VI. CIRCUIT ORGANIZATION

Much of the improvement in the functional characteristics of integrated circuits over the next decade can be expected to come from better circuit organization. Whenever many thousands of components are used to perform a function, the organization of those components becomes a major factor in their collective performance. Integrated circuits are now at a level of complexity where careful organization pays off handsomely, but we do not yet clearly understand the fundamental design constraints imposed by circuit technology.

The first thing one observes in a complex integrated circuit is that wires occupy most of its area. As a circuit increases in component count, the percentage of its area devoted to wires will also increase, unless the circuit is carefully organized. The reason is that not only do the number of wires increase in direct proportion to the component count, but also each wire tends, on average, to be longer. As Appendix A shows, for the upper limit of random interconnection, the space per component required for wiring increases linearly with the number of components; the area cost of *each component* thus goes up solely because of the number of components, not because of the component itself. The gains to be achieved by arranging components in rows and columns with local wiring between them therefore increase with increasing component count.

The second thing one observes about complex integrated circuits is that their ultimate computation speed is limited by the rate at which information can be transmitted from one place to another. But the rate at which information can be transmitted from one place to another is limited because the conductors have capacitance and must be driven by sources of finite resistance. There is an advantage, therefore, in physically arranging information so that data elements that must be combined in a computation are located close to each other and close to the circuits that perform the computation. The difficulty of computation tasks should not be thought of in terms of megabits transmitted or computed per second, but rather in terms of megabit *meters* per second.

Heretofore, we have designed computing equipment with a rather clear-cut physical as well as functional separation between memory and computing, a separation no longer warranted by our technology. We now have a strong motivation to commingle processing and memory functions in order to derive the most computational output from our equipment. The motivation is increased by the fact that both memory and computation are most economically provided by the very same semiconductor technology. Unfortunately, we know relatively little about how best to organize our logical elements to make use of these capabilities.

In the past, suggestions for combining computing and memory have been implemented in a context in which the fundamental costs of the functions have been drastically different, i.e., thousands of bits of memory could be obtained for the cost of a few computation functions. But for the chip, the costs of processing and memory are more nearly the same. Wiring costs dominate both of these functions, because wiring not only occupies most of the chip area but also introduces most of the propagation delay. This is a whole new context in which organization needs to be explored.

In the construction of large software systems, and in the design and fabrication of complex systems such as oil refineries and aircraft, we have learned that careful organization is essential to success. Whenever many thousands of parts are involved, and all of them must function perfectly for correct overall operation, the conception and planning of the interrelationships between the parts must be done with great care. In software, design concepts such as "structured programming" have been found very useful. Such concepts are really just reflections of good engineering practice: the form of the solution should follow the form of the problem, and parts of the design that are treated separately must be truly separable. In circuitry, as opposed to the logical design of software, conflicts for wiring space become a major design consideration, just as conflicts for piping space are a major consideration in the design of oil refineries, ships, and aircraft.

Careful organization of the design task, as well as of the design itself, may be required as the complexity of hardware design begins to approach the complexity of the software systems with which we have had such abysmal disasters. In fact, it may be argued that the distinctions between computer hardware and software are beginning to vanish. Software, after all, is a medium in which one can describe logical processes and have them performed; it is characterized by a very high design cost and a very low replication cost. Silicon microcircuit technology is also characterized by high design cost and low replication cost. While there are those who maintain that the solution to the software problem lies in improving integrated circuits, we believe that unless care is taken now, the design of the integrated circuit may itself become "the software problem."

We have found relatively little evidence that system designers are doing any fundamental thinking about organization. The theoretical results discussed in Appendix A are the only ones we know of that deal with the problem of organizing wiring. It would be nice if a body of theory about the geometric aspects of computing could be developed, and if that theory could be applied to practical devices. As circuit complexities involving several tens of thousands of components are reached, there is no doubt that improved organizations can make large differences in design cost and functional performance.