, more units contribute little to learning. With inflation in labor costs, the costs actually rise when the learning is flat. In order to provide a base for predicting the inflationary effect, the consumer price index has been plotted (Fig. CPI).

Learning curves don't appear to be understood beyond intuition. They are (empirical) observations that the amount of human energy, En, required to produce the n^{th} item is:

$$En = K \times n^{d}$$

where K and d are "learning constants". Thus, by producing more items, the repetitive nature of a task causes learning, and hence the time (and perhaps cost) to produce an item decreases with the number produced and not with the calendar time an object is produced.

In his study of technology progress, Fusfeld [1973] took six items, chose a measure of progress in the production thereof, and plotted that measure against cumulative units produced. In each case, he found a relationship of the form:

$$T_{i} = a \times i^{b}$$

where i is the number of units produced and Ti is the value of his selected technology progress measure at the ith unit.

The graph for turbojet engines, where he used fuel consumed per pound as the technology measure, is reproduced in Figure Turbo. The results for all six items studied are shown in the following table:

Item	Measure, Ti	Quantity produced(i)	Technology progress(b)	Change observed in study	Total change
light bulbs	lumens/bulb	10^{10}	.04;.19	33	80
automobiles	vehicle h.p.	$3x10^7;10^8$	.11;.74	10	6;13
titanium	p.s.i./$/16	3×10^8	.3;1;1.04	10	350
aircraft	max.speed	2×10^5	.33-1.2	6	56
turbojet engines	fuel consumed, weight	1.6×10^4	1.06	2	2.9×10^4
computers	mem.size x rate	10^5	2.51	10^9	3.5×10^{12}

Where two values are given for the technology progress constant, a second rate of progress was observed after a significant shift in the industry occurred. For example, in the automobile industry, such a shift occurred in the late 1920's when the acceptance of the automobile, the development of a new tire, and the expansion of the public road network operated concurrently to change the nature of the industry.

Examination of the table will reveal substantial variations in the technology progress constant from item to item. This is probably because most of the technologies represented above are mechanically oriented with associated physical limits. Computer technology is electronically oriented and has not yet reached its limits. In essence the table is comparing systems constrained by Newton's Laws with those determined by Maxwell's Equations.

Using the two formulas,

1) $$T(t) = K \times e^{ct}$$

and

2) $$T_i = a \times i^b$$

one can relate the Fusfeld results (2) to the earlier view of technology progress (1).

First, differentiate (1) with respect to time:

3) $$\underline{dT}/dt = Kce^{ct} = cT$$

$$\text{or } dT/T = c\ dt$$

and by differentiating (2) with respect to the quantity produced, i:

4) $$dT/di = abi^{b-1} = b\ (T/i)$$

$$\text{or } dT/T = b\ (di/i)$$

Since both models must produce the same results, DT/T must be the same for each and therefore can be equated to obtain:

5) $$c\ dt = (b/i)\ di$$

Therefore, the rate of production is:

6) $$di/dt = (c/b)\ i$$

This formula indicates that the production rate is a constant fraction of the total production to date - i.e. production occurs with exponential growth.

While the Fusfeld information is an interesting result, it does not explain why technology improves exponentially, nor does it explain why cost declines exponentially. Learning curves and an exponential increase in the quantity of items produced may depress cost. However, simple production learning does not account for the rapid technology changes in the integrated circuit, for example, where totally different production processes have been evolved to support the greater technology. It appears best to simply observe that the these situations <u>have</u> <u>been</u> true, and can be extrapolated to hold over the next few years because one can see ways by which each limit can be overcome.

Management scientists studying technology evolution [von Hippel, 1977] have made a number of observations about the sources of technology innovation:

1. Technical problem solving is correlated with business activity. Inventors tend to be stimulated by sales and slacken efforts when sales are low. While this might appear to be a counter-intuitive observation, it is not, because inventors are attracted to profitable industries and companies. When sales are low, the inventors move to a more "glamorous" company or industry. Further, inventors are supported by active, enterprising successful companies/industries; but inventors and ideas are unwelcome in companies/industries that are "stable", where it is felt "we already have our ideas," and "we know how to do it without new ideas".

2. Production alone does not stimulate innovation. A lesser number of inventions are stimulated by production needs, except in the auto industry, where "production is where it's at". Of these, the same user-supplier relationship is the best framework. The users of equipment (the producer for the end product) stimulate the production equipment suppliers.

The Influence of Technology Innovation on Cost

The cost of computing is the sum of costs which correspond to the various levels of integration described in Chapter 1 plus the operational costs. The levels were integrated circuits, boards, boxes, cabinets, operating systems, standard languages, special languages, applications components, and applications. In actual practice each additional level-of-integration is often looked at as overhead. Using standard accounting practice, the basic hardware cost, at the lowest level, is then multiplied by an overhead factor at each subsequent outer level. While an overhead-based model may work operationally for a stable set of technologies, such a model will not adequately allow for rapidly evolving technologies or the elimination of levels, for example. By examining each level, as this technology section and the packaging section attempt to do, observations can be made about the use and substitution of technology. More importantly, conclusions can be drawn about how structures are likely to evolve.

In preceding chapters and sections, semiconductor technology has been singled out as the main determinant of a computer's cost, performance, reliability and memory size. Magnetic storage technology is of equal importance because disk and tape memory densities evolve rapidly, even though the cost of a given physical unit does not. These units have become an increasingly major component of the price of computer systems as the costs of processors and primary memory have dropped, but their performance/price improvements have been notable, nonetheless.

Cost, Performance, and Economy of Scale

For most technologies used in the computer industry, there is a relationship between cost, performance, and economy of scale:

$$\text{performance} = k \times \text{cost}^{s} \times r^{t}$$

where
k = base case performance
s = economy of scale coefficient
r = rate of improvement of technology
t = calendar time

There are four options of the amount of economy of scale:

1. Economy of scale holds. A particular object can be implemented at any price, and the performance varies exponentially with price.

   $$\text{performance} = k \times \text{price}^{s};\ s>1$$

2. Linear price performance relationship.

a. performance = k x price

b. performance = base + K x price

3. Constant performance, price independent

performance = k

4. Only a particular device has been implemented. The performance (or size) is a linear sum of such devices.

performance = n x (k x price)

Sometimes, economy of scale effects are observed in situations where their applicability would not normally be expected. For example, assume a performance improvement feature exists that costs the same whether it is added to a large computer or added to a small computer. Adding that feature to a product that is already high priced will have a modest effect (say 5%) on the cost but a substantial effect (say 100%) on the performance. Adding the same constant cost feature to a lower cost product will have a substantial effect (say 200%) on the cost, but a performance effect similar (100%) to that obtained with the higher cost system (or prehaps even less). This condition is especially true in disks and computer systems. Use of a particular recording method employing costly logic for encoding/decoding, or addition of a cache memory is often employed to the high priced systems first. With time, and learning, the technique can then be applied to lower cost systems. For example, cache, a nearly perfect example of the constant cost add-on, first appeared in such large machines as the IBM 360/85 in 1968 and later migrated down to large minicomputers such as the 11/70 in 1975. On a research basis, cache even reached the small minicomputer, the cache-based PDP-8/E at Carnegie Mellon (page 00).

In Fig. Costvstime, the cost of the lowest price unit is kept to a minimum and decreasing while the cost of the mid range product continues to increase. The cost of the highest performance product increases the most, because it can afford the overhead costs. Looking at the basic technology metric, there are really three curves as shown in Fig. Techvstime. The first curve is applied to get the greatest improvement and be applied to the large price unit. With time the technology evolves and is reapplied to the first level copy in the middle range products (to most likely provide the best cost performance) and finally, several years later, the technique becomes commonplace and is applied on low cost products. The resultant cost/performance ratios are shown in Fig. Cost/tech.

In most industries, the management of technology by applying it to products in various price and performance ranges occurs in a more or less ordered fashion, but has not occurred to the extent that it has in the computer industry. This is probably because no other industries have evolved in the same rapid and broad fashion as have the computer and semiconductor industries. The computer industry is fundamentally driven by the semiconductor technology push on the one hand, and by IBM on the other. IBM follows the strategy of applying technology on an economy of scale basis. This permits the technology to be first tested at the high performance, high price, lower volume systems before being introduced in higher volume production. The following examples (from IBM) show this at work. In printing, the high price/low volume to low/price high/volume introduction cycle was followed in the use of dot matrix printing, chain printing, ink jet printing, and computer printing on flexible disks as a precursor to use as systems products using xerography. In magnetic storage, the cycle saw the basic technology for large disks as a precursor to the use of similar technology on smaller disks.

Technology Substitution

Since each constituent technology evolves at its own rate, the cost and performance of a system are roughly the additive and multiplication functions, respectively, of the parts. Usually when one component begins to dominate (e.g., packaging), then pressure occurs to more rapidly change and improve the technology to avoid the cost or performance bottleneck. Sometimes a slowly evolving technology is just eliminated as a substitute is found.

Some of the substitutions that have occurred:

1. Semiconductor memories are now used in place of core memories. Since the latter has evolved more slowly in terms of price decline, semiconductors are now used to the exclusion of cores. (This has not occurred where information must be retained in the memory during periods of time without power.)

2. Read-only semiconductor memories are now substituted for semiconductor logic elements.

3. In a similar way PLAs can be potentially substituted for ROMs and true content addressable memories can replace various read write and ROM memories.

4. The judicious use of CCD or bubble memory can cause drastic reduction (and quite possibly the elimination) of the use MOS random access memories for primary memory. The fixed head disk could be eliminated at the same time.

6. For small systems the main operational memories could be completely non-electromechanical; electromechanical memories (e.g., tape cassettes and floppies) would be used for loading files into the system and for archives. For yet lower cost systems, semiconductors ROMs could replace cassettes and floppies for program storage as in the programmable calculators.

After a while those components of computer system cost which are decreasing less rapidly than other components, or are remaining static, or are rising (like the packaging and power) may become a significant fraction of the total cost. Costs are additive and hence exponential improvements have disproportionate effects causing pressure for structural change.

For instance, although the PDP-8 is normally considered to be the first minicomputer, it post dates the CDC 160 (1960) and DEC's PDP-5 (1963). However, the PDP-8 was unique in its use of technology because:

1. It eliminated the full frame cabinets used by other systems. This also

presented a new computer style such that users could embed the computer in their own cabinets. A separated small box held the processor, memory and many options.

2. Automatic wire wrap technology was used to reduce printed circuit board interconnection cost. This also eliminated errors and reduced check out time.

3. Printed circuit board costs were reduced by using machine insertion of components.

4. The Teletype ASR33 (also used on PDP-5) was connected as the peripheral. It had a combined printer, keyboard, and paper tape i/o device (for program loading). It eliminated the paper tape reader and punch.

The Effect of the Research, Advanced Development Process of the State of the Art Line

The complete development process is pipelined as shown along the lines of Fig. 1 in Chapter 1, page 0 with the with the stages: research, applied research, advanced development (product breadboard), development, test, sell/build, and use. In this model, ideas and information flow through the various organizations in a process-like fashion, culminating in a product. Each product type has a different set of delays associated with the parts of the pipeline. At the end of the pipeline, the "education of use" delay occurs while the prospective customers are taught how the product meets their needs;

this delay culminates in market demand. For well defined, commodity-like products such as disks and primary memory, the education of use delay is zero as each user "knows" the product. For a new language, on the other hand, there is a large education of use delay and the market demand usually develops slowly.

The disk supply process is a good example of the pipeline nature of the development process. The technology (as measured by the number of bits per areal inch) doubles about every two years (i.e., the density improves 41% per year). IBM is estimated to invest about 100 million dollars per year in the development and associated manufacturing process pipelines. Because of this massive investment, the IBM disks essentially establish the state of the art line in a structure that is typified by Fig. Techvstime. Using the pipeline development process, development of competitive disks by other companies would lie somewhere about four to six years behind the state of the art line. This can be seen by looking at the development process and taking into account the delays through each stage. In order to be more competitive, the disk industry short circuits various delays by engaging in reverse engineering; this results in only two year lags. In reverse engineering, the tools are micrometers and reverse molds. At time of the first ship of a new product by the technology leader, the product is purchased by competitors and basically copied on a function per function basis. The more successful designs use pin for pin compatibility so as to take maximum advantage of the leader's design decisions.

From the process, it is also easy to see how merely copying competitive products guarantees products that will be two to four years behind leadership products and lagging the state of the art. Nonetheless, if there is a strong market function which operates to define products based on existing product use, and if the design and manufacturing process at the copying company is quite rapid, such a strategy can be effective. The copying process can be very effective for software products because while there are no delays associated with manufacture, the time to learn about the product provides a time window in which copiers can catch up with the leaders.

A high technology, exponentially increasing (volume) product is denoted by:

1. Exponential yearly cost improvement (price decline) rates through product technology improvements as measured by price decline of > 20% (e.g., disk/price this year = .8 last year's disk price, cpu = .79, memory = .7).

2. Short product life < 4 years.

3. Various types of learning curves. Some products require very little learning, while others require a great deal of learning or require re-learning because of personnel turnover or the frequent hiring of additional personnel.

<u>The Product Problem</u> (Behind the State-of-the-Art)

Typical product situations, including competitive "problems" can be seen in Fig. Product.cost. When a product is introduced to the market, it has a relationship to the state-of-the-art line. There are five possible situations:

1. ideal (on the state-of-the-art line)
2. advanced (moves below the line)
3. late (slip in time to the right)
4. expensive (more than expected in cost, straight above the line)
5. late and expensive (to the right and above the line)

Situations 3, 4, and 5 are product problems because they are behind the state-of-the-art line and hence less competitive. This implies increased sales costs, lower margins, loss of sales, etc. Note that a late product could be OK if somehow the cost were lower. Similarly an expensive product is OK if it appears earlier in time.

Time is Money (and vice versa)

Thus product problems can be solved by either:

1. movement in time (left) to get on the line; or
2. movement in cost (straight down) to get on the line

since

c = cost at time, t (in years)
b = base cost
r = rate of price decline

therefore, with exponential price declines a family of products over a long time will follow a cost curve, c.

$$c = b \times r^{t}$$

now

dc = change in cost above (or below) to get back to the state-of-the-art

dt = delay (or advance) in time to get back to the state-of-the-art line

let

$f = dc/c$ = fraction (%) of cost away from line

$f = 1 - r^{dt}$ poor cost, expressed as project slip

and

$dt = \ln (1 - f) / \ln (r)$ poor timing, expressed as poor cost

These formulas permit the interchange of time and money (cost).

For example, in disks or cpu's where r = .8 and ln.8 = .22

1. $f = 1 - .8^{dt}$
2. $dt = - 4.45 \times \ln (1 - f)$

(using 1.) A 1 year slip is equal to a 20% cost problem

(using 2.) A 10% cost increase is equal to a .47 year slip

Who does what, and their effect

By and large engineering, by establishing the product direction, has the greatest effect on the product. However, since most product problems may have multiple components, it's worth looking at each.

1. Timing

   a. Engineering schedule slips translate into a competitive cost problem as a sub state-of-the-art, late product.

   b. Manufacturing - ramping up the learning curve quickly by risk taking has a high payoff when considering the apparent cost or delay.

2. Cost

   A number of components and organizations contribute to the total product cost, as shown in Fig. Net.

   a. Engineering is perhaps the major determinant of cost by the product design - number of parts, ease of assembly, etc. The most common cost problems occur by continued product enhancement during the design stage to provide increased functionality (called "one-plussing the design"). One-plussing often occurs because the market had not been modeled before the design was begun, and without a model of the market, engineering is a ship without a rudder.

b. Manufacturing - direct labor and overhead really count. Making major changes in the design of a product or the location of manufacture for a product starts a new learning curve and serves to stretch the production time out, and the increased costs associated therewith put false pressure one engineering to design new products. One curve in Figure Net. shows the direct costs associated with manufacturing assembly and then some learning should take place as long as product volumes increase exponentially. New technology materials show the greatest cost improvement for computers, assuming that semiconductors and other electronic materials improve with time. Note that by capital equipment investment (tooling), there can be stepwise cost reductions in materials costs.

c. Inflation - while not a direct cost function, it combines with labor cost to negate the downward cost trends that were obtained from learning effects.

d. Note the costs are taken altogether. In terms of a sub state of the art product, the costs are compound. A one year late, 10% overcost product has the effect of being about 1-1/2 years late or about 30% too expensive.

3. Manufacturing learning

Learning curves and forgetting curves really matter. Left alone, a typical product may go down three alternative paths (see Fig. Product.price):

t

1. c = b x .95 decrease as little as 5%/year.

2. c = b = stay constant with little attention.

t

3. c = b x 1.06 = increase with inflation.

Mid Life Kicker for Product Rejuvination

By enhancing an existing product (the so called mid-life kicker) one can improve the cost/performance metric of a given product. This is non-trivial, and for certain products must be inherent (i.e., designed in). Under these conditions, improvements in cost go immediately to get the product back onto the state-of-the-art-line.

For example a factor of 2 in performance halves cost/performance. The effect, of doubling the density of a disk is to move the product back to the state-of-the-art line by a time shift. Plugging the factor of two improvement into the formulas:

$dt = 4.45 \times \ln (0.5) = 3.1$ years

This situation is shown in Fig. Product.improve and is compared with a 5%/year learning curve.

PACKAGING

Seymour Cray, in a lecture at Lawrence Livermore Laboratory in December 1974, described packaging as the most difficult part of the computer designer's job. The two major problems are heat removal and the thickness of the mat of wires which cover the backpanel. His rule of thumb indicates that with every generation of large computer, the size decreases by roughly a factor of five--and each generation takes roughly five years.

While it is easy to understand why Cray's super computers are so dominated by packaging, it is more interesting to examine the effect of packaging on small computers. At one extreme, the first hand-held scientific calculator, the HP35, was simply a new package for a common object, the calculator, which has been around about 100 years.

Although semiconductor density had been improving for several years, it was was not until densities were high enough to permit implementation of a calculator in a few chips, and not until those chips could be re-packaged in a particular fashion, that the hand-held calculator came into existence. Currently this embodiment is synonomous with the calculator name--in the future, the calculator might take on some other form (e.g., watch, pencil, voice actuated, hearing aid, notebook).

Packaging also seems to be the dominant reason for the PDP-8 and minicomputer phenomenon--although marketing, the coining of the name, and the ease of manufacture (also part of packaging) are alternative explanations. The packaging advantages of the PDP-8 over predecessor machines can be seen from the photographs in Part II.

The Packaging Design Problem

Packaging is the complete design activity of interconnecting a set of components via a mechanical structure in order to carry out a given function. In order to package a large structure such as a computer, the problem is further broken into a series of levels each with components that carry out a given function. Figure 3 shows the hierarchy of levels that have evolved these last twenty years for the DEC computers. There are eight levels which describe the component hierarchy resulting in a computer system.

For each packaging level there is a set of interrelated design activities as shown in Fig. 4. The activities are almost independent of the level at which they are carried out, and some design activities are carried out across several levels.

While the initial design activities indicated in Figure 4 are each aimed at solving a particular problem, the solving of one problem in computer engineering usually creates other problems as side effects. For example, the integrated circuits and other equipment that do information processing require power to operate. The power creates a safety hazard and is provided by power supplies that operate at less than 100% efficiency. These side effects create a need for designing insulators and providing methods of carrying the heat away from the power supply and the components being powered. In this way, cooling problems are created. Sometimes cooling is carried out using conduction to an outside surface so that it may be carried away by the air in a room, but most cooling is carried out by convection with a cabinet fan which carries air into the room so that the room air conditioning system is left with the problem of carrying the heat away. In this process, the fans create acoustical noise pollution in the room, making it more difficult for humans to work in the room. Furthermore, if the computer is used in an unusually harsh environment, a special heat exchanger is required in order to avoid contamination of the components within the computer by the pollutants present in the cooling air flow.

Finally, a particular package exhibits mechanical characteristics such as weight and size. These parameters directly affect manufacturing and shipment costs. They determine whether a system can be built, and whether it can be shipped in a certain size airplane, or carried by a particular distribution channel (e.g., parcel post or United Parcel). The dynamic characteristics determine the type of vehicle (special air ride van for electronic equipment) in which equipment must be shipped.

It is also necessary to examine the particular design parameter in order to determine whether it is a constant (meets the German VDE standard x), a goal (cheap as possible), or part of a more complex objective function (measured in bits/sec/$ or part of a system benchmark--jobs/sec/$).

The following table lists the various kinds of design activities and whether they deal with goals, constraints, or parts of more complex objective functions. The table also gives the dimensions of various metrics (e.g., cost, weight) available to measure the designs and many of these metrics are used in subsequent comparisons.

Table Design Activities, Metrics and Environment Setting Goals and Constraints

Design Activity	Determining Environment and [Metrics]
Primary function and performance (e.g., memory)	Market, next highest level (i.e. the consumer) of system [memory size (bits), operation-rate (bits/sec.)]
Human engineering	Human factors criteria, competitive market
Visual/Aesthetics	Market, other similar objects, the environment in which the object is to exist--usually only important at outer-most level
Acoustic noise	Government standards, operating environment, market [decibels in various frequency bands]
Mechanical	Shippability (e.g., air cargo container size, truck vibration), handling, assembly/disassembly time [weight, floor area, volume, expandability, acceleration, mechanical frequency response
EMI	
radiation input	Government standards, must operate within intended environment (e.g., high noise) [power vs. frequency]
Power	Operating environment, market [watts, voltage supply range]
Cooling and environment	Market, intended storage and operating environment, government standards [heat dissipation, temperature range, air flow, humidity range, salinity, dust particle, hazardous gas]
Safety	Government standards
Cost	
cost/metric ratios	[cost/performance (its function)--cost/bit and cost/bit/sec., cost/weight, cost/area, cost/volume, cost/watt]
density metrics	[weight/volume, watts/volume, operation-rate/volume]
power metrics	[operation-rate/watt; efficiency = power out/power in]
reliability	[reliability--failure rate (Mean Time Between Failures, Mean Time To Repair (MTTR)]

Given the basic design activities, one may now examine their interaction with the hierarchy of levels (i.e., the systems) being designed (see Table 00). This is done by looking at each level and examining the interaction of the design activities for that level with other design activities (e.g., function requires power, power requires cooling, cooling requires fans, fans create noise and noise requires noise suppression).

Table Interrelationship of Hierarchy of Levels and Design Activities

Design\Level of Packaging Activity\	Chip	Chip Pkg	Module	Backplane	Box	Cabinet	Computer Systems
Functional	logic------ electrical	----------	----------	------>		configuration options	selection of right components by user
	circuit design physical layout		physical layout	physical layout	what fits and operates	boxes, what configurations will operate	
Human interface						location of console, size for use	placement for use
Visual					visible, bought for integration	determines system appearance	set of cabs, attractive place to be
Acoustic				Airflow ------ vibration	----------	------>	quiet for operators and users
Mechanical	buildable;		shippable	serviceable ----	----------	------>	floor load
	signal transmission	----------	----------	----------	----------	------>	room size
EMI	noise coupling and rejection of ext.RFI		inter/intra module noise containment and shielding	coupling, RFI	RFI containment, external RFI shield		away from RFI input (outside operating range)

Power	special on-chip		dist. and regulation	dist. and regulation	control, dist. and regulation	interconnect- with computer system	by user. special power supplies for high availability
Cooling and other environment	chip to cooling special envir.	IC module cooling special envir.	IC to cooling	module	cooling & covering	source	interbox coupling to room air envir.
Safety			power----------> for various systems		determines safety if used at this level	determines user safety	
Dominant design activities	circuit logic	logic ----------------------->			mechanical, power, cooling, EMI, acoustic	configtion visual, shipping EMI, safety	user configation design

Note: The box and backplane levels can be considered as a single level. (Alternatively, the box level may be eliminated in large systems)

Computer Systems Level

The topmost level in the preceding table is the "computer system", which for the larger mini and "mainframe" 10 computers consists of a set of subsystems (processor, memories, etc.) within cabinets, housed in a room, and interconnected by cables. The functional design activity is the selection and interconnection of the cabinets, with a basic computer cabinet holding processor, memory, and interfaces to peripheral units. Disks, magnetic tape units, printers, and terminals occupy free standing cabinets. The functional design is usually carried out by the user and consists of selecting the right components to meet cost, speed, number of users, data-base size, language (programming), reliability and interface constraints. Aside from the functional design problem, cooling and power design are significant for larger computers. For smaller computers, accessibility, acoustic noise, and visual considerations are significant because these machines become part of a local environment--and must "fit in".

Cabinet Level

Since the cabinet is the lowest level component that users interface to and observe, the physical design, visual appearance, and human factors engineering are the dominant design activities. For the computer hardware designer, on the other hand, the component associated with the cabinet is often his largest system. His functional design efforts insure that the various components (i.e., boxes) that make up a cabinet level system will operate correctly when interconnected. Safety and EMI characteristics are important because the cabinet serves as the outermost place that shielding can be installed. Cooling and power distribution must be considered, since a number of different boxes may be mounted within the same cabinet. Finally, the mechanical structure of a cabinet must be designed to maintain its physical integrity when shipped.

Box Level

Box level functional design consists of taking one or more backplanes, the power supplies for the box, and any user interface such as an operator's console and interconnecting them mechanically (see Figure 11/05). For systems that are not sold at the box level, no separate box is required, and the power supply and backplanes are mounted directly in a cabinet (see

Figures 11/70.1) or other holding structure such as a desk or terminal case, so that box and backplane design are merged as one. If systems are sold at the box level, then the visual characteristics may be important; otherwise, the design is basically mechanical and consists of cooling, power distribution, and control of acoustic noise. The structure must be basically quite sound in order to protect the unit during shipment.

Backplane Level

This level of design is the final level of interconnection for the computer components that are designed to stand alone, such as a basic computer disk, or terminal. Backplane design is part of the computer's logical design. In second generation machines such as the PDP-7 (Figure 24A, Chapter 5), the backplane was wire wrapped. In the early 1970's printed circuit boards were used to interconnect modules (Figure BP). Figure 00, page 00 shows how the Omnibus backplane for PDP-8/A is used to interconnect the basic components within the box. Secondary design activities include holding, powering and cooling the modules so they will operate correctly. Since the signals are transmitted on the backplane, there is an EMI design problem. For industrial control systems whose function is to switch power, additional safety problems are created.

Module Level

In the second generation, module level design was a circuit design activity taking discrete circuits and interconnecting them to provide a given logic function. In the third and fourth generations, this interface between circuit and logic design moved to be within chip level design, so module level design became the process of dealing with physical layout design problems associated with logical design. This shift in roles (function) will be described below in the section discussing the interrelationship between the technology generations and packaging. The integrated circuits that perform the functions are assigned to different positions on the module. Module level design is basically electronic, so power, cooling, and EMI (cross talk) considerations dominate.

Integrated Circuit Package and Chip Level

Most integrated circuits used in the computer industry today are sold in a plastic or ceramic package configuration that has two rows of pins, and is hence called a Dual In-line Package, abbreviated DIP. The majority of the IC's in the module shown in Figure LSI11 are 16-pin DIPs. Because of the popularity of this packaging style, the terms IC, chip, and DIP are often used interchangably. This is not strictly correct, as an integrated circuit is actually a quarter inch square portion of semiconductor material (die or "chip") from a two inch to four inch diameter semiconductor wafer. Until multiple dice are packaged within a single DIP, creating different design interconnections at the IC and DIP levels, the IC and DIP can be discussed as a single level. In Figure LSI11, the center DIP of the four large DIPs has two dice mounted on it.

Logic design is a part of the functional design problem at the chip level. The task is to supply a terminal behavior that can be used at the next (module level) and, depending on the technology generation, this function can vary from a simple gate (in the early third generation) to a full computer (late in the fourth generation).

Other design activities at this level include generic (?) to electrical signal processing: power, heat dissipation, and EMI. Since some ICs are designed to operate in hostile environments, there is a considerable mechanical design activity associated with packaging, interconnection and manufacturing.

The Packaging Evolution

Figure 6 shows the relation of packaging and the computer classes for the various computer generations. For each new generation, there is a short, evolutionary transition phase, but ultimately the new technology is repackaged such that a complete information storage or processing component (bit, register, processor) occupies a small fraction of the space and costs a small fraction of the amount it did in the prior generation. Quite discrete events mark packaging characteristics of each generation, starting from 1 bit per vacuum tube chassis in the first generation and evolving to a complete computer on a single integrated circuit chip in the fifth generation. Not only the size of the packaging changed, but also the mounting methods. In the first generation, logical units were permanently mounted in racks, whereas in the later generations they were made removable for ease in servicing.

Whereas the time-line shows the packaging evolution of a complete computer, the following table shows how a particular component, now called the Universal Asynchronous Receiver-Transmitter (UART), has evolved with time.

Table __ Packaging Hierarchy Evolution for Universal Asynchronous Receiver Transmitter (UART) Telegraph Line Controller

early second	**late second**	**early third**	**late third**	**late fourth**
backplane				
modules	2 modules	module		
discrete	discrete	IC	IC	
circuit	circuit	chip	chip	chip area

The UART logic carries out the function of interfacing to a communications line carrying serial data and transforming the data to parallel on a character by character basis for entry into the rest of the computer system. The UART has three basic components: the serial/parallel conversion and buffering; the interfaces to both the computer and to the communication line; and the sequential controller for the circuit.

The UART is probably the first fourth generation computer component, since it is somewhat less complex than a processor, yet rich enough to be identifiable with a clean, standard interface.[1]

Footnote 1

Of historical note, DEC played a significant part in the development of the UART technology. With the PDP-1, the first UART function was designed using 500 Khz systems modules, and was used in a message switching application as described in chapter 00. The interface was called a line unit, and was subsequently repackaged in the late second generation to be on two extended systems modules (see Fig. 9). The UART function was also built into the PDP-8/I using two modules, but substantially smaller ones than those for the PDP-1. In the 680/I, a PDP-8/I driven message switch, the UART function was accomplished by programmed bit sampling. Late in the third generation (or at the beginning of the fourth generation), some designers from Solid State Data Systems, a small company on Long Island, worked with Vince Bastiani at DEC and developed a UART that occupied a single chip. This subsequently evolved to become a standard IC, and has been used throughout the industry, for example in the DL-11 communications line interface module for the PDP11.

The DEC Computer Packaging Generations

With this general background on packaging, one can examine the DEC packaging evolution more specifically and against the general archetype of Figure 3. Figure 12 shows how the hierarchies have changed with the technology generations. The figure is segmented into the different product groupings. A product is identified as being at a unique level if it is sold at the particular packaging level. The first DEC computers (i.e., PDP-1 to PDP-6) were sold at the cabinet level as complete hardware systems. Although the PDP-8 was available at the cabinet level for complete systems, it was significantly smaller than the previous machines and was principally sold at the mechanical box level so that it could be incorporated into other hardware systems and used within the packaging hierarchies of OEM customers. Subsequently computer systems evolved to be available at the backplane level (LSI-11), and at the module level (CMOS-8).

The original packaging hierarchy for most of DEC's second generation computers used a relatively common packaging scheme based on the PDP-1. The most significant change occurred in the late second generation when R-Series and B-Series Flip Chip modules (see Fig. 12) were introduced so that backplanes could be wire wrapped automatically, enabling lower cost and greater scale production.

The change to wire wrap technology also enabled the box-level production of computers. The change to wire wrap and two level (box and cabinet level) products is clear.

The change to IC packaging in the third generation gave rise to computers that are sold at the box level.

The CMOS-8 module, described more completely in Chapter xx, is a single-board complete computer with processor, 16 Kword memory, and all the optional controllers to directly interface up to five peripheral options. For a computer packaged in this fashion, a backplane is not used, and hence the backplane packaging level doesn't exist.

The LSI-11 is marketed at the three levels in Figure 12. Although components are sold as separate modules (e.g., communications line interfaces, additional primary memory), a complete system requires a backplane, thus the lowest level for the product is the backplane. For larger systems, a power supply is combined and placed in a metal box (the 11/03), and finally a complete system such as the 11V03 is available with terminal and mass storage in a single cabinet.

Specific Cabinet and Box-level Designs

Cabinet and box-level design is perhaps the most difficult part of computer design, yet it is perceived (incorrectly) to be trivial and hence deferred until last. It is also misunderstood because there is not a single discipline or complete set of well understood laws that govern the design. This deception arises from the fact that, on first glance, the only physical law is that not more than one thing can occupy the same space at a given time. However, a number of disciplines must be understood, including acoustical engineering, heat transfer, aesthetics, human engineering, RF engineering, and the understanding of package functionality. In addition to this understanding, each box type requires a separate specialized manufacturing process.

Obviously no one person fully understands all the disciplines necessary to correctly design a package. Herein lies the problem: Packaging is the integration of a number of separate disciplines. One needs to look at the basic packaging problem by first ennumerating the possibilities for placing modules within a particular box package. By being quite restrictive, one can build a grammar with six production rules that describe the possible DEC boxes. Even with this simple grammar, about three million possibilities can be generated. The large (combinatorial) number of possibilities arise from the basic size and way a box is held, the console mounting, cooling, power supply location and the way modules are mounted within the box. The packaging possibilities can then be explored by simply writing statements that express the alternatives for the separate decision dimensions.

A Grammar to Generate Three Million Box/Cabinet Alternatives

1) Size and Mounting

The box is (1/2 or full) cabinet depth by (3-3/4", 5-1/4", 10-1/2", 15-3/4" or 21") high, mounted with (fixed slides, right hand hinge, or left hand hinge).

2) Console

The console is (non-existent, simple power on/off, maintenance only, full feature maintenance/programming).

3) Cooling

The cooling is carried out by (box-level or cabinet-level)(fan or fans) producing (plenum or forced) air flow that is (normal or parallel) to the module mounting axis, which has eight possible orientations, or else the cooling is carried out by natural convection with air entering at the (top, bottom, right side, left side, front, or rear), and exhausting through the (top, bottom, right side, left side, front, or rear).

4) Power Supply Location

The external power supply is mounted (separately, attached behind the box, attached to the back of the cabinet, or attached on the front of the cabinet), or an internal power supply is mounted using a mounting scheme that is the (same as or different than) that of the modules, and is located at the (top, bottom, right side, left side, front, or rear) of the cabinet.

5) Module Mounting

There are (no, one, or many) backplanes for mounting modules.

6) Module Pins

The module pin axis is (top, bottom, right side, left side, front, or rear) mounted, oriented in the (horizonal or vertical) plane with the IC side facing (up, down, right, left, front, or rear).

One can state the six sentences (1 point in the decision space) that describe the PDP-8/A:

1. The box is 1/2 cabinet depth by 10-1/2" high with fixed mounting.

2. The console is for programming and maintenance.

3. Cooling is carried out by box-level fans giving forced air cooling flow parallel to the module mounting axis.

4. The box power supply uses a different mounting scheme and is located at the bottom of the box.

5. There is one backplane for mounting modules.

6. Modules are rear (wall) mounted, oriented in the horizontal plane with IC side facing up.

Of all the dimensions to consider in the design, perhaps the most important is how the box (or module mounting structure) is placed in a cabinet. This placement effects air flow, shippability, configurability, cable placement, and serviceability, and is a classical case of design tradeoffs. The scheme that provides the best metrics such as packaging density and weight, may have the poorest access for service, and the most undesireable cable connection characteristics. These characteristics are given in the following table:

Table __ Fixed, Drawer and Hinged Box/Cabinet Mounting

	SERVICE ACCESS	CABLING	DENSITY	COOLING	APPLICABILITY
Fixed	Good for either backplane or module, but not both unless a thin cabinet is used	Best (i.e., shortest)	Good for thin or rear cabinet PS mounting	Best (known)	Box not needed; box can be used
Drawer	One side access	Long + moves	Very high	Can be	High density, self contained
Drawer (with tilt) for service	Good	Longer + much more moves	Very high	Cooled*	
Drawer vertical mounting modules	Very good	Longer + moves	High		
Hinged (module backplane)	Very good	Short	Medium	Good (if fans are fixed to cage)	Separate box is awkward

*Density restricts cabinet airflow

Figure Cab shows the various boxes (top view) as they fit within a cabinet profile. In some cases there is no two-level cabinet-box hierarchy.

Packaging is largely a matter of designer preference because the only clear goal is manufacturing cost. Marketing considerations, especially for OEM use, drives toward getting the highest density to minimize volume, floor space and rack mounting height. Despite this, many designers prefer packaging schemes that are not the highest density, but rather are fixed so that the power, cooling, and EMI conditions can be well-understood and so that the cabling can be done more rigidly. This has been particularly true for the larger computers such as PDP10's.

Power Supplies

While logical functions can be performed using small quantities of electrons and thus accommodated in very small physical structures, the power to move those electrons at useful speeds comes from power supplies which do not scale down in size as readily as the logical functions they suppport. Power supply technology has not provided the impressive increases in capability per dollar or capability per cubic foot that semiconductor technology has. Power supplies involve such materials properties as voltage breakdown limits, dielectric constants, magnetic permeability, and heat conductivity. Since these properties vary with physical dimension, increased capabilities in terms of voltage breakdown rating, capacitance, inductance, or heat dissipation are gained by making the component physically larger.

The performance criteria for power supplies are predominantly driven by the application for which they are designed. These criteria are given in terms of various efficiencies of volume, weight, power conversion, and cost. It is somewhat difficult to compare the various supplies, as they are all available at different times, produced in different quantities, designed for different reliabilities, and available with different features.

For the computer industry, power supplies can be divided into three main categories: processor and memory power supplies, disk and tape power supplies, and terminal power supplies. Each of these product categories has a unique set of reguirements, which are summarized in Table CVPST.

Table CVPST: Characteristics of Various Power Supply Types

	Processor & Memory	Disk & Tape	Terminal
Power Requirements	250-2500 watts	100-500 watts	0-150 watts
Use	logic	very low noise for head electronics; high current for servos	high voltage for CRT; high current for mechnical motions
Quantity in System	low to medium	medium	high
Typical Cost Sensitivity	low	medium	high
Size	important, especially in boxed computers	not important	very important
Weight	relatively unimportant	not important	very important
Reliability	very important	important	important

Features	power line sensing battery backup

Three of the four efficiency measures, cost (in relative cost per watt), weight (in watts per pound), and volume (in watts per cubic inch), are plotted for processor power supplies in Figures P1 and P2. In Figure P1 the plots use a time axis, and in Figure P2 the plots use a watts-of-output axis. The fourth efficiency measure, power conversion (in watts out per watts in) is given in Figure P3, using a time axis.

The cost of a power system is very dependent on the unit electrical size and technology. The features required on the units such as power monitoring (AC low, DC low), battery backed-up power, and servicing aids also significantly influence the cost. Since the cost is size dependent, a relative metric, dollars per watt, is chosen for processor power supplies.

In the cost characteristics there are different bands of cost curves which are technology dependent. These bands of curves span new, mature, and obsolete technologies. For example, the cost of power supply technology until just recently was very dependent upon the prices of iron and copper and the cost of labor. Now, costs of power supply technology tend to track semiconductor costs due to the widespread use of line switching power supplies. There are also bands within the cost curves; these represent the size dependency; larger power supplies are the most cost effective, with one exception (figure P1a and P2a).

The size of power supplies for minicomputers has been important, especially

for the boxed versions. The volume occupied by logic has gone down for the constant functionality computer; however, power requirements have declined far less than logic volume, and hence power densities have increased. Whereas 250 watts used to suffice for a 10 1/2 by 19 by 25 inch box, 800 watts is now required, and the space for the power supply has barely increased. This has put substantial constraints on the weight and efficiency of power systems, and at times space utilization has been (inadvertently) traded off against cost, manufacturability, and serviceability.

In response to these space pressures, there has been a constant gain in volumetric efficiency (Figure P1b) over the years with the highly dense power supplies on the top of the band and the modular packaged units on the bottom. With the introduction of line switching power supplies, this curve made a quantum jump. The increase in volumetric efficiency, plotted relative to time in Figure P1b, is plotted relative to output wattage in Figure P2b.

Power supply technology not only determines volunetric efficiency, it also determines the weight of the unit. Here again the use of high frequency line switcher technology rather than low frequency transformer technology has produced marked results - in this case, two distinct curves.

The weight efficiency (Watts/lbs) is fairly constant over the years and showed a slight improvement as larger supplies were built (figure P2c).

Finally, Figure P3 shows how power supply efficiency is improving with time. Note that with direct line switching, efficiencies of 70% are to be expected. This efficiency permits the aforementioned increase in volumetric effciency since there is less heat to dissipate.

Modules

Since the function of modules is to interconnect and hold components, the metrics for modules are area (for mounting the components) and the cost of each circuit interconnection. For minicomputers, the emphasis has been to have larger modules with more components packed on a module as a means to lower the interconnection cost. Figure Mod.size shows the area of DEC modules and the number of external pins per module versus time. Since the integrated circuit densities have been always increasing, in effect providing lower interconnection costs, a given module automatically provides increased interconnects simply by packaging the same number of ICs on a module. Obviously, one does not want to credit this effect to improved module packaging. By increasing the components per module, the cost per interconnect can be reduced provided the cost to test the module increases less rapidly than the increase in components. The emphasis on module size is usually most intense for larger systems, where a relatively large number of modules are needed to form a complete system.

Until recently, the increase in module area was accompanied by increases in the number of pins available to interconnect to the backplane. In the case of the VAX 11/780 and the PDP-10, the number of pins did not increase significantly over previous designs, although the board area was 50% larger. In these cases, the number of ICs able to be cooled limits the density. In other cases either the number of pins or the module size limits the module's functionality. There are similar effects throughout the generations.

In the early second generation Systems Module designs, the number of pins and the circuit board area (in square inches) were about the same. Components were fairly large and loosely packed on modules. With the Flip-Chip series, circuits were modified to pack more, smaller, components on a single module, using automatic component insertion equipment, and some of the space consuming components (e.g. pulse transformers) of the earlier circuits were removed so that a module design was a better balance between area and pins. As a result, the early second generation Flip-Chip modules had higher packing densities than comparable Systems Modules.

With the beginning of the third generation, the need for more printed pins to the backplane was clear, since so many interconnections were made on the computer's backplane. The PDP8/I was the first DEC integrated circuit computer, and the packaging philosophy strictly followed that of the second generation. As a result, the sudden increase in component functions made the modules drastically lacking in pins. By putting pins on both sides of the module, the number of pins for a double height module (20 square inches)was increased from 36 to 72, but was still inadequate. Assuming that each IC has 14 signal pins and a module had 70 signal pins, only 5 ICs could be placed on a board and still have pins brought out to the backplane pins, although the twenty square inch area of the module could potentially hold 20 ICs.

Although the 8/I was packaged using the twenty square inch 72-pin modules, it was clear that another packaging scheme was necessary to utilize ICs, modules, pins, and backplanes. Thus when the 11/20 and 8/E were designed (about 1970), they used larger modules in order to carry the large number of intra-module interconnections required when many ICs were placed on a single module.

It is interesting to note that in the recent case of high density ICs, the LSI11/2, the module area was too large to have a single option on a module, and since the LSI11 bus only required a few signals, the number of pins was more than adequate. Here the modules are functionality limited, versus pin limited. The figure attempts to annotate situations as to whether pins or modules limited the design.

Although the size of the module is important in determining the systems that can be built, how they are serviced, and how they are manufactured, the important module metric is the cost per interconnection on the printed circuit board (and remainder of the system). Figure Mod.cost shows how this has varied with time. Here one can see that the introduction of Flip-Chip modules initially increased costs (because learning almost had to start over again).

Interconnection costs consist of the printed circuit board, the cost to insert the components on the module and the cost to test the module. Printed circuit board costs have been decreasing with time, reflecting both benefits of learning and the benefits of placing more ICs on a single module, giving a compound economy of scale effect. The cost to assemble the components on the module have decreased rapidly, reflecting the increasing use of automatic component insertion machines. Testing has not been a significant cost component in module manufacturing. (It does represent a substantial cost by the time the module has been integrated into a system at the customer's site. This is because DEC tests modules not only as modules, but also as sub-assemblies, and as parts of systems both in the factory and in the field.) The total cost per interconnection has been decreasing, but the trend may either stay constant or even increase as greater use of LSI decreases the number of total connections in a system, but makes the remaining interconnections more expensive to assemble and test.

Backplanes

Backplanes provide the next level of integration packaging above modules and are used to hold and interconnect a set of modules which form a computer or an option (e.g. processor, memory, ot peripheral controller). Figure Backplane.cost gives the relative cost of interconnection of backplane module pins (using the same scale as Figure Mod.cost). Here the cost per interconnection is roughly the same as with a printed circuit module interconnection. This can be somewhat misleading because backplanes require a negligible cost for testing and few failures occur during testing.

The figure shows various kinds of interconnection technologies. Even though there are exponential increases in quantities produced, the cost continues to increase in the long run, with only occasional downward steps in cost. The greatest cost decline occurred when interconnections were carried out using automatic wire wrap machinery, but the 8/E was equally significant by using a completely wave-soldered backplane. This figure is also useful in seeing how effectively the module pins were used (i.e. whether all available pins were used). Note the phenomenon described with the 8/I, where modules were clearly pin limited, shows up clearly.

Boxes and Cabinets

Since the function of the cabinet and box is to hold backplanes which in turn hold modules, which in turn hold circuit-level components, the metric of electronic enclosures is the number of printed circuit boards they hold. The earliest method of mounting was to place the backplanes directly in a six foot high cabinet which held 19 inch wide equipment in a 22 inch by 30 inch floor space and weighed about 185 pounds. Figure Cab (p. 83) shows the top view of the various cabinets used to hold module backplanes and boxes for minicomputers since 1960. The changes to the basic DEC six foot cabinet have been mainly for improved producibility. The latest (circa 1973) was to use riveted upright supporting members so the cabinet could be assembled easily without requiring bulk space for shipment and storage.

The original cabinet used the entire cabinet as an air plenum so that air was forced between the modules and out the front doors. When the PDP-7 used the same cabinet and the module mounting frame cut off the air flow, it was necssary to add fans to the back doors to blow air at the modules. Since cooling was one of the weak points in the 7, the 9 used a self-contained mounting and cooling structure in which air was circulated between the modules, with air pulled in from outside without going through the cabinet.

A second, later packaging method, initiated with the PDP-8, packaged the metal boxed minicomputer inside the six foot cabinet. Figure Boxes shows the significant boxes that have been used to package minicomputers both within the six foot cabinet and freestanding. The box packaging history begins with the PDP-8. The rows of the figure indicate the four ways (see page 00) that are available to access the circuitry (fixed, book, slides, and tilt for access). The 8 design was followed by the 8/S design which oriented the modules with the pins up for access to the backplane. By tilting (rotating) the box the handle side of the modules could be accessed. For the 8/I (not shown), modules were mounted in a vertical plane.

Several fixed backplane module mounting structures were formed beginning with the 8/A (1975). Note that over ten years have elapsed since minicomputers were mounted in a fixed structure in a cabinet (i.e. PDP5).

Heat

Although the volumetric measures of module area, and size of the cabinet are also important, the amount of heat the enclosure is capable of dissipating is the most important because of reliability. The following table gives some of the important metrics of several of the recent DEC computer boxes:

Table EBC: Expansion Box Characteristics

Box Model	Used On	Year	Weight (Lbs.)	Size (Cu.Ft)	Modules	Module Volume (Cu.Ft)	Heat In	Heat Density (KW/Ft3)	Space Efficiency
BA11-D	11/35	1974	100	2.6	24 hex	.93	0.7	.27	.35
BA11-E	11/45	1972	100	2.6	27 quad	.7	0.7	.27	.27
BA11-F	40&45&70	1972	260	5.3	44 hex	1.7	2.2	.42	.32
BA11-K	04&34&70	1974	110	2.6	24 hex	.93	1.0	.38	.36
BA11-L	11/04	1976	50	1.3	9 hex	.35	.55	.43	.27
BA11-M	11/03	1975	25	0.5	4 quad	.1	.25	.54	.24
BA11-N	11/03	1977	40	1.0	9 quad	.23	.24	.31	.22
BA11-P	11/60	1977	100	3.0	29 hex	1.1	1.1	.36	.22
BA8-CA	8/A	1975	117	2.4	20 quad	.52	1.2	.50	.22
H9300	8/A	1977	55	1.1	10 quad	.26	0.3	.26	.24
H9500	11/780	1978	344	43.4	67 ext. hex	3.7	6.0	.15	.10

Figure Heat.den gives the heat density for the various boxes. The amount of heat dissipated by the box is in terms of the kilowatts per cubic foot and has been relatively constant with time. There has been a great variation about a norm, and the very high heat dissipation of the first 8/A (due to high packing density and a relatively inefficient power supply) affected the next design to a lower density. The space utilization shows a similar picture, although the efficiency appears to be declining (see Figure Space.util). This decline is hardly noticeable and is even surprising in light of more efficient power supplies which make it possible to place more components in a given enclosure. The cost effectiveness of the average enclosure, as measured by the material cost, is declining with time as measured by the relative cost of materials per cubic foot of modules held (Figure Cost.payload).

The time chart gives a completely erroneous view of the situation, because economy of scale is not considered. Figure Box.cost shows how the relative cost of box materials varies with the volume (in number of hex modules). Here the upward trend of the previous figure is not apparent, but it merely occurs because later packages are for smaller numbers of modules.

AN OVERVIEW OF MANUFACTURING

Although the result of a design project is an entity which is manufactured, very little is written about manufacturing in computer engineering literature. Such literature generally discusses algorithms, logic design, and circuit technology. Yet for a computer to be commercially successful, it must be manufacturable, economically operable, and serviceable. Moreover, for most of the computer engineering discussed in this book, because the designs are intended for volume production, engineering costs are small (1-10%) compared with other product and lifecycle costs. The product cost is determined by the price of the components and the manufacturing process~~.~~; the ~~The~~ ~~lifecycle~~ stet ~~cost includes the purchase price, the operational costs, and service costs.~~ ¶ For production, machines must be easy to assemble and test, repair must be rapid, engineering changes must be introduced smoothly, and the production line cannot be held up because of shortages of components -- all parts of traditional manufacturing understanding.

A detailed discussion of manufacturing is clearly beyond the scope of this text. Information on test equipment is to be found mainly in the manuals published by test equipment manufacturers. References on quality control include [Juran, 1962] and [Grant and Leavenworth, 1972].

The Life Cycle of a Product

Figure Lifecycle shows a simplistic process flow for the major phases and milestones in the life of a product. In reality, planning and designs for many of the phases go on concurrently. The early research, advanced development, and definitional phases are not shown. Often, products proceed from the idea stage to the engineering breadboard and are then terminated because they do not meet original goals or because better ideas arise.

To accomodate changes, the engineering breadboard is usually built with wirewrapped boards rather than printed circuit boards, if the circuit technologies used permit the long wire lengths characteristic of wirewrapped boards. At or before the breadboard stage, schedules are made for manufacturing start up. Other organizations formulate and execute plans: systems engineering - for product test/verification; software engineering - for special software and verification; marketing - for promotion and product distribution; sales - for training; field service - for training and parts logistics; and software support.

After the engineering breadboard has been debugged, construction of engineering prototypes begins. The engineering prototypes prove the design using the actual printed circuit modules that will be used in manufacturing. Usually a number of prototypes are constructed, the number varying from ten to a hundred depending on the complexity, cost, and anticipated product volume. All processors and peripherals in the planned systems configurations are tested in conjunction with the prototypes. The complete system must meet the product specifications and must run all of the system software.

The requirement that all of the system software be run is an excellent supplement to the normal testing of prototypes, and is especially useful when the product being designed is a processor with a mature architecture, because more software is available for use in testing the prototype. Since the number of possible states and state sequences in a computer system is very large, a diagnostic test which exercises every one is impractical. Diagnostic programs and microdiagnostics therefore test a judiciously chosen subset of all states. Such programs are not perfect, however, and therefore system software is run as well. Thus, the more software that is available to test a prototype, the less likely it is that a design error will go unfound. The general problem of testing requires much more work before it can be considered mature. One would like to see, for example, the automatic generation of verification programs from an ISP description of the architecture being built.

Limited release (LR) to manufacturing is a major milestone. The product is placed under formal engineering change control; specifications and documentation are available for the product and manufacturing process. For the integrated circuits, sources of supply and testing procedures are in place. Process control tapes are ready for the numerically controlled machine tools, such as component insertion, backplane wiring, and printed-circuit board drilling machines. Any special tooling for the mechanical packaging has been obtained. Testing at all levels has been specified; test programs for computer-controlled testers have been written, special test equipment has been built, and diagnostic programs are ready.

Design maturity testing (DMT) with a number of engineering prototypes verifies the design and justifies the risk of a pilot run. Tests for reliability and functionality are conducted. Environmental testing (shock, temperature, humidity, static discharge, radiation, power interrupt, safety) is conducted at this stage.

The pilot run shakes down and verifies the actual manufacturing process by building a small number of units, at the manufacturing plant, using the product and process documentation.

Product announcement usually occurs during the DMT period but can occur at any time - often as early as limited release or as late as first customer shipment, depending on the marketing strategy. This strategy is clearly a function of the volume, novelty, and competitive needs.

Process maturity testing (PMT) verifies that the product is being manufactured with the desired cost, quality, and production rate. After PMT, the steady-state phase of manufacturing continues, with possible perturbations due to the introduction of product enhancements or process changes to lower product costs, until the product is phased out.

Manufacturing Process Flows

An overview of a manufacturing process is given in Figure Manufover, which shows how a product moves through the various factories. There are often different plants for boards, peripherals, memories, and central processors. Integration from the other stages and stock storage occurs at the stage called final assembly and test (FA&T). Here, the software system that is to be run, operations manuals, and other documentation is also integrated and tested.

Figure process60 gives the complete flow for a typical volume manufacturing line -- the 11/60 central processor facility in Aguadilla, Puerto Rico. Integrated circuits are manufactured elsewhere and are sent to be inserted, soldered in, and tested in the module manufacturing plant in nearby San German.

The manufacturing process includes extensive thermal cycling to ensure that component "infant mortality" cases are discovered early during manufacturing, because it is more expensive to find defective components at the later ~~larger~~, more integrated systems level. The temperature/humidity environmental chambers which house twelve or sixteen CPU's each are shown in Fig. TChamber. Figures 2224 and qvstation show small heated enclosures used to induce failure during the test and repair of modules.

Since testing occurs at each stage in the manufacturing process, dedicated logic must be added to the design, to provide physical access "probes" for the test equipment. To test a particular function, it must be specifiable, invokable, and observable. For example, the function of an adder can be clearly specified, but it cannot be easily invoked or observed if its inputs and outputs are etch runs on a printed circuit board. Several testing strategies are used: add signal lines from the adder to the backplane where there are adequate probe access points; probe directly onto the module etch or IC pins; and subsume the adder in a function whose inputs and outputs can be more easily controlled and observed. The problems of observation and control exist at all levels of integration. Examples of observation points at each level are given in Table Obslevels for the 11/60 computer.

The problem of testability must be addressed at design time. Providing access for testing always incurs added product cost (extra logic and module pins or circuit pins), but lowers manufacturing cost and field service costs. As gate density per chip continues to increase, the problem worsens. One solution, which is economical in i/o connections, is to design every storage element as a shift register which can be loaded in parallel (normal mode) or serially loaded (with an invoking state) or serially read (with the state to be observed). Eichelberger and Williams [1977] report on such a scheme for gate array designs. The individual shift register latches are connected to form one or more independent shift registers which are connected to the leads of the gate array package. At the module level, An example of using shift registers to conserve I/O pins is found in the VAX 11/780. Eight test points are fed to a shift register whose serial output is fed to a module pin. Twenty modules are then chained. The console subsystem, an LSI11, controls the reading of the shift registers and interperts the results.

In Fig. QVstation, the function being tested is a complete CPU which is packaged on several printed circuit modules. Behavior is being induced by PDP-11 programs and observed by inspect~~ing~~ion of the results in memory manually or by ~~using~~ a diagnostics program ~~or manually~~. However, a lower level of behavior can be observed (on the special display panel at the right) and controlled (by varying the clock rate of the CPU). The lights on the display panel are driven from backplane signals and show the contents of certain registers, e.g., micro-instruction register, program status register, and the ALU output register.

Table Obslevels: Examples of Observation Points at Each Structural Level for the PDP11/60 Computer

Level in computer hierarchy	**Observation point**	**Stage in manufacture of computer**	**Example**
electrical circuit	transistor contacts on metallization layer	semiconductor fabrication	wafer test with micro-probe
switching circuit	leads on I.C. package	incoming inspection of ICs	IC tester
register transfer	etch run	module	probe on PC board (module-specific test using GR tester)
register transfer	backplane	module	2224 memory

			exerciser for cache
PMS (Pc)	Unibus	CPU	Unibus voltage margin tester
PMS (Pc)	contents of memory	CPU	diagnostic programs at subsystem level, e.g., memory-management unit, or processor instruction set tests
PMS (C)	contents of memory	System integration	peripheral diagnostic programs
PMS (C)	Unibus	System integration	~~DECX-11~~ bus exerciser

In both Figs. QVstation and FA&T, the entire instruction set is tested; a central computer system loads the diagnostic programs and monitors their execution. Several hundred lines emanate from this system to all of the computers under test throughout the production line. The Unibus of the processor under test serves as the umbilical cord [Chapter 00, p. 00] to accept the diagnostic programs and to be monitored by making memory observations.

Modules are tested using several methods. One method (see step ~~a~~A of Fig. Process60), uses the GR tester of Figure GR. Here, every signal input and test point on the module is probed using a fixed "bed of nails" test probe. The tester then selects the desired input and test point. In another method, (see step ~~b~~B, Fig. Process60) the module under test is placed in a working processor to be verified.

Doane
Gale
Gonzales
Pey
Scanl
RES
Tender
Wld

Smith
Schelke

Following the inter-plant transfer, systems integration begins. Figure FA&T shows an 11/60 at the Westminster Plant. For some computer systems, Final Assembly and Test (FA&T) is carried out by having an individual technician follow a system through each of the three phases: incoming test of the CPU cabinet assembly, integration of peripherals, and final acceptance. Thus one or more technicians have individual responsibility for fabricating a single system in what is a traditional, hand-crafted, job-shop fashion.

Alternatively, FA&T is carried out in what is fundamentally a continuous, production line environment called the Common Systems Integration (CSI) line. Here, workstations exist at each stage of the production line, and are interconnected by conveyors.

~~Acknowledgements~~: Dave Widder, Lorin Gale, Hank Sch~~a~~elke, Memorex Corporation, Bob Peyton, Russ Doane, Steve Teicher, Joe Smith, Dick Gonzales.

"Statistical Quality Control", Grant and Leavenworth, Fourth Edition, McGraw-Hill, 1972.

"Quality Control Handbook", Juran, J. M., McGraw-Hill 1951; Second Edition, 1962.

Louise please insert in order in ~~following~~ list of references ~~[illegible]~~ below.

Acknowledgements

We gratefully acknowledge the following colleagues ~~valuable~~ who provided data for this chapter and ~~[illegible]~~ critiques of earlier drafts: Russ Doane, Rony Elia-Shaoul, Lorrin Gale, Dick Gonzales, Bob Peyton, Jim Scanlan, Henk Schalke, Joe Smith, Steve Teicher, and Dave Widder.

~~Memo~~ We ~~are~~ thank Memorex Corporation for ~~permission~~ ~~Memorex Corporation~~ ~~to use~~ FigureS Memtree, Area.Digital, Large.Disk.Price & Mem trends. [Heidi: does this go here or in front matter?]

Bhandarkar, D. P. and Juliussen, The Impact of Semiconductor Technology on Computer Systems.

Braeckelmann, W., Fritzsche, H., Kroos, F-K., Trinke, W., and Wilhelm, W. "A Masterslice LSI for Subnanosecond Random Logic", Digest of Technical Papers, 1977 Ent. Solid-State Circuits Conference, pp. 108-109.

Denning, P. J. "The Working Set Model for Program Behavior", Communications of the ACM, 11, May 1968.

Denning, P.J. "Virtual Memory", Computing Surveys, Sept. 1970, pp 153-189.

Eichelberger, E.B. and Williams, T. W. "A Logic Design Structure for LSI Testability", Proceedings of the 14th Design Automation Conference; June 20-22, 1977.

Fusfeld, A. R. The Technological Progress Function Technology Review, Feb. 1973, pp. 29-38

Gaskill, J.R., J. H. Flint, R. G. Meyer, L. J. Micheel and L. R. Weill, "Modular Single-Stage Universal Logic Gate, "IEEE Journal of Solid-State Circuits, SC-11, 4, p. 529-538, 1976.

Hodges, D.A. A Review and Projection of Semiconductor Components for Digital Storage, Proc IEEE, 63, 8, August 1975, pp 1136 - 1147.

Hodges, D. A., Progress in Electronic Technologies for Computers, National

Bureau of Standards Report T73219, March, 1977.

Kilburn, T., D. L. G. Edwards, M. J. Lanigan, and F. H. Sumner, "One-Level Storage System," IRE Trans., Vol. EC-11, No. 2, pp. 223-235, April 1962.

Landman, B. S. and Russo, R. L.
"On a Pin Versus Block Relationship for Partioning of Logic Graphs", IEEE Transactions on Computers, C-20, 12, Dec 1971, pp 1469-1479.

Liptay, J. S., Structural Aspects of the System/360 Model 85 - The Cache, IBM Systems Journal, 7, 1 (1968).

Logue, J. C., N. F. Brickman, F. Howley, J. W. Jones, and W. W. Wu, "Hardware Implementation of a Small System in Programmable Logic Arrays," pp. 110-119, March 1975.

Luecke, J. Overview of Semiconductor Technology Trends; IEEE Computer Society International Conference, Fall, 1976, pp. 52-55.

Masaki, A., Harada, Y., Chiba, T.
"200 Gate ECL Masterslice LSI",
ISSCC Digest of Technical Papers, pp. 62-63, Feb. 1974.

Moore, G. E., Microprocessors and Integrated Electronic Technology, Proc. IEEE, 64, 6, June 1976, pp. 837-84.

Nakano, T., O. Tomisawa, K. Anami, M. Ohmore, I. Okkura and M. Nakaya, "A 920 Gate Masterslice", Digest of Technical Papers, IEEE Solid-State Circuits Conference, 1978, pp. 64-65.

Noyce, R. N. "Large Scale Integration: What is Yet to Come?", Science, Volume 195, Number 4283, 18 March 1977, pages 1102-1106.

Noyce, R. N. "Microelectronics", Scientific American, Volume 237, Number 3, September 1977, pages 62-69.

Parke, N. G., Private Communication, 1978.

Patil, S.S. and Welch, T. An approach to using VLSI in digital systems. 5th Annual Symposium on Computer Architecture, April 1978, pp ~~[illegible]~~ 139-143

Phister, M. "Data Processing Technology and Economics", Santa Monica Publishing Co. Santa Monica, Calif. , 1974.

Russo, R. L. "On the Tradeoff Between Logic Performance and Circuit to Pin Ratios for LSI", IEEE Transactions on Computers C-21, 2, February 1971, pp 147-152.

Strecker, W.D. "Optimal Design of Memory Hierarchies", Proceedings of the Hawaii Systems Conference, 1978.

check → Toom Private Communication, 1977

Turn, Rein, Computers in the 1980s, Columbia University Press, N. Y., 1974.

vonHippel, E. The Dominant Role of the User in Semiconductor and Electronic Subassembly Process Innovation; IEEE Transactions on Engineering Management, vol. EM-24, No. 2, May 1977. pp. 60-71.

Chapter 2 Figures and Tables (that are to be made as figures)

Memtree	Family tree of memory/logic technology (courtesy of Memorex Corporation, Puthuff, 1977)
LMtimeline	Logic and memory technology evolution time line
ICtree	Family tree of digital Integrated Circuit functions
ALU	Arithmetic-Logic Unit logic diagram
ALUFCN	74181 Function table
AMD	AMD 2900 4-bit microprocessor slice block diagram (registers and data path)
Semiden	Component (transistor) per single Integrated Circuit die versus time
Compprice	Price per gate versus time
Gatecomp (table in text)	Circuit components (transistors) required for various logic functions in various technologies -- show also rom, ram, pla, ccd, flip flops, gates 2900, LSI-11, CMOS-8; includes N, P, IIL, CMOS, ECL< TTL, TTL/S
Memprice	Cost per bit of Integrated Circuit memory versus time (Noyce-Sci.-Am.)

~~Generality~~	~~Generality (use) and price of an Integrated Circuit versus its complexity~~
Reliab	Failure rate of silicon Integrated Circuits
Design.cost	Current design cost (or time) versus circuit density using various design methods
Semiuse	Manufacturing cost versus LSI circuit density for various design techniques
PLA	Signetics field programmable logic array (FPLA)
Memsizeperf	Memory size versus access time for various memories and yearly availability (Toome's data, TI)
Core.cost	Cost per bit of core memory estimated by various market surveys and future predications
Areal.digital	Areal density of various digital magnetic recording media (courtesy of Memorex Corp., 1978)
Areal.analog	~~Areal density of various analog magnetic recording media (courtesy of Memorex Corp., 1978)~~
Large.disk.price	Price per bit of large, moving head disks and semiconductor

	memories (courtesy of Memorex Corp., 1978)
Mem.trends	Memory trend 1975-1980 (courtesy of Memorex Corp., 1978)
Tapechar	Characteristics of various IBM magnetic tape units versus time
x ~~Tapeprice~~	Price of magnetic tape units (1978)
~~CRTprice~~	~~Price of various cathode ray tube terminals versus time~~
CPI	Consumer Price Index using 1967 as base
~~CPI.log~~	~~Consumer Price Index (plotted on semi-logarithm graph) using 1967 as base~~
Mktcy~~cl~~e Turbo	Technology progress functions for turbojet engines
~~Perf.size~~	~~Performance or size vs price at some time, t~~
Cost vs time	~~Cost of~~
Techvstime	
Cost/tech vs time	

~~Tradeoff System performance constrained by either memory size or performance~~

Product.cost Product cost versus time on the state-of-the-art line showing various situations

Net Cost of various components that contribute to product cost

Product.price Product cost versus time with manufacturing learning

Product.improve Product cost improvement by enhancement of cost/function

~~Note there is a reference to photos elsewhere in the book to illustrate the various packaging concepts.~~

~~Fig. 1. Packaging consists of placing boxes within smaller boxes...within boxes on an indefinite (not recursive) basis.~~

~~Fig. 2. A packaged system provides some function, but in addition gives a visual impression, usually requires cooling of some sort, has certain mechanical characteristics and an Electromagnetic Interface (EMT).~~

Fig. 3. Eight level packaging hierarchy for DEC second to fourth generation computer systems.

Fig. 4. Packaging is a set of closely interrelated design problems.

~~VAX.11/780 View of VAX 11/780 cabinet~~

11/05 PDP-11/05 (first version) Computer box showing modules, backplane (holding 9 hex modules), power supply, and fans.

11/70.1 Major components and assemblies of 11/70 processor and memory mounted in two standard DEC cabinets.

~~11/70.2 11/70 processor with maintenance cards~~

BP Printed circuit backplane cross-section

~~Pwrcable Power distribution cable harness for 11/70~~

~~Lab.sys DEC's first laboratory and systems modules~~

LSI11 LSI11 processor with 8 Kbytes of memory and microcode for commercial instruction set.

~~Fig. 5. Deleted~~

Fig. 6. Time line evolution of computer generations, packaging and classes (with examples).

~~Fig. 7. None~~

~~Fig. 8. None~~

Fig. 9. ~~4706 and~~ 4707 ~~receiver and~~ transmitter line unit~~s~~ of the late second generation.

~~(Get two modules from Mary Jane to photograph)~~

~~Fig. 10. DL11 line unit based on early fourth generation IC. (Need a photo of this module, Mary Jane is working on it.~~

~~Fig. 11. None~~

Fig. 12. DEC physical structure (packaging) hierarchies vs technology generations.

~~Fig. 13. Three views of PDP-8/A box.~~

Fig cab (from next page)

Fig. P1 a-c Cost, weight, and volumetric efficiencies versus time for various DEC computer power supplies

a) Cost efficiency (in relative cost/watt)

b) Weight efficiency (in watts/lb)

c) Volumetric efficiency (in watts/cubic inch)

Fig. P2 a-c Cost, weight, and volumetric efficiencies versus size for various DEC computer supplies

a) cost and efficiency (in relative cost/watt)

b) Weight efficiency (in watts/lb)

c) Volumetric efficiency (in watts/cubic inch)

Fig. P3 Power supply efficiency (watts out/watts in) versus time for various DEC computer power supplies.

Fig. Mod.size Module printed circuit board area and number of pins per module versus time for DEC modules.

Fig. Mod.cost Relative cost per interconnection on DEC printed circuit board modules versus time.

Fig. Backplane.cost Relative cost per possible and actual interconnection versus time for various DEC computer backplanes; also, pin density (in pins per square inch) versus time.

Fig. Cab Cabinets used to hold various DEC computers (in fixed, book, and box configurations)

Fig. Boxes Boxes used to hold various DEC PDP8 and PDP11 series minicomputers

Fig. Heat.den Heat density (Kw/ft3) of various DEC computer boxes

Fig. Space.util Space utilization (ft3 of modules/ft3) of various DEC computer boxes

Fig. Cost.pay Cost payload (relative cost of materials/ft3 of modules held) of various DEC computer boxes

Fig. Box.cost Relative cost of materials versus number of hex size modules for various DEC computer boxes

Figure	
Lifecycle	A simplified process flow for the major phases and milestones in the life of a product.
~~*Breadboard~~	
Manufover	Overview of manufacturing flow.
process60	The process flow for the 11/60 manufacturing plant in Aguadilla, P.R.
~~Bath~~	Chip failure rate plotted against time shows the three behavior regions. Thermal cycling is used to ensure that "infant mortality" cases in the first region are discovered early in the manufacturing flow.
*Tchamber	Thermal chambers. (photo)
*2224	2224 memory exerciser used for testing cache. (photo)
*QVstation	(photo)
*GR~~tester~~	(photo)
Obs	Placement of test points for observation and control.
*APT	Final acceptance prior to inter-plant transfer. (photo)
CSI	The common systems integration area at the Westminster Plant. (photo) -- From article in Materials Handling Engineering, July 1977
*spurs	(photo) [need new photo giving better view of system being built]
*Waferprobe	Probing test points on a wafer prior to dicing. (photo)

*These are photos -- copies of Polaroid photos are attached for allbut Breadboard and Waferprobe. CM is getting B/W glossies retaken in Aguadilla and Breadboard from the mill.

FA&T	(photo)

~~Index~~

Second edit

5/12/78

CHAPTER 2: TECHNOLOGY, PACKAGING AND MANUFACTURING

It is customary when reviewing the history of various industries to ascribe the events therein to either market pull or technology push. The history of the auto industry contains many good examples of market pull, such as the trends toward large cars, small cars, tail fins, and hood ornaments. The history of the computer industry, on the other hand, is almost solely one of technology push. Whereas the rest of this book gives examples of the effects of technology push, this chapter explores the individual elements that have made up the technology push.

Semiconductors have been the dominant factor in technology push within the computer industry, but magnetic recording density on disks and tapes has evolved rapidly too, and must be understood as a component of cost and a limit of system performance. Communications links have been left out of the discussion because during the past twenty years they have not evolved as rapidly as other technologies, nor have they been a major influencing factor on computer system design. The packaging (cabinets and boxes), interconnection, and power aspects are discussed, principally because these do not represent pushes but rather are large, high inertia objects that have to be ~~pushed~~ leaned against in the hope that they will eventually yield.

The semiconductor portion of the discussion begins by presenting a family tree of the possible technologies arranged according to the function they carry out and showing how these have evolved over the last two or three generations to affect computer engineering. The cost, density, performance, and reliability parameters are briefly reviewed, and then the application of semiconductors, using various logical design methods, is discussed with particular emphasis on how the semiconductor technology has pushed the design methods.

The discussion of ICs is from the user viewpoint because, until recently, DEC has always bought its ICs from the semiconductor manufacturers, and its engineers (users of ICs) have viewed the IC as a black box with a carefully defined set of electrical and functional parameters. Most design engineers will probably continue to hold that view (and be encouraged to do so), even though some ICs will be supplied by an in-house design and manufacturing facility. The advantages and disadvantages of intra-IC design will be discussed later in the chapter.

The discussion of the use of semiconductors in logic applications is followed by a section on memories for primary, secondary, and tertiary storage. The memory section is followed by a section containing some general observations about technology evolution: how technology is measured, why it evolves (or doesn't), cases of it being overthrown, and a general model for how its use in computers operates and is managed.

Packaging and interconnection are the physical structure by which computers are actually formed. The discussion in the packaging section builds on two of the models from Chapter 1: view 1 (structural levels) and view 3 (packaging levels of integration). The chapter concludes with a section that briefly describes the manufacturing process.

Semiconductor Logic and Memory

A single transistor circuit performing a primitive logic function within an integrated circuit (IC) is among the smallest, most complex of man-made objects. Alone, such a circuit is intrinsically trivial, but the fabrication process ~~for a set of structures~~ to form a complete integrated circuit is intricate a complex. To the digital IC users there are several relevant parameters:

1. The function an individual circuit performs within the IC, the aggregate function of the IC, and the functions that a complete IC family (such as the 7400-series) performs.

2. The number of primitive digital switching circuit functions per IC. This density is a measure of the capability of the IC process and the ingenuity of the designers.

3. Cost.

4. The speed of each circuit and the speed of the aggregate IC and set of ICs within a family. The semiconductor device family (Transistor-Transistor Logic = TTL, Schottky TTL = TTL/S, Emitter Coupled Logic = ECL, Metal Oxide Semiconductor = MOS) usually determines this performance.

5. The number of interconnections (pins) to communicate outside the IC.

6. The reliability. This parameter is a function of the ~~circuit~~ device technology,

the density, the number of pins, the operating temperature, the use (or misuse), and the maturity (experience) of the manufacturing process.

7. Power consumption and speed-power product ~~power/speed index~~. ~~The power/speed index is the power per switching element divided by the switching speed. Another~~ A frequently used metric is the "speed-power product", where the delay through a typical gate is multiplied by the power consumption of the gate. For a particular technology, the speed power product tends to be constant because ~~technologies that offer~~ short gate delays usually feature high power consumption. A technical advance that lowers the speed power product is considered noteworthy.

Figure ICtree shows a family tree (taxonomy) of the most common digital IC's. The tree is arranged such that the least complex functions are in the upper portion of the figure and the most complex are at the bottom of the figure. In addition, the tree is arranged to order the circuits by generation, starting with the second generation at ~~along~~ the left hand edge and progressing to the fifth generation ~~along the right hand edge~~. The circuits are clustered roughly by the regularity of the function being implemented, and whether there is memory associated with the function. The concept of circuit regularity is important in large scale integrated circuits because it is desirable to implement regular structures to minimize area-consuming interconnections and thus ~~to~~ simplify layout, simplify understanding, ~~and~~ aid testing, and reduce power.

As indicated in Figure ICtree, the branching of the IC family tree began in earnest at the beginning of the third generation. At that time, integrated circuit technology advances permitted collections of basic logic primitives (AND, NAND, etc.) and sequential circuit components (flip flops, registers, etc.) to occupy a single IC rather than an entire module. This had the benefit of providing a drastic reduction in size between the second and third generation computer designs, as shown most vividly by comparing the PDP-9 and PDP-15 (page 00), but had the drawback that modules now contained a wide variety of functions and were thus specialized.

As the densities began to improve to a hundred gates, the construction of complete arithmetic units on a single chip became possible. The earliest and most famous function, the 74181 arithmetic logic unit (ALU) (Fig. ALU, and Fig. ALUFCN) provided up to 32 functions of two 4-bit variables. By the fourth generation, it became possible to construct very large combinational circuits, such as a complete 16 x 16-bit multiplication circuit (the TRW multiplier) requiring about 5000 gates, on a single chip.

Progress during the fourth and fifth generations has not been without its problems, however. Without well-defined functions such as addition and multiplication, semiconductor suppliers cannot provide high density products in high volume because there are few large scale, general purpose universal functions. The alternative for the users is to interconnect simple logic circuits (AND gates, flip flops), but this does not permit efficient use of the technology and the cost per function remains high (about that of the third generation) because the printed circuit board and IC packaging costs (pins) limit the attendant cost reduction.

To address these problems, two methods of effectively customizing LSI logic are included in Fig ICtree and discussed in greater detail later in the chapter. These are the PLA and the gate array. The PLA (Programmable Logic Array) is an array, of AND-OR gates that can be interconnected to form the sum of products terms in a combinat~~ori~~ional design. Gate arrays are simply a large number of gates placed on the chip in fixed locations where they can be interconnected during the final metalization stages of semiconductor manufacture. ~~Gate arrays have been used extensively for several years by IBM, and Amdahl, and others for their biggest machines, but may soon appear in smaller computers.~~

There is a special branch of the tree shown in Figure ICtree for the purely memory functions. Memory is used in the processor as conventional memory, but can also be used as an alternative to conventional logic for performing combinational logica~~l~~ functions. For example, the inputs to a combinat~~or~~ionial function can be used as an address, and the output obtained by reading the contents of that address. Memory can also be used to implement sequential logic functions. For example, the memory can be used to hold state information for a microprogram. Since memories have so many fundamental uses, this branch is discussed separately in the memory section.

The remainder of the interesting ~~logical~~ functions include combinations both ~~of~~ logic and memory. There are various special functions such as ~~LPC (~~& Linear Predictive Coding~~)~~ & ~~encoding~~ algorithms for use in real time applications and data encryption algorithms for use in communication systems. One of the most useful communications functions, and the first one to use LSI, was the UART

(Universal Asynchronous Receiver Transmitter).

There is a special branch for bit-slice components that ~~can be combined together~~ are concatenated to form datapaths of arbitrary width. These are being used to construct most of today's high-speed digital systems, mid-range computers, and computer peripheral~~s~~ controllers. Although there have been several bit-slice families, the AMD 2900 series, whose register-transfer diagram~~s~~ ~~are~~ is shown in Figure AMD, has become the most widely used. Note that all the primitives of this series were present in the RTM family (Chapter 19), including the microprogrammed control unit referred to as the K(PCS) (Programmed Control Sequencer).

The final branch of the tree is the most complex, and is used to mark the fourth (microprocessor-on-a-chip) generation of technology and the beginning of the fifth (computer-on-a-chip) generation. The fourth generation is marked by a complete processor being packaged on a single silicon die, and the fifth generation, by this measure, has already begun since a complete computer (processor with memory) now occupies a single die. The evolution ~~in~~ along this branch ~~complexity during each generation simply permits larger~~ results in either longer word length processors or richer computers. ~~to be placed on one chip.~~ At the beginning of the fourth generation, a 4-bit processor was the benchmark, whereas toward the end of the fourth generation, either a complete 16-bit processor, ~~such as the PDP-11~~ or an ~~8~~ 8-bit processor with RAM and, could be placed on a chip.

Gates per Chip

The function performed by a chip is clearly dependent on the number of gates that can be placed on a chip. Thus, density in gates/chip is the single most important parameter determining chip functionality. From this measure, one can predict the functions likely to be implemented by just following the tree. It should be noted, that the whole tree is relatively alive and has dense areas of new branches everywhere except at the top, where unconnected gate and register structures have been relatively static. In the growing areas, as density increases sufficiently, a new branch grows. For example, the processor-on-a-chip started out as a 4-bit processor ~~(or rather as 2 chips for a single processor)~~ and then progressed to have 8-bit and 16-bit processors on a single chip. Similar effects can be observed with the arithmetic logic unit and with memories.

The number of gate circuits per chip not only determines chip functionality, it also is the measure of density as seen by a user (see Fig. Semiden). This metric is actually the product of the circuit area and the number of circuits per unit area. Progress in lithography has le~~a~~d to reduced conductor linewidths ~~and~~ with a corresponding reduction ~~of~~ in circuit size ~~to yield~~ and hence higher speeds and higher densities. Linewidths have decreased from 10 microns in early LSI chips to 6 microns in the LSI11 chips, and more recently to ~~3 or~~ 4 microns in Intel's 8086. Linewidths of less than a micron have been achieved at the research level, but require electron beam techniques instead of present photographic methods. The processing techniques to create the semiconductor materials have also been improved to have better manufacturing yields (and lower costs). Circuit and device innovation (such as reducing the number of transistors per memory cell) have also contributed to density and yield increases.

The model given in Fig. Semidens is exponential and correlates with previous observations that the number of bits per chip for a MOS technology memory doubles every two years according to the relationship:

$$\text{number of bits per chip} = 2^{t-1962}$$

There are separate curves, each following this relationship, for ROMs in prototype quantities, ROMs in production quantities, RAMs in prototype quantities, and RAMs in production quantities. Thus, depending upon the product and the maturity of its production process, one must scale the state-of-the-art line appropriately by one to three years according to the following rules:

bipolar read-write memories lag by two to three years

bipolar read-only memories lag by about one year

MOS read-only memories lead by one year

This gives the availability of various sizes of semiconductor memories as shown in Fig. LMtimeline.

The significance of these values is that they determine (technology push) when ~~where~~ certain architectures and implementations can occur. The chapter critiquing the PDP-11 uses this model to show how semiconductors accomplish this push.

Cost

The most important characteristic of ICs after density is cost. The cost of

ICs is probably the hardest of all the parameters to identify and predict because it is set by a complex marketplace. For circuits that have been in production for some time and for memory arrays, the price is essentially that of a commodity like eggs or bacon, and users generally consider these ICs as very similar to commodities, with the attendant benefits (cost) and problems (having a sufficient source of supply). In low volumes, IC prices are proportional to the die cost (which is proportional to the die area), but at higher volumes, assembly, testing, packaging, and distribution become the dominant cost factors. Furthermore, for those low volume circuits that have not yet reached commodity status, the prices also depend on the strategy of the supplier, e.g., whether he is willing to encourage competition.

Two curves are presented to reflect the price of various components (transistors) implemented in integrated circuits. Figure Compprice shows the price per gate for MOS and TTL circuits as a function of time and scale of integration (SSI, MSI, LSI). Table GateComp gives some idea of how circuit density (in elements) relates to actual function.

Insert α_{10}

The cost history of ICs is reflected very dramatically in the cost history of a special class of ICs, semiconductor memory. The semiconductor memory cost curves, given in Figure Memprice, are also interesting because of the important role of memory in past and future computer structures. As shown in the figure, the 1978 cost per bit was roughly .08 and .07 cents per bit for the 4 Kbit and 16 Kbit IC chips respectively, giving IC costs of $3.30 and $11.50 respectively. Two factors are at work to create the cost figures for an IC; these are density in bits per IC and cost per bit. The two factors have

No. of elements per function
(transistors for MOS, transistors + resistors for bipolar)

function	MOS			BIPOLAR		
	N	P	C MOS	ECL	TTL	I^2L
INVERTER	2	2	2	7	3	1
2 INPUT GATE	3	3	3 or 4	8	3	1
8 INPUTS GATE	9	9	9 or 16	14	3	2
R/S LATCH	6	6	6 or 8	12	6	2
MEMORY CELL (DYNAMIC)	2	2	2	-	-	2
MEMORY CELL (STATIC)	6	6	6	4-6	4-6	4
D ~~F.F.~~ Flip Flop	20	20	20 or 28	26	20	9
JK " "	20	20	20 or 36	-	26	11

Table Gatecomp The number of elements to implement various functions in different technologies

(~~from R.E.S. memo~~ 3/22/78)

not had equal influence in reducing costs because while the chip density has improved by a factor of two each (Fig. Semiden) year according to Moore [Noyce, 1977], the cost per bit (at the IC level) has declined by only a factor of two every two years. The line drawn in Noyce's [1977] Fig. Memprice has the equation:

$$\text{cost/bit (in cents)} = .3 \times .72^{t-1974}$$

It is also interesting that the cost compares favorably with the price decline observed in core memory over the period since 1960-1970 for the 18-bit computers, (page 00) and for the cost declines in both the PDP-11 and the PDP-8 (pages 00 and 00).

Performance

The performance for each semiconductor technology evolves at different rates depending on the cumulative learning associated with design and manufacturing process together with the marketplace pressure to have higher performance for the particular technology. One may hypothesize that each technology can be looked at as being relatively appealing or relevant to the particular design(er) styles associated with the computer market levels (view 4, page 00). One would expect the evolution to continue along the lines shown in the following table for the period around 1980.

Table SemChar: Characteristics of Dominant (1978) Semiconductor Technologies

Type	Evolution	Use

TTL (Transistor-	TTL	logic, bus interfacing
Transistor	TTL/Schottky	higher speed than plain TTL
Logic)	TTL/LS	same speed as TTL, but low power
ECL (Emitter	MECL II, III	very high performance
Coupled	MECL 10K, 100K	easier to work with
Logic		evolving to larger gate arrays
MOS (Metal-Oxide-	p-channel	low cost
Semiconductor	n-channel	greater densities, cost
		evolving to performance (memory)
		evolving to shorter channels (HMOS, DMOS, VMOS)
CMOS (Complementary MOS)		low power, higher speed, better noise immunity

DEC's use of the various IC technologies shown in Table SemChar is probably typical of most of the computer industry: TTL for ~~mid- and high-sized~~ minicomputers; ECL for the larger scale machines ~~(DECsystem 10)~~; MOS for memories, microprocessors, and specialized high density circuits; and CMOS for special microcomputers, especially those intended for battery operation.

Some of the lesser used technologies such as I^2L (Integrated-Injection Logic) and SOS (Silicon on Saphire) have been omitted. I^2L features high density and very low power consumption, but is slow. SOS MOS enhances CMOS speed by removing stray capacitance, making it comparable with TTL/LS speed, while retaining MOS ~~gate~~ density ~~complexity~~ capabilities. Both I^2L and SOS have been touted as replacements for various technologies shown in the table. But, if an entrenched technology has evolved for some time and continues to evolve, it is difficult for alternative technologies to displace it because of the cummulative investment in process technology and understanding. Semiconductors appear to be characteristic of other technologies in that usually only a single technology is applied to ~~used for~~ a given problem.

The early technologies, RTL (Resistor-Transistor Logic), TRL (Transistor-Resistor Logic), and DTL (Diode-Transistor Logic) have also been omitted. These technologies are important historically, because they were used in the first ICs. However, many manufacturers, including DEC, did not use them in computers (RTL was used in DEC K-Series modules) because they did not represent a sufficient advance over the discrete transistor circuits already being used. In addition, early circuits were packaged in flat packages and metal cans rather then in the dual in-line package used today, and automated manufacture using the components was thus not economically feasible.

Table Gatedelay gives the speed-power product and gate delay which are the two most useful measures of performance for the various technologies as they have evolved with time. The speed-power product metric ~~for a technology at a given time indicates~~ suggests that the user may tradeoff performance against power. There are limits to this tradeoff. Only about one watt can be dissipated by a traditional off-the-shelf IC package, (and tradition in IC package design has been strong). The table was formulated by Jerry Luecke of Texas Instruments (TI) at a time when I^2L technology had just been introduced (Oct. 1975) by TI:

Table Gatedelay: Gate Delay of Various Semiconductor Technologies [Luecke, 1976:53]

Year	Type of Logic	nano tp (nsec)	P_D ~~(Watts)~~ (milliwatts)	Speed-Power Product (Picojoules)
[1963	DTL	-	-	200]
[1964	RTL			180]

1965	TTL	10	10	100	
1967	TTL/H-series	5	20	100	
1968	TTL	30	1	30	
1970	TTL(Schottky)	3	20	60	
1972	TTL(low-power Schottky)	10	2	20	
1967	ECL	2	30	60	~~.9 1 .9~~
1974	ECL	0.7	43	30	
1970	PMOS	200	0.1	20	
1973	NMOS	100	0.1	10	
1973	CMOS	30	1.0	30	
1974	SOS	15	0.05	7.5	
[1976	NMOS	4	1	4]	
[1978	HMOS	~~x~~ 0·9	~~x~~ 1	~~x~~] 0·9	
1975	I^2L	35	0.085	3.0	
1976	I^2L	20	0.05	1.0	

Reliability

Over the past 15 years, the failure rate for standard ICs has been reduced by two orders of magnitude to the neighborhood of .01% per 1000 hours. This corresponds to 10^7 hours (about a millenium) mean time to failure (MTTF) per component. Figure Reliab, from a recent survey article by Hodges [1977b] shows the trend. The lower curves show the higher reliability obtained when more extensive testing and screening are employed. The improved MTTF of between 10^8 and 10^9 are obtained at a cost increase of 4 to 100 times per component.

I/O Connections

The number of pins per ~~#~~ IC package has risen relatively slowly to the point where 48 pins is just become accepted in 1978.

The LSI Dilemma

As indicated in the beginning of the discussion of Figure ICtree, a dilemma ~~involving a search of universal circuits~~ has developed in the manufacture of LSI circuits. The economics of the LSI circuit industry make it essential that IC suppliers produce circuits with a high degree of universality. This is because the learning curve of a manufacturing process causes cost to be inversely proportional to volume, and for a design to be sold in high volume, it must be usable in a large number of applications. However, the trend in circuit complexity, which allows semiconductor manufacturers to put more transistors on a constant die area each year, tends to increase specialization of function, lowering the volume and hence raising the price.

The LSI-product designer is therefore continually in search of universal primitives or building blocks. For a certain class of applications, such as controller applications, the microprocessor is a fine primitive and has been so exploited [Noyce, 1977]. For other applications, circuit complexity can embrace even higher functionality at the PMS level. The Intel 827X is an interesting example here: two processors, a 1.25 microsecond byte-processor and a 250 nanosecond bit-processor are combined in one LSI circuit.

Moore [1976] discusses the LSI dilemma in a paper on the role of the microprocessor in the evolution of microelectronic technology. He points out that a similar situation existed when ICs were first introduced. Users were reluctant to relinquish the design prerogative they had when they built circuits from discrete components. It was not until substantial price reductions were made that the impasse was broken. Then the cost advantages were sufficient to force users to adopt circuits that fit the technology.

~~The first high functionality, high universality circuit that comes to mind is the microprocessor on a chip~~. For many applications, including most computer systems, the microprocessor on a chip is not a cost-effective building block, and other solutions to the dilemma are found ~~used~~. For example, microprogramming is a systematic ~~highly general~~ way of generating control signals for data path elements, and table lookup is a highly general technique. Both methods are attractive because they use memory, an inherently low cost LSI circuit. Microprogramming, however, does have limitations. The extra level of interpretation extracts a performance penalty and some potential datapath parallelism is often given up to reduce cost. A more subtle, but practical, limitation is the development cost of microcode. Assuming the writing rate to be 700 microwords per man year for wide-word, unencoded (horizonal) micro machines, a desire to limit the effort to 20-24 manyears would limit the maximum control store size to about 16 Kwords. This maximum will tend to increase in the future, when the use of better microprogramming tools increases the microcode writing rate beyond the 700 microwords per man year figure given here.

At the RT level, the standard microprogramming design method is (conservatively) twice as expensive per instruction as conventional programming. Moreover, because microinstructions are usually not as powerful as conventional instructions, more microinstructions than conventional instructions are ~~will be~~ required to solve a given problem. These two factors, more expense per instruction and more code ~~instructions~~, ~~will~~ cause a microprogram to be 5 to 10 times as expensive as a conventional program to solve the same problem. ~~The payoff is that the internal speeds of a microprogrammed~~ microlevel machine ~~contro controller are at least a factor of 10 faster than a conventional mini.~~ However, microinstructions execute at least ten times faster.

The characteristics of microprocessor and ROM methods of tailoring ~~creating customized results from~~ universal LSI circuits are summarized, along with the characteristics of a number of other methods, in Table FunVar.

Table funvar

Building block	Technique for varying function	Degree of generality	Permanence of change
Computer module	program	v. high	none
microprocessor	program	high	low to medium
bit slice	microprogram	medium	medium
ROM	factory mask change	v. high	irreversible
PROM	field change	v. high	irreversible
EAROM	field change	v. high	low
PLA	factory mask change	medium	irreversible
FPLA	field change	medium	irreversible
gate array	factory mask change	medium	irreversible
RAM	write	v. high	none

The increased basic circuit functionality available at each new generation has not only been an important part of semiconductor design, but has also caused design methods to change with the generations. This book provides an examples, as summarized in the following table.

Table: Design Method versus Generation

Design Method:\Generation:	First	Second	Third	Fourth	Fifth	Examples in this book

Combinational and sequential; use of "standard" modules, IC's.	s	s	s			18-bit;PDP-8
Read only memory and PLA; microprogramming			s	m		PDP-9;PDP-11
Microprogramming with standard RT elements (high perf.); minor logical design				s	m	CMU-11
Programming using micros and logic for interfaces	p	p		s	x	LSI-11
PMS-level ~~PMS~~ design using completely specified and pre-designed microcomputer components					s	Cm*
Customized chip design and standard logical design (high performance)		m		m	m	LSI-11

s - the standard method for most digital systems
m - done by manufacturers of ~~basic equipment~~ integrated circuit
x - also used
p - prelude to micros, also done using minis

The design of most relatively high speed digital systems (including ~~low- to~~ std mid-range minicomputers) is carried out using standard register transfer ICs complete with data path and memory.

For higher performance computers, there is no alternative to using either tightly packed standard ICs or building a unique set of ICs using some form of customization. The high performance IBM and Amdahl machines, for example, use custom ECL circuits or gate arrays to improve packaging. Although Seymour Cray continues to build his high speed computers (the CDC 6600, 7600 and Cray 1) with no custom logic, he does so by using impressively dense modules with high density interconnection and freon cooling.

The current spectrum of IC's and their use is summarized in Table ICspectrum.

Table ICspectrum: A Spectrum of IC ~~Organization and Use~~ application in Various Computers

Application ~~Organization~~ ←	Technology ←	Unique Chips ←	Performance (MIPS) ←	Cost	Examples
Microcomputer	MOS (VLSI)	1	0.1	Lowest	Intel 8048, MOSTEK 3870
Microprocessor	MOS	1			Intel 8080, Zilog Z80, Motorola 6800
Microprocessor	MOS	2-4			DEC LSI-11, Fairchild F-8
Microprocessor	MOS	> 4			Burroughs B80, National IMP 16
Bit slice (micro-programmed)	TTL	few			DEC 11/34 Floating Point Unit
Gate array	TLL	most			Raytheon RP16, IBM Series 1
MSI	TTL	few			DEC VAX 11/780, 11/70, HP 3000
Gate array	ECL	all			IBM 370/168, Amdahl 470/v6
SSI	ECL (SSI)	std.	80	highest	CDC 7600, CRAY 1

The Changing Nature of System Design

With the advent of the processor on a chip, digital system design has been, or soon will be, converted completely to computer system design (PMS-level design). Problems such as controlling a CRT, controlling a lathe, building a billing machine, or implementing a word processing system become computer system design problems similar to those attacked over the first three generations. The hardware part of the design, the interface to the particular equipment, is straightforward. The major part of the design is the programming. Since the late 40's three complete generations have learned about computer design, especially programming. The first generation discovered and wrote about it. Then it was rediscovered and applied to minicomputer systems. This time, it is being learned by everyone who must use ~~and program the~~ a microcomputer. Each time, for each individual or organization, the story is about the same: people start off by programming (using binary, octal or hexadecimal codes) small tasks, using no structure or method of synchronizing the various multiple processes; the interrupt mechanism is learned, and the symbolic assembler is employed; and finally some more structured system--possibly an operating system is employed. Occasionally users move to higher level languages or macro assemblers.

In view of this cyclical history, it seems likely that current digital systems design practice, which consists of building simple hardware interfaces to relatively poorly defined busses together with programming the application, will be relatively short lived. The design method of the future will ~~be at~~ be with ~~the~~ PMS-level components, although at the moment it is still too difficult to be done reliably and cheaply by large numbers of engineers. The components from which the microcomputer systems will be formed will be significantly more advanced; they will use much better packaging, clearly defined busses, more general interfaces, and ~~base level~~ rudimentary operating systems, ~~The opera~~ ~~that are embedded in~~ The latt ~~hardware~~ by being placed in read only memory, ~~to~~ will give ~~the~~ a feeling of permanency so that users are less likely to embark on the expensive, unreliable rediscovery path. Standard components will be built which can be interfaced to a wide range of external systems using parameters that are specified by a field programming method instead of using logical design and building with interconnection on modules. In this way, the complexity of individual ICs can be increased and having a standard method for interconnection, higher volume and lower costs will result.

Design Costs Versus Unit Costs

Before discussing the alternatives associated with IC design, it is important to characterize the various costs. Figure Design.cost shows, at a crude level, what one might expect the relative design costs to be for various inter- and intra-IC design methods. Even the design cost is highly variable depending on the project size, its goals, the manufacturing volumes expected, and more importantly, on the computer aided design programs.

The lowest design cost is achieved by ~~staying~~ completely avoiding ~~away from~~ modifying the ICs(except for ~~programming~~ stet (patterning) (read only memories). There are two elements to the cost of read only memories, programming cost and parts cost. The programming cost has already been discussed on page 00, so this discussion will be limited to parts cost. There are two kinds of read only memories, the field programmable read only memory (PROM) and the factory programmable (masked) read only memory (ROM). PROM chips have a higher initial cost than ROMs but provide some inventory advantages in a manufacturing environment because a common stock of unprogrammed parts can be kept ~~divided up into various programmed parts rather than stocking a full supply of each required part~~. In many high volume applications, however, the cost of the extra testing steps involved in the common stock approach, plus the extra piece part costs for PROMs, cause masked ROMs to be preferable.

The design costs discussed in the preceeding paragraphs are summarized in Figure Semiuse, which shows the costs for conventional programming, costs for microprogramming and the design costs for ROM/PLA designs using combinat~~or~~ionial techniques rather than programming techniques. The most costly approach of all shown in Figure Semiuse is design using standard circuits and associated design techniques.

Design of ICs (Intra-IC Design)

Despite the prospects of higher design cost with custom ICs than standard ICs, and, in some cases, higher manufacturing cost, there are numerous reasons why a computer designer is often forced to design ICs. These are summarized in Tablereasons.

Tablereasons: Reasons to do Custom IC Design

1. A performance advantage ~~#~~ can be gained.
~~Performance bells and whistles can be added that give the product a selling advantage over a less-LSI competitor. The manufacturing "mix problem" can be simplified by building and stocking a single design, and then shipping features that are inhibited unless the customer ordered them.~~

~~2. Monthly charges for maintenance might be reduced if diagnosability or reliability is improved. Since the customer pays these charges over and over, a small reliability improvement at the system level might justify an entire custom LSI program for large-volume applications.~~

3. Diagnostic labor can be a high percentage of printed circuit board manufacturing cost. Diagnosis to the chip level can be speeded up by

2. Product life cycle costs can be lower if diagnosability and reliability features are added.

~~3. Manufacturing cost~~

features within the chip, and by a lower chip count, with a resultant lower manufacturing cost.

4. Data busses can be absorbed entirely within a chip to avoid bus interface costs. Even shortening a data bus from multi-board to single-board length may reduce cost and/or improve performance. ~~by reducing stored energy and its attendant drive/speed penalties~~.

5. Innovations concealed within a chip are difficult for competitors ~~the competition~~ to study and duplicate.

6. Performance barriers may be breakable only through custom LSI. ~~In CPU design especially, and perhaps for certain RAM interfacing jobs, a custom LSI approach is the only practical way to get around conflicting issues of size, power, capacitance, etc.~~

7. In some engineering environments there are extremely small amounts of space or very little power.

There are, however, some drawbacks to custom LSI design. These are listed stet ~~in Table~~ below ~~CusLsiDra.~~

Table CusLsiDra: Reasons Not to Do Custom LSI Design

1. For designs in the 100-500 equivalent gate complexity range, it may take up to a year to do the design with the existing, primative design tools.

2. For designs in the 100-500 equivalent gate complexity range, it may take up to $100,000 to do the design.

3. Unless substantial product volumes are obtained, the chip cost will be higher than ~~relative to~~ off-the-shelf chips.

4. The ~~A~~ decision ~~will have to be made~~ whether to have the design done by an outside vendor or within the company~~. This~~ can be ~~a~~ very complicated and expensive. ~~decision.~~

5. The logic design and logic partitioning ~~for LSI design~~ is different from that of conventional logic design; ~~and~~ designers ~~used to dealing with~~ ~~family w~~ familiar with conventional design will have to assimilate new knowledge to design LSI themselves or even to work ~~talk~~ with LSI designers.

The use of custom ICs to reduce the number of discrete components or to reduce the total number of ICs in a machine improves the reliability because the latter ~~reliability of a system~~ is mostly a function of the number of explicit physical connections, including the bond to the semiconductor die. Thus the likely reliability of two equal functionality designs can be compared by counting ~~discrete circuit pins,~~ IC pins, module pins, and connector pins.

Assuming that one decides to do IC design, the various design methods that might be used for various objects and densities are given in the following table.

The most straightforward (and extensively used) intra-IC design method is to modify an existing design. If this approach cannot be used, the next most straightforward method is to use arrays of gates and interconnect them to form the desired function. Design with gate arrays (also called masterslices) occurs in a completely defined environment because there is only one circuit from which the gate is formed; ~~and~~ the gate can thus be completely ~~parameterized and defined~~ characterized. The manufacture of gate arrays is fairly simple because ~~the fabrication of~~ all but the last few ~~semiconductor~~ wafer processing steps is identical for all designs, and therein lies their advantage. The customization, accomplished by interconnection of the gates by metal, is carried out last. Interconnection, ~~is~~ a well understood aspect of logic design, ~~and~~ is used to form the more complex macro structures (various flip flop types, decoders, adders) and then to form the higher levels of design by using arrays of gate arrays. ~~There is~~ A disadantage ~~to gate arrays, which~~ is that gate array design methods do not permit the high density possible with ~~the more custom~~ hand-packed methods because device placement is fixed,

It should be noted that gate array design is not a new idea brought about by the need for a simple method of customizing LSI. Rather, it was one of the design ~~philosophies~~ approaches advocated in the first few generations. The concept then was to have a single module containing a set of gates, and all subsequent logical design would be done in terms of that module. For example, flip flops would be constructed by interconnecting the gates. A design predicated on a single module type, simplifies the spare stocking and servicing aspects immensely, and it is possible to troubleshoot a problem by simply replacing circuits according to a pattern. Designers did not find it important enough to get these advantages at that time, however, so the gate array concept was set aside until it was rediscovered by LSI designers.

A representative gate array is a Raytheon RA-116. It has 300 Schottky TTL gates, of two cluster configurations, each repeated 12 times within the 160 mil x 160 mil chip:

Type 1:	3 external driver gates	(4-input NAND)
	5 internal driver gates	(3-input NAND)
	5 internal expansion gates	(3-input NAND)
Type 2:	2 external driver gates	(4-input NAND)
	5 internal driver gates	(3-input NAND)
	5 internal expansion gates	(3-input NAND)

Within each cluster, the expansion gates may be combined with the driver gates to form 7- or 8-input NAND gates and AND-OR-INVERT circuits with up to six product terms.

The gates have a typical propagation delay of 5-6 nanoseconds and dissipate

5.5-6 milliwatts per driver and 1 milliwatt per OR expander. Two metal layers are used for interconnect, and the resulting circuitry can be connected to the outside world by means of 56 external pins, including power and ground.

~~Since the designer can arbitrarily interconnect, he constructs flip flops, adders, decoders, etc.~~ Because the use of IC gate arrays is recent, data on package count reduction is scarce, but one informal study for the Raytheon RP-16 aerospace computer measured a 9 to 1 replacement ratio and an overall factor-of-three improvement over a system constructed with standard components [Parke, 1978].

A 920 gate MOS array of 3-input NORs has been reported by [Nakano, et al 1978]. Its 3 nanosecond gate delay illustrates the performance potential as MOS continues to be scaled down. For higher speed applications, ~~an~~ ECL ~~gate~~ arrays ~~can be~~ are used. These devices, with sub-nanosecond speeds, exploit the inherent properties of current mode logic to obtain a particularly flexible element [Gaskill et al., 1976]. Examples are the TI and Fairchild ECL 168.

Standard cell design is identical to the logical design of the first ~~few~~ two generations ~~(e.g., PDP-1)~~ stet since there is a previously designed, well-~~defined~~ characterized set of primitive components (AND gates, flip flops) in which the design is carried out. The advantage of the standard cell design methods is that special functions can be mixed on the chip in greater variety. There may also be a density advantage over gate arrays. However, in some schemes each cell occupies a different space and has a fixed shape. Careful planning of the cell arrangements is necessary to minimize loss of space. Hence, the improvement in packing density is not as substantial as ~~direct~~ comparisons ~~between standard cell technology and gate array technology~~ might ~~at~~ first indicate. In addition, if there are a large number of circuit types, their interconnection rules may not be characterized well enough to achieve a quick, cheap design that works the first time.

Custom design is in some ways an extension ~~a variant~~ of the standard cell, since designers typically have a set of favorite circuits which they interconnect to create designs for specified applications. With custom design the designer can (theoretically) specify a circuit for each use within a particular logical design. For example, upon observing that a particular gate or flip flop only drives a certain load, the designer can modify that gate or flip-flop to provide only the appropriate driving capability. Therefore, with custom design, the whole IC can theoretically be an optimum, since each part is no larger than it need be. The advantages are clearly size, cost, and speed. The design costs are high because each part can, in principle, be customized. The quality of the circuit design is totally dependent on the designer who must analyze each circuit geometry in terms of his expectation of performance, operating margins, etc. To the extent that this analysis is carried out, the circuit is clearly optimal.

Also on the graph is a hypothetical line for universal logic arrays. For at least 15 years, academicians have studied the possibility of designing a single array of logical design elements, or a collection of such arrays, that could be interconnected on a custom basis to carry out a given function. The gate array can be looked at as the simplest example of this type of design. While many are skeptical that such a device exists, a line representing it is placed on the graph as a target for those who search for the one truly universal logic array. Patil and Welch [1978] ~~p~~ have proposed a storage/logic array matched to VLSI gate densities.

Both Read Only Memory (ROM) and Programmable Read Only Memory (PROM) are commonly used, but trivial, forms of the truly universal arrays, because they can be used in a table look up fashion to create several functions of a number of input variables. For example, a 1,024 word ROM arranged in a 256 x 4-bit fashion can generate 4 independent functions of 8 variables. This is a distinct alternative for using a conventional gate structure to carry out combinational functions. A disadvantage of this method is that the required ROM size doubles for each additional input variable.

The PLA is a combinational circuit which remedies the disadvantages of the ROM implementation of combinatorial functions by allowing the use of product terms rather than completely decoding the input variables. Fig. PLA shows a typical circuit, which consists of separate AND and OR arrays. Inputs are connected to the AND array, and outputs are drawn from the OR array. Each row in the PLA can implement an AND function of selected inputs or their complements, thus forming a boolean product term, and the OR array can combine the product terms to implement any boolean function.

A simple application is operation-code decoding. For the PDP-11, the 16-bit Instruction Register could be directly connected to a PLA and the output thereof used to specify the address of the microprogram that execute~~d~~s that instruction. Three different types of operation-code decoding are customarily applied to PDP-11 instructions: source mode decoding, destination mode decoding, and instruction decoding. With a PLA implementation, a PLA could be used for each of these decoding operations, and only three chips would be required. A ROM implementation, on the other hand, would require 128x8 for address mode decoding and 64Kx8 for instruction decoding. Using 2Kx8 ROM's, 33 chips would be required. For this reason modern minicomputers, such as the PDP-11/34, use PLAs rather than ROMs or combinat~~o~~ionrial logic for instruction decoding. The technique is also extended downward into microcomputers such as the LSI-11, where PLAs are used to conserve the die area used by their control ~~units~~ sections.

The PLA becomes an even more useful building block when it is made field programmable -- the FPLA. The programmable connectors shown in Figure PLA are fusible nichrome links that are burned out when the unit is programmed.

When a register is added to the outputs of the PLA and incorporated in the same integrated circuit, a simple sequential machine is obtained in one package. Since register circuit packages are pin intensive, adding registers to PLAs (or ROMs) permits about a factor-of-two package count reduction in typical applications.

The first PLAs had propagation times of the order of 150 nanosec and were thus only suitable building blocks for slow, low-cost computers. Propagation times of 45 nanoseconds are quite common today, and the PLA is more widely used. An attractive application with these higher speed components is the replacement of the SSI and MSI packages used to implement the control logic for Unibus arbitration. Alternatively, such a structure may be more cost-effective when implemented in gate arrays.

A more complex application than instruction decoding has been documented in [Logue et al., 1975]. An IBM 7441 Buffered Terminal Control Unit was implemented using PLAs and compared with an SSI/MSI version. The PLA design included two sets of registers fed by the OR array (PLA outputs): one set fed back to the AND array (PLA inputs), and the other set held the PLA outputs. A factor-of-two reduction in printed circuit board count was obtained with the PLA version. The seven PLA's used in the design replaced 85% of the circuits in the SSI/MSI version. Of these circuits 48% were combinational logic, and 52% were sequential logic.

MEMORY TECHNOLOGY (AND SEMICONDUCTORS USED AS MEMORIES)

The previous section discussed the use of memory for microprogramming and table look up in logical design, but that is not the principal use of memory in the computer industry. Rather, the more typical use of memory components is to form a hierarchy of storage levels which hold information on a short term basis while a program runs and on a longer term basis as permanent files. This section will present the various parameters and discussion relevant to this use. Many different, sometimes exotic, technologies have been used for memories, and figure Memtree covers many of them. While ~~the principal~~ our principal focus will be on core and semiconductor memories, slower speed electromechanical memories (drums, disks, and tapes) will be considered superficially, as their performance and price improvements have also pushed the computer evolution. Since the typical uses for memory usually require read and write capabilities, write once or read only memory such as video disks will be excluded from the discussion.

Because memory is the simplest of components, it should be possible to discuss the technology ~~memory~~ using a minimal number of ~~measurement~~ parameters. Many of the parameters vary with time; this is particularly true of semiconductor memory price, which has declined at a rate of 28% per year compound, (which amounts to about 50% in two years). The price is expressed only as price/bit, but it is important to know the price (or size) of the total memory system for which that price applies because of the economy of scale. ~~In order to get the lowest price per bit, a user may be forced to a large system.~~

Performance for cyclical memories, both the electromechanical types such as disks and the electronic types such as bubbles, is expressed in two parameters: the time to access the start of a block of memory, and the number of bits that can be accessed per second after the transfer begins. The operational environmental parameters of power consumption, temperature, space and weight effect the utility of memories in various applications, and the reliability measures are needed in order to see how much redundancy must be placed in the memory to operate at a given level of availability and data integrity.

In summary, the relevant parameters for a given memory technology are:

1. state of development of the technology at the time the measurements are taken relative to the likely life span of the technology

2. price per bit

3. total memory size or total memory price

4. performance

   a. the access time to the first word of the block

   b. the time to transfer each word (data-rate) in the block

5. operational power, temperature, space, weight

6. volatility

7. reliability and repairability

~~A good example of a technology that is young relative to its expected total lifetime is semiconductor memory.~~ Figure Memprice gives past prices and expected future prices of semiconductor memory. These memories have declined in price every two years by a factor of two, and that rate of decline is expected to continue well into the 80's because ~~of continued increases in semiconductor densities~~ the technology is young relative to its expected lifetime. Figure Memsizeperf, a graph by Dean ~~Toomg~~ Toombs, Vice President of Engineering for Texas Instruments, shows memory size and performance improvements with time and includes Charge Coupled Devices (CCDs) and magnetic bubbles. These latter devices show slower performance figures than the other semiconductor memories because they are cyclically accessed in a fashion similar to disks.

<u>Core and Semiconductor Memory Technology for Primary Memory</u>

The core memory was developed early in the first generation for Whirlwind (1953) and remained the dominant primary memory component for computers until it began to be superseded by semiconductor technology. The advent of the 1 Kbit memory chip in 1972 started the demise, and the crossover point occurred for most memory designs with the availabiilty of the 4 Kbit semiconductor chip in 1974.

The description and operation of the core memory is given briefly in Chapter 0 on the PDP-8 (page 0), and several of the significant circuit and production innovations are given in Chapter 0 on the 18-bit computers (page 0).

Over the period since the early 60s, the price of core memory declined roughly at a rate of 19% per year. This decline can be seen in the 12-bit (page 0), 18-bit (page 0) and IBM 360/370 memory prices (since 1964). The price of PDP-10 ~~DECsystem 10~~ memory has declined at 30% per year (page 0), although it is unclear why. A possible reason is that the modular memory structure had a high overhead, and that with subsequent implementations the memory module size was increased, thereby giving an effective decrease in overhead electronics cost and a greater decrease in the cost per bit.

The cost of various memories was projected by several technology marketing groups in the period 1972-1974. Each study attempted to analyze and determine the core/semiconductor memory crossover point.

Three such studies are plotted in Fig. Core.cost along with Turn's [1974] memory price data and Noyce's [1977b] semiconductor memory cost (less overhead electronics) projection. Note that most crossover points were projected to be in 1974, whereas one study showed a 1977 crossover. Even though all studies were done at about the same time, the variation in the studies shows the problem of getting consistent data from technology forecasts.

Figure Memprice shows the cost of semiconductor memory decreasing at a rate of 28% per year.

While these graphs of core and semiconductor prices and performance permit an understanding of trends in the principal use areas for these devices, (processors, primary memory, cache, and small paging memories), additional information is needed for disk and tape memory in order to complete the memory hierarchy.

Disk Memories

Disk memories are a significant part of ~~most~~ total system~~s~~ costs in the mid~~dle~~-range minicomputer systems; ~~and~~ for larger systems, they dominate the system costs.

Although access time is determined by the rotational delays and ~~the moving~~ speed of head ~~arm speed~~ movement, the single metric that is most often used is simply memory capacity and the resultant cost/bit. In the subsequent section on memory hierarchies, it will be ~~shown~~ argued that performance parameters are less important because more higher speed memory can be ~~traded off~~ add to gain the same ~~system~~ overall ~~level~~ performance.

Memory capacity is measured in disk surface areal density (i.e., the number of bits per square inch) and is the product of the number of bits recorded along a track and the number of tracks of the disk. Figure Areal.Digital gives the areal recording densities using digital and analog recording methods. Figure Large.disk.price shows the price of the state-of-the-art line for large, multiple platter, moving head disks. Note that the price decline is a factor of 10 in 9 years, for a price of decline of x% per year.

Figure Memtrends shows the performance plotted atainst the price per bit for the technology in 1975 and 1980.

Magnetic Tape Units

Figure Tapechar shows the performance characteristics that are relevant for magnetic tape units. The data is for several IBM tape drives between 1952 and 1973. It shows that the first tape units started out at 75 inches per second and achieved a speed of 200 inches per second by 1973. While this amounts to only a 5% improvement per year in speed over a 21 year period, this is a rather impressive gain considering the physical mass movement problems involved. It is akin to an improvement in automobile speed (and associated braking capability) of a factor of three.

The bit density (in bits per linear inch) has improved from 100 to 6250 in the same period for a factor of 62.5, or 23% per year. Since both speed and density have improved, the tape data rate has improved by a factor of 167, or 29% per year.

Tape unit prices (see Fig. x) ~~are based on~~ follow the ~~various~~ usual design styles. Slow tape units (minitapes) are built for lowest cost. The most cost effective seem to be around 75 inches per second (the initial design), if one considers only the tape. High performance units, though disproportionately expensive, provide the best system cost effectiveness.

Memory Hierarchies

A memory hierarchy, according to Strecker [1978], "is a memory system built of a number of different memory technologies: relatively small amounts of fast, expensive technologies and relatively large amounts of slow, inexpensive technologies. Most programs possess the property of locality[1]: the tendency to access a small, slowly varying subset of the memory locations they can potentially access. By exploiting locality, a properly designed memory hierarchy results in most processor references being satisfied by the faster levels of the hierarchy and most memory locations residing in the inexpensive levels. Thus, in the limit a memory hierarchy approaches the performance of the fastest technology and the per bit cost of the least expensive technology."

The key to achieving maximum performance per dollar from a memory hierarchy is to develop algorithms for moving information back and forth between the various types of storage in a fashion which exploits locality as much as possible. Two examples of hierarchies which depend upon program locality for their effectiveness are the one-level store (demand paging) first seen on the Atlas computer [Kilburn, et al, 1962] and the cache suggested by wilkes [1965] and first seen on the IBM

360/85 [Liptay, 1968]. Because both of these are automatically managed (exploiting locality), they are transparent to the programmer. This is in contrast to the case where a programmer uses secondary memory for file storage, because in that case he explicitly references the medium, and its use is therefore no longer transparent.

The following table, in order of memory speed, lists the memories used in current day hierarchies. There is a continuum based on need together with memory technology size, cost, and performance parameters.

Table: Computer System Memory Component and Technology

Part	Transparency (to machine language programs)	Based-on
Microprogram memory	yes ①	very fast
Processor-state	no	very small, very fast register set (e.g., 16 words)
Alternative processor-state context	yes	same (so speed up processor context swaps)
Cache memory	yes	fast. used in larger machines for speed
Program mapping and	yes	small associative store
Primary (program) memory	no	relatively fast, and large depending on Pc speed
Paging memory	yes	can be electromechanical, e.g.,drum, fixed head disk, or moving head disk. Can be CCD or

		bubbles.
Local file memory	no	usually moving head disk, relatively slow, low cost
Archival files memory	yes (preferably)	very slow, very cheap to permit information to be kept forever

Microprogram Memories

Nearly every part of the hierarchy can be observed in the computers in this book. Chapter 0, 0, and 0 describe PDP-11 implementations than use microprogramming. These memories are transparent to the user, except in machines such as the PDP-11/60 which provide user microprogramming via a writeable control store. Mudge (chapter 0) describes the writeable control storage user aspects associated with the 11/60 and the user microprogramming. Chapter 0 describes similar possibilities in the LSI-11, although the writeable control store option was not available at the time the article was written.

In retrospect, DEC should have built upon the experience gained from the small read only memory used for the PDP-9 (1967) and exploited the idea earlier. In particular, a ~~ROM~~ microprogrammed implementation might have produced a lower cost PDP11/20 and might have been used to implement lower cost PDP-10s earlier.

In principle, it is possible to have a cache to hold microprograms~~, hence,~~, which ~~there could be~~ would add another level to the hierarchy. At the moment, this would probably be used only in high cost, high performance machines, because of the overhead cost of the loading mechanism and the cache control. However, like so many other technical advances, it will probably migrate down to lower cost machines.

Processor State Registers

To the machine language program, the number of registers in the processor state is a very non-transparent part of the architecture. This number is solely dictated by the availability of fast access, low cost registers. It is also occasionally the means of classifying architectures (e.g., single accumulator based, general register based, and stack based).

In 1964, even though registers were not available in single IC packages, the PDP-6 adopted the general register structure because the cost of registers was only a small part of the system cost (see chapter 00). In the chapter on the DECsystem 10 there is a discussion of whether an architecture should be implemented with general registers in an explicit (non-transparent) fashion, or whether the stack architecture should be used. Altough a stack architecture does not provide registers for the programmer to manage, most implementations do incur the cost of registers for the top few elements of the stack. The general register structure was adopted to give better program control of a small number of local variables and hence gain performance advantages. The change in register use from accumulator-based design to general-register-based design and the associated increase in the number of registers from one to eight or sixteen can be observed between the 12-bit and 18-bit designs and the later DECsystem 10 and PDP-11 designs.

Alternative Processor State Context Registers

As the technology improved, the number of registers increased, and the processor state storage was increased to provide multiple sets of registers to improve process context switching time.

Cache Memory

In the late 60s, the cache memory was introduced for large scale computers. This concept was applied to the KL10 and 11/70 in 1975 when the relatively large (1 Kbit), relatively fast (factor of 5 faster than previously available), memory chip was introduced. The cache is described and discussed extensively in chapters 00, 00 and 00, so the reader may want to peruse page 0 if not familiar with the concept. It derives much power by the fact that it is an automatic mechanism and hence transparent to the user. It is the best example of the use of the principle of memory locality. For example, a well designed cache of 4 Kbytes can hold enough local computational memory so that, independent of program size, 90% of the accesses to memory are via the cache.

Program Mapping and Segmentation

A similar memory circuit is required to manage (map) multiprogrammed systems by providing relocation and protection among various user programs. The requirements are similar to the cache and may be incorporated in the caching structure. The KI 10 used an associative memory for this mapping function, and the VAX 11/780 uses a 64 entry two-way associative memory.

Paging Memory

The Atlas computer [Kilburn, et al, 1962] was designed to have a single, one level, large memory. This structure ultimately evolved so that multiple users could each have a large virtual address and virtual machine. ~~However, the concept of~~ The paging mechanism works because ~~there is not equally random access to each page, but rather only local access to various parts of a~~ of the locality exhibited by program ~~by the processor at a given time~~ references. Denning pointed out the clustering of pages for a given program at a given time and introduced the notion of the working set [1968]. For most programs the number of pages accessed locally is small compared with the total program size. Initially a magnetic drum was used to implement the paging memory, but as disk technology began to dominate the drum, both fixed head and moving head disks (backed up with larger primary memories) were used as the paging memories. Denning's tutorial article [1970] is an excellent discussion of this section of the memory hierarchy. In the next few years, the relatively faster and cheaper CCD semiconductor memories and bubble memories are clearly ~~the~~ candidates for paging memmories.

Local File Memory and Archival File Memory

For medium sized to large scale systems there is no alternative to disks, with archival files on magnetic tapes. These permit files to be stored cheaply on an indefinite basis. For smaller systems there are usually fewer memory technologies used than in larger systems, because the smaller systems cannot afford the overhead costs (disk drives, tape drives, etc.) associated with the various technologies. At most, two levels of storage would probably exist as separate entities.

Alternatively, one might expect a combination of floppy disk, low cost tape, and magnetic bubbles to be used to reduce the primary memory size and at the same time provide file and archival memory. Currently the floppy disk operates as a single level memory. Here one can see two alternatives for technology tradeoff using the hierarchy: a tape or floppy disk can be used to provide removability, and archivability, whereas bubbles or CCD provide performance. The Strecker paper [1978] quoted at the beginning of this section on memory hierarchies elaborates on these concepts.

MEASURING (AND CREATING) TECHNOLOGY PROGRESS

The previous sections have presented technology in terms of exponentially decreasing prices and/or exponentially increasing performance. This section presents a basis for this constant change. The progress of a particular technology as a function of time, T(t) has been classically observed to be:

$$T(t) = K \times e^{ct}$$

This can be converted to a yearly improvement rate, r, by changing the base of the exponential to:

$$T(t) = T \quad \times r^{t-to}$$

where T = the base technology at t_o

and r = yearly increase (or decrease) in the technology metric

This is the same form used for declining (or increasing) cost from base c

$C = c \quad \times r^{t-to}$

Clearly there are manufactured goods that neither improve nor decrease in price exponentially, although many presumably could with the proper design and manufacturing tooling investments. The notion of price decline is completely tied to the cumulative learning curves of a) people building a product for a long time, b) process improvement based on learning to build it better, and c) design improvement by engineers based on learning from the history of design. Production learning per se is inadequate to drive cost and prices down because after an extremely long time in production, more units contribute little to learning. With inflation in labor costs, the costs actually rise when the learning is flat. In order to provide a base for predicting the inflationary effect, the consumer price index has been plotted (Fig. CPI)[1].

Learning curves don't appear to be understood beyond intuition. They are (empirical) observations that the amount of human energy, En, required to produce the nth item is:

$$En = K \text{ x } n^{d}$$

where K and d are "learning constants". Thus, by producing more items, the repetitive nature of a task causes learning, and hence the time (and perhaps cost) to produce an item decreases with the number produced and not with the calendar time an object is produced.

In his study of technology progress, Fusfeld [1973] took six items, chose a measure of progress in the production thereof, and plotted that measure against cumulative units produced. In each case, he found a relationship of the form:

$$T_i = a \text{ x } i^{b}$$

where i is the number of units produced and Ti is the value of his selected

[1] Note that all dollar values given in this book are in current dollars.

technology progress measure at the ith unit.

The graph for turbojet engines, where he used fuel consumed per pound as the technology measure, is reproduced in Figure Turbo. The results for all six items studied are shown in the following table:

Item	Measure, Ti	Quantity produced(i)	Technology progress(b)	Change observed in study	Total change
light bulbs	lumens/bulb	10^{10}	.04;.19	33	80
automobiles	vehicle h.p.	$3x10^7;10^8$	.11;.74	10	6;13
titanium	p.s.i./$/16	3×10^8	.3;1;1.04	10	350
aircraft	max.speed	2×10^5	.33-1.2	6	56
turbojet engines	fuel consumed, weight	1.6×10^4	1.06	2	2.9×10^4
computers	mem.size x rate	10^5	2.51	10^9	3.5×10^{12}

Where two values are given for the technology progress constant, a second rate of progress was observed after a significant shift in the industry occurred. For example, in the automobile industry, such a shift occurred in the late 1920's when the acceptance of the automobile, the development of a new tire, and the expansion of the public road network operated concurrently to change the nature of the industry.

Examination of the table will reveal substantial variations in the technology progress constant from item to item. This is probably because most of the technologies represented above are mechanically oriented with associated physical limits. Computer technology is electronically oriented and has not yet reached its limits. In essence the table is comparing systems constrained by Newton's Laws with those determined by Maxwell's Equations.

Using the two formulas,

1) $T(t) = K \times e^{ct}$

and

2) $T_i = a \times i^b$.

one can relate the Fusfeld results (2) to the earlier view of technology progress (1).

First, differentiate (1) with respect to time:

3) $\underline{dT}/dt = Kce^{ct} = cT$

$\text{or } dT/T = c\ dt$

and by differentiating (2) with respect to the quantity produced, i:

4) $dT/di = abi^{b-1} = b\ (T/i)$

$\text{or } dT/T = b\ (di/i)$

Since both models must produce the same results, DT/T must be the same for each and therefore can be equated to obtain:

5) $$c\,dt = (b/i)\,di$$

Therefore, the rate of production is:

6) $$di/dt = (c/b)\,i$$

This formula indicates that the production rate is a constant fraction of the total production to date - i.e. production occurs with exponential growth.

While the Fusfeld information is an interesting result, it does not explain why technology improves exponentially, nor does it explain why cost declines exponentially. Learning curves and an exponential increase in the quantity of items produced may depress cost. However, simple production learning does not account for the rapid technology changes in the integrated circuit, for example, where totally different production processes have been evolved to support the greater technology. It appears best to simply observe that the these situations have been true, and can be extrapolated to hold over the next few years because one can see ways by which each limit can be overcome.

Management scientists studying technology evolution [von Hippel, 1977] have made a number of observations about the sources of technology innovation:

1. Technical problem solving is correlated with business activity. Inventors tend to be stimulated by sales and slacken efforts when sales are low. While this might appear to be a counter-intuitive observation, it is not, because inventors are attracted to profitable industries and companies. When sales are low, the inventors move to a more "glamorous" company or industry. Further, inventors are supported by active, enterprising successful companies/industries; but inventors and ideas are unwelcome in companies/industries that are "stable", where it is felt "we already have our ideas," and "we know how to do it without new ideas".

2. Production alone does not stimulate innovation. A lesser number of inventions are stimulated by production needs, except in the auto industry, where "production is where it's at". Of these, the same user-supplier relationship is the best framework. The users of equipment (the producer for the end product) stimulate the production equipment suppliers.

The Influence of Technology Innovation on Cost

The cost of computing is the sum of costs which correspond to the various levels of integration described in Chapter 1 plus the operational costs. The levels were integrated circuits, boards, boxes, cabinets, operating systems, standard languages, special languages, applications components, and applications. In actual practice each additional level-of-integration is often looked at as overhead. Using standard accounting practice, the basic hardware cost, at the lowest level, is then multiplied by an overhead factor at each subsequent outer level. While an overhead-based model may work operationally for a stable set of technologies, such a model will not adequately allow for rapidly evolving technologies or the elimination of levels, for example. By examining each level, as this technology section and the packaging section attempt to do, observations can be made about the use and substitution of technology. More importantly, conclusions can be drawn about how structures are likely to evolve.

In preceding chapters and sections, semiconductor technology has been singled out as the main determinant of a computer's cost, performance, reliability and memory size. Magnetic storage technology is of equal importance because disk and tape memory densities evolve rapidly, even though the cost of a given physical unit does not. These units have become an increasingly major component of the price of computer systems as the costs of processors and primary memory have dropped, but their performance/price improvements have been notable, nonetheless.

Cost, Performance, and Economy of Scale

For most technologies used in the computer industry, there is a relationship between cost, performance, and economy of scale:

$$\text{performance} = k \times \text{cost}^{s} \times r^{t}$$

where k = base case performance
s = economy of scale coefficient
r = rate of improvement of technology
t = calendar time

There are four options of the amount of economy of scale:

1. Economy of scale holds. A particular object can be implemented at any price, and the performance varies exponentially with price.

   $\text{performance} = k \times \text{price}^{s}; s>1$

2. Linear price performance relationship.

a. performance = k x price

b. performance = base + K x price

3. Constant performance, price independent

performance = k

4. Only a particular device has been implemented. The performance (or size) is a linear sum of such devices.

performance = n x (k x price)

Sometimes, economy of scale effects are observed in situations where their applicability would not normally be expected. For example, assume a performance improvement feature exists that costs the same whether it is added to a large computer or added to a small computer. Adding that feature to a product that is already high priced will have a modest effect (say 5%) on the cost but a substantial effect (say 100%) on the performance. Adding the same constant cost feature to a lower cost product will have a substantial effect (say 200%) on the cost, but a performance effect similar (100%) to that obtained with the higher cost system (or prehaps even less). This condition is especially true in disks and computer systems. Use of a particular recording method employing costly logic for encoding/decoding, or addition of a cache memory is often employed to the high priced systems first. With time, and learning, the technique can then be applied to lower cost systems. For example, cache, a nearly perfect example of the constant cost add-on, first appeared in such large machines as the IBM 360/85 in 1968 and later migrated down to large minicomputers such as the 11/70 in 1975. On a research basis, cache even reached the small minicomputer, the cache-based PDP-8/E at Carnegie Mellon (page 00).

In Fig. Costvstime, the cost of the lowest price unit is kept to a minimum and decreasing while the cost of the mid range product continues to increase. The cost of the highest performance product increases the most, because it can afford the overhead costs. Looking at the basic technology metric, there are really three curves as shown in Fig. Techvstime. The first curve is applied to get the greatest improvement and be applied to the large price unit. With time the technology evolves and is reapplied to the first level copy in the middle range products (to most likely provide the best cost performance) and finally, several years later, the technique becomes commonplace and is applied on low cost products. The resultant cost/performance ratios are shown in Fig. Cost/tech.

In most industries, the management of technology by applying it to products in various price and performance ranges occurs in a more or less ordered fashion, but has not occurred to the extent that it has in the computer industry. This is probably because no other industries have evolved in the same rapid and broad fashion as have the computer and semiconductor industries. The computer industry is fundamentally driven by the semiconductor technology push on the one hand, and by IBM on the other. IBM follows the strategy of applying technology on an economy of scale basis. This permits the technology to be first tested at the high performance, high price, lower volume systems before being introduced in higher volume production. The following examples (from IBM) show this at work. In printing, the high price/low volume to low/price high/volume introduction cycle was followed in the use of dot matrix printing, chain printing, ink jet printing, and computer printing on flexible disks as a precursor to use as systems products using xerography. In magnetic storage, the cycle saw the basic technology for large disks as a precursor to the use of similar technology on smaller disks.

Technology Substitution

Since each constituent technology evolves at its own rate, the cost and performance of a system are roughly the additive and multiplication functions, respectively, of the parts. Usually when one component begins to dominate (e.g., packaging), then pressure occurs to more rapidly change and improve the technology to avoid the cost or performance bottleneck. Sometimes a slowly evolving technology is just eliminated as a substitute is found.

Some of the substitutions that have occurred:

1. Semiconductor memories are now used in place of core memories. Since the latter has evolved more slowly in terms of price decline, semiconductors are now used to the exclusion of cores. (This has not occurred where information must be retained in the memory during periods of time without power.)

2. Read-only semiconductor memories are now substituted for semiconductor logic elements.

3. In a similar way PLAs can be potentially substituted for ROMs and true content addressable memories can replace various read write and ROM memories.

4. The judicious use of CCD or bubble memory can cause drastic reduction (and quite possibly the elimination) of the use MOS random access memories for primary memory. The fixed head disk could be eliminated at the same time.

6. For small systems the main operational memories could be completely non-electromechanical; electromechanical memories (e.g., tape cassettes and floppies) would be used for loading files into the system and for archives. For yet lower cost systems, semiconductors ROMs could replace cassettes and floppies for program storage as in the programmable calculators.

After a while those components of computer system cost which are decreasing less rapidly than other components, or are remaining static, or are rising (like the packaging and power) may become a significant fraction of the total cost. Costs are additive and hence exponential improvements have disproportionate effects causing pressure for structural change.

For instance, although the PDP-8 is normally considered to be the first minicomputer, it post dates the CDC 160 (1960) and DEC's PDP-5 (1963). However, the PDP-8 was unique in its use of technology because:

1. It eliminated the full frame cabinets used by other systems. This also

presented a new computer style such that users could embed the computer in their own cabinets. A separated small box held the processor, memory and many options.

2. Automatic wire wrap technology was used to reduce printed circuit board interconnection cost. This also eliminated errors and reduced check out time.

3. Printed circuit board costs were reduced by using machine insertion of components.

4. The Teletype ASR33 (also used on PDP-5) was connected as the peripheral. It had a combined printer, keyboard, and paper tape i/o device (for program loading). It eliminated the paper tape reader and punch.

The Effect of the Research, Advanced Development Process of the State of the Art Line

The ~~complete~~ development process ~~is~~ can be modelled by a pipeline~~d~~ ~~as shown along the lines of Fig. 1 in Chapter 1, page 0 with the~~ with the following stages: research, applied research, advanced development (product breadboard), development, test, sell/build, and use. In this model, ideas and information flow through the various organizations in a process-like fashion, culminating in a product. Each product type has a different set of delays associated with the parts of the pipeline. At the end of the pipeline, the "education of use" delay occurs while the prospective customers are taught how the product meets their needs;

this delay culminates in market demand. For well defined, commodity-like products such as disks and primary memory, the education of use delay is zero as each user "knows" the product. For a new language, on the other hand, there is a large education of use delay and the market demand usually develops slowly.

The disk supply process is a good example of the pipeline nature of the development process. The technology (as measured by the number of bits per areal inch) doubles about every two years (i.e., the density improves 41% per year). IBM is estimated to invest about 100 million dollars per year in the development and associated manufacturing process pipelines. Because of this massive investment, the IBM disks essentially establish the state of the art line in a structure that is typified by Fig. Techvstime. Using the pipeline development process, development of competitive disks by other companies would lie somewhere about four to six years behind the state of the art line. This can be seen by looking at the development process and taking into account the delays through each stage. In order to be more competitive, the disk industry short circuits various delays by engaging in reverse engineering; this results in only two year lags. In reverse engineering, the tools are micrometers and reverse molds. At time of the first ship of a new product by the technology leader, the product is purchased by competitors and basically copied on a function per function basis. The more successful designs use pin for pin compatibility so as to take maximum advantage of the leader's design decisions.

From the process, it is also easy to see how merely copying competitive products guarantees products that will be two to four years behind leadership products and lagging the state of the art. Nonetheless, if there is a strong market function which operates to define products based on existing product use, and if the design and manufacturing process at the copying company is quite rapid, such a strategy can be effective. The copying process can be very effective for software products because while there are no delays associated with manufacture, the time to learn about the product provides a time window in which copiers can catch up with the leaders.

A high technology, exponentially increasing (volume) product is denoted by:

1. Exponential yearly cost improvement (price decline) rates through product technology improvements as measured by price decline of > 20% (e.g., disk/price this year = .8 last year's disk price, cpu = .79, memory = .7).
2. Short product life < 4 years.
3. Various types of learning curves. Some products require very little learning, while others require a great deal of learning or require re-learning because of personnel turnover or the frequent hiring of additional personnel.

The Product Problem (Behind the State-of-the-Art)

Typical product situations, including competitive "problems" can be seen in Fig. Product.cost. When a product is introduced to the market, it has a relationship to the state-of-the-art line. There are five possible situations:

1. ideal (on the state-of-the-art line)
2. advanced (moves below the line)
3. late (slip in time to the right)
4. expensive (more than expected in cost, straight above the line)
5. late and expensive (to the right and above the line)

Situations 3, 4, and 5 are product problems because they are behind the state-of-the-art line and hence less competitive. This implies increased sales costs, lower margins, loss of sales, etc. Note that a late product could be OK if somehow the cost were lower. Similarly an expensive product is OK if it appears earlier in time.

Time is Money (and vice versa)

Thus product problems can be solved by either:

1. movement in time (left) to get on the line; or
2. movement in cost (straight down) to get on the line

since

c = cost at time, t (in years)
b = base cost
r = rate of price decline

therefore, with exponential price declines a family of products over a long time will follow a cost curve, c.

$$c = b \times r^{t}$$

now

dc = change in cost above (or below) to get back to the state-of-the-art

dt = delay (or advance) in time to get back to the state-of-the-art line

let

f = dc/c = fraction (%) of cost away from line

$f = 1 - r^{dt}$ poor cost, expressed as
project slip

and

$dt = \ln(1 - f) / \ln(r)$ poor timing,
expressed as poor

cost

These formulas permit the interchange of time and money (cost).

For example, in disks or cpu's where r = .8 and ln.8 = .22

1. $f = 1 - .8^{dt}$
2. $dt = - 4.45 \times \ln (1 - f)$

(using 1.) A 1 year slip is equal to a 20% cost problem

(using 2.) A 10% cost increase is equal to a .47 year slip

Who does what, and their effect

By and large engineering, by establishing the product direction, has the greatest effect on the product. However, since most product problems may have multiple components, it's worth looking at each.

1. Timing

   a. Engineering schedule slips translate into a competitive cost problem as a sub state-of-the-art, late product.

   b. Manufacturing - ramping up the learning curve quickly by risk taking has a high payoff when considering the apparent cost or delay.

2. Cost

   A number of components and organizations contribute to the total product cost, as shown in Fig. Net.

   a. Engineering is perhaps the major determinant of cost by the product design - number of parts, ease of assembly, etc. The most common cost problems occur by continued product enhancement during the design stage to provide increased functionality (called "one-plussing the design"). One-plussing often occurs because the market had not been modeled before the design was begun, and without a model of the market, engineering is a ship without a rudder.

b. Manufacturing - direct labor and overhead really count. Making major changes in the design of a product or the location of manufacture for a product starts a new learning curve and serves to stretch the production time out, and the increased costs associated therewith,put false pressure one engineering to design new products. One curve in Figure Net. shows the direct costs associated with manufacturing assembly and then some learning should take place as long as product volumes increase exponentially. New technology materials show the greatest cost improvement for computers, assuming that semiconductors and other electronic materials improve with time. Note that by capital equipment investment (tooling), there can be stepwise cost reductions in materials costs.

c. Inflation - while not a direct cost function, it combines with labor cost to negate the downward cost trends that were obtained from learning effects.

d. Note the costs are taken altogether. In terms of a sub state of the art product, the costs are compound. A one year late, 10% overcost product has the effect of being about 1-1/2 years late or about 30% too expensive.

3. Manufacturing learning

Learning curves and forgetting curves really matter. Left alone, a typical product may go down three alternative paths (see Fig. Product.price):

1. $c = b \times .95^{t}$ decrease as little as 5%/year.

2. $c = b$ = stay constant with little attention.

3. $c = b \times 1.06^{t}$ = increase with inflation.

Mid Life Kicker for Product Rejuvination

By enhancing an existing product (the so called mid-life kicker) one can improve the cost/performance metric of a given product. This is non-trivial, and for certain products must be inherent (i.e., designed in). Under these conditions, improvements in cost go immediately to get the product back onto the state-of-the-art-line.

For example a factor of 2 in performance halves cost/performance. The effect, of doubling the density of a disk is to move the product back to the state-of-the-art line by a time shift. Plugging the factor of two improvement into the formulas:

dt = 4.45 x ln (0.5) = 3.1 years

This situation is shown in Fig. Product.improve and is compared with a 5%/year learning curve.

PACKAGING

Seymour Cray, in a lecture at Lawrence Livermore Laboratory in December 1974, described packaging as the most difficult part of the computer designer's job. The two major problems are heat removal and the thickness of the mat of wires which cover the back~~panel~~plane. His rule of thumb indicates that with every generation of large computer, the size decreases by roughly a factor of five--and each generation takes roughly five years.

While it is easy to understand why Cray's super computers are so dominated by packaging, it is more interesting to examine the effect of packaging on small computers. At one extreme, the first hand-held scientific calculator, the HP35, was simply a new package for a common object, the calculator, which has been around about 100 years.

Although semiconductor density had been improving for several years, it was was not until densities were high enough to permit implementation of a calculator in a few chips, and not until those chips could be re-packaged in a particular fashion, that the hand-held calculator came into existence. Currently this embodiment is synonomous with the calculator name--in the future, the calculator might take on some other form (e.g., watch, pencil, voice actuated, hearing aid, notebook).

Packaging also seems to be the dominant reason for the PDP-8 and minicomputer phenomenon--although marketing, the coining of the name, and the ease of manufacture (also part of packaging) are alternative explanations. The packaging advantages of the PDP-8 over predecessor machines can be seen from the photographs in Part II.

The Packaging Design Problem

Packaging is the complete design activity of interconnecting a set of components via a mechanical structure in order to carry out a given function. In order to package a large structure such as a computer, the problem is further broken into a series of levels each with components that carry out a given function. Figure 3 shows the hierarchy of levels that have evolved these last twenty years for the DEC computers. There are eight levels which describe the component hierarchy resulting in a computer system.

For each packaging level there is a set of interrelated design activities as shown in Fig. 4. The activities are almost independent of the level at which they are carried out, and some design activities are carried out across several levels.

While the initial design activities indicated in Figure 4 are each aimed at solving a particular problem, the solving of one problem in computer engineering usually creates other problems as side effects. For example, the integrated circuits and other equipment that do information processing require power to operate. The power creates a safety hazard and is provided by power supplies that operate at less than 100% efficiency. These side effects create a need for designing insulators and providing methods of carrying the heat away from the power supply and the components being powered. In this way, cooling problems are created. Sometimes cooling is carried out using conduction to an outside surface so that it may be carried away by the air in a room, but most cooling is carried out by convection with a cabinet fan which carries air into the room so that the room air conditioning system is left with the problem of carrying the heat away. In this process, the fans create acoustical noise pollution in the room, making it more difficult for humans to work in the room. Furthermore, if the computer is used in an unusually harsh environment, a special heat exchanger is required in order to avoid contamination of the components within the computer by the pollutants present in the cooling air flow.

Finally, a particular package exhibits mechanical characteristics such as weight and size. These parameters directly affect manufacturing and shipment costs. They determine whether a system can be built, and whether it can be shipped in a certain size airplane, or carried by a particular distribution channel (e.g., parcel post or United Parcel). The dynamic characteristics determine the type of vehicle (special air ride van for electronic equipment) in which equipment must be shipped.

It is also necessary to examine the particular design parameter in order to determine whether it is a ~~constant~~ constraint (meets ~~the German VDE~~ a particular government standard, for example ~~x~~), a goal (cheap as possible), or part of a more complex objective function (measured in bits/sec/$ or part of a system benchmark--jobs/sec/$).

The following table lists the various kinds of design activities and whether they deal with goals, constraints, or parts of more complex objective functions. The table also gives the dimensions of various metrics (e.g., cost, weight) available to measure the designs and many of these metrics are used in subsequent comparisons.

Table Design Activities, Metrics and Environment Setting Goals and Constraints

Design Activity	Determining Environment and [Metrics]
Primary function and performance (e.g., memory)	Market, next highest level (i.e. the consumer) of system [memory size (bits), operation-rate (bits/sec.)]
Human engineering	Human factors criteria, competitive market
Visual/Aesthetics	Market, other similar objects, the environment in which the object is to exist--usually only important at outer-most level
Acoustic noise	Government standards, operating environment, market [decibels in various frequency bands]
Mechanical	Shippability (e.g., air cargo container size, truck vibration), handling, assembly/disassembly time [weight, floor area, volume, expandability, acceleration, mechanical frequency response
EMI radiation input	Government standards, must operate within intended environment (e.g., high noise) [power vs. frequency]
Power	Operating environment, market [watts, voltage supply range]
Cooling and environment	Market, intended storage and operating environment, government standards [heat dissipation, temperature range, air flow, humidity range, salinity, dust particle, hazardous gas]
Safety	Government standards
Cost cost/metric ratios	[cost/performance (its function)--cost/bit and cost/bit/sec., cost/weight, cost/area, cost/volume, cost/watt]
density metrics	[weight/volume, watts/volume, operation-rate/volume]
power metrics	[operation-rate/watt; efficiency = power out/power in]
reliability	[reliability--failure rate (Mean Time Between Failures, Mean Time To Repair (MTTR)]

Given the basic design activities, one may now examine their interaction with the hierarchy of levels (i.e., the systems) being designed (see Table 00). This is done by looking at each level and examining the interaction of the design activities for that level with other design activities (e.g., function requires power, power requires cooling, cooling requires fans, fans create noise and noise requires noise suppression).

Table Interrelationship of Hierarchy of Levels and Design Activities

Design\Level of Packaging Activity\	Chip	Chip Pkg	Module	Backplane	Box	Cabinet	Computer Systems
Functional	logic-----------------------------------> electrical					configuration options	selection of right components by user
	circuit design physical layout		physical layout	physical layout	what fits and operates	boxes,. what configurations will operate	
Human interface						location of console, size for use	placement for use
Visual					visible, bought for integration	determines system appearance	set of cabs, attractive place to be
Acoustic				Airflow -------------------> vibration			quiet for operators and users
Mechanical	buildable;		shippable	serviceable ---------------->			floor load
	signal transmission --------------------------------------->						room size
EMI	noise coupling and rejection of ext.RFI		inter/intra module noise containment and shielding	coupling, RFI	RFI containment, external RFI shield		away from RFI input (outside operating range)

Power	special on-chip		dist. and regulation	dist. and regulation	control, dist. and regulation	interconnect with computer system	by user. special power supplies for high availability
Cooling and other environment	chip to cooling special envir.	IC module cooling special envir.	IC to cooling	module	cooling & covering	source	interbox coupling to room air envir.
Safety			power ----------> for various systems		determines safety if used at this level	determines user safety	
Dominant design activities	circuit logic	logic ----------------------->			mechanical, power, cooling, EMI, acoustic	configtion visual, shipping EMI, safety	user configation design

Note: The box and backplane levels can be considered as a single level. (Alternatively, the box level may be eliminated in large systems)

Computer Systems Level

The topmost level in the preceding table is the "computer system", which for the larger mini and "mainframe" 10 computers consists of a set of subsystems (processor, memories, etc.) within cabinets, housed in a room, and interconnected by cables. The functional design activity is the selection and interconnection of the cabinets, with a basic computer cabinet holding processor, memory, and interfaces to peripheral units. Disks, magnetic tape units, printers, and terminals occupy free standing cabinets. The functional design is usually carried out by the user and consists of selecting the right components tc meet cost, speed, number of users, data-base size, language (programming), reliability and interface constraints. Aside from the functional design problem, cooling and power design are significant for larger computers. For smaller computers, accessibility, acoustic noise, and visual considerations are significant because these machines become part of a local environment--and must "fit in".

Cabinet Level

Since the cabinet is the lowest level component that users interface to and observe, the physical design, visual appearance, and human factors engineering are the dominant design activities. For the computer hardware designer, on the other hand, the component associated with the cabinet is often his largest system. His functional design efforts insure that the various components (i.e., boxes) that make up a cabinet level system will operate correctly when interconnected. Safety and EMI characteristics are important because the cabinet serves as the outermost place that shielding can be installed. Cooling and power distribution must be considered, since a number of different boxes may be mounted within the same cabinet. Finally, the mechanical structure of a cabinet must be designed to maintain its physical integrity when shipped.

Box Level

Box level functional design consists of taking one or more backplanes, the power supplies for the box, and any user interface such as an operator's console and interconnecting them mechanically (see Figure 11/05). For systems that are not sold at the box level, no separate box is required, and the power supply and backplanes are mounted directly in a cabinet (see

Figures 11/70.1) or other holding structure such as a desk or terminal case, so that box and backplane design are merged as one. If systems are sold at the box level, then the visual characteristics may be important; otherwise, the design is basically mechanical and consists of cooling, power distribution, and control of acoustic noise. The structure must be basically quite sound in order to protect the unit during shipment.

Backplane Level

This level of design is the final level of interconnection for the computer components that are designed to stand alone, such as a basic computer disk, or terminal. Backplane design is part of the computer's logical design. In second generation machines such the PDP-7 (Figure 24A, Chapter 5), the backplane was wire wrapped. In the early 1970's printed circuit boards were used to interconnect modules (Figure BP). Figure 00, page 00 shows how the Omnibus backplane for PDP-8/A is used to interconnect the basic components within the box. Secondary design activities include holding, powering and cooling the modules so they will operate correctly. Since the signals are transmitted on the backplane, there is an EMI design problem. For industrial control systems whose function is to switch power, additional safety problems are created.

Module Level

In the second generation, module level design was a circuit design activity taking discrete circuits and interconnecting them to provide a given logic function. In the third and fourth generations, this interface between circuit and logic design moved to be within chip level design, so module level design became the process of dealing with physical layout design problems associated with logical design. This shift in roles (function) will be described below in the section discussing the interrelationship between the technology generations and packaging. The integrated circuits that perform the functions are assigned to different positions on the module. Module level design is basically electronic, so power, cooling, and EMI (cross talk) considerations dominate.

Integrated Circuit Package and Chip Level

Most integrated circuits used in the computer industry today are sold in a plastic or ceramic package configuration that has two rows of pins, and is hence called a Dual In-line Package, abbreviated DIP. The majority of the IC's in the module shown in Figure LSI11 are 16-pin DIPs. Because of the popularity of this packaging style, the terms IC, chip, and DIP are often used interchangably. This is not strictly correct, as an integrated circuit is actually a quarter inch square portion of semiconductor material (die or "chip") from a two inch to four inch diameter semiconductor wafer. Until multiple dice are packaged within a single DIP, creating different design interconnections at the IC and DIP levels, the IC and DIP can be discussed as a single level. In Figure LSI11, the center DIP of the four large DIPs has two dice mounted on it.

Logic design is a part of the functional design problem at the chip level. The task is to supply a terminal behavior that can be used at the next (module level) and, depending on the technology generation, this function can vary from a simple gate (in the early third generation) to a full computer (late in the fourth generation).

Other design activities at this level include generic (?) to electrical signal processing: power, heat dissipation, and EMI. Since some ICs are designed to operate in hostile environments, there is a considerable mechanical design activity associated with packaging, interconnection and manufacturing.

The Packaging Evolution

Figure 6 shows the relation of packaging and the computer classes for the various computer generations. For each new generation, there is a short, evolutionary transition phase, but ultimately the new technology is repackaged such that a complete information storage or processing component (bit, register, processor) occupies a small fraction of the space and costs a small fraction of the amount it did in the prior generation. Quite discrete events mark packaging characteristics of each generation, starting from 1 bit per vacuum tube chassis in the first generation and evolving to a complete computer on a single integrated circuit chip in the fifth generation. Not only the size of the packaging changed, but also the mounting methods. In the first generation, logical units were permanently mounted in racks, whereas in the later generations they were made removable for ease in servicing.

Whereas the time-line shows the packaging evolution of a complete computer, the following table shows how a particular component, now called the Universal Asynchronous Receiver-Transmitter (UART), has evolved with time.

Table __ Packaging Hierarchy Evolution for Universal Asynchronous Receiver Transmitter (UART) Telegraph Line Controller

early second	**late second**	**early third**	**late third**	**late fourth**
backplane				
modules	2 modules	module		
discrete	discrete	IC	IC	
circuit	circuit	chip	chip	chip area

The UART logic carries out the function of interfacing to a communications line carrying serial data and transforming the data to parallel on a character by character basis for entry into the rest of the computer system. The UART has three basic components: the serial/parallel conversion and buffering; the interfaces to both the computer and to the communication line; and the sequential controller for the circuit.

The UART is probably the first fourth generation computer component, since it is somewhat less complex than a processor, yet rich enough to be identifiable with a clean, standard interface.[1]

Footnote 1

single space as footnote

Of historical note, DEC played a significant part in the development of the UART technology. With the PDP-1, the first UART function was designed using 500 Khz systems modules, and was used in a message switching application as described in chapter 00. The interface was called a line unit, and was subsequently repackaged in the late second generation to be on two extended systems modules (see Fig. 9). The UART function was also built into the PDP-8/I using two modules, but substantially smaller ones than those for the PDP-1. In the 680/I, a PDP-8/I driven message switch, the UART function was accomplished by programmed bit sampling. Late in the third generation (or at the beginning of the fourth generation), some designers from Solid State Data Systems, a small company on Long Island, worked with Vince Bastiani at DEC and developed a UART that occupied a single chip. This subsequently evolved to become a standard IC, and has been used throughout the industry, for example in the DL-11 communications line interface module for the PDP11.

The DEC Computer Packaging Generations

With this general background on packaging, one can examine the DEC packaging evolution more specifically and against the general archetype of Figure 3. Figure 12 shows how the hierarchies have changed with the technology generations. The figure is segmented into the different product groupings. A product is identified as being at a unique level if it is sold at the particular packaging level. The first DEC computers (i.e., PDP-1 to PDP-6) were sold at the cabinet level as complete hardware systems. Although the PDP-8 was available at the cabinet level for complete systems, it was significantly smaller than the previous machines and was principally sold at the mechanical box level so that it could be incorporated into other hardware systems and used within the packaging hierarchies of OEM customers. Subsequently computer systems evolved to be available at the backplane level (LSI-11), and at the module level (CMOS-8).

The original packaging hierarchy for most of DEC's second generation computers used a relatively common packaging scheme based on the PDP-1. The most significant change occurred in the late second generation when R-Series and B-Series Flip Chip modules (see Fig. 12) were introduced so that backplanes could be wire wrapped automatically, enabling lower cost and greater scale production.

The change to wire wrap technology also enabled the box-level production of computers. The change to wire wrap and two level (box and cabinet level) products is clear in the second generation.

The change to IC packaging in the third generation gave rise to computers that are sold at the box level.

The CMOS-8 module, described more completely in Chapter xx, is a single-board complete computer with processor, 16 Kword memory, and all the optional controllers to directly interface up to five peripheral options. For a computer packaged in this fashion, a backplane is not used, and hence the backplane packaging level doesn't exist.

The LSI-11 is marketed at the three levels in Figure 12. Although components are sold as separate modules (e.g., communications line interfaces, additional primary memory), a complete system requires a backplane, thus the lowest level for the product is the backplane. For larger systems, a power supply is combined and placed in a metal box (the 11/03), and finally a complete system such as the 11V03 is available with terminal and mass storage in a single cabinet.

Specific Cabinet and Box-level Designs

Cabinet and box-level design is perhaps the most difficult part of computer design, yet it is perceived (incorrectly) to be trivial and hence deferred until last. It is also misunderstood because there is not a single discipline or complete set of well understood laws that govern the design. This deception arises from the fact that, on first glance, the only physical law is that not more than one thing can occupy the same space at a given time. However, a number of disciplines must be understood, including acoustical engineering, heat transfer, aesthetics, human engineering, RF engineering, and the understanding of package functionality. In addition to this understanding, each box type requires a separate specialized manufacturing process.

Obviously no one person fully understands all the disciplines necessary to correctly design a package. Herein lies the problem: _Packaging is the integration of a number of separate disciplines_. One needs to look at the basic packaging problem by first ennumerating the possibilities for placing modules within a particular box package. By being quite restrictive, one can build a grammar with six production rules that describe the possible DEC boxes. Even with this simple grammar, about three million possibilities can be generated. The large (combinatorial) number of possibilities arise from the basic size and way a box is held, the console mounting, cooling, power supply location and the way modules are mounted within the box. The packaging possibilities can then be explored by simply writing statements that express the alternatives for the separate decision dimensions.

A Grammar to Generate Three Million Box/Cabinet Alternatives

1) Size and Mounting

The box is (1/2 or full) cabinet depth by (3-3/4", 5-1/4", 10-1/2", 15-3/4" or 21") high, mounted with (fixed slides, right hand hinge, or left hand hinge).

2) Console

The console is (non-existent, simple power on/off, maintenance only, full feature maintenance/programming).

3) Cooling

The cooling is carried out by (box-level or cabinet-level)(fan or fans) producing (plenum or forced) air flow that is (normal or parallel) to the module mounting axis, which has eight possible orientations, or else the cooling is carried out by natural convection with air entering at the (top, bottom, right side, left side, front, or rear), and exhausting through the (top, bottom, right side, left side, front, or rear).

4) Power Supply Location

The external power supply is mounted (separately, attached behind the box, attached to the back of the cabinet, or attached on the front of the cabinet), or an internal power supply is mounted using a mounting scheme that is the (same as or different than) that of the modules, and is located at the (top, bottom, right side, left side, front, or rear) of the cabinet.

5) Module Mounting

There are (no, one, or many) backplanes for mounting modules.

6) Module Pins

The module pin axis is (top, bottom, right side, left side, front, or rear) mounted, oriented in the (horizonal or vertical) plane with the IC side facing (up, down, right, left, front, or rear).

One can state the six sentences (1 point in the decision space) that describe the PDP-8/A:

1. The box is 1/2 cabinet depth by 10-1/2" high with fixed mounting.

2. The console is for programming and maintenance.

3. Cooling is carried out by box-level fans giving forced air cooling flow parallel to the module mounting axis.

4. The box power supply uses a different mounting scheme and is located at the bottom of the box.

5. There is one backplane for mounting modules.

6. Modules are rear (wall) mounted, oriented in the horizontal plane with IC side facing up.

Of all the dimensions to consider in the design, perhaps the most important is how the box (or module mounting structure) is placed in a cabinet. This placement effects air flow, shippability, configurability, cable placement, and serviceability, and is a classical case of design tradeoffs. The scheme that provides the best metrics such as packaging density and weight, may have the poorest access for service, and the most undesireable cable connection characteristics. These characteristics are given in the following table:

Table __ Fixed, Drawer and Hinged Box/Cabinet Mounting

	SERVICE ACCESS	CABLING	DENSITY	COOLING	APPLICABILITY
Fixed	Good for either backplane or module, but not both unless a thin cabinet is used	Best (i.e., shortest)	Good for thin or rear cabinet PS mounting	Best (known)	Box not needed; box can be used
Drawer	One side access	Long + moves	Very high	Can be	High density, self contained
Drawer (with tilt) for service	Good	Longer + much more moves	Very high	Cooled*	
Drawer vertical mounting modules	Very good	Longer + moves	High		
Hinged (module backplane)	Very good	Short	Medium	Good (if fans are fixed to cage)	Separate box is awkward

*Density restricts cabinet airfl

Figure Cab shows the various boxes (top view) as they fit within a cabinet profile. In some cases there is no two-level cabinet-box hierarchy.

Packaging is largely a matter of designer preference because the only clear goal is manufacturing cost. Marketing considerations, especially for OEM use, drives toward getting the highest density to minimize volume, floor space and rack mounting height. Despite this, many designers prefer packaging schemes that are not the highest density, but rather are fixed so that the power, cooling, and EMI conditions can be well-understood and so that the cabling can be done more rigidly. This has been particularly true for the larger computers such as PDP10's.

Power Supplies

While logical functions can be performed using small quantities of electrons and thus accommodated in very small physical structures, the power to move those electrons at useful speeds comes from power supplies which do not scale down in size as readily as the logical functions they suppport. Power supply technology has not provided the impressive increases in capability per dollar or capability per cubic foot that semiconductor technology has. Power supplies involve such materials properties as voltage breakdown limits, dielectric constants, magnetic permeability, and heat conductivity. Since these properties vary with physical dimension, increased capabilities in terms of voltage breakdown rating, capacitance, inductance, or heat dissipation are gained by making the component physically larger.

The performance criteria for power supplies are predominantly driven by the application for which they are designed. These criteria are given in terms of various efficiencies of volume, weight, power conversion, and cost. It is somewhat difficult to compare the various supplies, as they are all available at different times, produced in different quantities, designed for different reliabilities, and available with different features.

For the computer industry, power supplies can be divided into three main categories: processor and memory power supplies, disk and tape power supplies, and terminal power supplies. Each of these product categories has a unique set of reguirements, which are summarized in Table CVPST.

Table CVPST: Characteristics of Various Power Supply Types

	Processor & Memory	Disk & Tape	Terminal
Power Requirements	250-2500 watts	100-500 watts	0-150 watts
Use	logic	very low noise for head electronics; high current for servos	high voltage for CRT; high current for mechnical motions
Quantity in System	low to medium	medium	high
Typical Cost Sensitivity	low	medium	high
Size	important, especially in boxed computers	not important	very important
Weight	relatively unimportant	not important	very important
Reliability	very important	important	important

Features	power line sensing battery backup

Three of the four efficiency measures, cost (in relative cost per watt), weight (in watts per pound), and volume (in watts per cubic inch), are plotted for processor power supplies in Figures P1 and P2. In Figure P1 the plots use a time axis, and in Figure P2 the plots use a watts-of-output axis. The fourth efficiency measure, power conversion (in watts out per watts in) is given in Figure P3, using a time axis.

The cost of a power system is very dependent on the unit electrical size and technology. The features required on the units such as power monitoring (AC low, DC low), battery backed-up power, and servicing aids also significantly influence the cost. Since the cost is size dependent, a relative metric, dollars per watt, is chosen for processor power supplies.

In the cost characteristics there are different bands of cost curves which are technology dependent. These bands of curves span new, mature, and obsolete technologies. For example, the cost of power supply technology until just recently was very dependent upon the prices of iron and copper and the cost of labor. Now, costs of power supply technology tend to track semiconductor costs due to the widespread use of line switching power supplies. There are also bands within the cost curves; these represent the size dependency; larger power supplies are the most cost effective, with one exception (figure P1a and P2a).

The size of power supplies for minicomputers has been important, especially

[illegible]

Features power line sensing
battery backup

Three of the four efficiency measures, cost (in relative cost per watt), weight (in watts per pound), and volume (in watts per cubic inch), are plotted for processor power supplies in Figures P1 and P2. In Figure P1 the plots use a time axis, and in Figure P2 the plots use a watts-of-output axis. The fourth efficiency measure, power conversion (in watts out per watts in) is given in Figure P3, using a time axis.

The cost of a power system is very dependent on the unit electrical size and technology. The features required on the units such as power monitoring (AC low, DC low), battery backed-up power, and servicing aids also significantly influence the cost. Since the cost is size dependent, a relative metric, dollars per watt, is chosen for processor power supplies.

In the cost characteristics there are different bands of cost curves which are technology dependent. These bands of curves span new, mature, and obsolete technologies. For example, the cost of power supply technology until just recently was very dependent upon the prices of iron and copper and the cost of labor. Now, costs of power supply technology tend to track semiconductor costs due to the widespread use of line switching power supplies. There are also bands within the cost curves; these represent the size dependency; larger power supplies are the most cost effective, with one exception (figure P1a and P2a).

The size of power supplies for minicomputers has been important, especially

for the boxed versions. The volume occupied by logic has gone down for the constant functionality computer; however, power requirements have declined far less than logic volume, and hence power densities have increased. Whereas 250 watts used to suffice for a 10 1/2 by 19 by 25 inch box, 800 watts is now required, and the space for the power supply has barely increased. This has put substantial constraints on the weight and efficiency of power systems, and at times space utilization has been (inadvertently) traded off against cost, manufacturability, and serviceability.

In response to these space pressures, there has been a constant gain in volumetric efficiency (Figure P1b) over the years with the highly dense power supplies on the top of the band and the modular packaged units on the bottom. With the introduction of line switching power supplies, this curve made a quantum jump. The increase in volumetric efficiency, plotted relative to time in Figure P1b, is plotted relative to output wattage in Figure P2b.

Power supply technology not only determines volunetric efficiency, it also determines the weight of the unit. Here again the use of high frequency line switcher technology rather than low frequency transformer technology has produced marked results - in this case, two distinct curves.

The weight efficiency (Watts/lbs) is fairly constant over the years and showed a slight improvement as larger supplies were built (figure P2c).

Finally, Figure P3 shows how power supply efficiency is improving with time. Note that with direct line switching, efficiencies of 70% are to be expected. This efficiency permits the aforementioned increase in volumetric effciency since there is less heat to dissipate.

Modules

Since the function of modules is to interconnect and hold components, the metrics for modules are area (for mounting the components) and the cost of each circuit interconnection. For minicomputers, the emphasis has been to have larger modules with more components packed on a module as a means to lower the interconnection cost. Figure Mod.size shows the area of DEC modules and the number of external pins per module versus time. Since the integrated circuit densities have been always increasing, in effect providing lower interconnection costs, a given module automatically provides increased interconnects simply by packaging the same number of ICs on a module. Obviously, one does not want to credit this effect to improved module packaging. By increasing the components per module, the cost per interconnect can be reduced provided the cost to test the module increases less rapidly than the increase in components. The emphasis on module size is usually most intense for larger systems, where a relatively large number of modules are needed to form a complete system.

Until recently, the increase in module area was accompanied by increases in the number of pins available to interconnect to the backplane. In the case of the VAX 11/780 and the PDP-10, the number of pins did not increase significantly over previous designs, although the board area was 50% larger. In these cases, the number of ICs able to be cooled limits the density. In other cases either the number of pins or the module size limits the module's functionality. There are similar effects throughout the generations.

In the early second generation Systems Module designs, the number of pins and the circuit board area (in square inches) were about the same. Components were fairly large and loosely packed on modules. With the Flip-Chip series, circuits were modified to pack more, smaller, components on a single module, using automatic component insertion equipment, and some of the space consuming components (e.g. pulse transformers) of the earlier circuits were removed so that a module design was a better balance between area and pins. As a result, the early second generation Flip-Chip modules had higher packing densities than comparable Systems Modules.

With the beginning of the third generation, the need for more printed pins to the backplane was clear, since so many interconnections were made on the computer's backplane. The PDP8/I was the first DEC integrated circuit computer, and the packaging philosophy strictly followed that of the second generation. As a result, the sudden increase in component functions made the modules drastically lacking in pins. By putting pins on both sides of the module, the number of pins for a double height module (20 square inches)was increased from 36 to 72, but was still inadequate. Assuming that each IC has 14 signal pins and a module had 70 signal pins, only 5 ICs could be placed on a board and still have pins brought out to the backplane pins, although the twenty square inch area of the module could potentially hold 20 ICs.

Although the 8/I was packaged using the twenty square inch 72-pin modules, it was clear that another packaging scheme was necessary to utilize ICs, modules, pins, and backplanes. Thus when the 11/20 and 8/E were designed (about 1970), they used larger modules in order to carry the large number of intra-module interconnections required when many ICs were placed on a single module.

It is interesting to note that in the recent case of high density ICs, the LSI11/2, the module area was too large to have a single option on a module, and since the LSI11 bus only required a few signals, the number of pins was more than adequate. Here the modules are functionality limited, versus pin limited. The figure attempts to annotate situations as to whether pins or modules limited the design.

Although the size of the module is important in determining the systems that can be built, how they are serviced, and how they are manufactured, the important module metric is the cost per interconnection on the printed circuit board (and remainder of the system). Figure Mod.cost shows how this has varied with time. Here one can see that the introduction of Flip-Chip modules initially increased costs (because learning almost had to start over again).

Interconnection costs consist of the printed circuit board, the cost to insert the components on the module and the cost to test the module. Printed circuit board costs have been decreasing with time, reflecting both benefits of learning and the benefits of placing more ICs on a single module, giving a compound economy of scale effect. The cost to assemble the components on the module have decreased rapidly, reflecting the increasing use of automatic component insertion machines. Testing has not been a significant cost component in module manufacturing. (It does represent a substantial cost by the time the module has been integrated into a system at the customer's site. This is because DEC tests modules not only as modules, but also as sub-assemblies, and as parts of systems both in the factory and in the field.) The total cost per interconnection has been decreasing, but the trend may either stay constant or even increase as greater use of LSI decreases the number of total connections in a system, but makes the remaining interconnections more expensive to assemble and test.

Backplanes

Backplanes provide the next level of integration packaging above modules and are used to hold and interconnect a set of modules which form a computer or an option (e.g. processor, memory, ot peripheral controller). Figure Backplane.cost gives the relative cost of interconnection of backplane module pins (using the same scale as Figure Mod.cost). Here the cost per interconnection is roughly the same as with a printed circuit module interconnection. This can be somewhat misleading because backplanes require a negligible cost for testing and few failures occur during testing.

The figure shows various kinds of interconnection technologies. Even though there are exponential increases in quantities produced, the cost continues to increase in the long run, with only occasional downward steps in cost. The greatest cost decline occurred when interconnections were carried out using automatic wire wrap machinery, but the 8/E was equally significant by using a completely wave-soldered backplane. This figure is also useful in seeing how effectively the module pins were used (i.e. whether all available pins were used). Note the phenomenon described with the 8/I, where modules were clearly pin limited, shows up clearly.

Boxes and Cabinets

Since the function of the cabinet and box is to hold backplanes which in turn hold modules, which in turn hold circuit-level components, the metric of electronic enclosures is the number of printed circuit boards they hold. The earliest method of mounting was to place the backplanes directly in a six foot high cabinet which held 19 inch wide equipment in a 22 inch by 30 inch floor space and weighed about 185 pounds. Figure Cab (p. 83) shows the top view of the various cabinets used to hold module backplanes and boxes for minicomputers since 1960. The changes to the basic DEC six foot cabinet have been mainly for improved producibility. The latest (circa 1973) was to use riveted upright supporting members so the cabinet could be assembled easily without requiring bulk space for shipment and storage.

The original cabinet used the entire cabinet as an air plenum so that air was forced between the modules and out the front doors. When the PDP-7 used the same cabinet and the module mounting frame cut off the air flow, it was necssary to add fans to the back doors to blow air at the modules. Since cooling was one of the weak points in the 7, the 9 used a self-contained mounting and cooling structure in which air was circulated between the modules, with air pulled in from outside without going through the cabinet.

A second, later packaging method, initiated with the PDP-8, packaged the metal boxed minicomputer inside the six foot cabinet. Figure Boxes shows the significant boxes that have been used to package minicomputers both within the six foot cabinet and freestanding. The box packaging history begins with the PDP-8. The rows of the figure indicate the four ways (see page 00) that are available to access the circuitry (fixed, book, slides, and tilt for access). The 8 design was followed by the 8/S design which oriented the modules with the pins up for access to the backplane. By tilting (rotating) the box the handle side of the modules could be accessed. For the 8/I (not shown), modules were mounted in a vertical plane.

Several fixed backplane module mounting structures were formed beginning with the 8/A (1975). Note that over ten years have elapsed since minicomputers were mounted in a fixed structure in a cabinet (i.e. PDP5).

Heat

Although the volumetric measures of module area, and size of the cabinet are also important, the amount of heat the enclosure is capable of dissipating is the most important because of reliability. The following table gives some of the important metrics of several of the recent DEC computer boxes:

Table EBC: Expansion Box Characteristics

Box Model	Used On	Year	Weight (Lbs.)	Size (Cu.Ft)	Modules	Module Volume (Cu.Ft)	Heat In	Heat Density (KW/Ft3)	Space Efficiency
BA11-D	11/35	1974	100	2.6	24 hex	.93	0.7	.27	.35
BA11-E	11/45	1972	100	2.6	27 quad	.7	0.7	.27	.27
BA11-F	40&45&70	1972	260	5.3	44 hex	1.7	2.2	.42	.32
BA11-K	04&34&70	1974	110	2.6	24 hex	.93	1.0	.38	.36
BA11-L	11/04	1976	50	1.3	9 hex	.35	.55	.43	.27
BA11-M	11/03	1975	25	0.5	4 quad	.1	.25	.54	.24
BA11-N	11/03	1977	40	1.0	9 quad	.23	.24	.31	.22
BA11-P	11/60	1977	100	3.0	29 hex	1.1	1.1	.36	.22
BA8-CA	8/A	1975	117	2.4	20 quad	.52	1.2	.50	.22
H9300	8/A	1977	55	1.1	10 quad	.26	0.3	.26	.24
H9500	11/780	1978	344	43.4	67 ext. hex	3.7	6.0	.15	.10

Figure Heat.den gives the heat density for the various boxes. The amount of heat dissipated by the box is in terms of the kilowatts per cubic foot and has been relatively constant with time. There has been a great variation about a norm, and the very high heat dissipation of the first 8/A (due to high packing density and a relatively inefficient power supply) affected the next design to a lower density. The space utilization shows a similar picture, although the efficiency appears to be declining (see Figure Space.util). This decline is hardly noticeable and is even surprising in light of more efficient power supplies which make it possible to place more components in a given enclosure. The cost effectiveness of the average enclosure, as measured by the material cost, is declining with time as measured by the relative cost of materials per cubic foot of modules held (Figure Cost.payload).

The time chart gives a completely erroneous view of the situation, because economy of scale is not considered. Figure Box.cost shows how the relative cost of box materials varies with the volume (in number of hex modules). Here the upward trend of the previous figure is not apparent, but it merely occurs because later packages are for smaller numbers of modules.

AN OVERVIEW OF MANUFACTURING

Although the result of a design project is an entity which is manufactured, very little is written about manufacturing in computer engineering literature. Such literature generally discusses algorithms, logic design, and circuit technology. Yet for a computer to be commercially successful, it must be manufacturable, economically operable, and serviceable. Moreover, for most of the computer engineering discussed in this book, because the designs are intended for volume production, engineering costs are small (1-10%) compared with other product and lifecycle costs. The product cost is determined by the price of the components and the manufacturing process~~.~~; ~~The~~ the ~~lifecycle cost~~ stet ~~includes the purchase price, the operational costs, and service costs.~~ ¶ For production, machines must be easy to assemble and test, repair must be rapid, engineering changes must be introduced smoothly, and the production line cannot be held up because of shortages of components -- all parts of traditional manufacturing understanding.

A detailed discussion of manufacturing is clearly beyond the scope of this text. Information on test equipment is to be found mainly in the manuals published by test equipment manufacturers. References on quality control include [Juran, 1962] and [Grant and Leavenworth, 1972].

The Life Cycle of a Product

Figure Lifecycle shows a simplistic process flow for the major phases and milestones in the life of a product. In reality, planning and designs for many of the phases go on concurrently. The early research, advanced development, and definitional phases are not shown. Often, products proceed from the idea stage to the engineering breadboard and are then terminated because they do not meet original goals or because better ideas arise.

To accomodate changes, the engineering breadboard is usually built with wirewrapped boards rather than printed circuit boards, if the circuit technologies used permit the long wire lengths characteristic of wirewrapped boards. At or before the breadboard stage, schedules are made for manufacturing start up. Other organizations formulate and execute plans: systems engineering - for product test/verification; software engineering - for special software and verification; marketing - for promotion and product distribution; sales - for training; field service - for training and parts logistics; and software support.

After the engineering breadboard has been debugged, construction of engineering prototypes begins. The engineering prototypes prove the design using the actual printed circuit modules that will be used in manufacturing. Usually a number of prototypes are constructed, the number varying from ten to a hundred depending on the complexity, cost, and anticipated product volume. All processors and peripherals in the planned systems configurations are tested in conjunction with the prototypes. The complete system must meet the product specifications and must run all of the system software.

The requirement that all of the system software be run is an excellent supplement to the normal testing of prototypes, and is especially useful when the product being designed is a processor with a mature architecture, because more software is available for use in testing the prototype. Since the number of possible states and state sequences in a computer system is very large, a diagnostic test which exercises every one is impractical. Diagnostic programs and microdiagnostics therefore test a judiciously chosen subset of all states. Such programs are not perfect, however, and therefore system software is run as well. Thus, the more software that is available to test a prototype, the less likely it is that a design error will go unfound. The general problem of testing requires much more work before it can be considered mature. One would like to see, for example, the automatic generation of verification programs from an ISP description of the architecture being built.

Limited release (LR) to manufacturing is a major milestone. The product is placed under formal engineering change control; specifications and documentation are available for the product and manufacturing process. For the integrated circuits, sources of supply and testing procedures are in place. Process control tapes are ready for the numerically controlled machine tools, such as component insertion, backplane wiring, and printed-circuit board drilling machines. Any special tooling for the mechanical packaging has been obtained. Testing at all levels has been specified; test programs for computer-controlled testers have been written, special test equipment has been built, and diagnostic programs are ready.

Design maturity testing (DMT) with a number of engineering prototypes verifies the design and justifies the risk of a pilot run. Tests for reliability and functionality are conducted. Environmental testing (shock, temperature, humidity, static discharge, radiation, power interrupt, safety) is conducted at this stage.

The pilot run shakes down and verifies the actual manufacturing process by building a small number of units, at the manufacturing plant, using the product and process documentation.

Product announcement usually occurs during the DMT period but can occur at any time - often as early as limited release or as late as first customer shipment, depending on the marketing strategy. This strategy is clearly a function of the volume, novelty, and competitive needs.

Process maturity testing (PMT) verifies that the product is being manufactured with the desired cost, quality, and production rate. After PMT, the steady-state phase of manufacturing continues, with possible perturbations due to the introduction of product enhancements or process changes to lower product costs, until the product is phased out.

Manufacturing Process Flows

An overview of a manufacturing process is given in Figure Manufover, which shows how a product moves through the various factories. There are often different plants for boards, peripherals, memories, and central processors. Integration from the other stages and stock storage occurs at the stage called final assembly and test (FA&T). Here, the software system that is to be run, operations manuals, and other documentation is also integrated and tested.

Figure process60 gives the complete flow for a typical volume manufacturing line -- the 11/60 central processor facility in Aguadilla, Puerto Rico. Integrated circuits are manufactured elsewhere and are sent to be inserted, soldered in, and tested in the module manufacturing plant in nearby San German.

The manufacturing process includes extensive thermal cycling to ensure that component "infant mortality" cases are discovered early during manufacturing, because it is more expensive to find defective components at the later ~~larger~~, more integrated systems level. The temperature/humidity environmental chambers which house twelve or sixteen CPU's each are shown in Fig. TChamber. Figures 2224 and qvstation show small heated enclosures used to induce failure during the test and repair of modules.

Since testing occurs at each stage in the manufacturing process, dedicated logic must be added to the design, to provide physical access "probes" for the test equipment. To test a particular function, it must be specifiable, invokable, and observable. For example, the function of an adder can be clearly specified,, but it cannot be easily invoked or observed if its inputs and outputs are etch runs on a printed circuit board. Several testing strategies are used: add signal lines from the adder to the backplane where there are adequate probe access points; probe directly onto the module etch or IC pins; and subsume the adder in a function whose inputs and outputs can be more easily controlled and observed. The problems of observation and control exist at all levels of integration. Examples of observation points at each level are given in Table Obslevels for the 11/60 computer.

The problem of testability must be addressed at design time. Providing access for testing always incurs added product cost (extra logic and module pins or circuit pins), but lowers manufacturing cost and field service costs. As gate density per chip continues to increase, the problem worsens. One solution, which is economical in i/o connections, is to design every storage element as a shift register which can be loaded in parallel (normal mode) or serially loaded (with an invoking state) or serially read (with the state to be observed). Eichelberger and Williams [1977] report on such a scheme for gate array designs. The individual shift register latches are connected to form one or more independent shift registers which are connected to the leads of the gate array package. At the module level, An example of using shift registers to conserve I/O pins is found in the VAX 11/780. Eight test points are fed to a shift register whose serial output is fed to a module pin. Twenty modules are then chained. The console subsystem, an LSI11, controls the reading of the shift registers and interperts the results.

In Fig. QVstation,, the function being tested is a complete CPU which is packaged on several printed circuit modules. Behavior is being induced by PDP-11 programs and observed by inspect~~ing~~ion of the results in memory (manually or by ~~using~~ a diagnostics program ~~or manually~~). However, a lower level of behavior can be observed (on the special display panel at the right) and controlled (by varying the clock rate of the CPU). The lights on the display panel are driven from backplane signals and show the contents of certain registers, e.g., micro-instruction register, program status register, and the ALU output register.

Table Obslevels: Examples of Observation Points at Each Structural Level for the PDP11/60 Computer

Level in computer hierarchy	**Observation point**	**Stage in manufacture of computer**	**Example**
electrical circuit	transistor contacts on metallization layer	semiconductor fabrication	wafer test with micro-probe
switching circuit	leads on I.C. package	incoming inspection of ICs	IC tester
register transfer	etch run	module	probe on PC board (module-specific test using GR tester)
register transfer	backplane	module	2224 memory

			exerciser for cache
PMS (Pc)	Unibus	CPU	Unibus voltage margin tester
PMS (Pc)	contents of memory	CPU	diagnostic programs at subsystem level, e.g., memory-management unit, or processor instruction set tests
PMS (C)	contents of memory	System integration	peripheral diagnostic programs
PMS (C)	Unibus	System integration	~~DECX-11~~ bus exerciser

In both Figs. QVstation and FA&T, the entire instruction set is tested; a central computer system loads the diagnostic programs and monitors their execution. Several hundred lines emanate from this system to all of the computers under test throughout the production line. The Unibus of the processor under test serves as the umbilical cord [Chapter 00, p. 00] to accept the diagnostic programs and to be monitored by making memory observations.

Modules are tested using several methods. One method (see step ~~a~~ A of Fig. Process60), uses the GR tester of Figure GR. Here, every signal input and test point on the module is probed using a fixed "bed of nails" test probe. The tester then selects the desired input and test point. In another method, (see step ~~b~~ B, Fig. Process60) the module under test is placed in a working processor to be verified.

Following the inter-plant transfer, systems integration begins. Figure FA&T shows an 11/60 at the Westminster Plant. For some computer systems, Final Assembly and Test (FA&T) is carried out by having an individual technician follow a system through each of the three phases: incoming test of the CPU cabinet,assembly, integration of peripherals, and final acceptance. Thus one or more technicians have individual responsibility for fabricating a single system in what is a traditional, hand-crafted, job-shop fashion.

Alternatively, FA&T is carried out in what is fundamentally a continuous, production line environment called the Common Systems Integration (CSI) line. Here, workstations exist at each stage of the production line, and are interconnected by conveyors.

Acknowledgements: Dave Widder, Lorin Gale, Hank Schalke, Memorex Corporation, Bob Peyton, Russ Doane, Steve Teicher, Joe Smith, Dick Gonzales.

"Statistical Quality Control", Grant and Leavenworth, Fourth Edition, McGraw-Hill, 1972.

"Quality Control Handbook", Juran, J. M., McGraw-Hill 1951; Second Edition, 1962.

Louise please insert in order in ~~[illegible]~~ list of references ~~[illegible]~~ below

Acknowledgements

We gratefully acknowledge the following ~~valuable~~ colleagues who provided data for this chapter and ~~critiques~~ critique of earlier drafts: Russ ~~E~~ Doane, Rony Elia-Shaoul, Lorin Gale, Dick Gonzales, Bob Peyton, Jim Scanlan, Henk Schalke, Joe Smith, Steve Teicher, and Dave Widder.

~~Memo~~ We ~~are~~ thank Memorex Corporation for ~~permission~~ ~~Memorex Corporation~~ ~~to use~~ Figures Memtree, Area. Digital, Large. Disk. Price & Mem trends.

[Heidi: does this go here or in front matter?

~~Bhandarkar, D. P. and Juliussen, The Impact of Semiconductor Technol on Computer Systems.~~

Braeckelmann, W., Fritzsche, H., Kroos, F-K., Trinke, W., and Wilhelm, W.
"A Masterslice LSI for Subnanosecond Random Logic", Digest of Technical Papers, 1977 Ent. Solid-State Circuits Conference, pp. 108-109.

Bhandarkar, D.P "Dynamic MOS Memories: Serial or Random Access?" IEEE Compcon Spring 1978, Digest of Papers, pp 162-164, Feb 1978.

Denning, P. J. "The Working Set Model for Program Behavior", Communications of the ACM, 11, May 1968.

Denning, P.J. "Virtual Memory", Computing Surveys, Sept. 1970, pp 153-189.

Eichelberger, E.B. and Williams, T. W.
"A Logic Design Structure for LSI Testability", Proceedings of the 14th Design Automation Conference; June 20-22, 1977.

Fusfeld, A. R. The Technological Progress Function Technology Review, Feb. 1973, pp. 29-38

Gaskill, J.R., J. H. Flint, R. G. Meyer, L. J. Micheel and L. R. Weill,
"Modular Single-Stage Universal Logic Gate, "IEEE Journal of Solid-State Circuits, SC-11, 4, p. 529-538, 1976.

Hodges, D.A. A Review and Projection of Semiconductor Components for Digital Storage, Proc IEEE, 63, 8, August 1975, pp 1136 - 114

Hodges, D. A., Progress in Electronic Technologies for Computers, National

Bureau of Standards Report T73219, March, 1977.

Kilburn, T., D. L. G. Edwards, M. J. Lanigan, and F. H. Sumner, "One-Level Storage System," IRE Trans., Vol. EC-11, No. 2, pp. 223-235, April 1962.

Landman, B. S. and Russo, R. L. "On a Pin Versus Block Relationship for Partioning of Logic Graphs", IEEE Transactions on Computers, C-20, 12, Dec 1971, pp 1469-1479.

Liptay, J. S., Structural Aspects of the System/360 Model 85 - The Cache, IBM Systems Journal, 7, 1 (1968).

Logue, J. C., N. F. Brickman, F. Howley, J. W. Jones, and W. W. Wu, "Hardware Implementation of a Small System in Programmable Logic Arrays," pp. 110-119, March 1975.

Luecke, J. Overview of Semiconductor Technology Trends; IEEE Computer Society International Conference, Fall, 1976, pp. 52-55.

Masaki, A., Harada, Y., Chiba, T. "200 Gate ECL Masterslice LSI", ISSCC Digest of Technical Papers, pp. 62-63, Feb. 1974.

Moore, G. E., Microprocessors and Integrated Electronic Technology, Proc. IEEE, 64, 6, June 1976, pp. 837-84.

Nakano, T., O. Tomisawa, K. Anami, M. Ohmore, I. Okkura and M. Nakaya, "A 920 Gate Masterslice", Digest of Technical Papers, IEEE Solid-State Circuits Conference, 1978, pp. 64-65.

Noyce, R. N. "Large Scale Integration: What is Yet to Come?", Science, Volume 195, Number 4283, 18 March 1977, pages 1102-1106.

Noyce, R. N. "Microelectronics", Scientific American, Volume 237, Number 3, September 1977, pages 62-69.

Parke, N. G., Private Communication, 1978.

Patil, S.S. and Welch, T. An approach to using VLSI in digital system 5th Annual Symposium on Computer Architecture, April 1978, pp 139-143

Phister, M. "Data Processing Technology and Economics", Santa Monica Publishing Co. Santa Monica, Calif. , 1974.

Russo, R. L. "On the Tradeoff Between Logic Performance and Circuit to Pin Ratios for LSI", IEEE Transactions on Computers C-21, 2, February 1971, pp 147-152.

Strecker, W.D. "Optimal Design of Memory Hierarchies", Proceedings of the Hawaii Systems Conference, 1978.

check → Toombs, D. Private Communication, 1977

Turn, Rein, Computers in the 1980s, Columbia University Press, N. Y., 1974.

vonHippel, E. The Dominant Role of the User in Semiconductor and Electronic Subassembly Process Innovation; IEEE Transactions on Engineering Management, vol. EM-24, No. 2, May 1977. pp. 60-71.

Wilkes, M.V.

Chapter 2 Figures and Tables (that are to be made as figures)

To α on next page Memtree	Family tree of memory/logic technology (courtesy of Memorex Corporation, Puthuff, 1977)
LMtimeline	Logic and memory technology evolution time line
ICtree	Family tree of digital Integrated Circuit functions
ALU	Arithmetic-Logic Unit logic diagram
ALUFCN	74181 Function table
AMD	AMD 2900 4-bit microprocessor slice block diagram (registers and data path)
Semiden	Component (transistor) per single Integrated Circuit die versus time
Compprice	Price per gate versus time
Gatecomp (table in text)	Circuit components (transistors) required for various logic functions in various technologies -- show also rom, ram, pla, ccd, flip flops, gates 2900, LSI-11, CMOS-8; includes N, P, IIL, CMOS, ECL< TTL, TTL/S
Memprice	Cost per bit of Integrated Circuit memory versus time (Noyce-Sci.-Am.)

~~Generality~~	~~Generality (use) and price of an Integrated Circuit versus its complexity~~
Reliab	Failure rate of silicon Integrated Circuits
Design.cost	Current design cost (or time) versus circuit density using various design methods
Semiuse	Manufacturing cost versus LSI circuit density for various design techniques
PLA	Signetics field programmable logic array (FPLA)
α → Memsizeperf	Memory size versus access time for various memories and yearly availability (Toome's data, TI)
Core.cost	Cost per bit of core memory estimated by various market surveys and future predications
Areal.digital	Areal density of various digital magnetic recording media (courtesy of Memorex Corp., 1978)
~~Areal.analog~~	~~Areal density of various analog magnetic recording media (courtesy of Memorex Corp., 1978)~~
Large.disk.price	Price per bit of large, moving head disks and semiconductor

	memories (courtesy of Memorex Corp., 1978)
Mem.trends	Memory trend 1975-1980 (courtesy of Memorex Corp., 1978)
Tapechar	Characteristics of various IBM magnetic tape units versus time
x ~~Tapeprice~~	Price of magnetic tape units (1978)
CRTprice	~~Price of various cathode ray tube terminals versus time~~
CPI	Consumer Price Index using 1967 as base
CPI.log	~~Consumer Price Index (plotted on semi-logarithm graph) using 1967 as base~~
Mkt~~cycle~~ Turbo	Technology progress functions for turbojet engines
~~Perf.size~~	~~Performance or size vs price at some time, t~~
Cost vs time	~~Cost of~~
Techvstime	
Cost/tech vs time	

~~Tradeoff System performance constrained by either memory size or performance~~

Product.cost Product cost versus time on the state-of-the-art line showing various situations

Net Cost of various components that contribute to product cost

Product.price Product cost versus time with manufacturing learning

Product.improve Product cost improvement by enhancement of cost/function

~~Note there is a reference to photos elsewhere in the book to illustrate the various packaging concepts..~~

~~Fig. 1. Packaging consists of placing boxes within smaller boxes...within boxes on an indefinite (not recursive) basis.~~

~~Fig. 2. A packaged system provides some function, but in addition gives a visual impression, usually requires cooling of some sort, has certain mechanical characteristics and an Electromagnetic Interface (EMT).~~

Fig. 3. Eight level packaging hierarchy for DEC second to fourth generation computer systems.

Fig. 4. Packaging is a set of closely interrelated design problems.

~~VAX.11/780 View of VAX 11/780 cabinet~~

11/05 PDP-11/05 (first version) Computer box showing modules, backplane (holding 9 hex modules), power supply, and fans.

11/70.1 Major components and assemblies of 11/70 processor and memory mounted in two standard DEC cabinets.

~~11/70.2 11/70 processor with maintenance cards~~

BP Printed circuit backplane cross-section

~~Pwrcable Power distribution cable harness for 11/70~~

~~Lab.sys DEC's first laboratory and systems modules~~

LSI11 LSI11 processor with 8 Kbytes of memory and microcode for commercial instruction set.

~~Fig. 5. Deleted~~

Fig. 6. Time line evolution of computer generations, packaging and classes (with examples).

~~Fig. 7. None~~

~~Fig. 8. None~~

Fig. 9. ~~4706 and~~ 4707 ~~receiver and~~ transmitter line unit~~s~~ of the late second generation.

~~(Get two modules from Mary Jane to photograph)~~

~~Fig. 10. DL11 line unit based on early fourth generation IC. (Need a photo of this module, Mary Jane is working on it.~~

~~Fig. 11. None~~

Fig. 12. DEC physical structure (packaging) hierarchies vs technology generations.

~~Fig. 13. Three views of PDP-8/A box.~~

Fig Cab (from next page)

Fig. P1 a-c Cost, weight, and volumetric efficiencies versus time for various DEC computer power supplies

a) Cost efficiency (in relative cost/watt)

b) Weight efficiency (in watts/lb)

c) Volumetric efficiency (in watts/cubic inch)

Fig. P2 a-c Cost, weight, and volumetric efficiencies versus size for various DEC computer supplies

a) cost and efficiency (in relative cost/watt)

b) Weight efficiency (in watts/lb)

c) Volumetric efficiency (in watts/cubic inch)

Fig. P3 Power supply efficiency (watts out/watts in) versus time for various DEC computer power supplies.

Fig. Mod.size Module printed circuit board area and number of pins per module versus time for DEC modules.

Fig. Mod.cost Relative cost per interconnection on DEC printed circuit board modules versus time.

Fig. Backplane.cost Relative cost per possible and actual interconnection versus time for various DEC computer backplanes; also, pin density (in pins per square inch) versus time.

Fig. Cab Cabinets used to hold various DEC computers (in fixed, book, and box configurations)

Fig. Boxes Boxes used to hold various DEC PDP8 and PDP11 series minicomputers

Fig. Heat.den Heat density (Kw/ft3) of various DEC computer boxes

Fig. Space.util Space utilization (ft3 of modules/ft3) of various DEC computer boxes

Fig. Cost.pay Cost payload (relative cost of materials/ft3 of modules held) of various DEC computer boxes

Fig. Box.cost Relative cost of materials versus number of hex size modules for various DEC computer boxes

Figure

Lifecycle A simplified process flow for the major phases and milestones in the life of a product.

~~*Breadboard~~

Manufover Overview of manufacturing flow.

process60 The process flow for the 11/60 manufacturing plant in Aguadilla, P.R.

~~Bath Chip failure rate plotted against time shows the three behavior regions. Thermal cycling is used to ensure that "infant mortality" cases in the first region are discovered early in the manufacturing flow.~~

*Tchamber Thermal chambers. (photo)

*2224 2224 memory exerciser used for testing cache. (photo)

*QVstation (photo)

GR~~tester~~ (photo)

~~Obs Placement of test points for observation and control.~~

~~*APT Final acceptance prior to inter-plant transfer. (photo)~~

~~CSI The common systems integration area at the Westminster Plant. (photo) -- From article in Materials Handling Engineering, July 1977~~

~~*spurs (photo) [need new photo giving better view of system being built]~~

~~*Waferprobe Probing test points on a wafer prior to dicing. (photo)~~

~~*These are photos -- copies of Polaroid photos are attached for allbut Breadboard and Waferprobe. CM is getting B/W glossies retaken in Aguadilla and Breadboard from the mill.~~

FA&T (photo)