TRENDS IN MICROCOMPUTER TECHNOLOGY

J. E. Bass
Rockwell International
Microelectronic Device Division
3310 Miraloma Avenue
Anaheim, CA 92803

1.0 INTRODUCTION

This paper identifies current trends in the microcomputer industry and projects expected developments over the next five years. The intent of this paper is to give the system designer an insight into the microcomputer tools available to him in new system implementations. Since all product development activities are driven by market requirements, a brief analysis of major microcomputer market segments and their product requirements are given. Then the various system factors of architectural design, software requirements, process technology, increasing system complexity, and pricing are considered. From these, projections for microcomputer characteristics for the 1982 time frame are made.

2.0 MARKET ANALYSIS

Microcomputers were originally developed in 1970-71 for use in desk top calculators. The primary incentive was to reduce the requirements for custom MOS circuit developments which were expensive and time consuming to develop. The original 4-bit microcomputers from Rockwell (PPS-4) and Intel (MCS 4004) served these needs well. Other industries were quick to pick up the microprocessor approach to product design and the industry was off and

running. Today, the marketplace has expanded into four major segments: the market served by single chip microcomputers in consumer type applications; the market served by 4-bit and low cost 8-bit microprocessors in control and transaction products; the market served by 8-bit microprocessors in instrumentation, byte handling, and intelligent terminal applications; and the market served by high performance 8-bit and 16-bit microprocessors in computational applications. Each market segment has unique characteristics which will influence future microprocessor development activities for products directed toward that market.

The microcomputer market projection through 1982 is shown in Figure 1. The sales dollars represent a 25% compound growth rate per year. Taking into account projected price declines, this represents a 40-50% per year compound growth rate in device volume. The total market can be segmented into nine general areas; these are shown in Figure 2. The dollar size of each market segment is shown in Figure 3 to give you a perspective of the potential of each market segment. This potential dollar revenue is the driving force on product development dollars available for microprocessor development. A more interesting way of looking at the market is in terms of the market sizes for each class of microcomputer. Generally speaking, several classes of microcomputers are required to fully address each market segment. For purposes of this discussion, the classes will be defined as (1) single chip devices, (2) 4-bit/8-bit controllers, (3) 8-bit byte handlers, and (4) 8-bit/16-bit

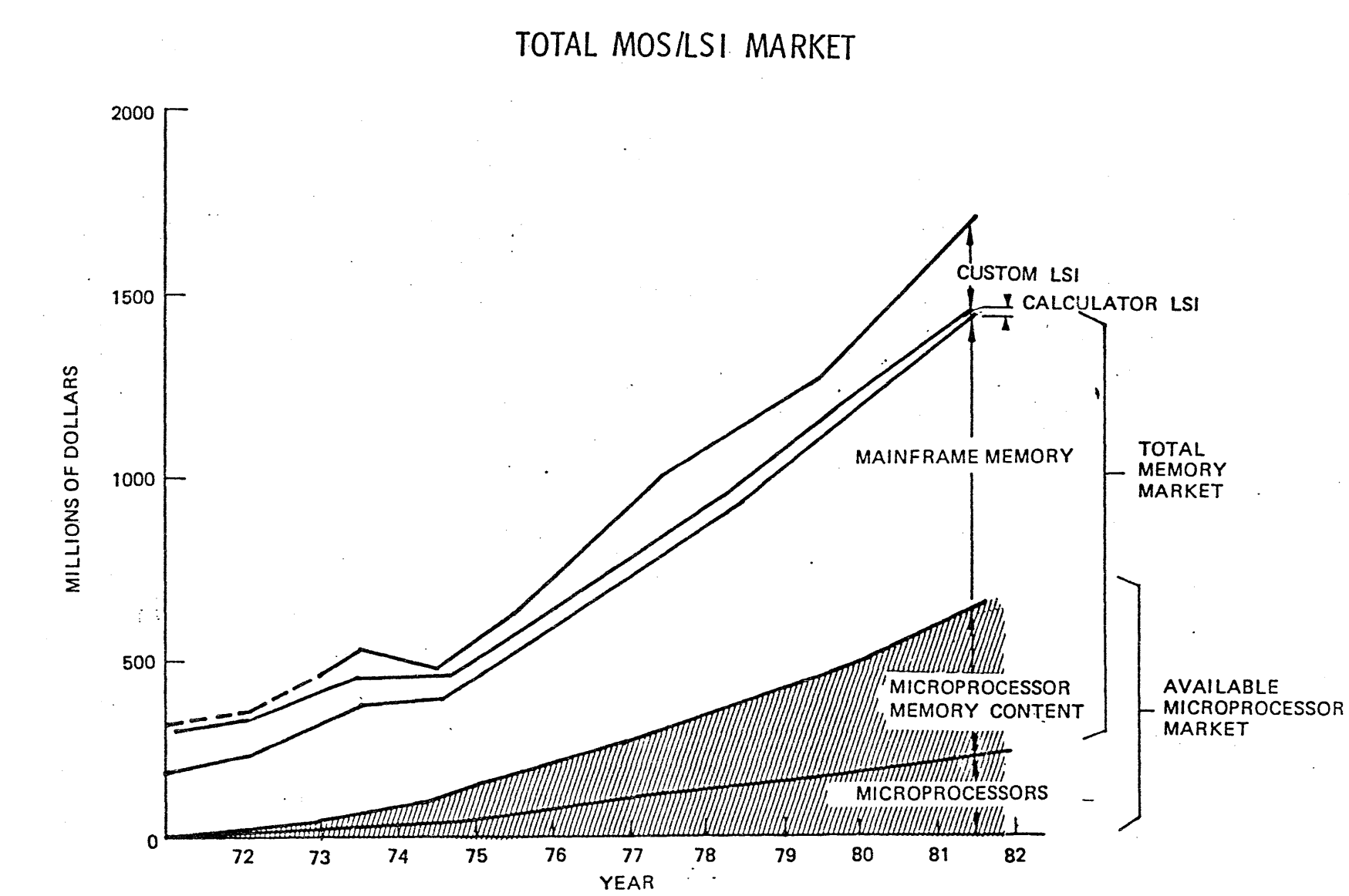

TOTAL MOS/LSI MARKET
2000
1500
1000
500
0
MILLIONS OF DOLLARS
72
73
74
75
76
77
78
79
80
81
82
YEAR
CUSTOM LSI
CALCULATOR LSI
MAINFRAME MEMORY
TOTAL MEMORY MARKET
MICROPROCESSOR MEMORY CONTENT
AVAILABLE MICROPROCESSOR MARKET
MICROPROCESSORS
SOURCES: GNOSTIC CONCEPTS
DIGITAL IC STUDY
MDD FORECAST
TOTAL U. S. PRODUCTION OF MOS PRODUCTS

Figure 1

MARKET SEGMENTATION

TRANSACTION EQUIPMENT	MONEY HANDLING TERMINALS, ARCADE GAMES, MERCHANDISING EQUIP.
OFFICE EQUIPMENT	OFF-LINE BUSINESS EQUIPMENT, FACSIMILE, WORD PROCESSORS
CONSUMER	GAMES, APPLIANCES, PERSONAL/HOME PRODUCTS
E D P	PERIPHERAL CONTROLLERS, INTELLIGENT TERMINALS, MEMORY
COMMUNICATIONS	TELEPHONE, RADIO, PABX, SWITCHING, COMMUNICATIONS, MODEMS
INDUSTRIAL CONTROLS	MACHINE AND PROCESS CONTROLS, MATERIAL HANDLING, TESTING
INSTRUMENTATION	LABORATORY, MEDICAL, TEST EQUIPMENT
MILITARY/AVIONICS	EXTENDED ENVIRONMENT COMMUNICATIONS, NAVIGATION AND CONTROL, WEAPONS SYSTEMS
AUTOMOTIVE	ENGINE, TRANSMISSION, ENVIRONMENTAL CONTROLS, SAFETY, CONVENIENCE

Figure 2

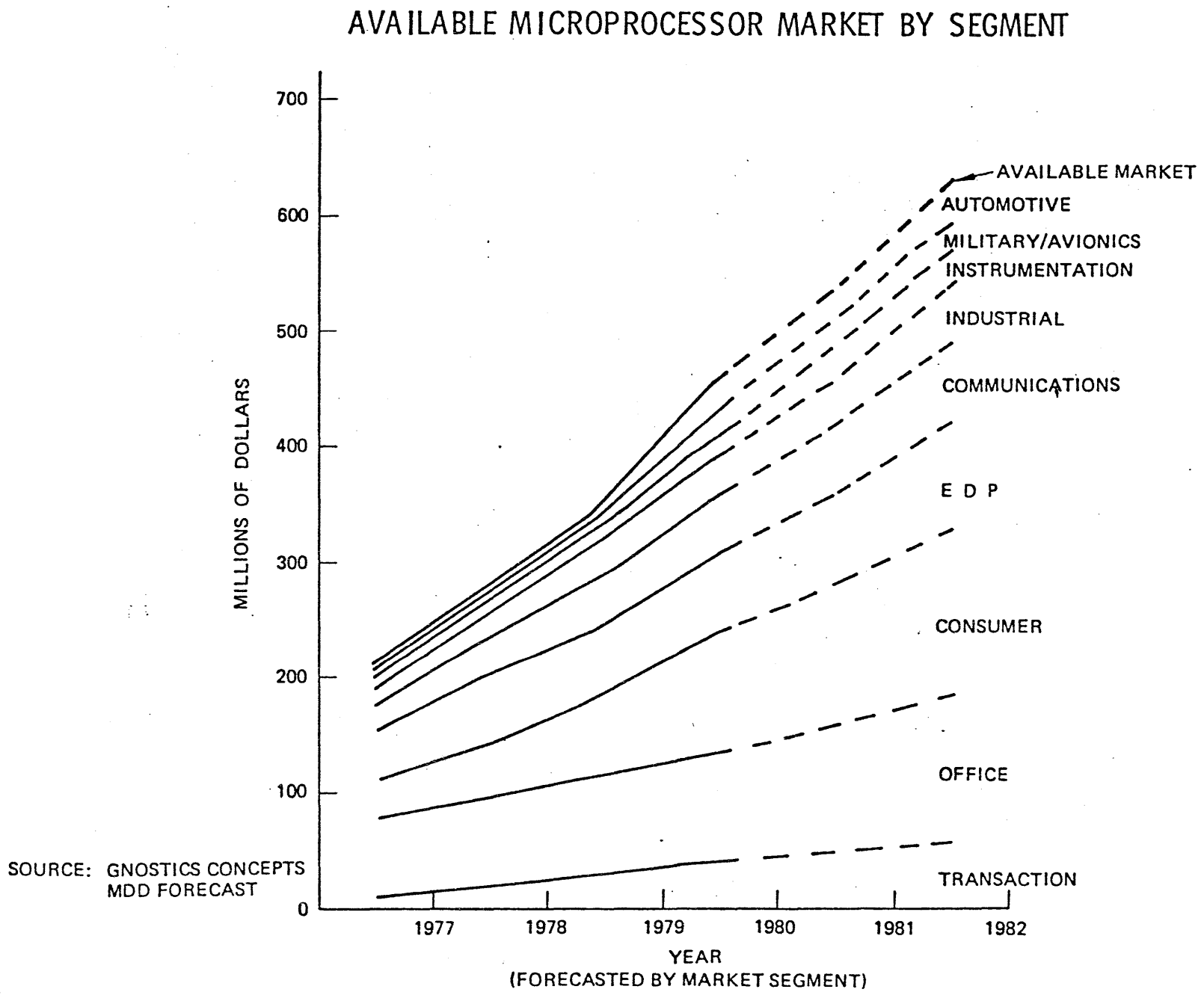

AVAILABLE MICROPROCESSOR MARKET BY SEGMENT
700
600
500
400
300
200
100
0
MILLIONS OF DOLLARS
AVAILABLE MARKET
AUTOMOTIVE
MILITARY/AVIONICS
INSTRUMENTATION
INDUSTRIAL
COMMUNICATIONS
E D P
CONSUMER
OFFICE
TRANSACTION
1977
1978
1979
1980
1981
1982
YEAR
(FORECASTED BY MARKET SEGMENT)
SOURCE: GNOSTICS CONCEPTS
MDD FORECAST

Figure 3

computation oriented microcomputers. The percent of total market dollars for each of these classes for 1977 is given in Figure 4. Of more interest, possibly, is the number of systems shipped for each segment. Shipments of single chip devices in 1977 will be in the one million category; shipments in the 4-bit/8-bit controller area will be in the two to three million category; shipments in the 8-bit byte handler area will be in the 300-400 thousand category, and 8-bit/16-bit computation market will be in the 10-20 thousand category.

The single chip microcomputers represent the newest and fastest growing class of microcomputers. Applications consist of a broad range of products in the consumer and the low cost controls area. The largest single application is currently in microwave oven controllers. These one chip devices provide a very high degree of flexibility in product features over the more commonly used electro-mechanical timers. The important aspect of this example from a systems point of view is that rather powerful microcomputers are now competing with simple electro-mechanical devices. These microcomputers which sell for $3-$10 in volume today contain a CPU which has an instruction repertoire of 40-50 instructions with 10-12 microsecond execution rates, up to 31 I/O lines, up to 2K by 8 ROM, and up to 128 x 4 RAM. These devices are being designed into sprinkler controllers, gasoline pumps, bathroom scales, high-fi turntables, home security systems, and an almost endless variety of other products. The capability of

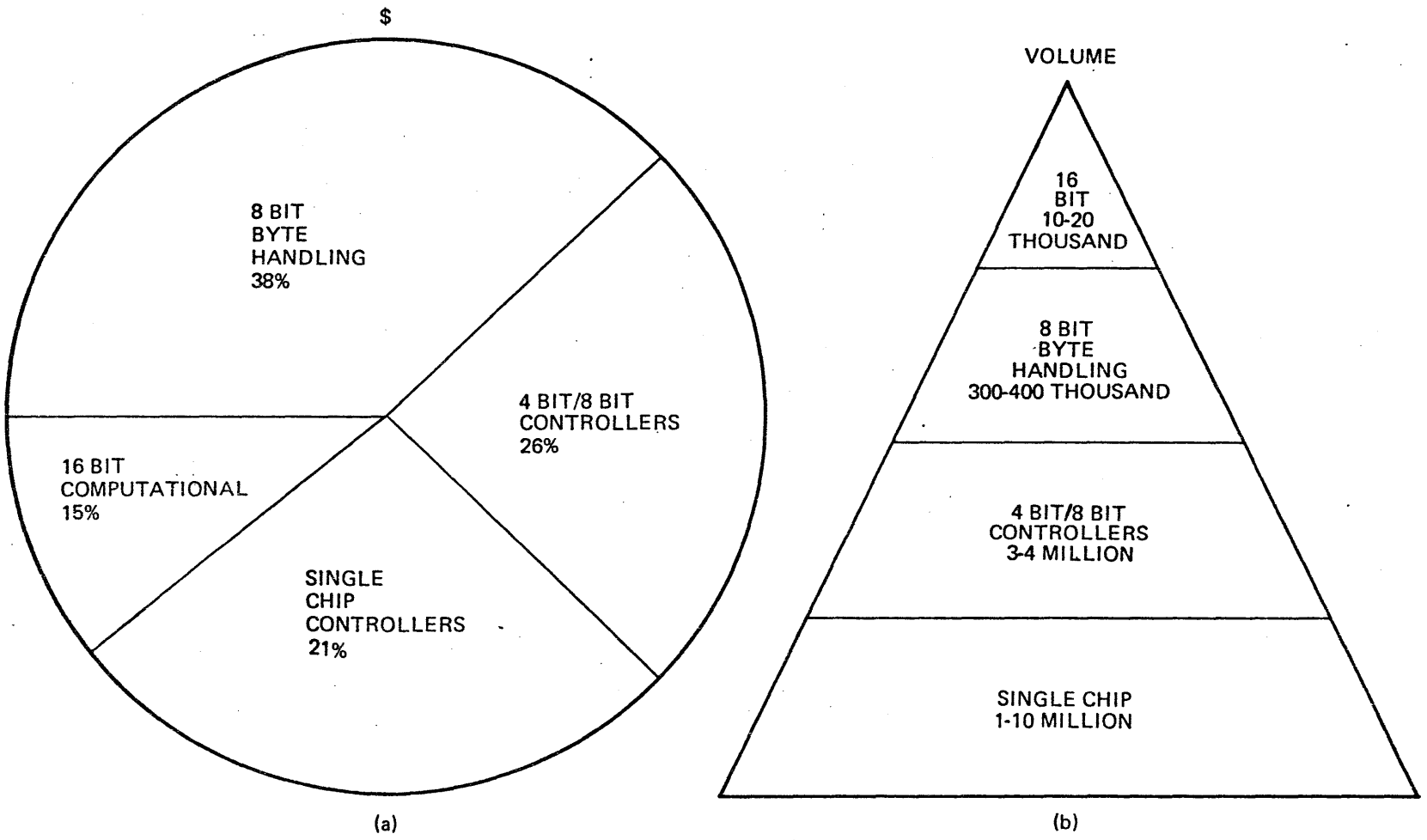

MAJOR MARKET SEGMENTS
FOR MICROPROCESSORS CLASSES
1977
$
8 BIT
BYTE
HANDLING
38%
4 BIT/8 BIT
CONTROLLERS
26%
16 BIT
COMPUTATIONAL
15%
SINGLE
CHIP
CONTROLLERS
21%
(a)
VOLUME
16
BIT
10-20
THOUSAND
8 BIT
BYTE
HANDLING
300-400 THOUSAND
4 BIT/8 BIT
CONTROLLERS
3-4 MILLION
SINGLE CHIP
1-10 MILLION
(b)

Figure 4

these low cost microcomputers must be considered in any new system design activity since they represent a new building block in electronic functions. Any equipments, for example, which require switch settings and a few digits of display are a candidate for a one-chip microcomputer just to handle these mundane functions. Then, higher level tasks and functions can easily be added to the system at little additional cost. When we get a little smarter, systemwise, to take advantage of multiple low cost distributed microcomputers, we should expect a very large increase in system cost/performance and in system features.

The low cost 4-bit/8-bit controllers represent the largest segment of the market today in terms of microprocessor volume. These devices are used primarily in electronic cash registers, transaction terminals, pinball machines, desk top calculators, taxi meters, billing and accounting machines, and as controllers for typewriters and low-speed printers. Characteristically, a microcomputer of this class today will consist of the CPU with 50-70 instructions with an execution rate of 4-12 microseconds, 2 to 6K of ROM, 128-512 words of RAM, and from 24 to 48 input/output lines. Normally, the microcomputer system consists of 2-6 devices. System functions are to service keyboards and displays, drive printers, provide limited computational ability, and provide some means of communication with other units. In most instances, real time control functions are required. Contrary to widely held opinion, the 4-bit microcomputer is actually better than the 8-bit

microcomputer in many applications handling numeric data. The RAM memory is organized in its natural numeric form, 4-bit words so that data computaion and manipulation is much easier than in a packed 8-bit word. Also, the instruction length of the 4-bit microcomputer is the same as that of the 8-bit microcomputer. Instruction lengths vary from 8 to 24 bits in both instances, so a 4-bit microcomputer can have just as powerful instruction set as an 8-bit system. The 4/8 bit classification essentially refers only to the organization of the data memory. Of course, 8-bit organizations are better for alphanumeric data handling.

Another distinct class of microcomputers is the medium performance 8-bit microprocessor used essentially in byte handling applications. This class of microcomputer has been given the most publicity and is the most visible to the technical community although its volumes of shipments are considerably lower than the previously discussed product classes. Typical applications are intelligent terminals, instrumentation, test equipment, low performance peripheral controllers, printer controllers, word processors, and communication controllers. The typical characteristics of this type microcomputer today are a CPU with from 70 to 120 instructions, instruction execution times of from 2-8 microseconds, 4 to 12K of ROM, 1-2K of RAM and 16-24 input/output lines. System functions are to service keyboards and displays, drive printers and floppy disks, provide communications control and protocol, and perform a medium amount of computation and data operations.

The remaining class of microcomputer is the higher performance 8 bit and 16 bit microprocessors. These are aimed at the sub-minicomputer type of product where computational capability is the primary function. Todays typical characteristics are a CPU with 100-200 instructions, instruction execution times of 2-6 microseconds, 8 to 24K of ROM, 2-8K of RAM, and 24-36 I/O lines. Typical applications are currently primarily in instrumentation applications. The lack of good products in this class has limited its growth, but recent product offerings will change this in the near future.

3.0 System Design Considerations

Early microcomputer designs placed most of their emphasis on CPU considerations and gave input/output considerations a much lower priority. Early device designers were not systems oriented and, as a result, applied most of their efforts to developing a CPU along the lines of a minicomputer type of architecture. Fortunately, these designs were general enough so that they could be used in many applications, but they were certainly not an optimum design for the types of jobs people want to do with microcomputers. Ninety-five percent of all potential microcomputer applications are not related to performing traditional minicomputer functions in a slower and less expensive manner, but are significantly different in application.

As these first microcomputers were applied to specific products, the importance to I/O capability became apparent because of the nature of the problems being solved. Data was being sensed and input, limited computations were being made, and the transformed data was being output. In most cases, the application was a real

time control function with many inputs and outputs such as in a printer controller or a cash register. We soon learned a fact that the big computer systems people had known all along. This was that the CPU is just a small, albeit an important, element of an overall computer system; furthermore, the I/O subsystems may be higher in dollar content than the mainframe and can affect system throughput just as much, if not more, than the CPU. As a result, a series of programmable parallel and serial I/O devices were soon developed by some major microcomputer vendors. These programmable I/O devices could be configured for various input/output configurations by menas of control words loaded by the CPU; they also performed all data transfer handshaking protocol. Some vendors, Rockwell being the leader in this particular area, went a step further and developed peripheral controllers such as keyboard/display controllers and various low speed printer controllers which were functionally independant of the CPU. The CPU controls the transfer of data and status information to and from these peripheral controllers; the controllers then perform all detailed peripheral control functions independant of the CPU. This permits the CPU to operate in a system executive mode; the tasks were set up and monitored by the CPU, but the CPU was free to execute other tasks while the detailed task was being executed by the peripheral controller device. Thus, the use of distributed processing design techniques in microcomputer systems began and offered substantial benefits for the user over the CPU oriented approach. First, the MOS/LSI peripheral controller represented a significant cost advantage over peripheral controllers implemented with discrete logic. The same

benefits of MOS/LSI derived in CPU implementations also apply to peripheral controllers, i.e., low cost, low power, and functional size reductions. Second, the intelligent peripheral controller overcame the inherent disadvantage of MOS/LSI, that of lower speed operation compared to equivalent discrete logic implementations, by providing parallelism in executing system functions. Thirdly, the use of intelligent peripheral and I/O controllers significantly reduced the software complexity of real time operations. The simultaneous real time control of several system functions by the CPU can result in very complex software and greatly complicate the addition of future system functions. In CPU oriented systems, a law of nature working against you is that as the percentage utilization factor of the CPU gets above 60 to 70 percent, the software complexity for real time functions tends to go up in some exponential manner. In microcomputers, it is extremely important to minimize software complexity since a large majority of people writing software for them are converted logic designers, not professional programmers. The software required to support an intelligent MOS/LSI printer controller, for example, consists of a simple 5 to 6 instruction loop transferring data to be printed to that device. In these cases, we are essentially using dedicated MOS/LSI logic to replace complex software; this is practical due to the low cost of today's MOS/LSI devices. This approach will be used more and more in the future. It makes economic sense in the end product, it shortens user product development cycles, and it minimizes the effects of a shortage of programmers which exists today.

The next logical step beyond intelligent controllers in the distributed processing trend in microcomputers is the use of multiple microprocessors in the same system. In this approach, multiple CPU's are used to perform various subassignments in the overall system design. Much effort has been expended by computer system designers to solve the general multiprocessor problems of programming and real time task allocation; this is, of course, a very difficult problem which requires much additional effort. In the microcomputer applications, however, the system is designed and dedicated to a fixed set of tasks. The specific, dedicated tasks are easily assigned and coded independently, to a large degree; a solution to the generalized problem is not required in these specific cases. The multi-processor approach offers a significant increase in performance over a single microprocessor, and additionally simplifies overall software requirements since the multiple CPU's are not loaded nearly as heavily as a single CPU would be in performing the total job.

Another aspect of distributing intelligence throughout the microcomputer system is the integration of interrupt and DMA handling techniques into the CPU and various I/O and peripheral controller devices. Again, methods of dispersing the workload outside of the CPU helps prevent CPU bottlenecks and their attendant problems. In this philosophy, interrupts become self-identifying requiring no polling and interrupt requests may be generated by any device and each device has its own built-in method of priorization. In this concept, DMA requests and terminations can also be generated by I/O and peripheral devices.

Because of the low cost attributes of MOS/LSI, the use of DMA techniques should be carefully re-evaluated. Previously, DMA has been

associated only with high speed data transfers; it has been an expensive option in minicomputers which has limited its philosophy of usage. Now, in a microcomputer, it is so inexpensive that it can be considered simply as a means of reducing system overhead and software complexity. A typical DMA controller for a microcomputer should cost less than $1.00 per channel controller which has automatic start, termination, and prioritization logic. A sizeable simplification of system software and reduction in CPU overhead can be achieved even in very slow data transfer situations.

Instruction sets on initial microcomputers were pretty much a scaled down, basic copy of classical minicomputer instruction sets. As device functional densities increased, instruction sets grew under the influence of two forces. The first force is to add additional traditional minicomputer instructions; the second force is to add instructions which more uniquely address specific requirements for microcomputer types of applications. Typical of these are bit manipulation instructions, block move instructions, and additional addressing modes for all instructions. Bit manipulation instructions become very important in microcomputers because of the heavy I/O control aspects of most applications. The use of memory mapped I/O ports with bit set, reset, and bit test instructions provide a new advance in bit oriented I/O control applications. Similarly, bit rotation instructions can be utilized to effectively scan multiple output lines. One trend in instruction sets is to remove the instruction "action" from the CPU registers to memory locations or I/O locations, at least from the programmers point of view. In these type instructions, multiple functions such as load register A, set bit 2, and store

register A are all coded as a single "Set Bit 2 in memory location AAAA" for example. These macro type instructions are becoming very common and represent a first step in utilizing more complex control logic to obtain, in some degree, the effect of a higher level programming language.

The major challenge of system architects in the future is to judiciously combine and integrate hardware and software techniques to reduce by an order of magnitude the initial effort required to program a microcomputer application.

4.0 Software Development

Most software development for microcomputers has been done in assembly language by a mixture of both programmers and logic designers. Program development equipment provided by the microcomputer supplier has been the primary means of program development. This equipment provides resident assembly capability as well as system emulation and debug capability.

The simplicity of the few MOS/LSI hardware devices required for a product application tends to mislead people into underestimating the complexity of the software task. In spite of the fact that you can hold the 5 or 6 microprocessor chips in the palm of your hand, programming that microcomputer system can be very difficult, especially for the logic designer who is making his first attempt at programming a general purpose computer. His problem is compounded in that he must commit his program, in most cases, to ROM codes which cannot be easily changed. To be honest, many first time users would probably not have opted for a microcomputer solution had they known the size of the software task beforehand.

The difficulty of programming microcomputers must be overcome before the potential of microcomputers can be fully realized.

The growth of the microcomputer market is being hampered today by the lack of available people to program their microcomputers. Most of our customers have several other projects they want to start just as soon as they have people to put on the new project.

The high front end cost of software plus the limited number of programmers available has resulted in a large amount of effort on higher level languages in an attempt to simplify the programming problem. A sub-set of PL/1 has been adopted by Intel and named PLM; others have followed Intel's course with their own versions. Several problems exist with today's higher level languages, however. First, the available languages have evolved from large computer system languages such as PL/1, and they don't exactly fit the kinds of jobs you need to do with microcomputers in primarily real time control applications. They may be useful in developing some subroutines, but they are not yet general purpose enough to do the full job. As a result, they have achieved only limited success. Current higher level languages seem sufficient to give a first cut at a portion of a program; then the programmer must incorporate functions not handled by the higher level language, and then optimize the overall code. Since the designer must worry about detailed timing and real time bit and I/O operations, he may be better off not using the higher level language at all.

Logic designers seem to have a desire to understand bit patterns and bit level activities more so than programmers. It is initially surprising to see many people coding in machine language, but if the program is less than 2K, this seems to be the preferred method.

A higher level language defined more specifically for the class of problems being solved with the majority of microcomputers is required. Characteristics of this language would be bit manipulation commands such as set, clear, test, and rotate. Interrupt handling capability with associated stack and unstack commands for register manipulation are also required. Also needed is a full complement of data representations as well as string manipulation commands. Links to assembly language routines are essential as is resident operation. Several languages incorporating some or all these features are under development and will overcome many of the limitations of earlier languages.

We would expect to see the importance of software given its rightful position in the design of the next generation of microcomputers. An integrated design in which specific hardware will be added to accommodate software tasks is necessary to reduce the overall problem. Conceptually, at least, the first step might be to define the higher level language and then develop the hardware around that definition.

Another simpler solution in many cases is simply expand the macro-instruction concept. This can be done simply with today's microcomputers to address specific applications. Rockwell, for example, has developed PIGOL, which is a macro type language for pinball game applications. The pinball machine designer has a series of programmed sub-routines that handle all scoring and play situations; he merely writes simple subroutine calls to link together the various subroutines that are required as a result of real time play activity. Other industries are being explored for applicability.

5.0 Technology Development

The microcomputers of 1971 were built with PMOS metal gate technology and consisted of 4 bit CPU's. The CPU die sizes were approximately 140x150 mils on a side; total die area was correspondingly 21-23K mils. These CPUs had instruction sets of 40-50 instructions with instruction execution times of from 4-25 microseconds. Minimum working dimensions which could be reliably yielded in volume production was approximately 10 μm.

The microcomputers of 1976-77 are now being built with NMOS silicon gate technology and are primarily 8 bit CPUs. CPU die sizes are approximately 180x190 mils on a side; die areas are 30-40K mils. The typical size of the instruction sets are 60-80 instructions with instruction execution times of from 2-10 microseconds. Typical minimum working dimensions are from 4-6 μm.

In analyzing technology development trends, there are several inter-related parameters which must be considered. The first, and most important parameter, is the minimum working dimensions which can be held in volume production. This relates to the tolerances which can be successfully controlled through all the steps of semiconductor production such as mask making, exposing, etching, and metallization. These minimum dimensions in controlling line widths and line spacings are directly related to the smallest component which can be fabricated; the smaller the MOS component, the higher the functional density of the device and the higher its speed of operation. The relationships of working dimensions (in microns) and clock speeds as a function of time is illustrated in Figure 5. As can be seen, the industry has been successful in reducing working

MINIMUM WORKING DIMENSIONS/CLOCK RATE VERSUS TIME

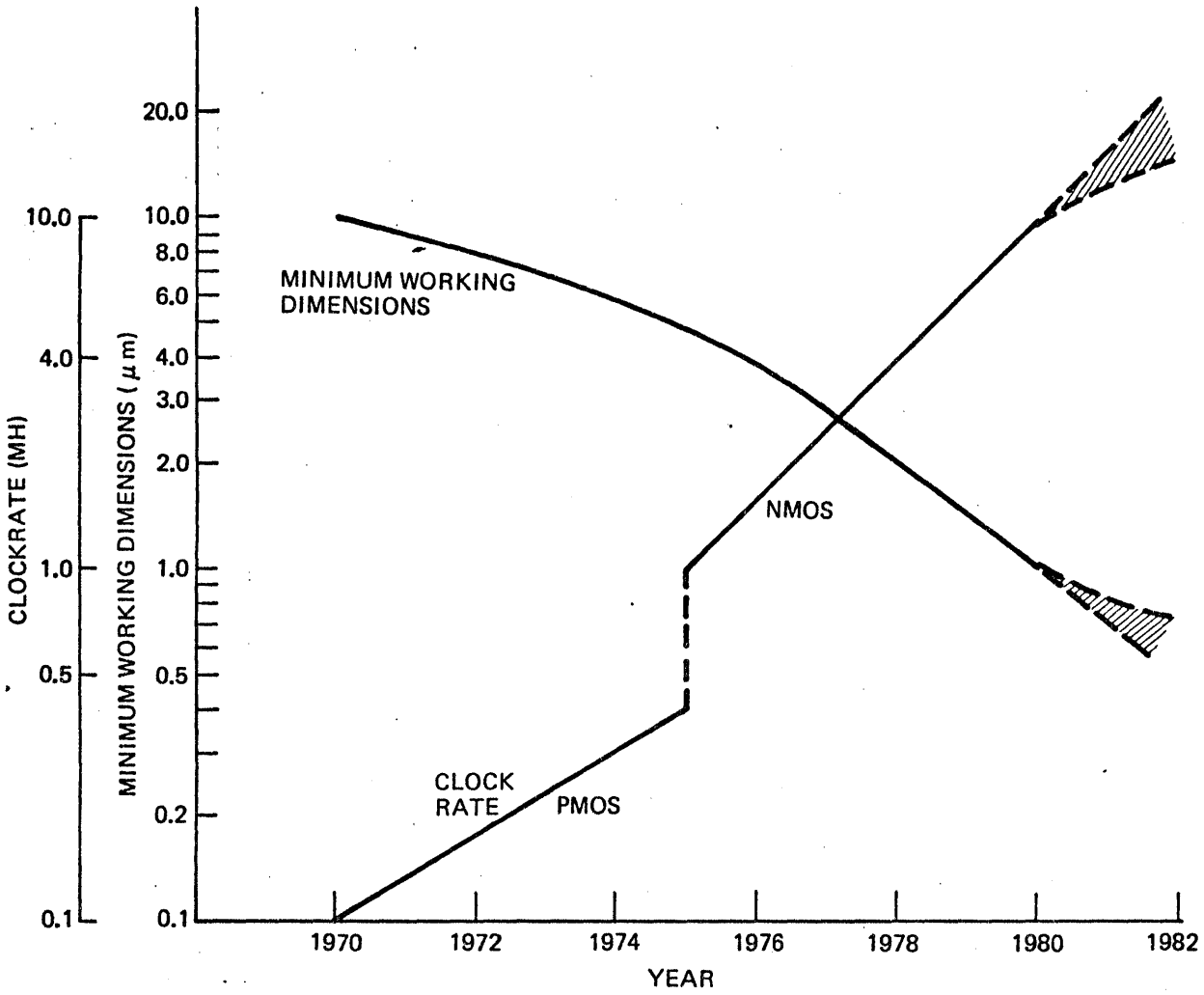


Figure 5

SCALING PRINCIPLES:		
DIMENSIONS	–	n:1
DENSITY	–	n^2:1
DELAY	–	1:n
POWER	–	1:n^2
ENERGY/LOGIC EVENT	–	1:n^3

Figure 6

dimensions by a factor of about 3 since 1970 and is projecting another factor of 3 improvement by 1980. The impact of reduced working dimensions, or scaling, is shown in Figure 6. Density goes up as the square of the scaling factor, speed goes up inversely proportional, and power goes down inversely proportional to the square of the scaling factor. The resulting decrease in the area required for a single MOS component is shown in Figure 7. We currently have pushed our current production photolithography techniques near their limits to achieve today's 5-6 micron technology, but new techniques are emerging from the labs which will enable us to move on down the curve to 1 micron technology by 1980.

This next step in increasing circuit density is referred to as VLSI technology, or very large scale integration. VLSI technology consists of two aspects. First is a dry etching technique to replace today's wet chemical etching. The use of dry plasma etching significantly improves etching control and will permit working to 2 micron line widths. To get beyond 2 microns, we must resort to new pattern generation equipment because our current optical pattern generation equipment is not adequate. This will require the development of electron beam pattern generators for pattern definitions. These techniques exist today and are used in bubble memory development work where you must deal with sub-micron gaps for controlling the magnetic bubbles. New techniques for vastly increasing the speed of pattern formation are required, however, before they can be used in production semiconductor fabrication processes. The use of electron beam pattern generation and X-ray lithography techniques hold promise of sub-micron components which will make today's technology look pretty archaic.

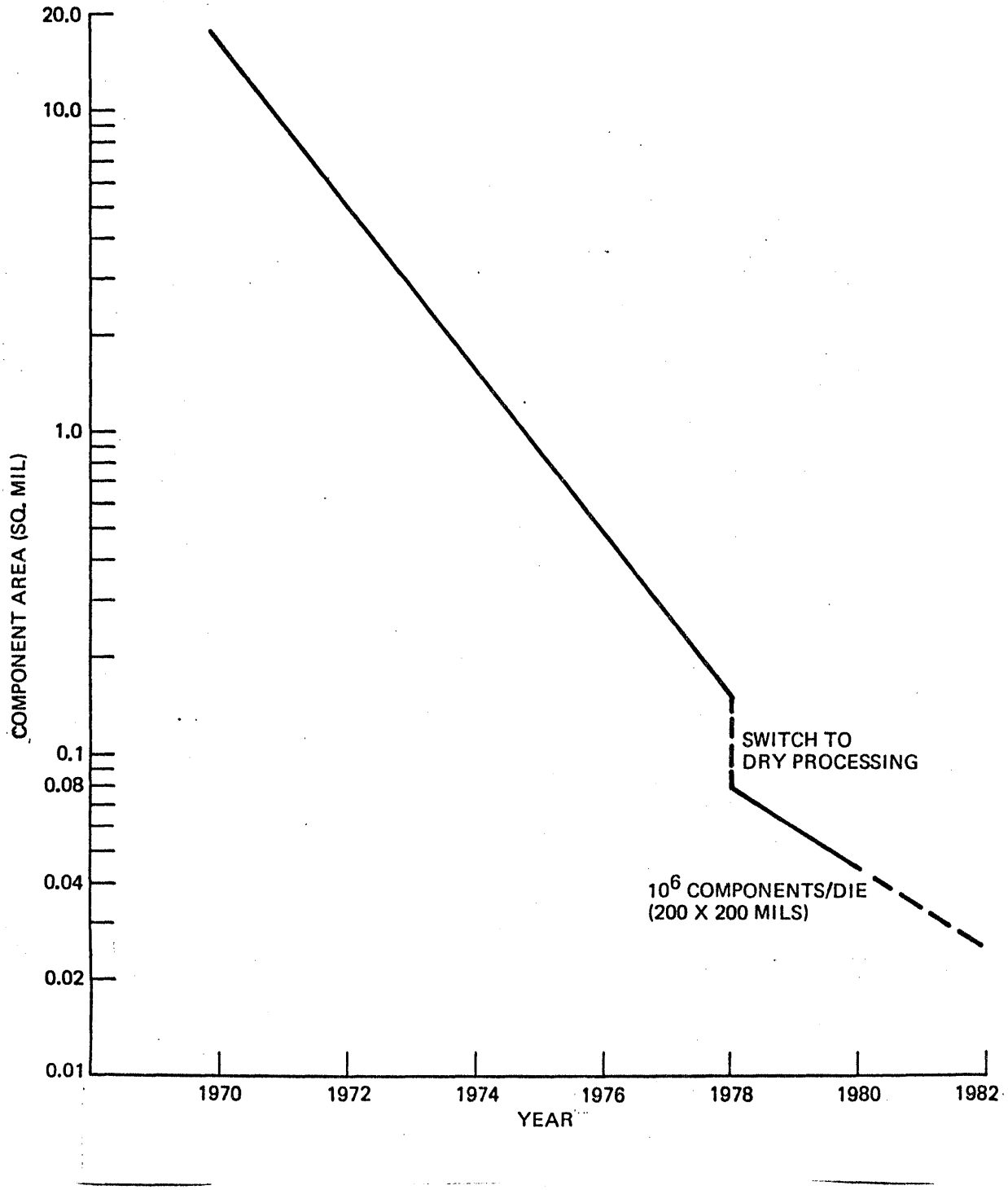

MOS/LSI COMPONENT AREA
(SQ MILS) VERSUS TIME
20.0
10.0
1.0
0.1
0.08
0.04
0.02
0.01
COMPONENT AREA (SQ. MIL)
SWITCH TO
DRY PROCESSING
10^6 COMPONENTS/DIE
(200 X 200 MILS)
1970
1972
1974
1976
1978
1980
1982
YEAR

Figure 7

The second aspect of technology development relates to economically producible die size as a function of time. Figure 8 indicates the trend in die size producibility. As producible die size increases, more functions per device can be included. This, in turn, reduces the number of devices required to build a system. The key parameter affecting producible die size is defect density per unit area. Defect density directly affects the number of good die per wafer and is a function of size of each die. The larger the die size, the higher the probability that a defect will be present within the die; if a defect falls in the die area, you have a non-working device. Defects arise from the basic silicon material and the photoresist chemicals used. As the manufacturing system becomes more tightly controlled, defect density drops and the economically producible die size increases.

The third aspect of technology development relates to the size of the wafer processed. The industry started with 2 inch wafers in 1969 and has progressed to 3 inch wafers today. Most manufacturers are now planning 4 inch wafers by 1978. The wafer size is a key economic factor to the semiconductor manufacturer. Wafer size determines the maximum number of potentially good die per unit factory time. For example, a 2 inch wafer has about 150 candidate die for a die size of 140x150 mils; a 3 inch wafer has about 350 candidate die of the same size. For a 20% yield, the 2 inch wafer yields 30 good parts; the 3 inch wafer yields 70 good parts, assuming the same overall yield. The economics of production are obvious from this example; the 3 inch wafer facility gives over twice the number of good parts per processing time.

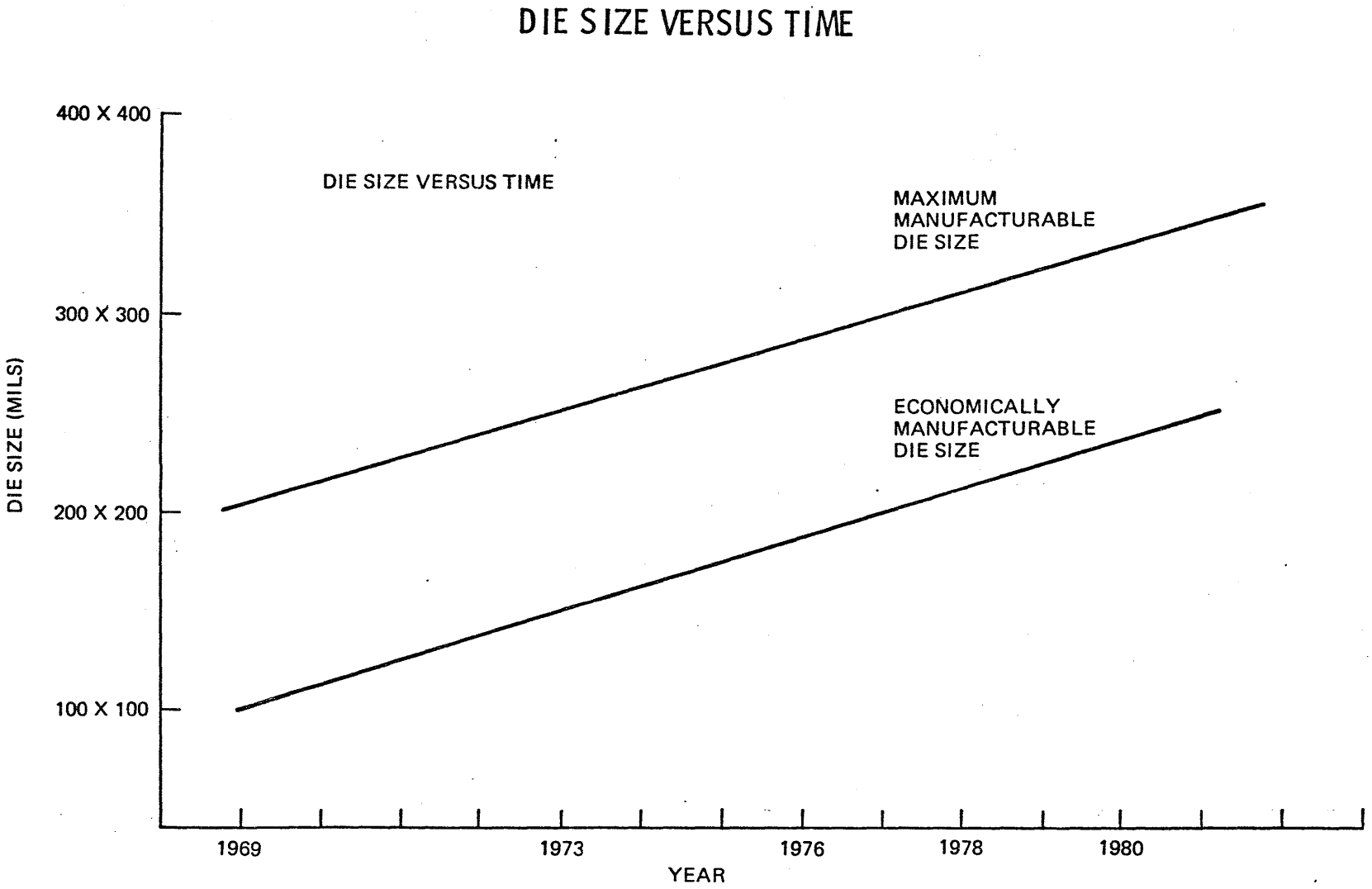

DIE SIZE VERSUS TIME
400 X 400
300 X 300
200 X 200
100 X 100
DIE SIZE (MILS)
DIE SIZE VERSUS TIME
MAXIMUM
MANUFACTURABLE
DIE SIZE
ECONOMICALLY
MANUFACTURABLE
DIE SIZE
1969
1973
1976
1978
1980
YEAR

Figure 8

Since fixed costs are such a large portion of total factory costs, the size of the wafers processed is a dominant factor in the economics of producing MOS/LSI. Figure 9 illustrates projected wafer processing technology versus time and provides the number of die per wafer for various size die and various size wafers. The example illustrated shows an increase in die candidates from 240 to 420 when going from a 3 inch wafer to a 4 inch wafer for a 180 X 190 mil die size. This only represents one factor in productivity; it does illustrate the potential for further cost reductions in MOS/LSI devices, however.

Another very important aspect of technology development is in packaging. The impact on microcomputer products that the physical package contributes is very considerable. There are never enough pins; you always need one more. The lack of, or availability of, pins on the package can play a major role in overall architectural design and device partitioning. The increasing functional density capability on silicon requires some innovative packaging design for increased pin density, if we are to fully benefit.

DIE PER WAFER VERSUS WAFER SIZE

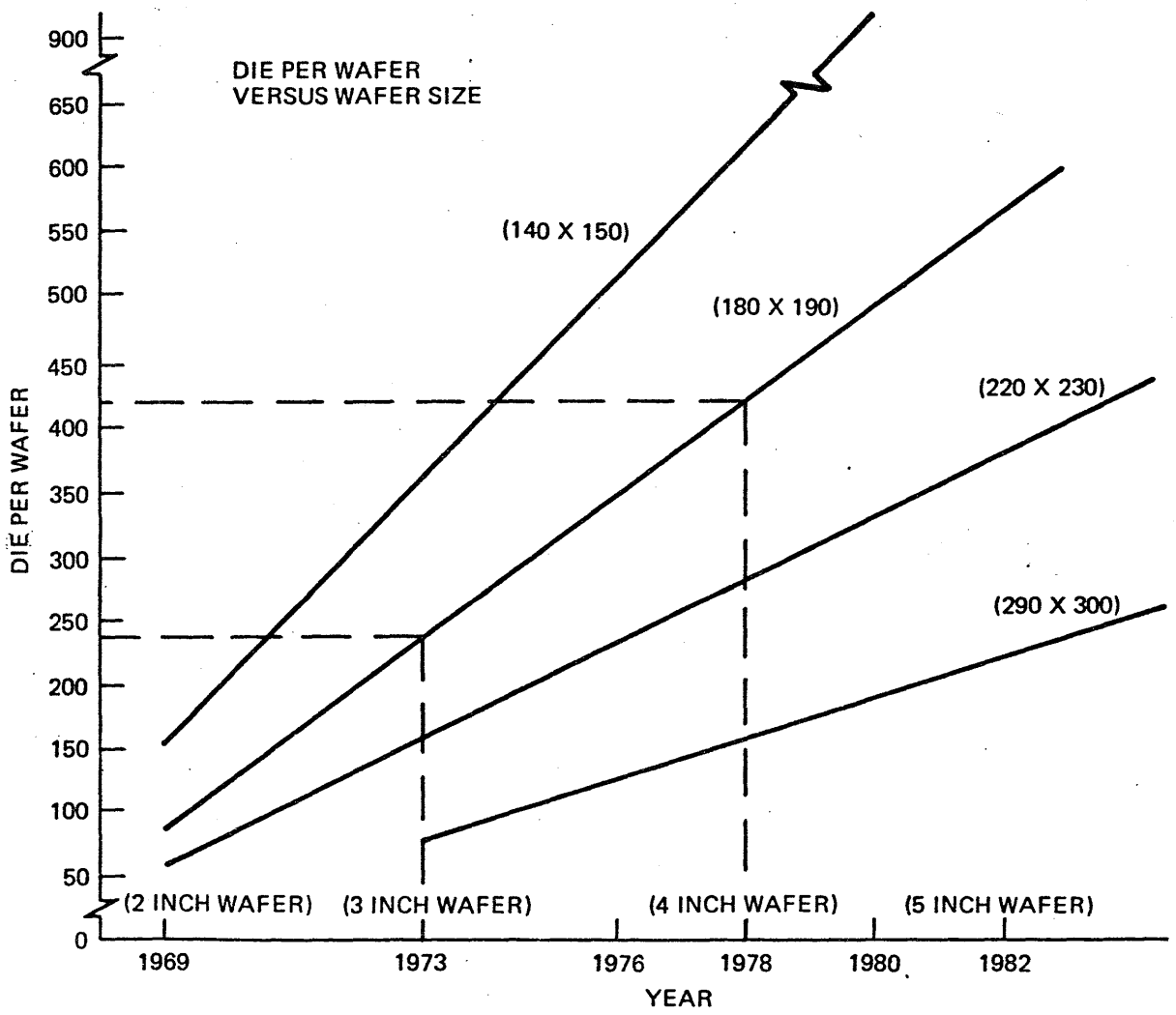


Figure 9

6.0 Future Microcomputer Developments

We can begin to get a feel as to where microcomputers are headed if we integrate all of the technology trends and product characteristics we've discussed. In summary, these are increasing die sizes and yields, increased functional density per square mil of silicon, increased speed of circuitry, improved system architectures, and the dedication of hardware to reduce the software task.

Our method of prognostication will be to configure five typical microcomputer products which represent MOS/LSI capability by the early 1980's. The intent is to project capabilities and not to project actual products. Our semiconductor capability may far outstrip our ability to design and configure systems, for we are talking about full systems on a single chip in the 1980's. This has rather far reaching implications in all aspects of the electronics industry. Also, the ability to generate masks directly on silicon from a computer controlled electron beam pattern generator over large surfaces, creates possibilities that haven't been fully explored.

The ability to combine two or more devices into a single device produces several synergistic effects. First, the cost of one or more packages and packaging labor are saved, although the single package may now require more pins. Second, the need for one set of drivers and bonding pads required to get off chip can be eliminated when devices are combined; this can amount to 15-20 percent of overall chip area. Third, the opportunity to improve performance is much higher if all circuitry can be contained within a chip.

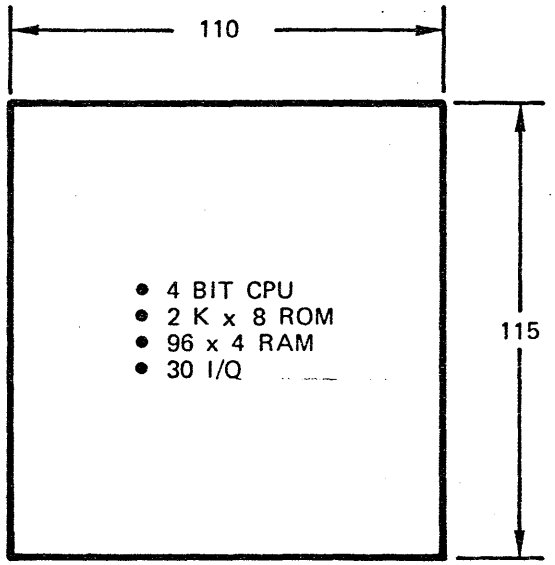
The first product we want to describe is a low end 4 bit microcomputer designed for applications such as appliance controllers.

Specifications for this product is shown in Figure 10. Estimates for the sizes of various sections of the microcomputer is given. On a small device of this size, input/output circuitry and bonding pads consume a large percentage of chip area; VLSI techniques really can't help too much in this area. Power dissipations for such a device should be under 10 milliwatts; low power standby techniques for RAM memory should give the facilities for non-volatile memory with a very small standby battery. The cost of such a microcomputer should be in the $0.25 - $0.50 range in medium volumes by the mid-1980's.

This places the low end microcomputer down into the cost level of today's MSI/TTL gate package by 1981.

The second microcomputer is a single chip 8 bit microcomputer with characteristics as shown in Figure 11. This single chip microcomputer replaces a 6-8 chip system of 1977 and has 3-4 times the performance. Clock rate is projected to be 10 MH. The device size shown should permit prices in the $0.50-$1.00 range for medium quantities in the mid-1980's. With the 4K X 8 ROM capability, on chip, a macro-oriented language can be implemented for the less complex applications.

The third microcomputer projected is an expandable 8 bit controller with limited ROM and RAM on chip as well as I/O, multiplexed A/D conversion interval timers, and peripheral controller functions as illustrated in Figure 12. Devices of this size permit full systems on a chip capability with prices in the $2-$4 range. The capability represented on this device portends another major shift in contributed values in the electronics industry. With the possible exception of possibly additional ROM and RAM, it will now be possible to place the total electronics for a sophisticated data acquisition

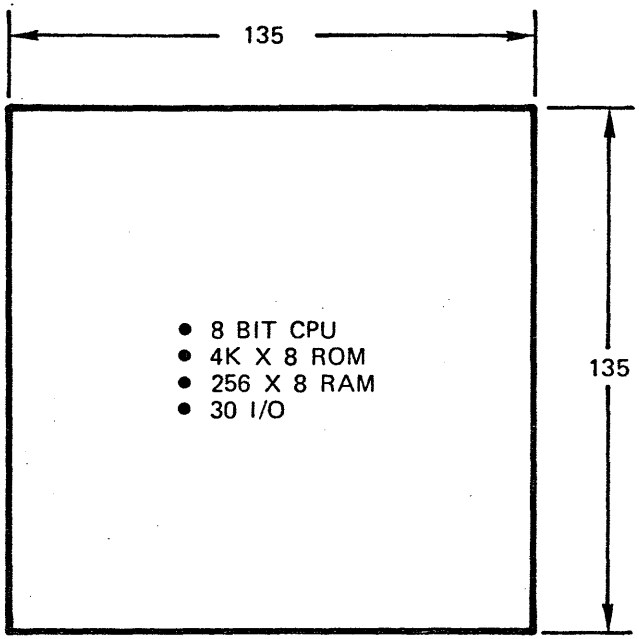


CPU	–	2,000 SQ. MIL
ROM	–	2,500 SQ. MIL
RAM	–	800 SQ. MIL
I/O	–	1,500 SQ. MIL
DRIVERS/PADS	–	5,700 SQ. MIL
TOTAL	–	12,500 SQ. MILS

INSTRUCTIONS : 50
CLOCK : 5 MH.

LOW END SINGLE CHIP CONTROLLER : 1980–82

Figure 10



CPU	–	2,700 SQ. MILS
ROM	–	4,000 SQ. MILS
RAM	–	2,000 SQ. MILS
I/O	–	1,500 SQ. MILS
DRIVERS/PADS	–	8,000 SQ. MILS
TOTAL	–	18,200 SQ. MILS

INSTRUCTIONS : 100
CLOCK : 10 MH.

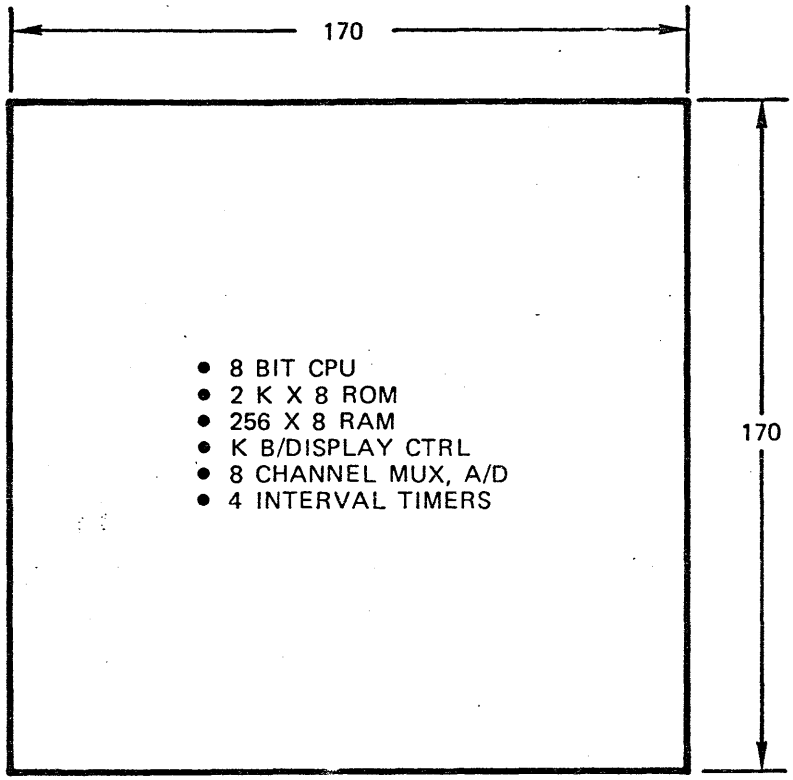
SINGLE CHIP CONTROLLER : 1980–82

Figure 11

system, electronic cash register, or transaction terminal on a single chip. The area in this example identified for the 8 channel multiplexed A/D converter could be used just as well for a 1200 BPS modem function for the terminal applications. For a simple terminal, communication protocol can be handled in software.

The next example shown in Figure 13 represents the general purpose 8 BIT processor typical of the PPS-8/8080/6800 class of product. The device contains all system support functions required for a sophisticated microcomputer system. Our hypothetical example includes 8 channels of DMA control, an interrupt processor, two programmable peripheral or I/O controllers, 4 internal timers, a floppy disc controller, and a serial communications protocol controller. This device will be supported with 32K X 8 ROM chips and 64K bit (8K X 8) RAM devices. The overall device size should allow this device to sell in the $3 - $4 in the mid 1980's. Direct execution of higher level language programs should be feasible. In the worst case, a single ROM device should handle a resident compiler. Note that this device outperforms all of today's minicomputers. It has an instruction set of about 250 instructions and an execution time of roughly 500 nanoseconds.

The final product example addresses the high end of the microcomputer market and the low end of the minicomputer market. Figure 14 illustrates its features. Typical applications would be use as a data base processor for supporting a department store, with several hundred POS terminals, or as a store and forward message switches/concentrator. The device contains a high performance 16 bit CPU, an associative index file or catche memory, a 16 channel DMA controller, a data encriptor/decriptor processor to handle the NBS algorithm, a communication

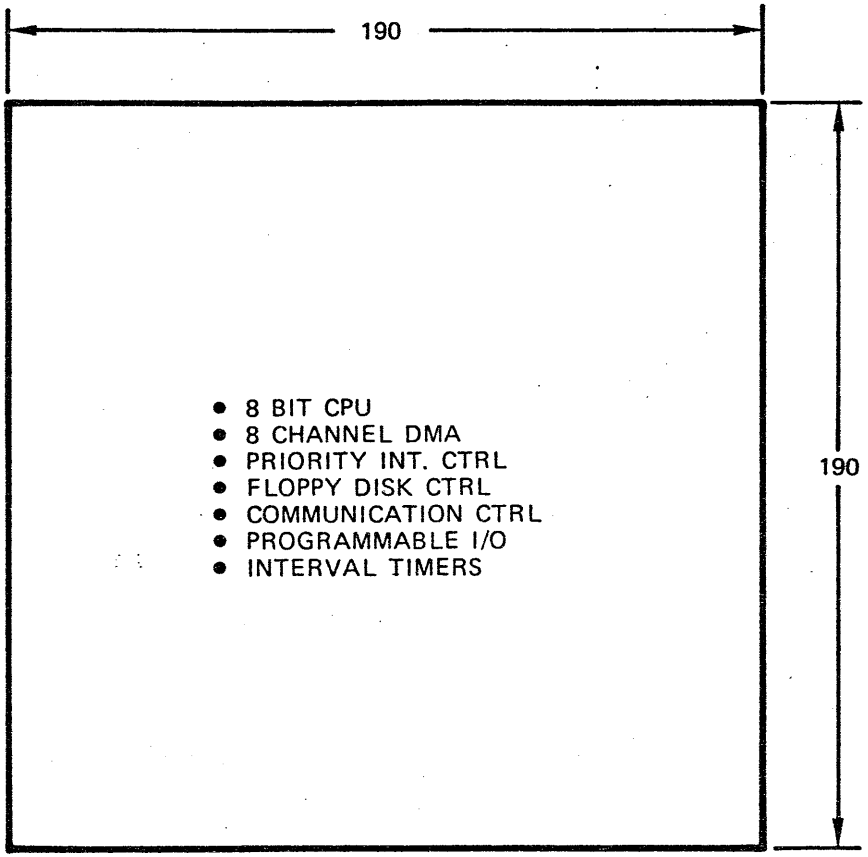


CPU	–	4,500 SQ. MIL
ROM	–	2,500 SQ. MIL
RAM	–	2,000 SQ. MIL
MUX, A/D	–	3,000 SQ. MIL
KB/DISPLAY GTRL	–	2,400 SQ. MIL
INTERVAL TIMER	–	2,500 SQ. MIL
I/O	–	1,500 SQ. MIL
DRIVERS/PADS	–	10,000 SQ. MIL
TOTAL	–	28,400 SQ. MIL

INSTRUCTIONS	:	150
CLOCK	:	10 MH

8 BIT MULTI–CHIP CONTROLLER: 1980–82

Figure 12

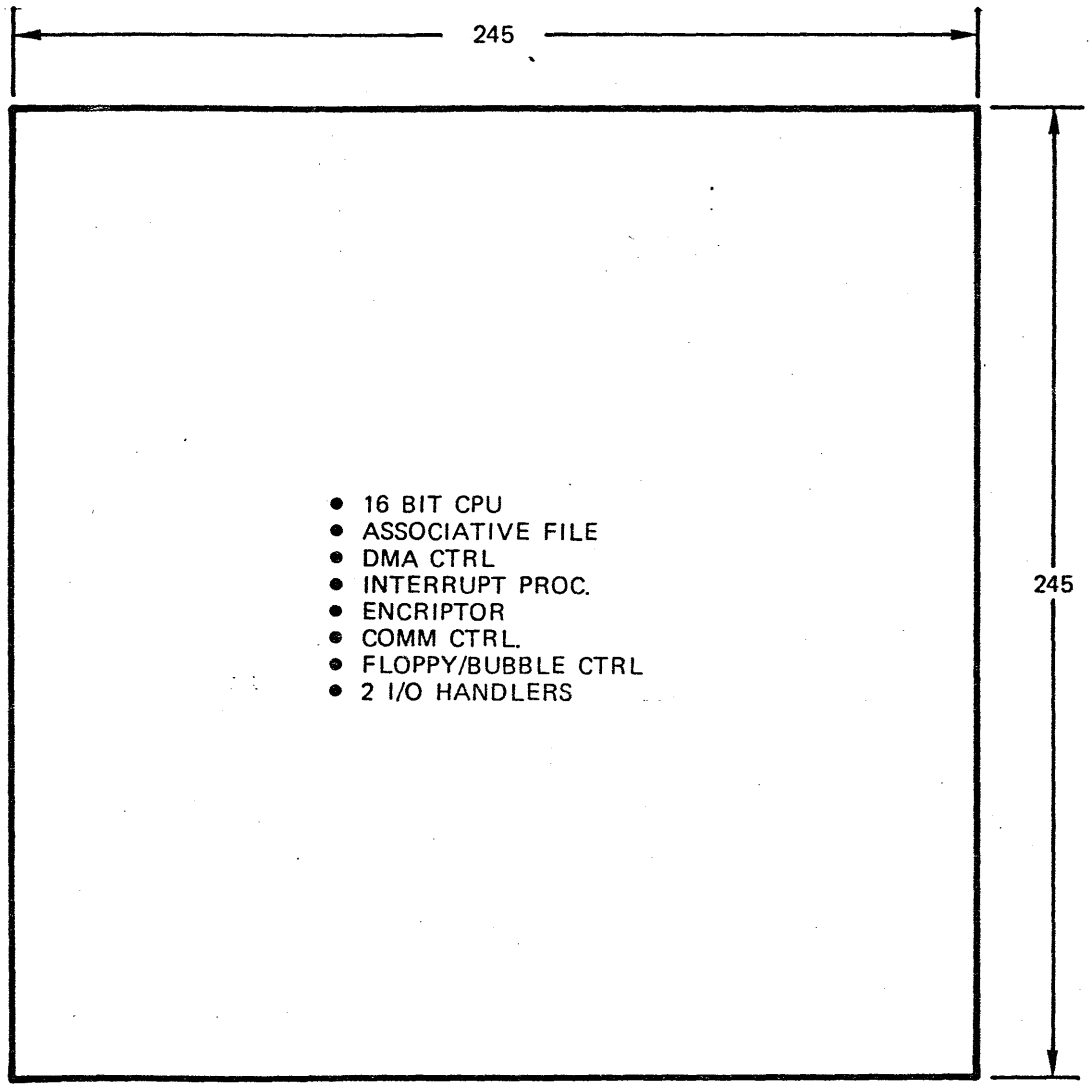


CPU	–	6,000	SQ. MIL.
DMA	–	2,500	SQ. MIL.
PRIORITY INT. GTRL	–	1,400	SQ. MIL.
INTERVAL TIMERS	–	2,500	SQ. MIL
FLOPPY CTRL	–	4,000	SQ. MIL.
COMM CTRL	–	3,500	SQ. MIL
2 PROG. I/O	–	4,800	SQ. MIL
DRIVERS/PADS	–	11,000	SQ. MIL
TOTAL	–	35,900	SQ. MILS

INSTRUCTIONS : 250
CLOCK : 10 MH.

8 BIT PROCESSOR: 1980–82

Figure 13



CPU	–	12,000	SQ. MILS
ASSOCIATIVE FILE	–	12,000	SQ. MILS
DMA	–	3,000	SQ. MILS
ENCRIPTOR	–	8,000	SQ. MILS
COMM CTRL	–	3,500	SQ. MILS
I/O HANDLERS	–	4,800	SQ. MILS
FLOPPY/BUBBLE CTRL	–	4,000	SQ. MILS
I/O	–	1,500	SQ. MILS
DRIVERS/PADS	–	12,000	SQ. MILS
TOTAL	–	60,800	SQ. MILS

INSTRUCTIONS : 300–400
CLOCK : 10 MH.

16 BIT MICROPROCESSOR : 1980-82

Figure 14

protocol processor, a disk or bubble memory controller, and two I/O controllers. This product could be expected to sell in the $15 - $20 dollar range in mid-1980.

The intent of these examples is to point out the rapid advances to be expected in semiconductor hardware capability and costs. The onus is on the software and systems people to be just as creative and productive as the semi-conductor industry so that these hardware advances may be fully exploited. The costs projected can only be realized if volume markets are generated through innovative applications.

The decrease in cost per function in electronics equipments has been dramatic over the past 15 years, something like a 10,000 fold decrease. Dr. Gordon Moore of Intel has put this advance in perspective with an analogy with the automotive industry. He says "If the auto industry had advanced at the same rate we'd have cars capable of traveling at 100,000 miles per hour and getting 50,000 miles per gallon. It would be cheaper to throw away a Rolls Royce than to park it downtown for an evening, but reliable enough to be passed on from generation to generation." The parameters which we have discussed offer another times 100 decrease in cost per function over 1977 capability. We have a real challenge, systemwise, to take advantage of this impending advance. Any new systems designs must factor in the considerations of continuing new microcomputer technology developments.