$10^{**}4$ gates by 1979. By the middle 1980s we should be able to achieve the equivalent of 10**5 gates on a single IC. This would allow a 32-bit parallel CPU and 2**15 bytes of random access memory on the IC. Dedicated read/write memory circuits containing 2**20 bits should be available by the mid 1980s. Read only memories are usually able to achieve twice the density of read/write memories.

The speed of such a computer should be on the order of 10**6 instructions per second. Speed will be limited primarily by power consumption and the number of external connections permitted. The cost of a random logic IC should not deviate significantly from the current $10. Memory circuits should be available for about $15.

While the improvement in disc storage has not been as dramatic as those in the random logic and main memory areas, disc characteristics have also improved significantly over the past 20 years. Again there is no indication that the rate of improvement is diminishing. Disc storage density was approximately 8000 bits per square inch in 1960. By 1970 it had increased to 8x10**5 bits per square inch. By 1985 we should be able to achieve 250x10**6 bits per square inch. Unfortunately, similar improvements in seek time and rotation delay do not seem to be forthcoming.

In summary, we see technology advancing so fast that current concerns such as hardware cost and CPU efficiency will become meaningless in the foreseeable future. But we see no similar reduction in the cost of software. Similar arguments lead one to question the long term viability of the current mid-range computer systems (e.g., 1110s, 6700s, and 158s) in their present form. With $100 computer and $10**7 operating systems, it would appear that simple, one user at a time, batch systems and personnel computers would be advantageous. But there is no indication that computers will "look" differently than they do now.

Distributed Microprogram Architecture, Dr. Gideon Frieder

The advent of dynamic microprogramming, and the extensive usage of interpretation which resulted from it, enables us to re-evaluate our motions of current systems in the sense that it is not necessary to assume fixed control functions and fixed instruction sets.

This, coupled with the new technologies, will in my mind bring forth the following type of developments:

1. Distributed loci of control, each of a nature which will be dynamically adaptable to the task at hand, controlling a hierarchy of storages.

2. Multiple central processors, with dynamically selectable instruction sets, adopting to the task at hand. The instruction will be defined by microcode, although I do not foresee free user microcoding. Rather, I foresee a rich repertoire of sets of instructions, carefully prepared and evaluated by the vendor to fit the tasks for which the machine is marketed.

3. A dramatic change in operating system design, motivated by the need to support different virtual machines and simplified by the fact that new technologies, in my mind, will bring forth a phenomenon which I call "zero time I/O". By this I mean to denote a situation in which the requested information item is almost always in memory, the word almost being interpreted in the same statistical sense that cache supported system has the data in the cache when it is requested. I belive that one can include in the instruction set special instructions which will enable proper hierarchical staging and pre-paging. These instructions will be produced by the compilers following a global flow analysis and will use a considerable amount of residual control and control storage locations to accumulate both static and dynamic data, to be used in pre-paging.

Unconventional Architecture, Dr. Wayne T. Wilner

Just as one-way streets sometimes make it inconvenient to get from here to there, the primitiveness of computer operations and the procrustean inflexibility of systems and devices can make it difficult to get from having an information-processing problem to having a computer's solution to it. Conventional architecture is a one-way street which makes it easy to solve a very limited class of problems and difficult to solve the rest.

Conventional architecture reflects itself into programming languages and into the way in which we use computers. If we attack current deficiencies by changing architecture, then we can expect monumental changes to be required in software and methods of use as well.

It has been suggested that computers which are simply much faster and much cheaper will relieve the difficulties. Speed and economy do nothing to relieve the primitiveness and inflexibility of computers. They merely make the cost of such agony smaller and available to more people.