

R. D. Traub
J. P. Roth

Potential Significance to Neurophysiology of Design Algorithms for Digital Computers

Abstract: A set of algorithms has been developed within the computer industry that aids in the design and comparison of large switching circuits. To apply these algorithms to the nervous system in a realistic way, it is necessary to find neurons where (a) the electrophysiological data exist to allow construction of a plausible model; and where (b) the behavior of the neuron is not dependent on its distant past, so that the cell may be approximated by a switching circuit. It is also necessary to find areas of the nervous system whose structure resembles that of so-called regular designs; a regular design is a logic circuit in which feedback loops are constrained to take certain simple forms. In cases where local feedback loops are an intricate part of the design (as is true in virtually all areas of the mammalian central nervous system), this constrains one to analyze the behavior of the area in question only over time intervals sufficiently short so that such feedback loops can be ignored. Certain areas, such as the retina, are excluded because the individual neurons behave nondiscretely and are intimately coupled together (i.e., the retina does not resemble a regular design).

Introduction

Algorithms have been developed in the computer industry that can, among other things, determine the equivalence of switching circuits. These algorithms are used in the analysis of computer hardware designs containing large numbers of switching elements [1-5]. Since these algorithms are available for the analysis of circuitry of great complexity, the question can be raised as to whether such algorithms might be useful in studying the nervous system. This is a matter of importance now that it is becoming technically feasible to model large networks of neurons [6-8].

In this Communication, we consider some of the properties and applications of computer design algorithms. We then discuss how these properties constrain us to select for modeling certain types of neurons—namely, those whose behavior resembles that of switching circuits—and certain areas of the nervous system—areas whose behavior over short time intervals is of interest and in which the neurons are not “tightly coupled” together.

Properties of design algorithms

A large modern digital computer contains switching circuitry of great complexity. This circuitry must be designed reliably and flexibly, i.e., in such a way as to allow modifications after the design process has begun; it must also allow easy repair after a hardware failure in the finished product.

To these ends, a set of algorithms—called herein “design algorithms”—has been developed. Some of these

design algorithms determine whether two combinational circuits (without feedback loops) are equivalent, i.e., give the same outputs for the same inputs [3, 4]; or in their more general form, whether two “regular designs” (defined below) are equivalent [1]. In addition, other design algorithms exist [2] that accept as input a functional specification of a circuit (in the form of a program written in the special purpose computer language PL/R, a variant of PL/I) and generate as output a logic design that realizes the given specification. The designer can define his circuit in a compact, easily understood notation; translation to a logic design at the hardware level is done automatically by computer. If a change be made in the resulting design, the algorithm VERIFY [4] can determine if the new circuit is functionally equivalent to the old one. If a failure occurs in a switching element, a particular algorithm called the D-algorithm generates a set of inputs to the circuit that distinguishes (i.e., makes an output differ between) the circuit with the failed element and the “correct” circuit.

We now characterize the class of circuits to which design algorithms can be applied.

In their present form, design algorithms can be applied only to Boolean circuits, i.e., those in which each wire in the circuit carries either a 0 or a 1. The circuits are built up from “gates.” A gate, or switching element, is a device that accepts a number of Boolean inputs; each output of the gate is some Boolean function of the inputs. By identifying outputs of certain gates with the inputs of others, circuits of essentially arbitrary complexity are

constructed. Gates are assumed to operate synchronously and determinately. Time does not enter explicitly. In order to introduce time, one or more clock pulses are included among the inputs to selected gates.

For design algorithms to apply to circuits with feedback, the circuit must be in a special form, called a regular design [1, 8]. A regular design is a circuit with the following property: either it consists entirely of acyclic logic (i.e., contains no feedback loops); or it consists of blocks of acyclic logic connected to one another through registers, together possibly with lines from the outputs of acyclic logic blocks to the inputs of the registers. The registers also receive inputs from nonoverlapping clocks. Regular designs have the property that their behavior is always well-defined (see Fig. 1 in [8]).

Another important property of the design algorithms must be mentioned. A major purpose is to determine whether *two* designs are equivalent. Thus, for example, the D-algorithm compares two designs differing in one gate, or in the value assigned to one line. (Design algorithms can, however, be used to aid in the initial specification of a single design.) Note also that "equivalent" here means "yielding *exactly* the same outputs for the same inputs."

Methods for constructing models of neurons and nerve cell assemblies

We mention three different approaches to modeling nerve cells. The first method is oriented toward the physiology of the single cell and examines such matters as electrotonic architecture and the distribution of ionic conductances over the surface of the cell [8-13]. The second method looks at small assemblies of cells and seeks to explain extracellular potentials in terms of the interconnections among the cells [14]. Both of these methods lead to models with a number of differential equations, and both involve events in the cell membrane governed by processes with small time constants (a millisecond or less). However, another approach involves the study of a large network of nerve cells, in which each cell is represented in a relatively simple way and the questions of interest concern the global behavior of the network [6-8, 15].

Models that contain differential equations involve special problems for the application of design algorithms, as discussed below.

Problems with design algorithms applied to models involving differential equations

In order to apply design algorithms, we must start with a model consisting of elements which change state synchronously every time step. If the model contains differential equations, this constrains one to take a time step of the order of the integration step. The integration step

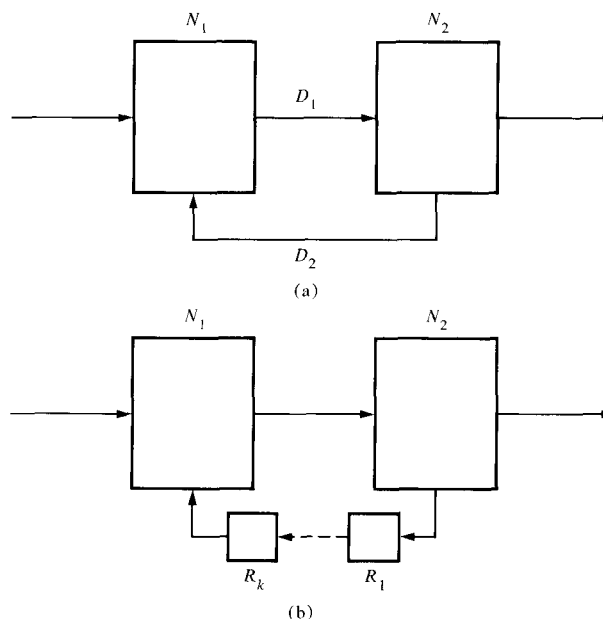


Figure 1 The "Loose-Coupling Criterion" and conversion to a regular design. (a) N_1 and N_2 represent acyclic logic blocks that model two different neurons, with D_1 being the delay for passage of bits from N_1 to N_2 , and D_2 the delay for passage of bits in the reverse direction. Blocks N_1 and N_2 are loosely coupled if and only if D_2 is long compared with D_1 . (b) If D_2 is long compared with D_1 , we can pass bits from N_2 back to N_1 through a series of registers $R_1, \dots, R_k$. This results in a regular design (see text).

depends on how much membrane-biophysical detail is included; the order of magnitude can be estimated by noting the time constants for some of the membrane properties of central nervous system (CNS) neurons. Thus, the membrane RC time constant is of the order of 3 to 10 ms, while the Hodgkin-Huxley state variable m for a fast cell such as a Renshaw spinal interneuron may have a time constant as small as 25 μ s [12].

One might then envision the following approach: To a logic circuit L, take as input variables all of the state variables of the neuronal network model (the variables being encoded in binary form). Circuit L itself is the logic circuit which integrates the state variables through one time step, in the manner specified by our original differential equations model.

The problem here is that of treating logic and numerical analysis simultaneously. Inevitable inaccuracies in the reference model, or the use of different integration schemes, will yield models whose behavior is qualitatively similar but whose equivalent logic circuits are distinct. The design algorithm approach makes every bit in the circuit just as important as every other bit, while from the point of view of the original model this may not be true at all.

Possibilities for application of design algorithms to networks where the neurons are represented as combinational elements

In some cases, it may be possible to model neurons as combinational elements, an approach first suggested by McCulloch and Pitts [16]. For this to work, the output of the neurons in question should be in the form of discrete impulses, ruling out cells that function by graded potentials [17]; such cells occur, for example, in the retina. In addition, the neurons should behave instantaneously relative to the time step in which the overall model functions. If the neurons contain conductances with long time courses, e.g., the slow potassium conductance of tonic spinal motoneurons, this may necessitate inclusion of a memory with each cell, which in effect records the status of this conductance. These same observations apply to synaptic junctions, where processes of relatively long duration can occur (an example being the synapses on Renshaw spinal interneurons from motoneuron axon collaterals, where excitatory post-synaptic potentials of the order of 20 ms duration can be produced).

Assuming that a particular neuronal circuit under investigation does consist of neurons and synapses that can be represented as combinational elements, the next requirement for the application of design algorithms is that the circuit be regular in the sense defined above. What this means is that any feedback loops from one element in the circuit to another must pass through a register pair. It will be possible to introduce such registers into the model if and only if we have the "Loose-Coupling Criterion": Delays along lines carrying information around a feedback loop must be long compared with the delays for switching states in neurons and synapses and for transfer of information forward in the circuit (Fig. 1). This criterion may be met in circuits where all feedback takes place through loops containing lengths of axon; but it will not be met if there are synapses between two neurons carrying information in opposite directions, thus coupling the cells together with a short time constant—see, for example, [Chap. 9 of 18] for a discussion of such "reciprocal synapses" in the thalamus. The Loose-Coupling Criterion will also be violated if electric field interactions between neighboring neurons are important, as may possibly occur in certain brainstem and spinal motor nuclei [19, 20].

In the design of hardware for logic circuits in digital computers, great care is used to ensure 1) that the individual circuits operate in physical isolation from one another, 2) that information passes from one logic circuit to another only at specified times, and 3) that feedback only takes place with appropriate delays. It may be that similar design criteria are met in selected regions of the mammalian central nervous system, but such criteria

are certainly *not* fulfilled everywhere. Examples mentioned above include the retina and thalamus, but there are other areas as well in mammalian brain where these design criteria are not met, including the olfactory bulb. Nevertheless, it is likely that major areas of the brain exist where the Loose-Coupling Criterion can be applied at least approximately.

Specification of detail in neuronal vs logic circuitry

The application of design algorithms to a circuit requires that the circuit be defined both rigorously and completely; the algorithms are then used to test for logical equivalence of different versions of a given circuit, to find tests that distinguish the behavior of a given circuit from the behavior of that circuit with a failed element, etc. Specification at this level of detail of behavior of neural circuits involves, however, two major problems. One problem is ignorance, discussed below. The other, probably deeper, problem is whether it makes sense to represent neuronal circuitry in such a fashion that changes in the behavior of single lines or switching elements matter. One feels that the brain must have redundancy built into it because of unreliability in nerve cells [21], membrane noise, resistance to trauma and fatigue, and the like. Redundancy of this sort would make uninteresting the problem of distinguishing the behavior of two circuits that differ in only one bit in one line—presumably, what really matters are patterns of behavior that involve large numbers of elements.

This is a problem on which we feel it best to reserve judgment, until greater understanding of brains is achieved. We make two observations, however. First, redundancy of a system per se does not rule out the applicability of design algorithms; many logic circuits actually used in computers in fact contain redundancy. Second, even if it be true that only macroscopic patterns of activity in thousands of nerve cells have any "real" importance, it may be possible to understand such patterns only after a painstaking and detailed analysis of circuits in which each action potential *is* important.

The problem of ignorance

Few nerve cells are understood in the kind of detail that one would like for a convincing logical model. Much of this difficulty arises from problems in recording with electrodes from identified small neurons. In addition, quantitative theoretical studies have not kept pace with the experimental information already available on large neurons.

Unfortunately, no real neuronal circuit is known that consists only of large, easily comprehensible cells. Nevertheless, possibilities exist for making reasonable simplifying approximations, as has been done in the work on the cerebellar cortex of Pellionisz, Llinás, and Perkel

[8]. It is possible that the cerebellum and perhaps also the hippocampus will offer the best opportunity for application of the methods described above.

Conclusions

Methods have been developed in the computer industry for the analysis of large circuits, and it is tempting to consider the possibility of applying these methods to the nervous system. However, the designer of logic circuits for a digital computer can assume that the physical realization of his work will fulfill certain design criteria, e.g., isolation of the switching elements from one another. These design criteria make the analysis of the circuitry by logical tools meaningful.

It is now known that certain areas of the mammalian brain do not fulfill these criteria, in the sense that the behavior of neighboring nerve cells may be "tightly coupled." Nevertheless, the possibility still exists for the application of design algorithms to other areas of the nervous system, the major barrier here being ignorance of the physiology of small nerve cells.

References

1. J. P. Roth, "Generation and Verification of Hardware Designs at High Level," *Research Report RC 5779*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1976.
2. H. Halliwell and J. P. Roth, "System for Computer Design," *IBM Tech. Disclosure Bull.* **17**, 1517 (1974).
3. J. P. Roth, "Diagnosis of Automata Failures: A Calculus and a Method," *IBM J. Res. Develop.* **10**, 278 (1966).
4. J. P. Roth, "Verify," *Research Report RC 4682*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1974.
5. J. P. Roth, W. G. Bouricius, and P. R. Schneider, "Programmed Algorithms to Compute Tests to Detect and Distinguish Between Failures in Logic Circuits," *IEEE Trans. Comput.* **EC-16**, 567 (1967).
6. A. Pellionisz, "Computer Simulation of the Pattern Transfer of Large Cerebellar Neuronal Fields," *Acta Biochim. et Biophys. Acad. Sci. Hung.* **5**, 71 (1970).
7. A. Pellionisz and J. Szentagothai, "Dynamic Single Unit Simulation of a Realistic Cerebellar Network Model," *Brain Res.* **49**, 83 (1973).
8. A. Pellionisz, R. Llinás, and D. H. Perkel, "A Computer Model of the Cerebellar Cortex of the Frog," *Neurosci.* **2**, 19 (1977).
9. W. Rall, "Theory of Physiological Properties of Dendrites," *Ann. N.Y. Acad. Sci.* **96**, 1071 (1962).
10. F. A. Dodge, Jr. and J. W. Cooley, "Action Potential of the Motoneuron," *IBM J. Res. Develop.* **17**, 219 (1973).
11. R. D. Traub, "Motoneurons of Different Geometry and the Size Principle," *Biol. Cybernetics* **25**, 163 (1977).
12. R. D. Traub, "Repetitive Firing of Renshaw Spinal Interneurons," *Biol. Cybernetics*, to be published.
13. R. D. Traub and R. Llinás, "Spatial Distribution of Ionic Conductances in Normal and Axotomized Motoneurons," *Neurosci.* (to be published).
14. W. Rall and G. M. Shepherd, "Theoretical Reconstruction of Field Potentials and Dendrodendritic Synaptic Interactions in Olfactory Bulb," *J. Neurophysiol.* **31**, 884 (1968).
15. M. N. Nass and L. N. Cooper, "A Theory for the Development of Feature Detecting Cells in Visual Cortex," *Biol. Cybernetics* **19**, 1 (1975).
16. W. S. McCulloch and W. Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity," *Bull. Math. Biophys.* **5**, 115 (1943).
17. F. O. Schmitt, P. Dev, and B. H. Smith, "Electrotonic Processing of Information by Brain Cells," *Science* **193**, 114 (1976).
18. G. M. Shepherd, *The Synaptic Organization of the Brain*, Oxford University Press, London, 1974.
19. R. Baker and W. Precht, "Electrophysiological Properties of Trochlear Motoneurons as Revealed by IVth Nerve Stimulation," *Exp. Brain Res.* **14**, 127 (1972).
20. P. G. Nelson, "Interaction Between Spinal Motoneurons of the Cat," *J. Neurophysiol.* **29**, 275 (1966).
21. S. Winograd and J. D. Cowan, *Reliable Computation in the Presence of Noise*, M.I.T. Press, Cambridge, MA, 1963.

Received May 2, 1977

The authors are located at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598.