T H E   U N I V E R S I T Y   O F   M I C H I G A N

COLLEGE OF LITERATURE, SCIENCE, AND THE ARTS
Communication Sciences Program

Technical Report

UNIVERSALITY IN THE VON NEUMANN CELLULAR MODEL

J. W. Thatcher

ORA Projects 03105, 06376, 06689

under contract with:

U. S. ARMY RESEARCH OFFICE (DURHAM)
GRANT NO. DA-ARO(D)-31-124-G588, PROJECT NO. 4049-M
DURHAM, NORTH CAROLINA
and
NATIONAL SCIENCE FOUNDATION
GRANT NO. GP-2539
WASHINGTON, D.C.
and
DEPARTMENT OF THE NAVY
OFFICE OF NAVAL RESEARCH
CONTRACT NO. Nonr-1224(21)
WASHINGTON, D.C.

administered through:

OFFICE OF RESEARCH ADMINISTRATION   ANN ARBOR

September 1964

RESEARCH PROGRESS REPORT

Title: "Universality in the Von Neumann Cellular Model," J. W. Thatcher, University of Michigan Technical Report 03105-30-T, September 1964; Nonr-1224(21), NR 049-114.

Background: The Logic of Computers Group of the Communication Sciences Program of The University of Michigan is investigating the application of logic and mathematics to the design of computing automata. The detailed working out of the von Neumann growing automata design forms a part of this investigation.

Condensed Report Contents: The von Neumann cellular automaton model is described and designs within this model are presented for objects that behave like tape and constructing units. An algorithm is developed for embedding in the cellular structure any automaton which effectively manipulates the tape and constructing units. The algorithm is based on a very simple language in which the behavior of such machines can be described. Finally, universality of construction and computation as well as automaton self-reproduction are discussed relative to the von Neumann model.

For Further Information: The complete report is available in the major Navy technical libraries and can be obtained from the Defense Documentation Center. A few copies are available for distribution by the author.

ACKNOWLEDGEMENTS

This research was motivated by a seminar on von Neumann's contributions to the theory of automata conducted by Professor A.W. Burks at the University of Michigan in the Fall of 1961. Many of the ideas presented here were developed while the author was with the Logic of Computers Group at the University of Michigan. Support for this research was provided by the following government agencies:

Office of Naval Research contract Nonr 1224(21)

National Science Foundation grant GP-2539

U.S. Army Research Office (Durham) grant DA-ARO(D)-31-124-G588

A preliminary draft of this report was prepared at the University of Michigan. Since the author joined the staff at IBM Thomas J. Watson Research Center, there have been major revisions in the original report. These revisions include, in particular, adapting the designs to a programming model of computation and construction. This simplified presentation of the computational process required considerable modification in the tape unit design.

The author gratefully acknowledges the support received through the Logic of Computers Group from the contracts and grants listed above. In addition the author wishes to express his appreciation to Professor Burks for his continued encouragement and to Stephen Hedetniemi of the Logic of Computers Group who offered many constructive criticisms of the original manuscript. The author would also like to thank Eric Wagner, of IBM Thomas J. Watson Research Center, for many helpful comments and suggestions based on a reading of the revised report.

TABLE OF CONTENTS

LIST OF FIGURES

PAGE

LIST OF TABLES

INTRODUCTION

The objective of this report is to consider the design of certain universal and self-reproducing automata in the cellular model proposed by John von Neumann in his manuscript, "The Theory of Automata: Reproduction, Construction, Homogeniety".(1) Von Neumann's model can be described in an informal way as an infinite chess board, each square of which contains a copy of the same (twenty-nine state) finite automaton. One decides on some initial configuration of states for the automata in every one of the squares of the board. This determines the state of the model at time $t = 0$, or alternatively, it determines a particular cellular automaton embedded in the model. At subsequent discrete units of time, the state of the automaton in each square at time $t+1$ is uniquely determined by its state at time t and the states, also at time t, of its four (non-diagonal) neighbors. Thus, with the choice of an initial configuration, the history of the model for all subsequent time is completely determined.

(1) This unpublished manuscript is to be edited by A.W. Burks and published by the University of Illinois Press. For a description and discussion of von Neumann's model see A.W. Burks, "Cellular Automata", Proceedings of the International Symposium on the Theory of Relay Systems and Finite Automata, Moscow (1962). Two papers of general interest relative to the problems considered here and to cellular automata in particular are A.W. Burks, "Programming and the Theory of Automata", Computer Programming and Formal Systems, (P. Braffort and D. Hirshberg, Eds.) Studies in Logic Series, North-Holland (1962), and E.F. Moore, "Machine Models of Self-Reproduction", Mathematical Problems in the Biological Sciences, Proceedings of Symposia in Applied Mathematics, Volume XIV, American Mathematical Society (1962). The latter paper includes an extensive bibliography of the earlier papers in this area.

By appropriately interpreting successive configurations of states at times $t = 0, 1, 2, \ldots$, one can consider the cellular model (or embedded cellular automaton) to be carrying out constructional and computational tasks. In particular, starting with some initial configuration of states at time $t = 0$, if at some later time there is an identical copy of this configuration (in addition to the original), then it is not unreasonable to say that automaton self-reproduction has taken place.

In fact, von Neumann formulated this model to study exactly such situations - to study the complex problem of self-reproduction.(2) In the unpublished manuscript, besides describing the model, von Neumann undertakes to design a self-reproducing automaton. We will, in effect, carry out such a design in the following sections of this report. The general organization of our self-reproducing automaton is similar to von Neumann's proposed design, both involving a "central control unit", a "tape unit", and a "constructing unit". One of the crucial differences is in the design of the tape unit. With his method, von Neumann was forced to consider difficult and cumbersome problems of timing which resulted in an extremely complex construction. These complications could easily be discouraging, either for one attempting to design an automaton

(2) Discussions of the problem of automaton self-reproduction are to be found in some of von Neumann's earlier published works including: "The General and Logical Theory of Automata", Cerebral Mechanisms in Behavior, Proceedings of the Hixon Symposium, (L.A. Jeffress, Ed.) Wiley (1951) 1-31, and The Computer and the Brain, Yale University Press, 1958.

within the system or for one reading von Neumann's presentation.[3]

The elimination of the timing problems results in considerable simplification of the tape unit design. Some handwritten notes were attached to the von Neumann manuscript which specified, in diagramatic form, a method by which an automaton embedded in the cellular system would construct secondary automata in the surrounding area. Considering these notes it is difficult to imagine that von Neumann did not realize the possibility of this simplification, for the method of construction by an automaton is analogous to the method of tape modification which eliminates the necessity of timing in the tape unit control.

Von Neumann's manuscript was never finished. It terminates with the completed tape unit design and the notes on the construction method. In essence, the tape unit and the method of construction are the primary obstacles to the proof of the existence of a self-reproducing automaton. Hence, relative to the existence question, this report serves only to present the proof in a completed form.

Before getting on to the detailed construction, it is our intention to make precise the problems that are being attacked and the nature of the solutions. In order to do this it seems reasonable to consider these problems with respect to a general class of cellular models of which the

(3) This is a relative matter. The constructions of succeeding sections are burdened with technical details.

von Neumann model is a special case.(4)

It is certainly not our purpose to provide a general theory of cellular automata in the brief discussion of this introductory section. Indeed, it is sincerely hoped that the emphatic statements, the proposed definitions and the suggested problems will serve to underscore the need for, and the very interesting nature of, such a theory. Besides these particulars of a more formal nature, there are many areas of interpretation and of application which support the author's contention of the importance of further research into this subject. Von Neumann, as the originator of this area of study, was not interested in cellular automata as mathematical objects of study, but instead, was attempting to find a manageable way of treating in detail the problem of how to make machines reproduce themselves, hoping to shed some light, not only on the intrinsic capabilities of machines, but also on the problems of biological self-reproduction. John Holland, whose research was certainly influenced by

(4) As will be clear, the generalization presented here can be carried a good deal further. The only paper known to the author directed specifically at generalizing the concept of iterative or cellular automata is that by J.H. Holland, "Universal Embedding Spaces for Automata", Technical Report 06114-1-T, ORA, the University of Michigan, January, 1964. (To be published in a festschrift for Norbert Wiener, Netherlands Central Institute for Brain Research, Amsterdam). Moore's discussion (op.cit.) of cellular, or as he calls them tesselation, models is in a general, but informal, framework. The problems of self-reproduction and construction are treated with a recursive-function-theoretic approach by Myhill in "The Abstract Theory of Self-Reproduction", Notes on the Automata Theory Course, the University of Michigan Engineering Summer Conference, 1963.

von Neumann's work, describes a certain class of "modular computers", appropriately called the class of Holland Machines.[5] But again, the formulation was not an end in itself, but directed toward a general theory of adaptive systems. The ability, within a cellular model, to talk about populations of automata interacting among themselves, and with the environment, makes such a formulation particularly appropriate. Perhaps paradoxical, but certainly supporting the thesis of the importance of cellular automata theory, is the interest in Holland Machines that has been evident on the part of people interested in building computers--on the part of people interested in investigating radical new concepts in machine organization. In another vein, although the theories of finite automata and Turing machines contribute to a general theory of computation and computers, they are undenyably insufficient in many very essential respects. There are features of the theory of cellular automata which indicate the possibilities of interesting applications. With cellular automata, computation *is* structure in a given cellular system and can be studied as such. The general and very vague problem of "how" computations take place, being insignificant in current computability theory, might have a reasonable formulation within the area of cellular automata.

(5) See J.H. Holland, "Iterative Circuit Computers", *Proceedings of the Western Joint Computer Conference* (1959) and "Outline for a Logical Theory of Adaptive Systems", *Journal for the Association of Computing Machinery* (1962) 297-314. Other interesting papers on Holland type machines include, E.G. Wagner, "An Approach to Modular Computers , I: Spider automata and Embedded Automata", IBM Research Report, RC 1107 (1964), W.T. Comfort, "Highly Parallel Machines", *Proceedings of the Workshop on Computer Organization*, Spartan Books (1963), and H.L. Garner and J.S. Squire, "Iterative Circuit Computers", appearing in the same *Proceedings*.

We continue now with the definition of a class of cellular models to be discussed in the remainder of this section. All the models of this class have the same "geometry". Taking I to be the set of integers, the geometry is represented by the index set I x I together with the four neighbor functions:

$$u(m,n) = \langle m+1, n\rangle, \quad d(m,n) = \langle m-1, n\rangle,$$
$$r(m,n) = \langle m, n+1\rangle, \quad \ell(m,n) = \langle m, n-1\rangle.$$

A particular cellular model is determined by a finite automaton to be associated with each cell $\alpha \in I \times I$. This finite automaton is specified by a triple, $\langle V, v_o, f\rangle$, where V is the set of (local or cellular) states, v_o is a distinguished element of V called the quiescent state and f is the (local) transition function from five-tuples of elements of V into V. There is one restriction on the transition function, namely, $f(v_o, \ldots, v_o) = v_o$.

A configuration or state of the cellular model is a function from I x I into V which has finite support relative to v_o. Thus, a configuration s is an assignment of a cellular state $s(\alpha)$ to each cell α in I x I such that only a finite number of cells are assigned states other than the quiescent state.

Let C be the class of all configurations of a cellular model. We then define the transition function, F, of the model as a function from C into C. It is defined in terms of the local transition function as follows:

$$F(s)(\alpha) = f(s(\alpha), s(u(\alpha)), s(d(\alpha)), s(r(\alpha)), s(\ell(\alpha))).$$

Interpreting a configuration as a state of the cellular model at time t, the transition function F determines the state of the model at time t+1. The equation above is a formal statement of the fact that the state of cell α at time t+1 is determined (by the local transition function) on the basis of the state of that cell at time t together with the states, at time t, of the four neighbors of α.

Choosing an initial configuration s to specify the state of the cellular model at time t=0, the study of cellular models must involve the interpretation of the sequence,

$$s,\ F(s),\ F^2(s), \ldots, F^t(s), \ldots,$$

as exhibiting, in some way, computational or constructional processes. This is perhaps one of the most difficult areas in the theory. The tendency, as evidenced in the literature and, in fact, in the later sections of this report, is to base this interpretation on the standard sequential representation of the computational process. We will consider, in the sequel, interpretations which are somewhat nonstandard. But before doing this, some additional notation and terminology will be helpful in talking about the cellular systems.

The <u>support of s</u>, written sup(s), is, as indicated above, the set of $\alpha \in I \times I$ such that $s(\alpha) \neq v_o$. s' is a <u>subconfiguration</u> of s if $s \mid sup(s') = s' \mid sup(s')$. Two configurations s and s' are <u>disjoint</u> if $sup(s) \cap sup(s') = \emptyset$. We will use the term <u>area</u> to mean an arbitrary subset of $I \times I$ and will say that an area U and a configuration s are disjoint if $\sup(s) \cap U = \emptyset$. For a pair of disjoint configurations s and s', $s \cup s'$ is the function defined in the natural way:

$$s \cup s'(\alpha) = \begin{cases} s(\alpha) & \text{if } \alpha \in \sup(s) \\ s'(\alpha) & \text{if } \alpha \in \sup(s') \\ v_0 & \text{otherwise,} \end{cases}$$

For configurations s and s' and an area U, we will write $s|U = s'$ as an abbreviation for the statement that $s|U = s'|U$ and that $s'|U^c$ is everywhere quiescent. The <u>translation</u> s_δ of a configuration s by an element $\delta \in I \times I$ is defined as follows:

$$s_\delta(\alpha) = s(\alpha - \delta),$$

where the operation is component-wise subtraction.

A configuration is called <u>passive</u> if $F(s) = s$. s is <u>completely passive</u> if every subconfiguration of s is passive.(6) Finally, s is <u>stable</u> if there exists a t such that $F^t(s)$ is passive. The problem of determining whether or not a given configuration is passive or completely passive is effective. On the otherhand (as would be expected) there is, in general, no effective method for determining whether or not a configuration is stable. The halting problem for Turing machines can be reduced to the "stability

(6) It will be clear that there exist passive, but not completely passive, configurations for the von Neumann model. This results from our definition of subconfiguration.

problem" for the von Neumann model (Section 8).

In order to talk about self-reproduction and in general about construction within a cellular model, we must be able to interpret a sequence of states of the model in such a way as to say that a particular configuration has or has not been constructed after a certain number of time steps. It seems quite reasonable to require that we interpret as constructed only those configurations appearing in areas which, at time $t = 0$, were entirely quiescent. Thus the following definition is proposed: a configuration s constructs a configuration s' if there exist an area U, disjoint from s, and a time t such that $s' = F^t(s)|U$.

A configuration s is self-reproducing if there exists a translation δ such that s constructs s_δ.

Consider the cellular model with $V = \{0, 1\}$, $v_o = 0$ and for all v_i,

$$f(v_1, v_2, v_3, v_4, 1) = 1$$

$$f(v_1, v_2, v_3, v_4, 0) = v_1.$$

In words, a cell α remains in its state of time t if its neighbor to the left is in state 0. On the other hand, if the state of the left-hand neighbor is 1, then the state of α is 1 at time $t+1$.

In this example, every configuration is self-reproducing. Yet in the von Neumann model, even with the definition given, self-reproduction is non-trivial. Why this happens, how these cases can be distinguished and variations on the definition of self-reproduction are very interesting problem areas. The main result, initiating the research into cellular automata, is given by

Proposition I: There exist self-reproducing configurations for the von Neumann model. (Section 8)

We continue now to the general problem of construction. That a configuration can exist only at time $t=0$ (a Garden-of-Eden configuration) implies, in Moore's treatment, that the configuration is not constructable. This is not the case with the definition presented here.(7) Some (but not all) Garden-of-Eden configurations can be constructed in the von Neumann model.

For the von Neumann model we will show the following:

Proposition II: There exist non-constructable configurations, (Section 2.3).

Proposition III: All completely passive configurations are constructable (Section 5.6).

It is probably not the case that all passive configurations are constructable, but the answer to this question is not known to the author.

Since we are now to consider universal cellular automata, it is necessary to develope the analog of the Turing machine tape for the

(7) Op. cit. This difference lies in the fact that his areas of construction must, in effect, be closed or filled-in. In the cited paper, Moore proves a necessary condition for the existence of Garden-of-Eden configurations. A converse to this theorem is given in Myhill, "A Converse to Moore's Garden-of-Eden Theorem", Automata Theory Notes, University of Michigan Engineering Summer Conferences, Ann Arbor, 1963, pp. 421-422.

cellular theory. The interpretation of Turing machines, or in general of effective computational processes, within the theory of cellular automata must necessarily involve the representation of both tape and control automaton as configurations. Thus, this general theory must at least take into account, and hopefully make precise, the distinction between the "passive" tape and the "active" control.

The Turing machine tape, in the usual treatment, serves two functions: (1) to provide a potentially infinite memory and (2) to provide a method of representing information for the control automaton to process. Because any automaton embedded in cellular space is itself potentially infinite, it is likely that the analog of "tape" for the cellular theory will involve essentially the problem of the representation of information. To that end, it is quite reasonable to propose the following definition: A configuration i is a <u>tape for a configuration</u> s if i is completely passive and disjoint from s. If there are completely passive configurations other than the trivial one ($i(\alpha) = v_o$ for all α) then the unrestricted definition of a tape as a completely passive configuration certainly permits the encoding of any information that could be required. Indeed, the initial reaction is very likely to be that this definition is too broad. It can be argued that any restriction of the conditions of the definition would be artificial.

When i is a tape for s, we will use the notation s(i) to denote the configuration $s \cup i$ as defined above.

A configuration s is a <u>universal constructor</u> for a class S' of configurations if for every $s' \in S'$ there exists a tape i such that s(i) constructs s'. Note that there is no universal constructor for the example mentioned above because the only tape is the trivial one (all quiescent). If the definition of tape is expanded to include all configurations then the trivial configuration is a universal constructor for this model. For the von Neumann model we will show:

<u>Proposition IV</u>: There exists a universal constructor for the class of completely passive configurations in a fixed quadrant of the plane. (Section 5.6).

This universal constructor is extremely simple, in fact, it can be taken to be one cell. The encoding of configurations is, however, complicated, requiring about 1400 times as many cells as the configuration to be constructed.

As the last general topic, we want to consider computation and universal computers within the cellular model. We will say that a tape i is <u>in an area U</u> if $i|U^c$ is everywhere quiescent. For an area U, a configuration s disjoint from U computes the partial tape function φ given as follows: For every tape i in U,

$\varphi(i) = F^t(s \cup i)\,|\,U$ if $F^t(s \cup i)$ is stable and

$F^t(s \cup i)\,|\,U$ is a tape,

$\varphi(i)$ is undefined otherwise.

Let T' be a subset of the set of all tapes. The alphabet V' of T' is that subset of V which occurs in T'. A partial function φ from T' into T' is said to be Turing-computable if there exists a Turing machine operating on a 2-dimensional tape with the symbol alphabet V' which computes φ.

A cellular model is computation universal if there exists an area U and an infinite subset T' of the set of all tapes in U such that every Turing-computable function from T' into T' is computable by some configuration of the model.

Proposition V: The von Neumann model is computation universal (Section 8).[8]

For a tape function φ and a translation δ, φ_δ is the translate of φ: $\varphi(i) = i' \longleftrightarrow \varphi_\delta(i_\delta) = i'_\delta$. Again, let U be an area and T' an infinite subset of the tapes in U. A configuration s, disjoint from U, is a universal computer if for every partial tape function φ from T' into T' which is computable in the model, there exists a tape $i \in T'$ and a translation δ such that s(i) computes φ_δ.

Proposition VI: There exists a universal computer in the von Neumann model (Section 8).

(8) C.Y. Lee, in an unpublished manuscript "Synthesis of a Cellular Universal Machine Using the 29 State Model of von Neumann", August, 1963, proves the computation universality of the von Neumann model. An extremely important difference however is that Lee allows initial configurations with infinite support.

The von Neumann model is described in detail in Section 2. In Section 3, we specify certain schema for configurations which will be useful as subconfigurations or parts of more complicated automata to be embedded in the model. In Section 4 we discuss a "programming language" for describing the behavior of a certain class of automata operating in the cellular system. This operation, motivated by Turing machine theory, is described in terms of a finite automaton controlling a tape and a construction device. The "constructor" is designed in Section 5 and in Section 6 designs are given for the tape and tape control units.

Several of the interesting propositions stated above are verified in Section 8. This results from a description in the previous section, of a class of configurations called the class of von Neumann machines. These configurations are designed by means of an algorithm from the class of programs discussed in Section 4. As a result of the generality of this algorithm, the existence of universal computers, constructors and self-reproducing von Neumann machines is easily verified.

Dealing with specific coordinates in the cellular model is a bother. We will present designs of cellular systems - configurations - by states marked in a checker-board-like rectangular area. Such a diagram can be considered to be a schema for configurations, the particular instances being obtained by assigning a coordinate to the lower left hand corner to the rectangle. For the most part in the sequel, we will be ignoring coordinates, identifying a configuration and any translation of that configuration.

2. THE CELLULUAR MODEL

2.1 The Underlying Cellular Structure

The basic structure or geometry of the cellular model is an infinite rectangular grid. Each cell of the grid is considered to be a finite automaton with 29 states. With integral values of time, the state of the cell α at time $t+1$ is a function of the state of that cell at time t and the states of the four neighbors at time t.

The four neighbors of a cell α will, for convenience, be denoted $a(\alpha)$ for $a \epsilon \{u, d, r, \ell\}$ in the reasonable correspondence. The notation a^{-1} has its natural interpretation.

2.2 The Twenty-Nine States

The 29 states can be conveniently grouped into five categories which are described below. In the discussion of each of the five groups, we indicate the behavior of the states in the group. This is made precise with the statement of the transition rule in the following Section. It must be emphasized that the discussion (A) through (E) is informal and that complete understanding of the transition function of the 29 state automaton is available through an analysis of the transitive rule of Section 2.3.

(A) At any time all but a finite number of cells are assumed to be in an unexcitable or <u>quiescent state</u> U. Areas totally in the state U are areas for potential growth or construction controlled by other areas which are in states other than the quiescent state.

(B) There are eight <u>ordinary transmission states</u>, $T^{0}_{a,\eta}$ where $a \in \{u, d, r, \ell\}$, specifying the direction of transmission, and $\eta \in \{0,1\}$ which differentiates between the excited and passive or unexcited transmission state. $T^{0}_{a,0}$ is an unexcited ordinary transmission state in the a-direction. Under interpretation, the ordinary transmission states act both as directed wires and as disjunction elements, with unit delay. Excitation ($\eta = 1$) can be interpreted as a pulse that will be transmitted to the neighbor in the a-direction at the next unit of time. The unexcited ordinary transmission states are indicated by single arrows in the diagrams of this report. A 1 over the arrow indicates excitation.

(C) There are four <u>confluent states</u>, $C_{\epsilon\eta}$, where ϵ and η are either 0 or 1. C_{00} is the unexcited confluent state designated by C in the diagrams. As was mentioned in (B), the ordinary transmission states act as directed wires, transmitting excitation from cell to cell with one unit of delay. A confluent state in a chain of transmission states acts as a two unit delay. An excited transmission state pointing at a cell in state C_{00} at time t results in the state C_{10} of that cell at time $t+1$. At time $t+2$ the state will be C_{01}. Finally, at time $t+3$ the state will be C_{00}. In addition, any ordinary transmission state (or special transmission state-- see (D) below) adjacent to, but not pointing at, this cell will be excited at time $t+3$. Thus the excitation of the ordinary transmission state is "passed through" the confluent state with one extra unit of delay. With more than one ordinary transmission state pointing towards a cell in state C_{00}, all must be excited in order to have the state transition to C_{10}. Thus the confluent state acts as a conjunction element. The transition after C_{10} is as indicated above. The confluent state C_{11} occurs when the excitation condition is met for two time steps in succession.

In summary, there are four uses of the confluent state; (1) as a fan-out element, (2) as a two unit delay, (3) as a conjunction element and (4) as an element to excite unexcited special transmission states which we now describe.

(D) The eight <u>special transmission states</u> $T^1_{a,\eta}$ for $a \in \{u, d, r, \ell\}$ and $\eta \in \{0, 1\}$ are similar in operation to the ordinary transmission states. They are almost dual to the ordinary transmission states in their role in construction and destruction. The "almost" comes from the fact that an excited special transmission state pointing in the direction of a confluent state produces the quiescent state U in place of the excited confluent state at the next time step. Otherwise, a special transmission state can convert an ordinary transmission state to U and conversely. The symbols $\Uparrow$, $\Downarrow$, $\Longrightarrow$ and $\Longleftarrow$ are used to denote unexcited $(\eta = 0)$ special transmission states in the diagrams.

(E) Finally there are eight transient or <u>sensitized states</u>, S_Σ where $\Sigma \in \{\theta, 0, 1, 00, 01, 10, 11, 000\}$. These will be intermediate states for the transformation of a quiescent state U into one of the ten unexcited states.

<u>2.3 The Transition Rule</u>

To completely specify the transition function of the 29 state automaton in tabular form would require about 20 million entries. As an alternative we introduce the notation of a binary predicate for each of the 29 states. Thus $U(\alpha, t)$ is true if cell α is in state U at time t. We will give a listing of the necessary and sufficient conditions for each of the predicates $R(\alpha, t)$ where R ranges over all of the 29 states. Some comment is given with most of these conditions. The quantified variable, c, ranges over the set of directions, $\{u, d, r, \ell\}$. For the variable ϵ which takes the values 0 and 1, $\overline{\epsilon}$ denotes the complement. Free variables should be interpreted universally.

In each case where a subscript or superscript is omitted, this is an abbreviation for the corresponding disjunction. For example,

$$T^\epsilon_a(\alpha, t) =_{df} T^\epsilon_{a,0}(\alpha, t) \vee T^\epsilon_{a,1}(\alpha, t).$$

Before getting into the actual transition rule we will define the destruct and excite conditions for the transmission and confluent states and, in so doing, will introduce some abbreviations which will then appear in the statement of the transition rule.

(1) Destruct condition for a transmission state:

$$D(T^{\epsilon}_{a}) =_{df} \exists c[T^{\bar{\epsilon}}_{c,1}(c^{-1}(\alpha), t-1)]$$

The destruct condition for an ordinary (special) transmission state at α is that there exist an excited special (ordinary) transmission state at a neighbor of α, pointing at α.

(2) Destruct condition for a confluent state:

$$D(C) =_{df} \exists c\ [T^{1}_{c,1}(c^{-1}(\alpha), t-1)]$$

(3) Excite condition for a transmission state:

$$E(T^{\epsilon}_{a}) =_{df} \exists c\ [c^{-1} \neq a \wedge [(T^{\epsilon}_{c,1}(c^{-1}(\alpha), t-1) \vee C_{01}(c^{-1}(\alpha), t-1) \vee C_{11}(c^{-1}(\alpha), t-1)]\ ]$$

Thus, the excite condition for an ordinary (special) transmission state is the existence of an excited ordinary (special) transmission state or an excited confluent state ($C_{\eta 1}, \eta \in \{0, 1\}$) at a neighbor of α. The transmission state must be pointed towards α, the confluent state is not directional.

(4) The excite condition for a confluent state:

$$E(C) =_{df} \exists c\ [T^{0}_{c}(c^{-1}(\alpha), t-1)] \wedge \forall c\ [T^{0}_{c}(c^{-1}(\alpha), t-1) \longrightarrow T^{0}_{c,1}(c^{-1}(\alpha), t-1)]\ .$$

The excite condition for the confluent state is that all ordinary transmission states pointed towards it must be excited.

We can now write out the transition rule.

$$(5)\quad T^{\epsilon}_{a,1}(\alpha,t) \longleftrightarrow T^{\epsilon}_{a}(\alpha,t-1) \wedge E(T^{\epsilon}_{a}) \wedge \neg D(T^{\epsilon}_{a})$$

Thus, a cell in ordinary (special) transmission state (excited or unexcited) at time t-1 will be in excited transmission state at time t if there is an excite condition and no destruct condition (the destruct condition takes precedence throughout).

The rule for the confluent state is exactly as discussed above:

$$(6)\quad C_{1\eta}(\alpha,t) \longleftrightarrow [C_{\eta 0}(\alpha,t-1) \vee C_{\eta 1}(\alpha,t-1)] \wedge E(C) \wedge \neg D(C)$$

$$(6')\quad C_{01}(\alpha,t) \longleftrightarrow [C_{10}(\alpha,t-1) \vee C_{11}(\alpha,t-1)] \wedge \neg E(C) \wedge \neg D(C)$$

We now have necessary and sufficient conditions for all of the excited states.

Since the unexcited transmission and confluent states can arise from the construction operation, we will next specify that part of the transition rule governing the transition from the quiescent state through the sensitized states to the unexcited transmission and confluent states. To simplify the formulas we abbreviate with the following notation:

$$(7)\quad B_1(\alpha) =_{df} \exists c\, [T_{c,1}(c^{-1}(\alpha), t-1)]$$

$$(8)\quad B_0(\alpha) =_{df} \neg B_1(\alpha).$$

$B_1(\alpha)$ holds if (at time t-1) there is an excited transmission state (special or ordinary) pointing towards α.

(9) $S_\theta(\alpha, t) \longleftrightarrow U(\alpha, t-1) \wedge B_1(\alpha)$.

If an excited transmission state is pointed toward a quiescent state, this cell assumes the sensitized state S_θ at the next time step. The next two rules determine the sequence of transient states; the final outcome of which will be one of the unexcited states.

(10) $S_{\Sigma 1}(\alpha, t) \longleftrightarrow S_\Sigma(\alpha, t-1) \wedge B_1(\alpha)$ $[\Sigma = \theta, 0, 1]$

(11) $S_{\Sigma 0}(\alpha, t) \longleftrightarrow S_\Sigma(\alpha, t-1) \wedge B_0(\alpha)$ $[\Sigma = \theta, 0, 1, 00]$

Here $\theta 0 = 0$ and $\theta 1 = 1$. (10) and (11) provide rules for passing from S_θ through one if the possible sequences terminating in S_Σ with $\Sigma = 00, 01, 10, 11,$ or 000. We can now write the rule for the unexcited states.

(12) $T^\epsilon_{a,0}(\alpha, t) \longleftrightarrow [S_\Sigma(\alpha, t-1) \wedge B_\eta(\alpha)]$
$\vee [T^\epsilon_a \ (\alpha, t-1) \wedge \neg E(T^\epsilon_a) \wedge \neg D(T^\epsilon_a)]$

where the following relationships between a, ϵ and Σ, η hold.

$\epsilon = 0$	Σ	η
$(\uparrow)u$	000	1
$(\rightarrow)r$	000	0
$(\downarrow)d$	01	0
$(\leftarrow)\ell$	00	1

Ordinary

$\epsilon = 1$	Σ	η
$(\Uparrow)u$	10	0
$(\Rightarrow)r$	01	1
$(\Downarrow)d$	11	0
$(\Leftarrow)\ell$	10	1

Special

Now, the corresponding rule for the confluent state:

$$(13)\quad C_{00}(\alpha,t) \longleftrightarrow [S_{11}(\alpha,t-1) \wedge B_1(\alpha)]$$
$$\vee\, [[C_{01}(\alpha,t-1) \vee C_{00}(\alpha,t-1)] \wedge \neg D(C) \wedge \neg E(C)]\ .$$

Finally we state the rule for the quiescent state.

$$(14)\quad U(\alpha,t) \longleftrightarrow [U(\alpha,t-1) \wedge B_0(\alpha)] \vee [C(\alpha,t-1) \wedge D(C)]$$
$$\vee\, [T(\alpha,t-1) \wedge D(T)]\ .$$

Therefore a cell is in the quiescent state at time t just in case it was in the quiescent state at t-1 and there were no excited transmission states pointing towards it at t-1, or it was in any of the confluent or transmission states which encountered destruct conditions at t-1.

The actual functioning of this transition rule will very soon become clear to the reader after examining some of the simple examples which are presented in the next section.

Any configuration consisting only of the ten unexcited states $(C_{00}, T^{\epsilon}_{0,\alpha}, U)$ is completely passive. An example of a passive and not completely passive configuration is a cycle of excited ordinary transmission states. The state of the configuration does not change with time, but any proper subconfiguration will have an excited ordinary transmission state pointing at a quiescent state. Such a configuration is not passive.

A non-constructable configuration is obtained by completely surrounding the sensitized state S_{θ} by unexcited (C_{00}) confluent states. It is easily verified from the transistion rule that this configuration can occur as a subconfiguration only at time $t = 0$.

3. BASIC ORGANS

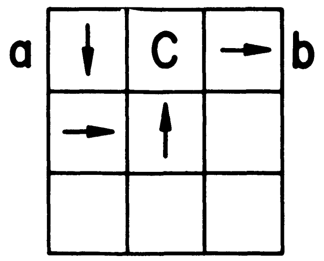
3.1 Introduction

In a manner analoguous to the organization of present day computing equipment, we will consider a structural hierarchy of an automaton to be embedded in the von Neumann cellular system. The cellular level, which is the lowest level in this hierarchy, has been discussed above. In this section we will consider the design of four general purpose components, corresponding to the second level. Each will be used repeatedly in the design of the components of the third level of the organizational hierarchy. The divisions on this last level include the tape unit, the constructing unit, and the supervisory (control) unit.

The construction of the basic organs differs from that used by von Neumann. This difference is reflected primarily in a uniformity of the size and delay of these organs resulting in some simplification of connection.

3.2 Notation and Conventions

The diagrams referred to in the sequel are all drawn with unexcited or passive states using the symbolism explained in the previous section. Cells not labeled are assumed to be in the quiescent state U. A junction of cells can be considered to be a monadic predicate over time. The convention is most easily described using examples. In Figure 3.2.1 the junction a is considered to be an input, the junction b is the output.



A CYCLE IN THE CELLULAR SYSTEM

Figure 3.2.1

a(t) has the value 1 (true) if the adjacent cell to the right will be excited at t+1, i.e., a(t) is true when the cell to the left is in one of the states $T^0_{r,1}$, C_{01} or C_{11} at time t. Similarly b(t) is true if the ordinary transmission state on the left is excited at time t. With this convention we can analyze the behavior of subconfigurations in the cellular model. From Figure 3.2.1 it can be verified that

$$a(t) \longrightarrow \forall n\ b(t+5n+6)$$

or alternatively,

$$b(t) \longrightarrow \exists n\ a(t-5n-6).$$

It will be necessary in the discussion below to speak of binary sequences as inputs or outputs of subconfigurations. In general the notation $i_1 \ldots i_n$ will be used for such a sequence. To say that the sequence $i_1 \ldots i_n$ occurs at input a at time t is the same as the set of equivalences:

$$a(t) \longleftrightarrow i_1$$

$$\ldots$$

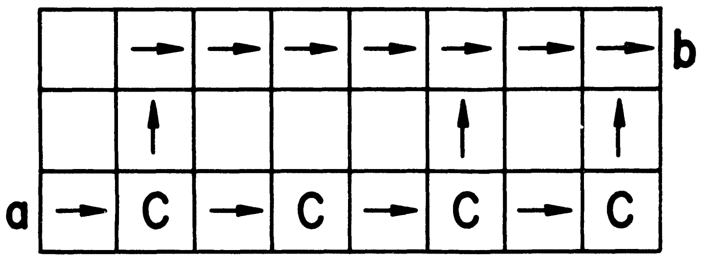
$$a(t+n-1) \longleftrightarrow i_n .$$

Here the i_k are interpreted as the propositional constants. Similarly $i_1 \ldots i_n$ as an output sequence at time t for some junction b is interpreted as $b(t+k-1) \Longleftrightarrow i_k$.

3.3 The Pulser

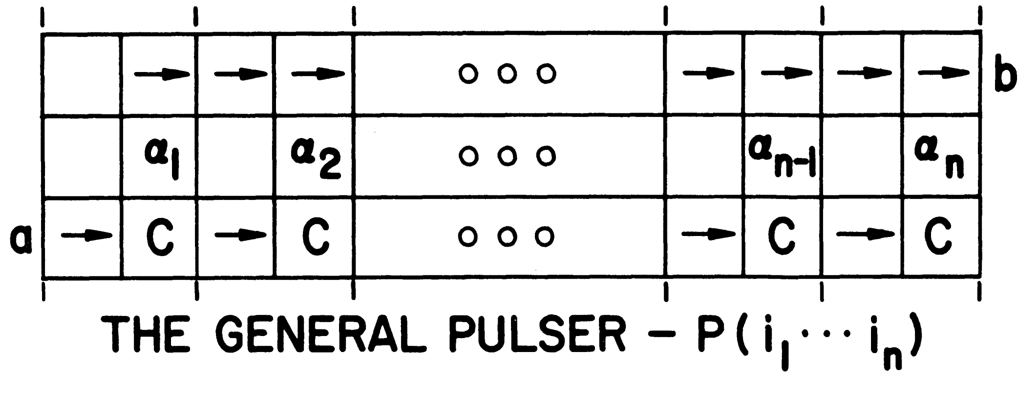
The notation for the general pulser is $P(i_1 \ldots i_n)$ where $i_1 \ldots i_n$ is the characteristic and n is the order of the characteristic. The pulser has one input and one output. It is to be constructed so that a single input pulse at time t produces the output $i_1 \ldots i_n$ at time $t+\Delta$. The delay Δ will be a function of the order of the characteristic.

The output sequence is formed in the pulser by multiplexing in an appropriate manner. As an example the pulser P(1011) is constructed in Figure 3.3.1. The generalized pulser is drawn in Figure



THE PULSER - P(1011)

Figure 3.3.1



DELAY $=2n+3$

LENGTH $=2n$

HEIGHT $=3$

$i_K = 1$ THEN $\alpha_K = \uparrow$

$i_K = 0$ THEN $\alpha_K = U$

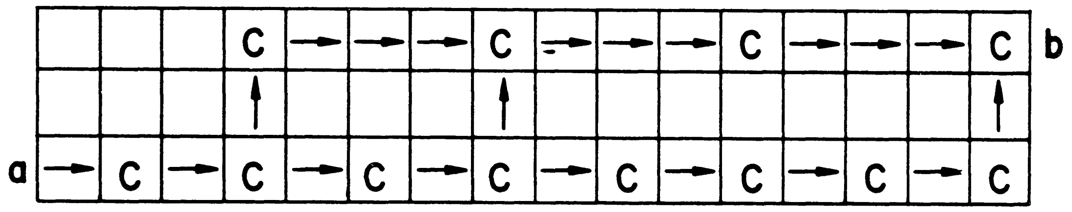
Figure 3.3.2

3.3.2. In this figure, if i_k of the characteristic is 1 then α_k is an ordinary transmission state pointing up. Otherwise, α_k is the quiescent state U. If $\alpha_k = \uparrow$ the the input pulse from a is multiplexed through this transmission state. The delay of the pulser can be calculated by considering the shortest possible path through the pulser. This path through α_1 has delay 2n+3. Hence, an input 1 at a time t results in the output $i_1 \ldots i_n$ at t+2n+3.

The pulser will be useful for two purposes. First, the pulser will be used for the coding of commands in the main channel which will connect the major functional units of a von Neumann machine (Section 7), and second, it will be used to create sequences of pulses that will perform the tape and constructing operations (Sections 5 and 6).

3.4 The Decoder

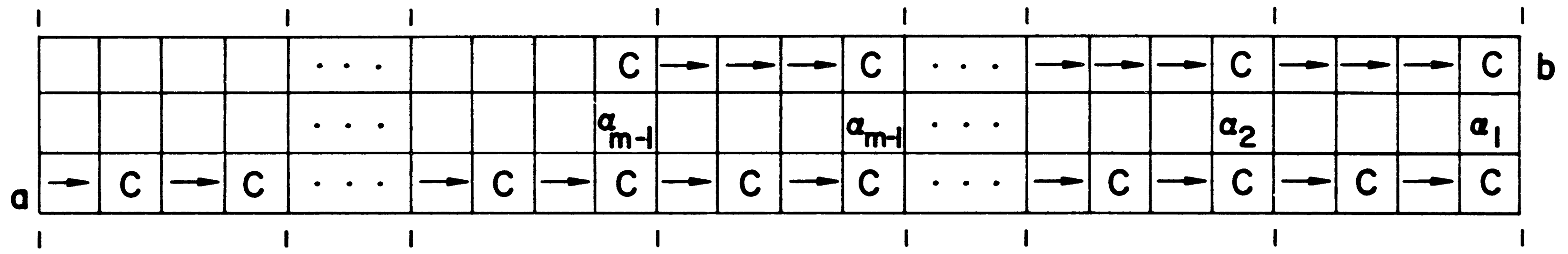
It is necessary in the coding scheme for the general von Neumann machine to be able to detect the existence of particular sequences of input pulses. The decoder, $D(i_1 \ldots i_n)$, partially fulfills this task in that it will detect any input sequence $j_1 \ldots j_n$ which is bit-wise implied by the characteristic of the decoder. That is, the response of the decoder $D(i_1 \ldots i_n)$ to the input sequence $j_1 \ldots j_n$ will be 1 if $i_k \longrightarrow j_k$ $(1 \leq k \leq n)$ and the response will be 0 otherwise.



THE DECODER - D(1011)

Figure 3.4.1

An example of a decoder is constructed in Figure 3.4.1 and the general decoder is drawn in Figure 3.4.2 (the method of construction is valid only in case there is a k such that $i_k = 1$). As with the pulser, the decoder is constructed in segments. The desired output is produced by successive confluences occurring at the confluent state in the upper right-hand corner of each segment.



THE GENERAL DECODER $D(i_1 \cdots i_n)$

DELAY = 6n+3
LENGTH = 4n
HEIGHT = 3

m is the largest index such that $i_K=0, m<K\leq n$.
$i_K = 1$ then $\alpha_K = \uparrow$
$i_K = 0$ then $\alpha_K = U$

Figure 3.4.2

Using the notation of Section 3.2, it can be verified that

$$b(t+27) \Longleftrightarrow a(t) \wedge a(t+2) \wedge a(t+3)$$

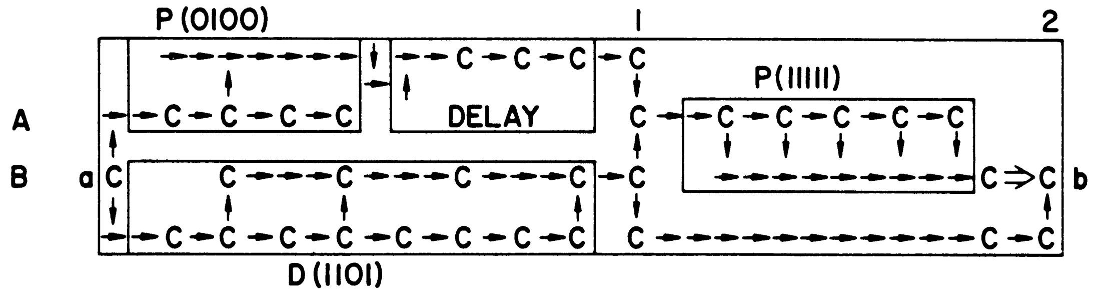
for D(1011) in Figure 3.4.1. This expression is independent of $a(t+1)$, hence whether $a(t+1)$ or $\neg a(t+1)$, the output $b(t+27)$ will be produced. In the general case we have:

$$b(t+6n+3) \Longleftrightarrow \bigwedge_{k=1}^{n} [i_k \longrightarrow a(t+k-1)] .$$

For two sequences, Π_1 and Π_2 of length n, let $\Pi_1 > \Pi_2$ if and only if Π_1 bit-wise implies Π_2. It is possible to restrict the set of allowed input sequences such that $\Pi_1 > \Pi_2$ only in case $\Pi_1 = \Pi_2$. If this method is used decoders will serve to detect any of the input sequences without ambiguity. This method was used by von Neumann and will be applied in Section 6 for coding within the tape unit. When large classes of input sequences are desired and if detection is to be unique, it is convenient to use the recognizer for decoding purposes.

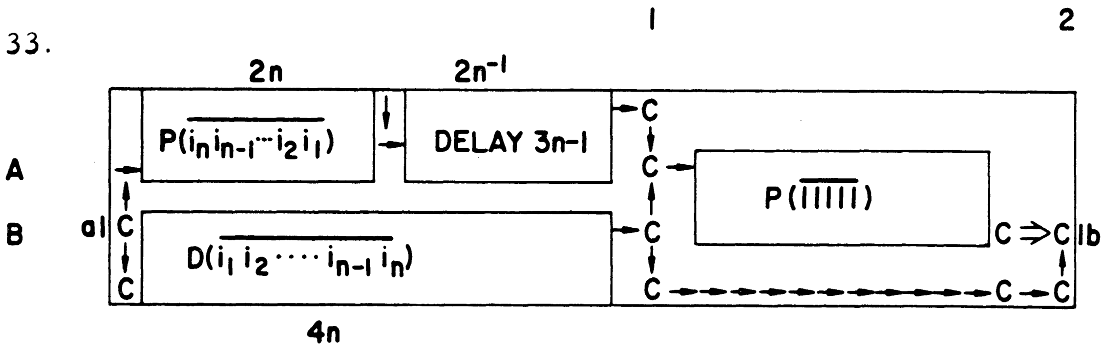
3.5 The Recognizer

The recognizer $R(i_1 \ldots i_n)$ with characteristic $i_1 \ldots i_n$ of order n is to produce an output 1 if and only if the input sequence is $i_1 \ldots i_n$. The recognizer might be considered intermediate between the second and third levels of the structural hierarchy. In its general design (Figure 3.5.2) two pulsers and a decoder are components of the recognizer.



THE RECOGNIZER - R(1101)

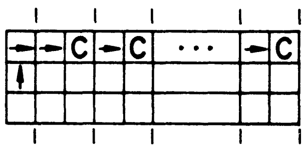
Figure 3.5.1



THE GENERAL RECOGNIZER $R(i_1 i_2 \cdots i_n)$

DELAY 6n+33
LENGTH 4n+17
HEIGHT 7

Figure 3.5.2



DELAY AREA FOR $R(i_1 \ldots i_n)$

Figure 3.5.3

For elements of the characteristic, $\bar{i}_k$ denotes the negation or complement of i_k. Hence, in the example (Figure 3.5.1) the recognizer has components D(1101), P(0100) and the control pulser P(11111).

The important cell for the analysis of the recognizer is the confluent state with coordinates (A,1) in Figure 3.5.1. In the example only the sequences 1111 and 1011 will be detected by the decoder D(1011). Therefore, in all other cases, there will be no pulses past the column labeled 1. However if 1111 occurs, the undesired $i_2 = 1$ will produce an input to the confluent state (A, 1) at exactly the same time that the decoder output reaches (A, 1). The resulting output of the confluent state will stimulate the control pulser P(11111). The output pulse of the decoder arrives at the confluent state (B, 2) at the same time that a special pulse (from the special transmission state pointing to the right) converts the C to a U. By the precedence of the transition rule, there will be no output. The remaining pulses from P(11111) convert U back into the state C.

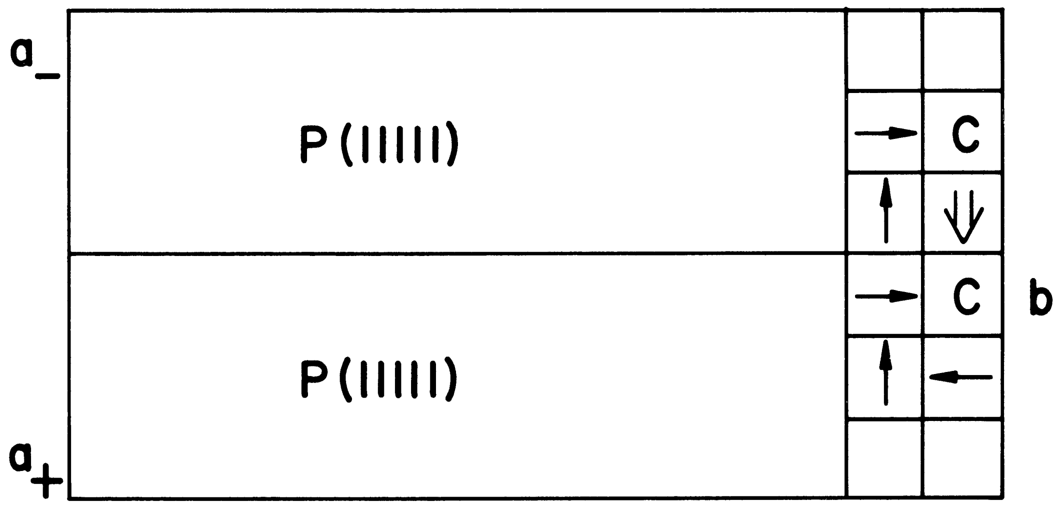
In general the output of the confluent state A-1 (Figure 3.5.2) is represented by:

$$\bigvee_{\kappa=1}^{n} (j_k \wedge \bar{i}_k) \wedge \bigwedge_{k=1}^{n} (i_k \longrightarrow j_k).$$

Therefore, the output of the recognizer at b occurs if and only if the input sequence is identical to the characterisitc of the recognizer.

3.6 The Periodic Pulser

The last general purpose component to be constructed is the periodic pulser for which the notation PP(1) will be used. This organ has two inputs labeled a_+ and a_- and one output which is labeled b. An input 1 at time t at a_+ "turns on" the pulser so that at $t+\Delta_+$ an unending sequence of 1's will appear at b. An input at a_- "turns off" the pulser. That is, if a 1 occurs at a_- at time t then at $t+\Delta_-$ the sequence of 1's will be terminated.



DELAY a_+ 16 LENGTH 12

DELAY a_- 18 HEIGHT 6

THE PERIODIC PULSER-PP(1)

Figure 3.6.1

The general plan of construction is as follows. In the von Neumann cellular system, the smallest cycle (Figure 3.2.1) which can be constructed with an output has a period of five. The input pulse at a_+ will therefore produce a sequence 11111 which "fills" the cycle with excited states. The input a_- destroys the confluent state of the cycle. The number of steps for which the confluent state is inoperative will be five, hence all the pulses in the cycle will be ineffectual and the sequence will be terminated.

The periodic pulser is drawn in Figure 3.6.1. As with the recognizer it is constructed of previously described units. Note that the upper pulser is inverted. This operation of inverting a component is always possible when no special transmission state is involved. In general, any component may be rotated to any possible position under these conditions. In addition, if there do exist special transmission states in the component (e.g., $R(i_1 \ldots i_n)$ and $PP(1)$) the statement above is still valid so long as construction is involved only in passing to and from confluent states.

4. INTERLUDE

In order to give further motivation for the designs to follow we will consider a formulation of effective computation and construction as a slight extension of standard Turing machine formulation. Because it is so convenient we will use the program method of describing Turing machine behavior, both for computation and construction.(9)

The model (as drawn in Figure 4.1) consists of a finite control automaton attached to a read-write head (via the tape arm) which scans a one-way infinite linear tape (here called the tape area). The alphabet of possible tape symbols is the usual $\{0, 1\}$ and in any computation of this device, it is assumed that all but a finite number of squares of the tape are initially 0. This much of the model corresponds to the usual concept of a Turing machine; the control automaton, can, through the read-write head, sense or modify the contents of the square being scanned and can move the scanning position one square to the right or left.

The control automaton is also attached to a constructing head (via the constructing arm) which scans a 2-dimensional constructing area. The symbol alphabet for the constructing area is a finite set V_c containing a distinguished element U. The initial condition of the constructing area is assumed to be everywhere U. This area is a write only tape. Thus the control automaton can, through the constructing arm, write any of the symbols of V_c on the scanned square or move the constructing head one

(9) H. Wang, "A Variant to Turing's Theory of Computing Machines", J. of Symbolic Logic, 4, (1957) 63-92.

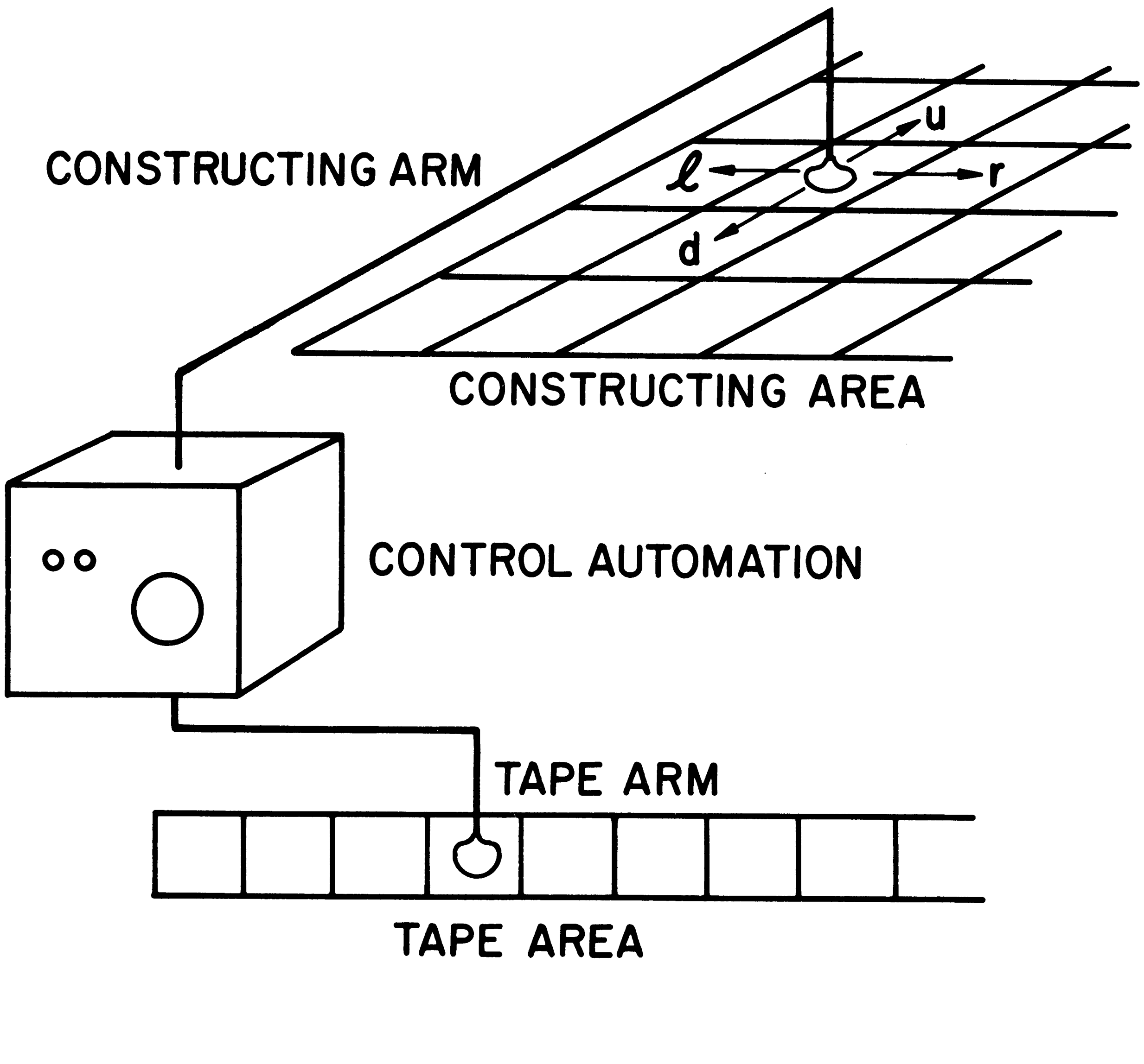


Figure 4.1 A Constructing Turing Machine

square in any of the four directions.

The control automaton operates according to a finite sequence of instructions (a program) taken from the following two lists.

Computing Instructions:

+	Move right one square on the tape.
-	Move left one square on the tape.
e	Place a 0 on the square under scan (erase).
m	Place a 1 on the square under scan (mark).
tn	If the current scanned square is 1, proceed to instruction n. Otherwise continue in sequence.
*	Stop.

Construction Instructions:

u, r, d, ℓ	Move the constructing arm one square in the direction specified.
Bx	Where x is an element of V_c, Bx requires the construction of x in the cell currently scanned by the constructing arm.

The first list is a standard list for Turing machines. The list for construction includes an instruction for each of the allowed operations as indicated above. A program in the language being described is simply a sequence of instructions, $f_1, \ldots, f_n$, with the following two restrictions:

(i) if f_i is tk then $k \leq n$

(ii) $f_n = *$.

Given an initial tape (the input), the control automaton begins with the first instruction and continues to perform the operations on the tape and constructing area as dictated by the program.

Any effective computation and construction can be carried out by a control automaton operating according to a program in the language described (this corresponds to Turing's thesis).

Our objective now is to embed this model of effective computation and construction in the von Neumann cellular system. Fortunately, for Turing machine theoreticians, it is not necessary to consider how the control automaton sequences and executes the instructions of its program or how reading, writing and motion of read-write head are accomplished. These, however, are our major problems.

For a given program, the corresponding embedded system will consist of three major functioning units: (1) the control automaton, hereafter called the supervisory unit, (2) the tape unit with tape and (3) the constructing unit and a constructing arm. These units must be interconnected to allow for the transmission of information from the supervisory unit to the tape and constructing units and from the tape unit to the supervisory unit. The general arrangement of this system-called a von Neumann machine-is pictured in Figure 7.1.1. The design of the tape and constructing units is fixed (Sections 5 and 6) whereas the supervisory unit depends on a given program. Thus in Section 7 an algorithm is presented which, as a function of a program or sequence of instructions taken from the two lists above, produces the design of a supervisory unit such that the resulting configuration of supervisory, tape and constructing units behaves as the program dictates.

The design of the constructing unit will be carried out first because the method of construction as well as the organization of the constructing unit control will be useful in the design of the tape unit. In specifying these designs we will adopt several conventions that are convenient in talking about the von Neumann model. A row of transmission states, either ordinary or special, acts like a wire over which pulse sequences can be transmitted and in fact, this is the method by which information is transmitted between components of an automaton embedded in the von Neumann cellular structure. Thus we will speak loosely of sequences, originating from pulsers, being transmitted along certain channels or from certain designated points to others. With similar lack of rigor we will refer to the tape reading head and the constructing arm as moving from one position to another. This, of course, requires the destruction of the state configuration representing the unit in the area that it occupies and the reconstruction of that same configuration in some new area one square removed. The methods for accomplishing this are described in the next section,

5. THE CONSTRUCTING UNIT

5.1 Introduction

It would be possible to design the constructing unit so that a von Neumann machine would have the potential of constructing any finite array of unexcited states (i.e., any inactive finite automaton) at any position in the surrounding cells of the infinite plane. However, in the design to be presented here the location of the construction is restricted to a quadrant of the plane as indicated in Figure 5.2.1. The process of construction will be accomplished by an array of cells in ordinary and special states extending out from the constructing unit. This array will be called the constructing arm. Construction of the unexcited states will take place at the end of this arm.

There will be two basic types of operation for the constructing unit. First, there must be the mechanism to manipulate (extend and contract) the constructing arm, and secondly, it must be possible to construct each of the 10 unexcited states at the end of the arm.

It will be assumed that all the cells with which the constructing arm comes in contact are initially in the quiescent state U.

5.2 The Constructing Arm - Inputs to the Constructing Unit

The organization of the constructing arm is indicated in Figure 5.2.1. It consists of two rows of cells; a row of special transmission states and a row of ordinary transmission states. The inputs to these channels are S and O respectively. The point where construction

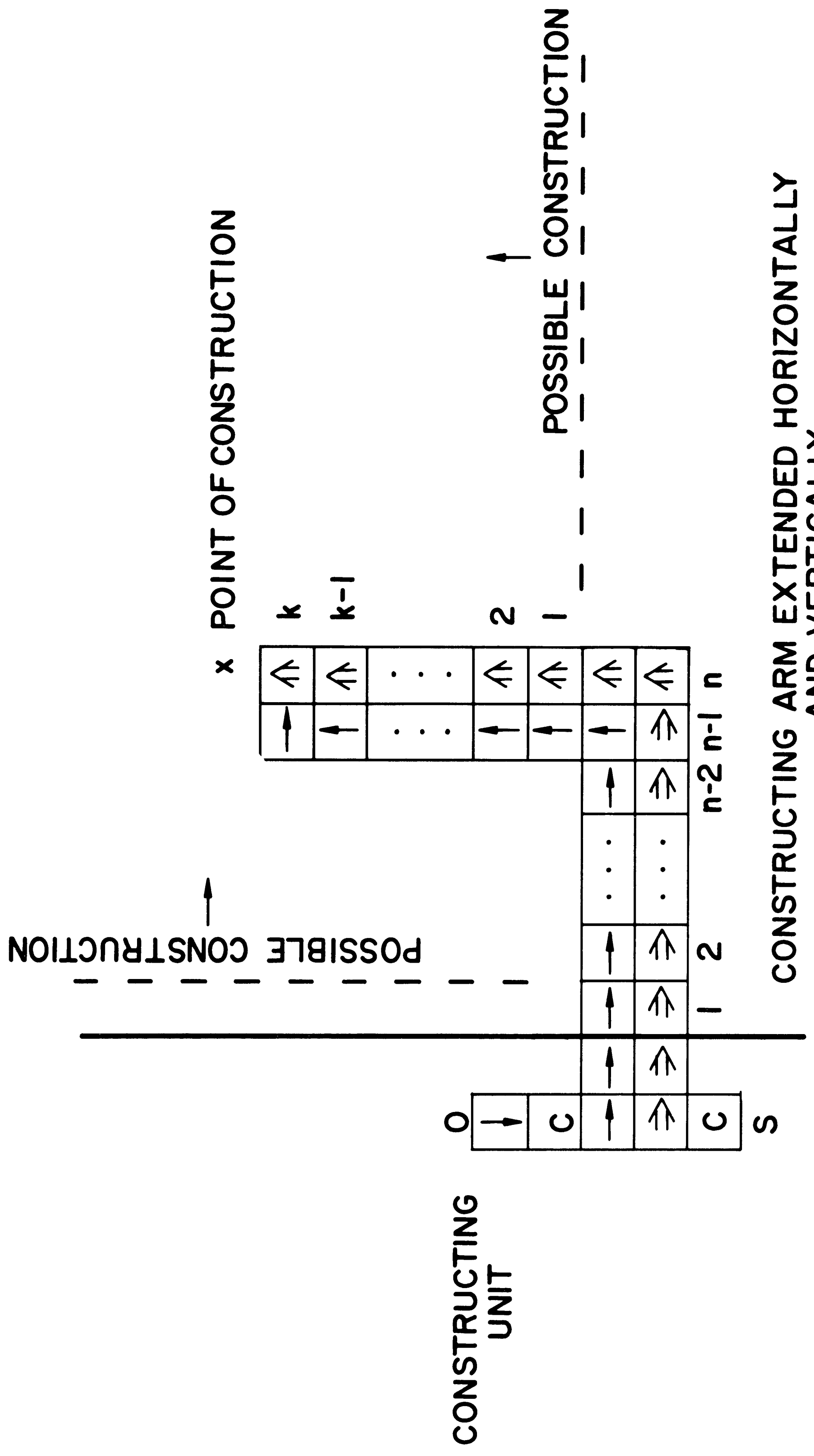

POINT OF CONSTRUCTION
POSSIBLE CONSTRUCTION
POSSIBLE CONSTRUCTION
CONSTRUCTING UNIT
CONSTRUCTING ARM EXTENDED HORIZONTALLY AND VERTICALLY
x
k
k-1
2
1
n
n-1
n-2
2
1
O
C
C
S

Figure 5.2.1

takes place is indicated at the end of the arm in the output direction of the last special transmission state.

There will be four inputs to the constructing unit specifying the positioning of the arm:

Advance the arm vertically one square	$\underline{u}$
Advance the arm horizontally one square	$\underline{r}$
Retract the arm one square in the vertical direction	$\underline{d}$
Retract the arm one square in the horizontal direction	$\underline{\ell}$

(The inputs to the various units are symbolized by the corresponding "instruction" underlined).

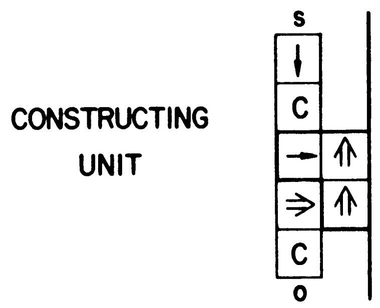


Figure 5.2.2 Initial Configuration of the Constructing Arm

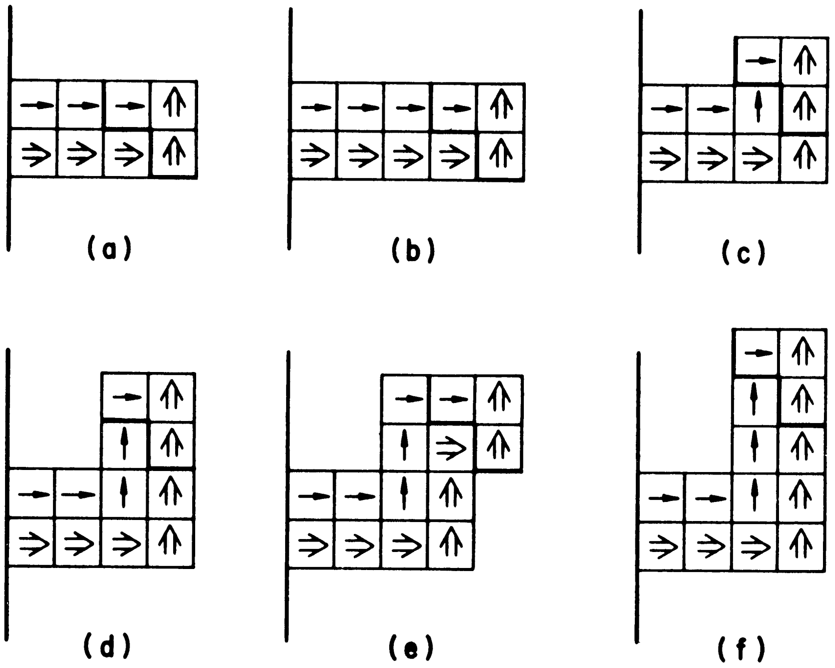


Figure 5.2.3

In Figure 5.2.1 the arm has been advanced horizontally n times and vertically k times from the initial position indicated in Figure 5.2.2. This configuration could be achieved by n $\underline{r}$ inputs followed by k $\underline{u}$ inputs to the constructing unit.

In Figure 5.2.3 several arm configurations are drawn to demonstrate the effects of the various inputs. Note that the three terminal cells which are outlined in Figure 5.2.2 have the same orientation in every position of the arm. These three cells will be called the "constructing head" analogous to the readwrite head of a tape. We will say that the arm is in a horizontal position if the inputs

Relations Between the Positions and Inputs for the Constructing Arm

Inputs to the Constructing Unit

Labeling in Figure 5.2.3	r	*l*	u	d
a	b	not shown	c	impossible
b	not shown	a	not shown	impossible
c	not shown	impossible	d	a
d	e	impossible	f	c
e	not shown	d	not shown	impossible
f	not shown	impossible	not shown	e

Table 5.2.1

to the constructing head are from the left and in a vertical position if the inputs are from below. Hence in Figures 5.2.1, 5.2.3c, d, and f the arm is in the vertical position whereas the arm is in the horizontal position in Figures 5.2.3a, b, and e.

When the arm is in the horizontal position of Figure 5.2.3a (achieved by four r inputs), the r input moves the constructing point one square to the right (Figure 5.2.3b) whereas a u input moves the constructing point one square in the vertical direction (Figure 5.2.3c). The entries in Table 5.2.1 are the results obtained from the input that labels the column when the arm is in the position of the figure that labels the row (these latter labels refer to Figure 5.2.3). Note that some input-state configurations are impossible since they will result in erroneous modifications. The incompatibility exists when the arm is in a horizontal position and a d input is given or when the arm is in a vertical position and an ℓ input is required.

With the design of the arm as presented, the constructing operations are all trivial. It is only necessary to transmit a five digit construction sequence along the special channel of the arm. For example, assume that an unexcited confluent state is desired at the point of construction in Figure 5.2.1. Then by the transition rule, the sequence 1111 as an input at S would perform the desired construction. Ten inputs for cell construction are required. (An

Constructing Unit Inputs

Notation	Function	Sequence to Constructing Arm
$\underline{r}$	Horizontal Advance	See Table 5.3.5
$\underline{\ell}$	Horizontal Retreat	See Table 5.3.5
$\underline{u}$	Vertical Advance	See Table 5.3.5
$\underline{d}$	Vertical Retreat	See Table 5.3.5
$\underline{B\rightarrow}$	Build $\rightarrow$	10000
$\underline{B\uparrow}$	Build $\uparrow$	10001
$\underline{B\leftarrow}$	Build $\leftarrow$	10010
$\underline{B\downarrow}$	Build $\downarrow$	10100
$\underline{B\Rightarrow}$	Build $\Rightarrow$	10110
$\underline{B\Uparrow}$	Build $\Uparrow$	11000
$\underline{B\Leftarrow}$	Build $\Leftarrow$	11010
$\underline{B\Downarrow}$	Build $\Downarrow$	11100
$\underline{BC}$	Build C	11110
$\underline{BU}$	Build U	00000
$\underline{s}$	Stimulate	See Table 5.3.6

Table 5.2.2

input is included for constructing the quiescent state U. This last is really not necessary because of the assumption that the states surrounding the construction arm are quiescent. The construct U input is included for the uniformity that it permits in programming construction.) These inputs are specified in Table 5.2.2 along with the sequences that must be transmitted into the special channel. The arm positioning inputs are also included in this table.

The input labeled s (stimulate) is included for the possibility of "starting" the constructed automaton. The required sequence is discussed at the end of Section 5.3.

5.3 Positioning the Constructing Arm

An arm positioning input to the constructing unit will simultaneously stimulate two pulsers corresponding to the appropriate operation. One pulse sequence will be transmitted along the special transmission channel (with input S in Figure 5.2.1) and the other pulse sequence will be transmitted down the ordinary channel, (input O in 5.2.1).

Construction operations (state changes) are carried out by these sequences at the end of the arm. One important point in this method is that the initial pulses of the two sequences are at the same place in the corresponding channels at the same time. Hence if the ordinary sequence is $i_1 \ldots i_m$ and the special sequence is

$j_1 \ldots j_m$, then i_1 and j_1 will arrive at S and O, respectively, at the same time. It is easily verified that i_1 and j_1 will also arrive at the corresponding inputs to the constructing head at the same time. Using this convention the construction originating from the ordinary and special channels may, in effect, be alternated while maintaining a single rigidly timed sequence on each of the two input channels. For example, if $i_k \ldots i_{k+r}$ is a construction sequence for the ordinary channel then the corresponding subsequence for the special channel will be 0...0 (r+1 times).

It is simply a matter of detail to write down the sequences that perform the desired operations. They are presented in the following tables and figures.

	Required Sequences	Modification at the End of the Constructing Arm
$\underline{r}$	Table 5.3.1	Figure 5.3.1
$\underline{\ell}$	Table 5.3.2	Figure 5.3.2
$\underline{u}$	Table 5.3.3	Figure 5.3.3
$\underline{d}$	Table 5.3.4	Figure 5.3.4

The figures show the end of the construcing arm during the positioning.

We will analyze one of these operations in detail. Choosing the horizontal advance (Figure 5.3.1 and Table 5.3.1), each row of the Table corresponds to a basic step in the procedure amounting to the modification of a single cell at the end of the arm. The corresponding figure

Pulse Sequences for Horizontal Advance

Step	Special	Ordinary	Cell	Old State	New State
b	00000	11010	(2, 2)	⇑	↓
c	00000000	11011000	(3, 2)	⇑	⇒
d	1100	0000	(3, 3)	U	⇑′
e	1101	0000	(2, 3)	U	⇐
f	110000000	000000000	(2, 2)	↓	→
g	0000	1100	(2, 3)	⇐	⇑

Table 5.3.1

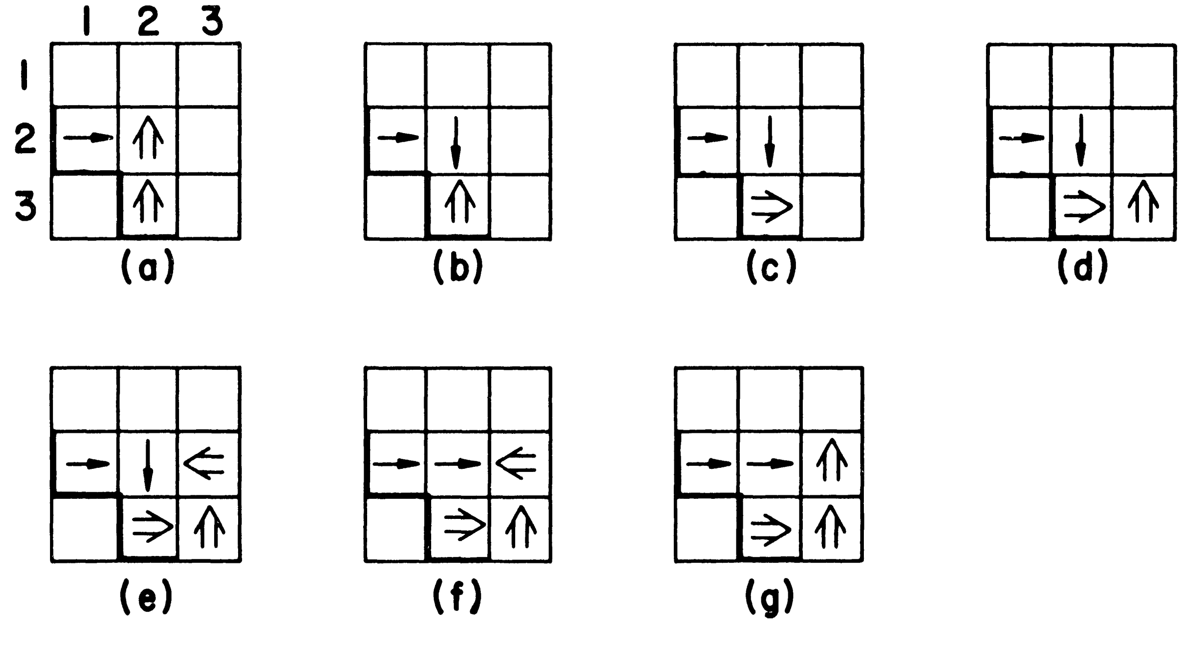


Figure 5.3.1

Pulse Sequences for Horizontal Retreat

Step	Special	Ordinary	Cell	Old State	New State
b	00000	11010	(2, 3)	⇑	↓
c	00000	11001	(3, 3)	**⇑**	←
d	0000000000	1110000000	(3, 2)	**⇒**	⇑
e	11011	00000	(2, 2)	→	**⇒**
f	11110	00000	(2, 3)	↓	⇓
g	10	00	(3, 3)	←	U
h	000000	110000	(2, 2)	**⇒**	**→**
i	00	10	(2, 3)	⇓	U
j	11100	00000	(2, 2)	**→**	⇑

Table 5.3.2

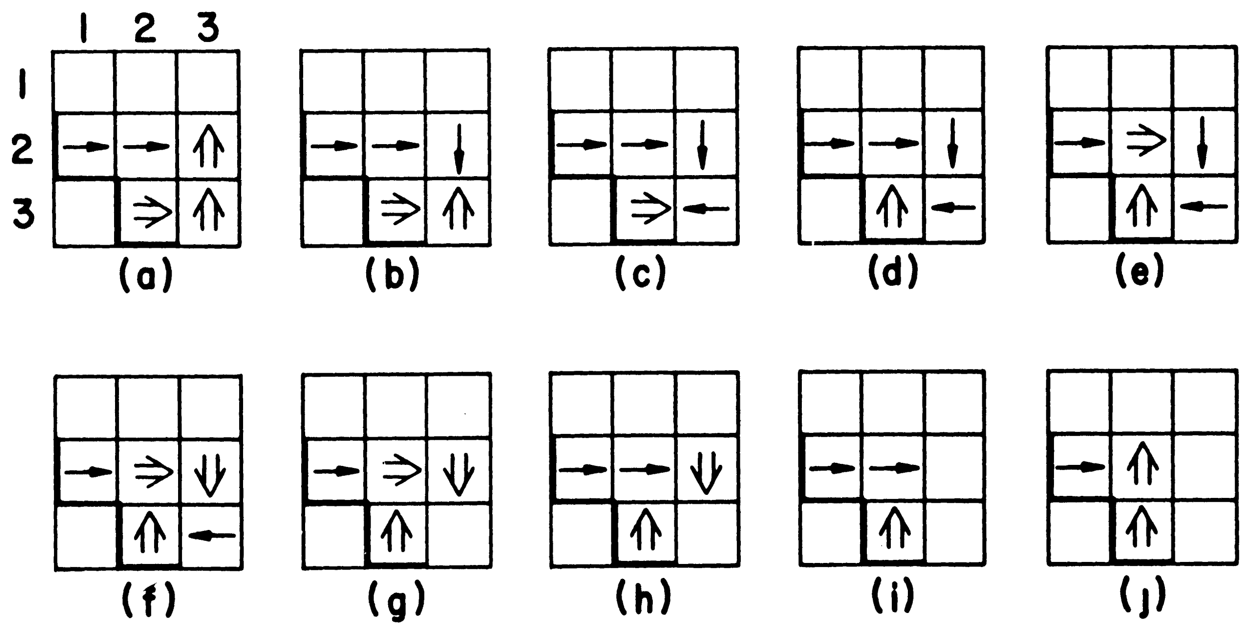


Figure 5.3.2

Pulse Sequences for Vertical Advance

Step	Special	Ordinary	Cell	Old State	New State
b	1101	0000	(1, 2)	U	⇐
c	1110	0000	(1, 1)	U	⇓
d	110001000	000000000	(2, 1)	→	↑
e	000000	110000	(1, 1)	⇓	→
f	00000	11100	(1, 2)	⇐	⇑

Table 5.3.3

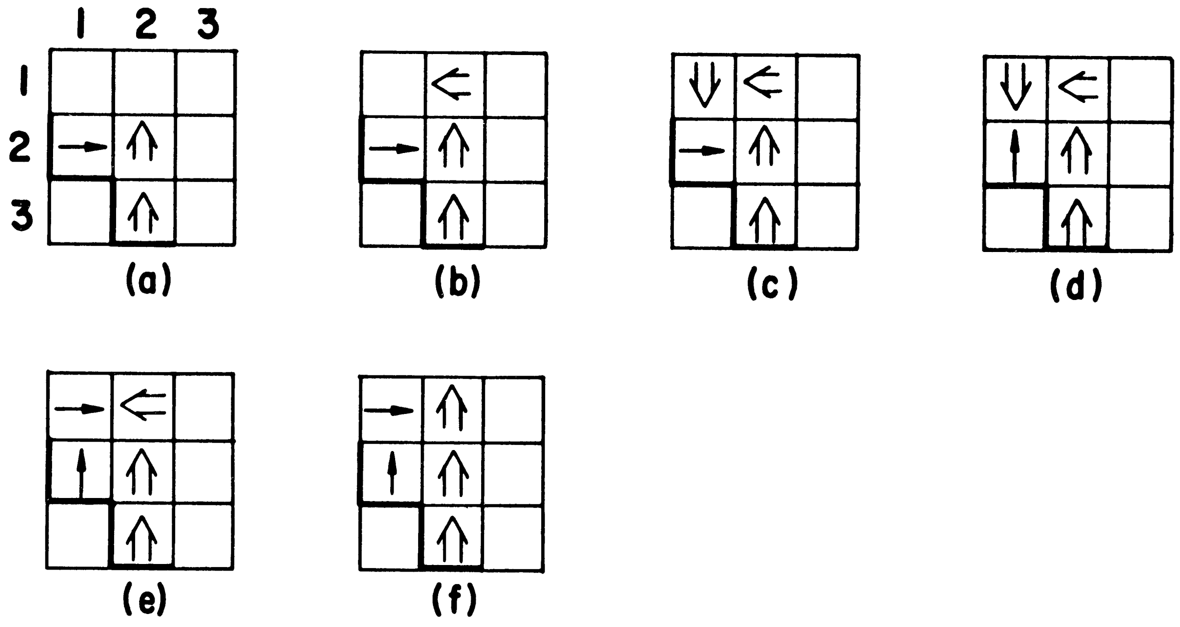


Figure 5.3.3

Pulse Sequences for Vertical Retreat

Step	Special	Ordinary	Cell	Old State	New State
b	00000	11010	(1, 2)	⇑	↓
c	000000000	111010000	(2, 2)	⇑	⇐
d	11000000	00000000	(2, 1)	↑	→
e	0000000	1110000	(2, 2)	⇐	⇑
f	11101	00000	(1, 2)	↓	⇐
g	1	0	(1, 1)	→	U
h	110001	00000	(2, 2)	⇑	↑
i	10	00	(1, 2)	⇐	U
j	00000	11100	(2, 2)	↑	⇑

Table 5.3.4

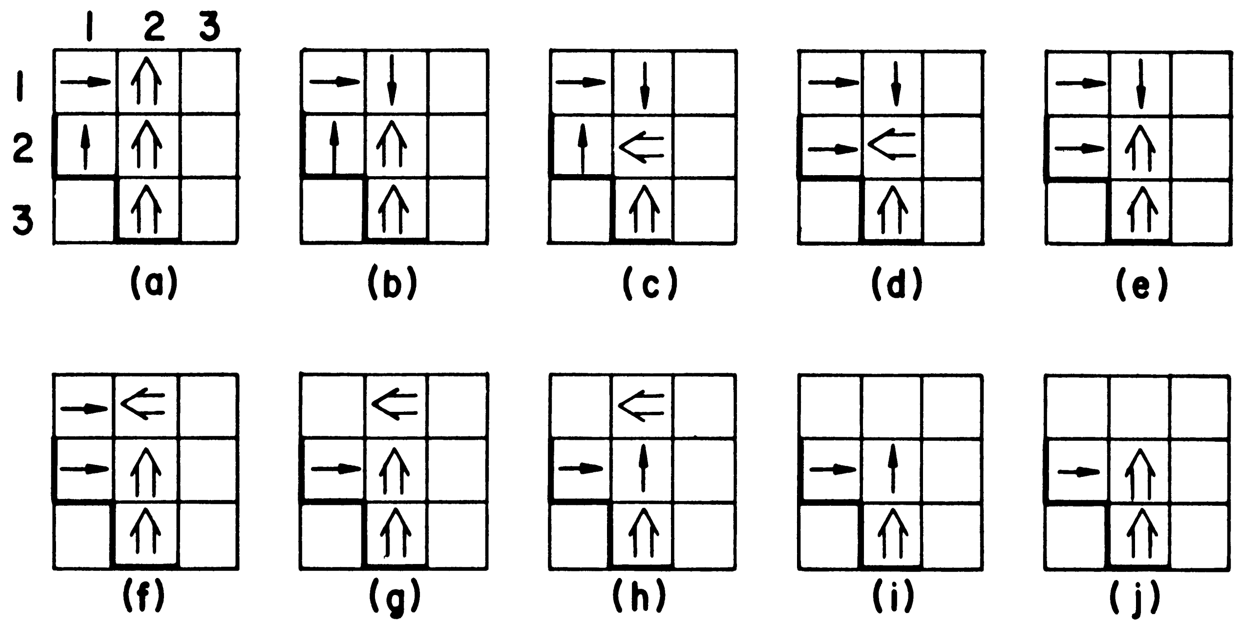


Figure 5.3.4

shows the end of the arm as a result of the step.

In the table the cells are referred to by the notation (m, n) designating the mth row and the nth column as numbered in Figure 5.3.1a.

The constructing head is initially in the configuration of 5.3.1a. The inputs to the constructing head are from the left if the arm is in the horizontal position and from below if the arm is in the vertical position. Regardless of the position, however, the ordinary cell (2, 1) and the special cell (3, 2) are the two cells that receive the input.

The result of step b, which is shown in Figure 5.3.1b, is to change the unexcited special transmission state of (2, 2) to an ordinary transmission state pointing down. This is accomplished by the sequence 11010 from the ordinary transmission state (2, 1) as dictated by the transition rule, part (12). The initial 1 converts $\Uparrow$ to U. The second 1 causes the transition from U to S_θ and the sequence 010 then results in the ordinary transition state pointing down. Note that the corresponding special sequence is 00000, the special channel being, in effect, inoperative while the modification is taking place through the ordinary channel. The result of step c (Figure 5.3.1c) is to change (3, 2) from $\uparrow$ to $\Rightarrow$. The 11011 as input to (1, 2) yields this result. The next step (to modify cell (3, 3)) requires input through the special channel and cell (3, 2). Thus we must effect a delay to insure that (3, 2) has been converted to a special transmission state before its input is received. The three units of delay resulting from the terminal 000 of the input sequence 11011000 to the ordinary channel are sufficient for this purpose. Continuing, in step d the sequence 1100 through the special channel changes U to $\Uparrow$ in (3, 3), and in step e, 1101 converts (2, 3) from U to $\Leftarrow$. Finally 110000 through the special channel results in $\longrightarrow$ in cell (2, 2). Again, we need to switch to operation through the other channel and this requires the additional three units of delay to complete the transition of (2, 2) to the ordinary transmission state pointing to the right. In the last step, with the sequence 1100 entering through the ordinary channel, cell (2, 3) is converted to $\Uparrow$. The construction point of the arm has thus moved one square to the right.

These comments apply analogously for Tables 5. 3. 2-5. 3. 4 and the other positioning processes. The actual sequences are exhibited in full detail in Table 5. 3. 5. In this table the notation $\Pi^{S}_{\underline{r}}$ denotes the pulse sequence for the special channel (S) for the operation $\underline{r}$. Similarly $\Pi^{0}_{\underline{r}}$ is the horizontal advance sequence for the ordinary channel. These sequences are formed by concatenating the sequences indicated in Tables 5. 3. 1 through 5. 3. 4.

The constructing arm and associated positioning operations will be used in the design of the tape unit. And it is convenient there to have the constructing point fed by a special transmission state. This

Pulse Sequences for Positioning the Constructing Arm

Horizontal Advance $\underline{r}$

$\pi_{\underline{r}}^{0}$ = 110101101100000000000000000000110000000000000000

$\pi_{\underline{r}}^{S}$ = 000000000000011001101110000000000000000000000000

Horizontal Retreat $\underline{\ell}$

$\pi_{\underline{\ell}}^{0}$ = 110101100111100000000000000000001100001000000000

$\pi_{\underline{\ell}}^{S}$ = 000000000000000000001101111110100000000011100000

Vertical Advance $\underline{u}$

$\pi_{\underline{u}}^{0}$ = 000000000000000001100001110000000000000000000000

$\pi_{\underline{u}}^{S}$ = 110111101100010000000000000000000000000000000000

Vertical Retreat $\underline{d}$

$\pi_{\underline{d}}^{0}$ = 110101110100000000000011100000000000000000011100

$\pi_{\underline{d}}^{S}$ = 000000000000001100000000000001110111100011000000

Table 5.3.5

causes only one inconvenience in the design of the constructing unit: specifying the sequences required to "start" a constructed automaton. In this process it will be assumed that the constructing arm is positioned under an unexcited ordinary transmission or confluent state and that an adjacent excited ordinary transmission state will initiate the action of the constructed automaton. Thus the $\underline{s}$ (stimulate) input to the constructing unit must cause sequences to be transmitted through the ordinary channel of the arm to convert the constructing point cell from $\Uparrow$ to $\uparrow$, then to activate the ordinary transmission state. Finally the constructing head must be restored by converting $\uparrow$ to a special transmission state pointing up. The specific sequences are again straightforward and are indicated in Table 5.3.6.

Sequences for the $\underline{s}$ Input to the Constructing Unit

Step	Special	Ordinary	Cell	Old State	New State
1	000000	110001	(2, 2)	$\Uparrow$	$\uparrow$
2	0	1	(2, 2)	$\uparrow$	$\uparrow_1$
3	11100	00000	(2, 2)	$\uparrow$	$\Uparrow$

$$\pi^0_{\underline{s}} = 110001100000$$

$$\pi^S_{\underline{s}} = 000000011100$$

Table 5.3.6

5.4 The Constructing Unit

With the method of construction specified, the design of the constructing unit is simple. The complete unit is drawn in Figure 5.4.1. We see there that the constructing unit has the four positioning inputs and 10 constructing inputs. Each constructing input stimulates the correct pulser, the corresponding sequence (see Table 5.2.2) being transmitted directly to the special channel of the constructing arm. We must be sure that the positioning inputs stimulate the correct pulsers. This is accomplished by assigning a distinct code to each input:

$\underline{s}$	<—>	100001
$\underline{r}$	<—>	100010
$\underline{\ell}$	<—>	100100
$\underline{u}$	<—>	101000
$\underline{d}$	<—>	110000.

These codes, being incomparable with respect to bit-wise implication, can be separated using decoders. Thus, each positioning input stimulates the pulser of its code and the code sequence is accepted by exactly the pair of decoders for the positioning operation. Because of the location of the components of the constructing unit, the input reaches the special sequence pulser at exactly the same time that the input reaches the ordinary sequence pulser. Likewise, the inputs to the constructing arm are simultaneous.

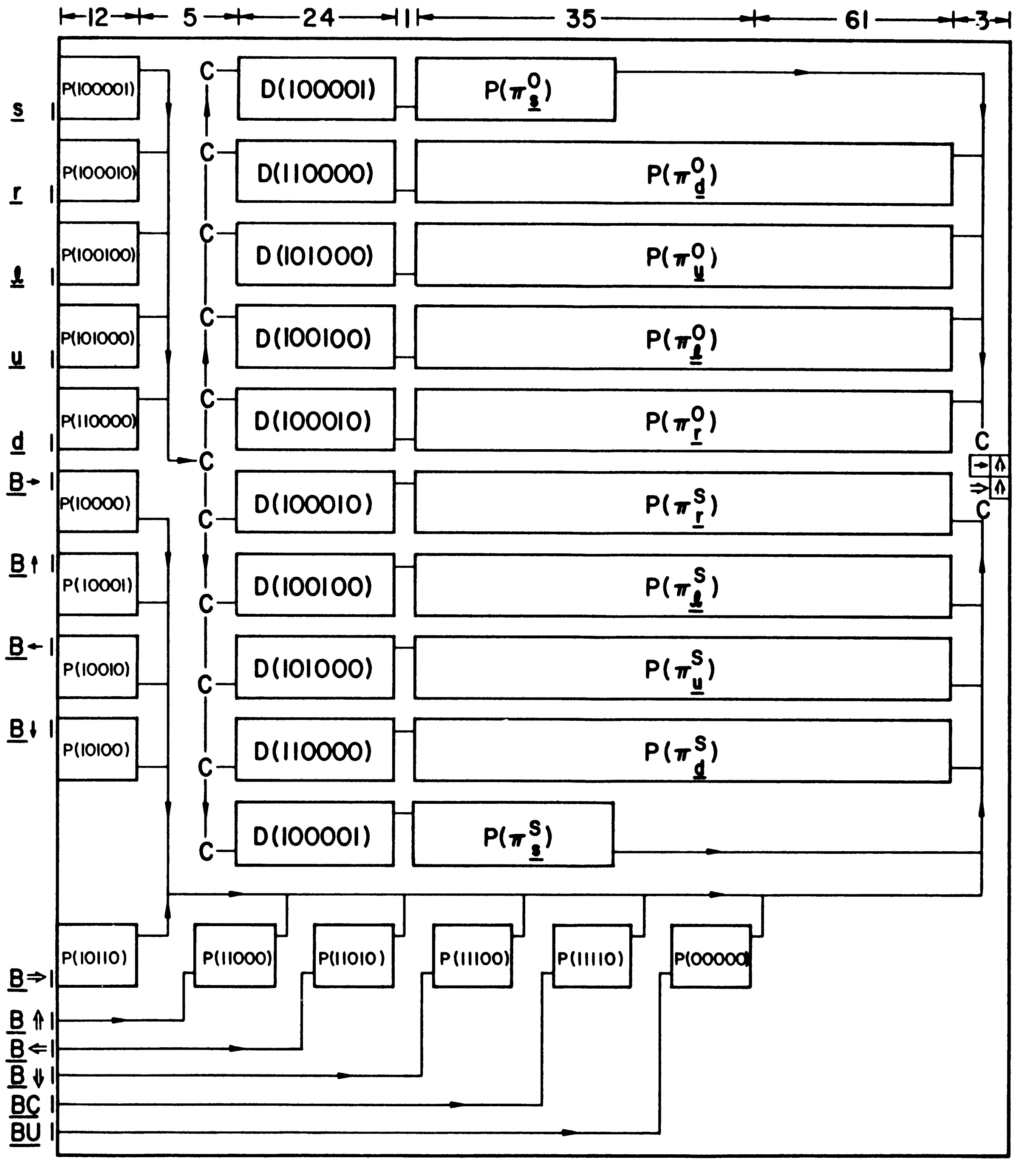


Figure 5.4.1 The Constructing Unit

The dimensions of the constructing unit are:

Length 139

Height 33

The timing in the constructing unit is analyzed in the next section.

5.5 Timing in the Constructing Unit

The exact delay from the time of an input stimulus to the completion of the designated operation can be computed for each input from Figure 5.4.1 and the configuration of the arm. For a given position of the arm, the total number of advances required to achieve the position will be called the length of the arm. If n is the length, then the delay for a positioning input is on the order of n+260 and that for a construction input is approximately n+160.

We are primarily interested in the required spacing between inputs. To simplify the calculation, we will be certain of correct functioning if a second input never arrives until the first working sequence appears at the output of its pulser. Calculating for the horizontal advance input this yields a required delay of 220. Hence, if at least 220 time steps separate each input to the constructing unit, we can be sure that there will be no malfunctioning.

5.6 Programming for Construction

We now have the idea of a constructing area and constructing arm in the cellular system and with this we will show that it is possible to construct all completely passive configurations. The Turing machine model of Section 4 has the decided advantage that there was assumed to be no interaction between the constructing area and the constructing arm. Since our constructing arm is almost always part of the constructing area, it is necessary to do all constructing from the top down.(10)

Any completely passive configuration can be described by a rectangular array $|m_{ij}]$ for $i=1,\ldots,n$ and $j=1,\ldots,k$ where n is the height of and k is the length. Each m_{ij} is one of the 10 unexcited states. Let m_{11} be the upper left-hand corner cell. To completely determine the construction of the configuration we need only specify the array $|m_{ij}|$ and the location of m_{11} relative to the origin of the constructing arm. To specify this point, let h be the distance to the right and v be the vertical distance required to position the constructing arm with the point of construction under the location of m_{11}. Because of the restriction on the area of construction, we must have $h \geq 2$ and $v \geq k-1$.(11)

Therefore, a complete description for construction by our constructing device is given by:

$$(h, v, k, n\ |m_{ij}|).$$

What program can we use to accomplish this construction? This task is <u>extremely simple</u>. The entire program (in the constructing language of

(10) It is fairly clear that anything that could be effectively constructed according to the model of Section 4 could also be constructed "from the top down".

(11) To fix values we can take $h=2$ and $v=k-1$.

Section 4) including the final retraction of the arm is as follows (the notation n[r] means n successive r instructions):

$$\begin{array}{l}
h[r],\ v[u] \\
(k-1)[r],\ Bm_{1k}, \ell, Bm_{1k-1}, \ell, \ldots, Bm_{12}, \ell, Bm_{11}, d, \\
(k-1)[r],\ Bm_{2k}, \ell, Bm_{2k-1}, \ell, \ldots, Bm_{22}, \ell, Bm_{21}, d, \\
\qquad \ldots \\
(k-1)[r],\ Bm_{nk}, \ell, Bm_{nk-1}, \ell, \ldots, Bm_{n2}, \ell, Bm_{n1}, \\
(v-n+1)[d],\ h[\ell]
\end{array}$$

The property of self-reproduction that we may require of an automaton demands considerably more structure than the restricted goal of universal construction. If we consider only universality of construction, then the constructing arm with an appropriately coded input can construct any completely passive configuration.

In particular, we are now in a position to verify Proposition IV, the existence of a universal constructor for completely passive configurations. The universal constructor is drawn in Figure 5.6.1. For any completely passive configuration, the tape for the constructor consists of a pair of pulsers together with some connecting ordinary transmission states (this "tape" is a completely passive configuration). For a given configuration s, let $|m_{ij}]$ be an array describing s (a rectangular array including s) and write the program, $f_1, f_2, \ldots, f_p$, for constructing this array as

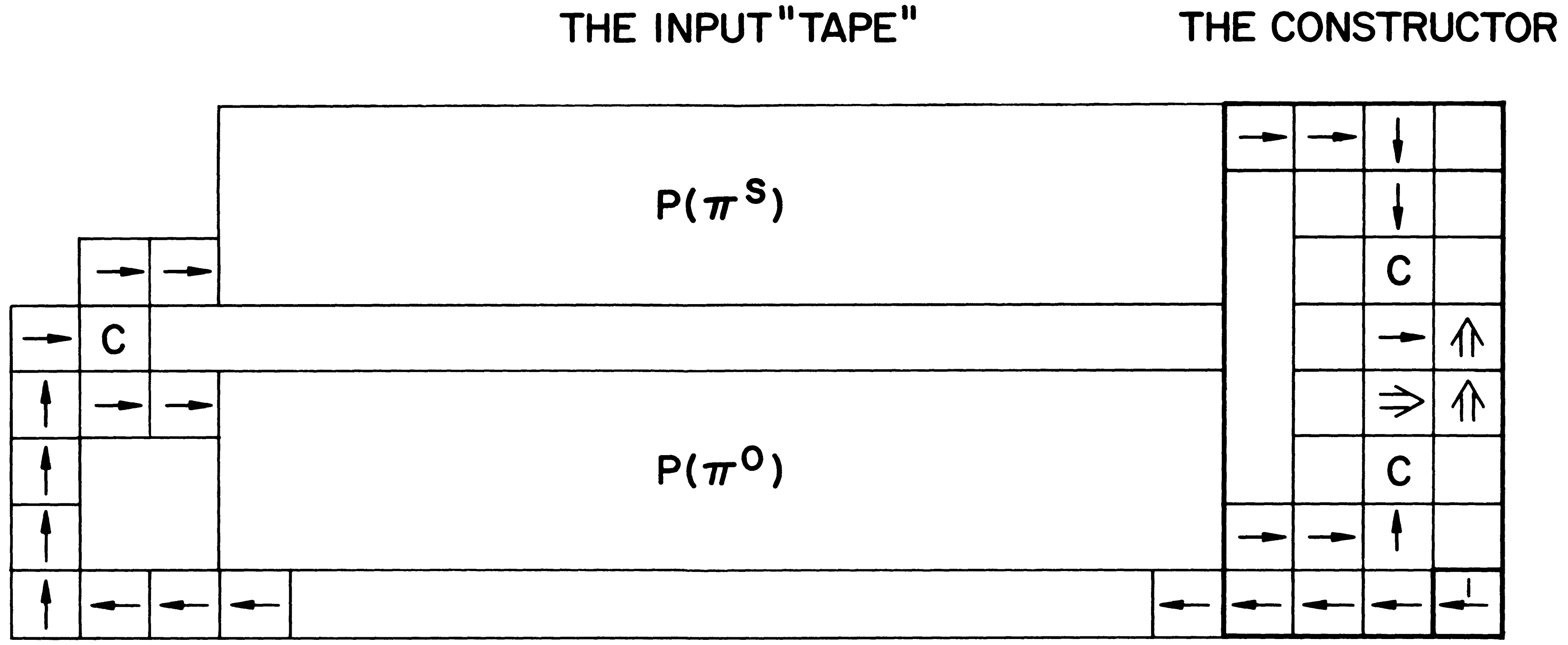


Figure 5.6.1 A Universal Constructor

indicated above. For each instruction f_i in this program obtain the corresponding sequences $\Pi^S_{f_i}$ and $\Pi^O_{f_i}$ required for the operations of the constructing arm (from Tables 5.2.2 and 5.3.5). Then form the two extremely long sequences:

$$\Pi^S = \Pi^S_{f_1} \frown 0^4 \frown \Pi^S_{f_2} \frown 0^4 \frown \dots \Pi^S_{f_{m-1}} \frown 0^4 \frown \Pi^S_{f_m}$$

and

$$\Pi^O = \Pi^O_{f_1} \frown 0^4 \frown \Pi^O_{f_2} \frown 0^4 \dots \Pi^O_{f_{m-1}} \frown 0^4 \frown \Pi^O_{f_m} .$$

where 0^4 is a sequence of four zeros and $\frown$ denotes concatenation. The insertion of the four zeros insures that the construction from the previous sequence is completed before the next one arrives at the constructing head. Finally, using the method of Section 3.3 code the input tape of the universal constructor by forming the pulsers $P(\Pi^S)$ and $P(\Pi^O)$ together with the connecting ordinary transmission states as shown in Figure 5.6.1.

The right-hand corner cell of the universal constructor is an excited ordinary transmission state (this alone could be taken to be the universal constructor with the rest being part of the tape). The excited state serves to stimulate the two pulsers which in turn perform the construction operations as described by the program. In this way any completely passive configuration can be constructed in the quadrant to the right and above the point of origin on the constructing arm. By moving the constructor around we can, of course, construct any completely passive configuration (Proposition III).

The method employed here will not work if self-reproduction is desired. Applying the algorithm above, the tape for an m x n array has dimensions on the order of 200mn x 7, an area 1400 times as great as that being constructed. We can very easily make a copy of the universal constructor of Figure 5.6.1 but we can not copy the tape with which the universal constructor constructs itself -- actually we can make a copy of that tape but this requires an even larger tape and so on. The verification of the existence of a self-reproducing configuration must be left to Section 8.

5.7 Extensions of the Constructing Unit Design

The assumption that the cells surrounding the constructing arm are all quiescent is not necessary. With a more complicated constructing arm (Figure 5.7.1) it is possible to obtain proper movements and constructions even though the state of the cell to be modified is not known.

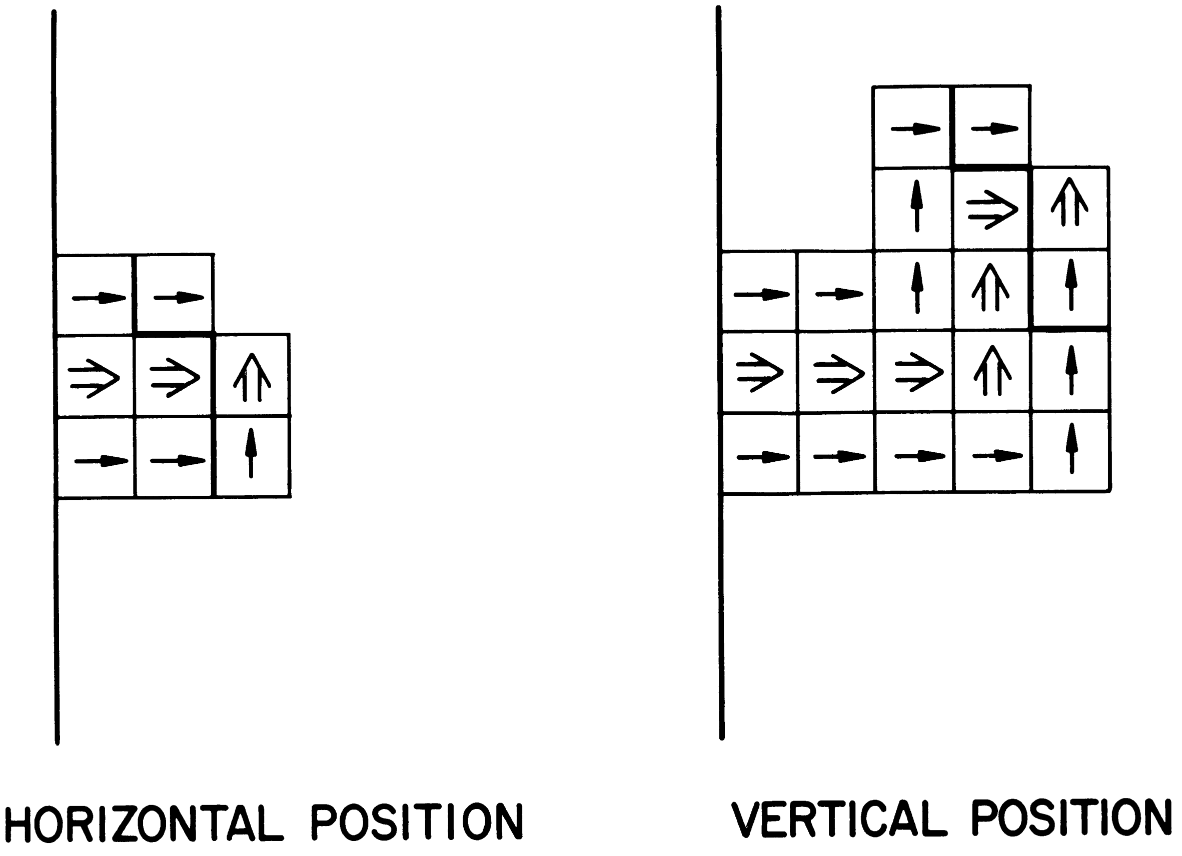


Figure 5.7.1 Alternative Constructing Arm

The principle applied is based on the fact that construction (or destruction) takes precedence, in the transition rule, over transmission. Thus if both a special and an ordinary transmission state are pointing at a given cell, a simultaneous stimulous to that cell from both ordinary and special transmission states will result in modification to the state U. (See Section 6.4).

There is, of course, no way to avoid the possibility of bombardment of the constructing arm from active neighboring cells. Such a situation would completely disrupt the operation of the constructor in most cases. This is, in fact, a feature of the von Neumann model in the sense that the only immutable situation of a given cell is one for which there is constant input from two adjacent cells, one in special transmission and the other in ordinary transmission state. The geometry does not allow a boundary of such cells and thus any automaton embedded in the cellular space is subject to possible destruction.

In the design presented above, the construction was restricted to the quadrant above and to the right of the constructing unit. This also is an unnecessary restriction. It is possible to add commands to the repertory of the constructing unit to specify reversal of direction of the constructing arm. As with the commands for the unit described above, these additional commands could be given at the wrong times causing erroneous operation.

6. THE TAPE UNIT

6.1 Introduction

In this section the arbitrarily extendable tape and the tape control will be designed. Together they constitute the tape unit for a von Neumann machine. It is in the design of the tape unit that this presentation differs most strikingly from von Neumann's.

The details of von Neumann's construction will not be reviewed here but the essential difference may be described as follows. We can consider a read-write head positioned over the n^{th} square of the tape. In the von Neumann construction, as well as the one presented here, the process of reading the n^{th} cell is independent of n. A certain sequence of pulses is sent to the read-write head which in turn recognizes the contents of this cell. An appropriate sequence specifying the contents is returned to the tape control unit. The delay involved (from control to read-write head and back to control) is a function of n but the process, that is, the sequence transmitted to the read-write head, is always the same, regardless of the position of the cell being interrogated.

In the von Neumann construction, however, the process resulting in the modification of the tape contents and the motion of the read-write head are functions of the position of the read-write head, i.e., functions of n. For example, it is necessary during the lengthening (motion to the right) process to transmit n sequences of a particular type to the read-write head. Hence it is necessary for the control to "know" n.

To accomplish this von Neumann introduces a timing loop which is lengthened or shortened as the read-write head of the tape is moved to the right or left. This method of operation is the primary cause of the complexity to be found in von Neumann's design. It gives rise to timing problems within the control unit which are extremely complicated.

In the construction presented here the modification, lengthening and shortening of the tape are independent of the position of the read-write head. As in the case of reading, these operations are performed by transmitting specific sequences of pulses to the read-write head. This method results in tremendous simplification over von Neumann's design.

6.2 External Connections to the Tape Unit

It is necessary to specify what information is to be received and transmitted by the tape unit. There will be two possible tape symbols corresponding to 0 and 1. Thus there will be two outputs, X0 and X1, one of which is activated after a read input has been executed and they designate that the current scanned square is 0 and 1 respectively. The five inputs, +, -, e, m, R correspond to the five active instructions for Turing machines (see Section 4) with the exception that the decision of the transfer instruction must be handled by the supervisory unit and the R input is used to initiate the read operation on the tape.

6.3 The Tape

The tape is drawn in Figure 6.3.1. It consists of an array of six rows of cells. The bottom two, labeled O_B and S_B, are precisely the same as the constructing arm (extended horizontally) as was described in the last section. Similarly, the top two rows form a "dual-inverted" constructing arm. The row labeled R is the return over which the information concerning the contents of the current tape square is transmitted. Finally, the row labeled X is the actual storage or tape area. The cells outlined in Figure 6.3.1 can be considered to be the reading head. It should be clear that motion of the reading head can easily be accomplished by using the horizontal positioning operations of the construction arms.

Each storage cell, X_n, will assume one of two possible states. The unexcited ordinary transmission state pointing down $(\downarrow)$ will represent 1 on the tape and the quiescent state (U) will represent 0.

Because of the duality of the two types of constructing arms we will not further consider any detailed manipulation of the dual-invented arm. It is assumed that one could carry out the computations of Section 6 to obtain the required sequences for such manipulations.

6.4 Tape Operations (+, -, e, m)

As in the design of the constructing unit it is necessary here to specify what sequences are to be transmitted down the channels

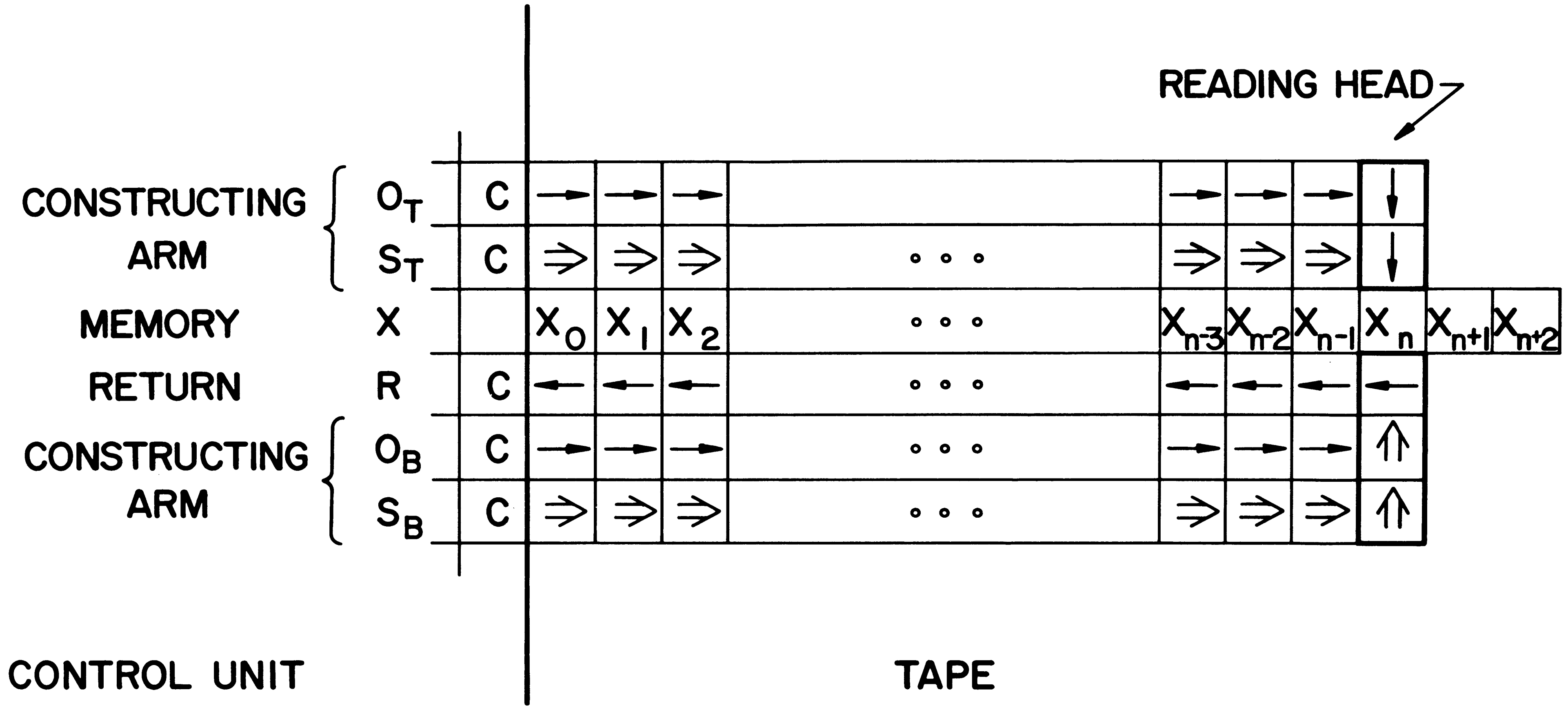


Figure 6.3.1 The tape for a von Neumann Machine

S_T, O_T, S_B, O_B of the tape in order to perform the required operations. Then (Section 6.6) the design of the tape control unit will amount to deciding on a geometrical arrangement for the required pulsers insuring that each sequence arrives at the corresponding channel at the right time. The last step in the control design, which consists of connecting the inputs to the correct pulsers, is accomplished with a coded channel as was the case with the constructing unit.

With the exception of modifying the R channel, motion of the read-write head (+, -) is accomplished by the corresponding operations on the constructing arm. But modification of the R channel consists simply of converting U to $\leftarrow$ for motion to the right and converting $\leftarrow$ to U for motion to the left. These conversions, conveniently, are to be carried out at the constructing point of the lower constructing arm. Thus, taking + for an example, the inputs to the top constructing arm consist of the sequences required for moving the constructing point one square to the right ($\underline{r}$). The same is true for the bottom constructing arm except that the special sequence is followed by 1001 to build a $\leftarrow$ at the point of construction.

The required sequences are listed in Table 6.4.1. Here $\bar{\pi}^{S}_{\underline{r}}$ denotes the sequence for the special channel for the $\underline{r}$ input for the dual-invented arm.

Input Sequences for Positioning the Reading Head

	${}_{\pi}{}^{0}{}_{T}$	${}_{\pi}{}^{S}{}_{T}$	${}_{\pi}{}^{0}{}_{B}$	${}_{\pi}{}^{S}{}_{B}$
+	$\bar{\pi}^{0}_{\underline{r}}$	$\bar{\pi}^{S}_{\underline{r}}$	$\pi^{0}_{\underline{r}}\ 0^{4}$	$\pi^{S}_{\underline{r}}\ 1001$
-	$\bar{\pi}^{0}_{\underline{\ell}}$	$\bar{\pi}^{S}_{\underline{\ell}}$	$0^{3}\ \pi^{0}_{\underline{\ell}}$	$100\ \pi^{S}_{\underline{\ell}}$

Table 6.4.1

Modifying the contents of the n th cell of the tape would be easy if we knew the state of X_n before the process began. An organization is possible that would, in effect, accomplish this. The tape modification input (e̲ or m̲) could first initiate a read operation (see Section 6.5) and on the basis of the information thus obtained the tape could be modified in a relatively straightforward manner.

This proposed method has a serious draw-back. The amount of time required to complete the tape modification, in particular, the delay necessary before another tape unit input could safely be given- would be a function of the length of the tape.

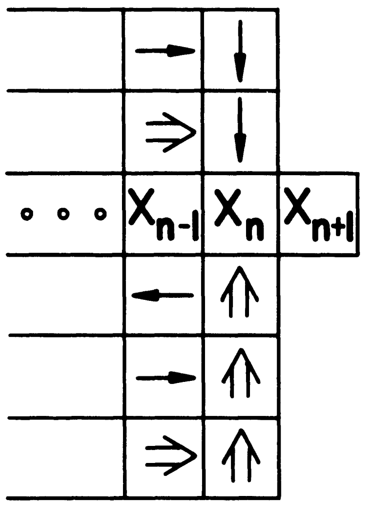
Alternatively, a known configuration can be produced at the end of tape starting from some unknown configuration. The method is first to obtain the arrangement at the end of the tape as pictured in Figure 6 4.1a. This is accomplished by step a of Table 6.4.2.

Pulse Sequences for Tape Modification

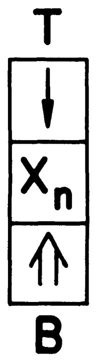
Step	0_T	0_B	S_B	Cell	Old State	New State
a	00000	00000	11100	R_n	$\leftarrow$	$\Uparrow$
b	01100001	00000000	11000010	X_n	?	U
c(<u>m</u>)		0000	1010	X_n	U	$\downarrow$
c(<u>e</u>)		0000	0000	X_n	U	U
d		110001	000000	O_{B^n}	$\Uparrow$	$\downarrow$
e		11001	00000	R_n	$\Uparrow$	$\leftarrow$
f		00000	11100	O_{B^n}	$\downarrow$	$\Uparrow$

Table 6.4.2

Because of precidence of "distruction" in the transitive rule there is a sequence π_U such that when π_U is fed simultaneously into T and B of Figure 6.4.1b, the resulting state of X_n will be the quiescent state regardless of the way X_n started. This sequence is detailed in Figure 6.4.1 c and the operation is step b of table 6.4.2. Thus both



(a)



(b)

INPUT AT BOTH T and B	1	1	0	0	0	0	1	
$X_n \neq U$	U	S_e	S_0	S_{00}	S_{000}	→	U	
$X_n = U$	S_e	S_1	S_{10}	⇑	⇑	⇑	U	

(c)

Figure 6.4.1 Process for Tape Modification

$\underline{e}$ and $\underline{m}$ inputs will cause the conversion X_n to U. For the $\underline{m}$ input then, the sequence 1010 along O_T will convert X_n to $\downarrow$. Finally the reading head must be restored to the original configuration as accomplished in steps d-f of Table 6.4.2. Again, as in Section 5.3, the full sequences $\pi_{\underline{e}}^{O_T}$, $\pi_{\underline{e}}^{O_B}$, etc., can be obtained from Table 6.4.2 by concatenating the sequences in the appropriate columns.

6.5 The Read Operation

The basic idea of the read operation as it appeared in von Neumann's manuscript is a very neat and simple one. If the pulse sequence 10101 is transmitted through the top ordinary channel (O_T), the return through the R channel to the control unit will be 1 or 10101 according to whether X_n was U or $\downarrow$ respectively. In the tape control unit then, it is simply necessary to distinguish between 10101 and 00100 (we will use recognizers) and produce a pulse at the corresponding output. This readout is, however, distructive and we are, in our programming motivated control, assuming that the tape contents are not destroyed by reading. Thus this basic scheme must be modified to permit restoring the correct value to X_n if it is altered.

Note that the situation is similar to that in tape modification (e, m) because the process of reading and then acting on the information obtained takes an amount of time which is a function of the length of the tape. Here, however, it causes no trouble since the supervisory unit

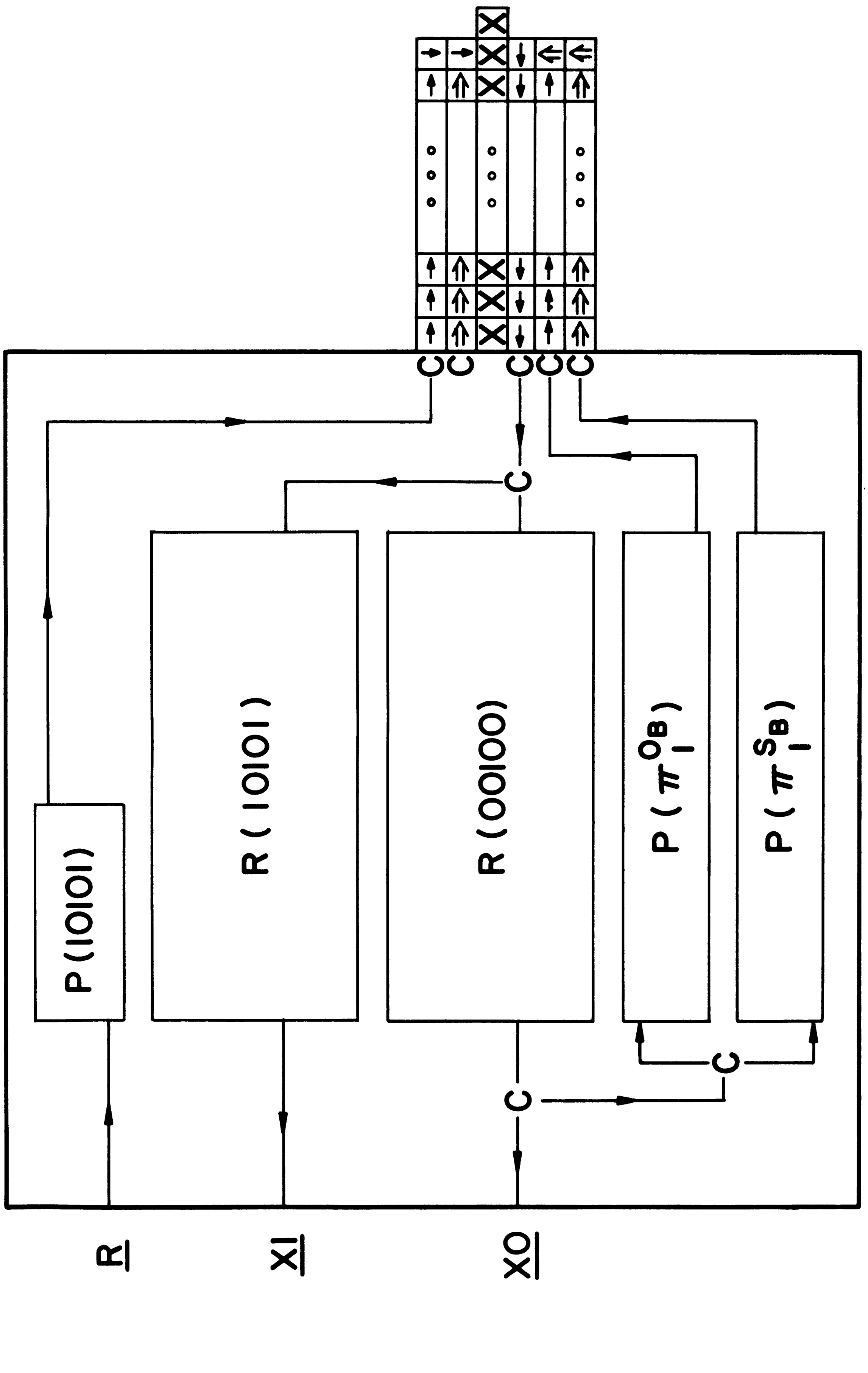


Figure 6.5.1 Control Unit for Reading Only

must receive the output of the tape unit ($\underline{X0}$ or $\underline{X1}$) before further inputs are furnished. Thus the amount of delay that is variable with the length of the tape is taken up before the tape unit output is produced.

For illustrating purposes a control unit for reading only is drawn in Figure 6.5.1. Here $\pi_1^{O_B}$ and $\pi_1^{S_B}$ are obtained from Table 6.4.2 by concatenating the appropriate sequences in the columns labeled O_B and S_B respectively. Step b is omitted because the contents of X_n is known to be an ordinary transmission state pointing down. In place of $c(\underline{e})$ the sequence 1000 is inserted for S_B. This has the effect of "erasing" X_n on which a 1 has been "written".

6.6 The Tape Control Unit

There are only two minor problems remaining in order to complete the construction of the tape unit.

(a) The proper input ($\underline{e}, \underline{m}, \underline{+}, \underline{-}, \underline{R}$) must stimulate the required pulsers and no others.

(b) It is necessary to insure that the initial pulses of the special and ordinary sequences for both lower and upper constructing arms arrive at the appropriate inputs to the tape at the same instant.

The first problem is solved by a simple coding of the inputs. The coding we shall use is as follows:

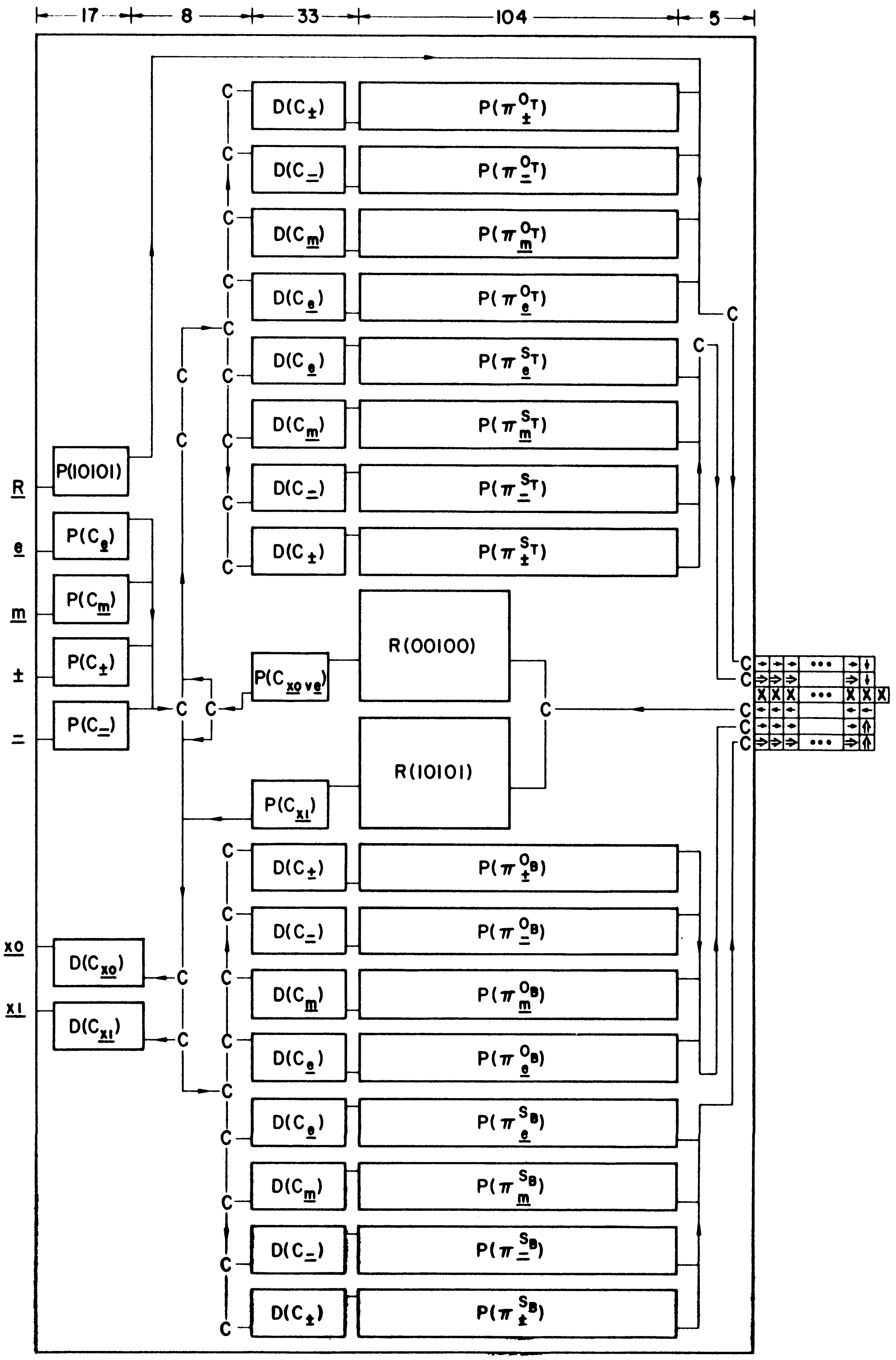


Figure 6.6.1 The Tape Unit

$\underline{e}$	<–>	$C_{\underline{e}} = 10000011$
$\underline{m}$	<–>	$C_{\underline{m}} = 10000101$
$\underline{+}$	<–>	$C_{\underline{+}} = 10001001$
$\underline{-}$	<–>	$C_{\underline{-}} = 10010001$
$\underline{X0}$	<–>	$C_{\underline{X0}} = 10100001$
$\underline{X1}$	<–>	$C_{\underline{X1}} = 11000001$

The complete tape unit is drawn in Figure 6.6.1. By appropriate physical placement of the pulsers, problem (b) above is easily handled. For example an input at $\underline{e}$ results in the sequence $C_{\underline{e}} = 10000011$ being transmitted into the coded channel. This sequence is multiplexed to four copies yielding inputs to all decoders but outputs are produced only by the four decoders, $D(C_{\underline{e}})$, and these outputs are simultaneous. Thus the four sequences $\pi_e^{O_T}$, $\pi_e^{S_T}$, $\pi_e^{O_B}$, $\pi_e^{S_B}$, are formed for input to the tape. By the positioning and the delays introduced in the top transmission line, these sequences reach the constructing heads of the two constructing arms at the same time.

The $\underline{R}$ input pulses 10101 to be sent down the top ordinary channel. According to whether the tape square under scan is 0 or 1, the return is 00100 or 10101 respectively. If 10101 is returned it is recognized by R(10101) and the code $C_{\underline{X1}}$ is fed into the coded channel and accepted only by the decoder D(11000001) to produce an output at $\underline{X1}$. If 00100 is returned, however, X_n was U and now is $\downarrow$. Thus the code 10100011 is transmitted into the channel. This sequence being bitwise implied by both $C_{\underline{X0}}$ and $C_{\underline{e}}$ (and only these) is recognized to give the output $\underline{X0}$ but also initiates the erase operation just as if the input $\underline{e}$ has been given. In this way X_n is returned to U.

The size of the tape control unit is,

height 83

length 107

The timing for the tape unit is discussed in the next section.

6.7 Tape Unit Timing

Again we could give a detailed analysis of timing for tape unit operation. The important question is the delay required between successive inputs to any of $\underline{+}$, $\underline{-}$, $\underline{e}$, or $\underline{m}$ in order to isure proper operation.

As in the case of the constructing unit we choose a large margin of safety, calculating the maximum time required for an input to the tape unit to cause the corresponding working sequences to appear as inputs to the tape. Thus taking $\underline{+}$ for example, an input at + results in $C_{\underline{+}}$ at the decoders at t + 57. The output of the decoders at t + 108 produces the corresponding pulse sequence output at t + 216. Thus the input to the tape occurs at t + 257.

In the design of the supervisory unit a delay greater than 300 separates inputs to the constructing and the tape units. Thus we can be sure that there will be no overlapping sequences to cause malfunction in these units.

7. THE SCHEME FOR VON NEUMANN MACHINES

7.1 Introduction

One of the best things that can be said for the designs of the tape and constructing units given in the preceding sections is that on the basis of these designs the remainder of our presentation is quite direct.

The organization of a general von Neumann machine is shown in Figure 7.1.1. Here the tape, constructing and supervisory units are connected via a coded channel. We begin in Section 7.2 to describe the coded channel and in Section 7.3 the synthesis algorithm is given for converting an arbitrary program in the language of Section 4 into the design of a supervisory unit which will control the tape and constructing units as dictated by the program.

7.2 The Coded Channel

The coded channel can be viewed as consisting of three parts. Pulsers are used for coding information to be transmitted into the channel. Recognizers decode this information. The channel itself consists of a single path of ordinary transmission states interrupted with confluent states for the outputs from the channel. Thus in Figure 7.1.1, pulse sequences travel in a clock-wise direction through the channel, but due to the break in the path at the tape unit, they cannot make a complete cycle. The decoding areas indicated in the diagram are sets of recognizers that receive signals from the channel, and the coding areas are sets of pulsers which transmit sequences into the channel.

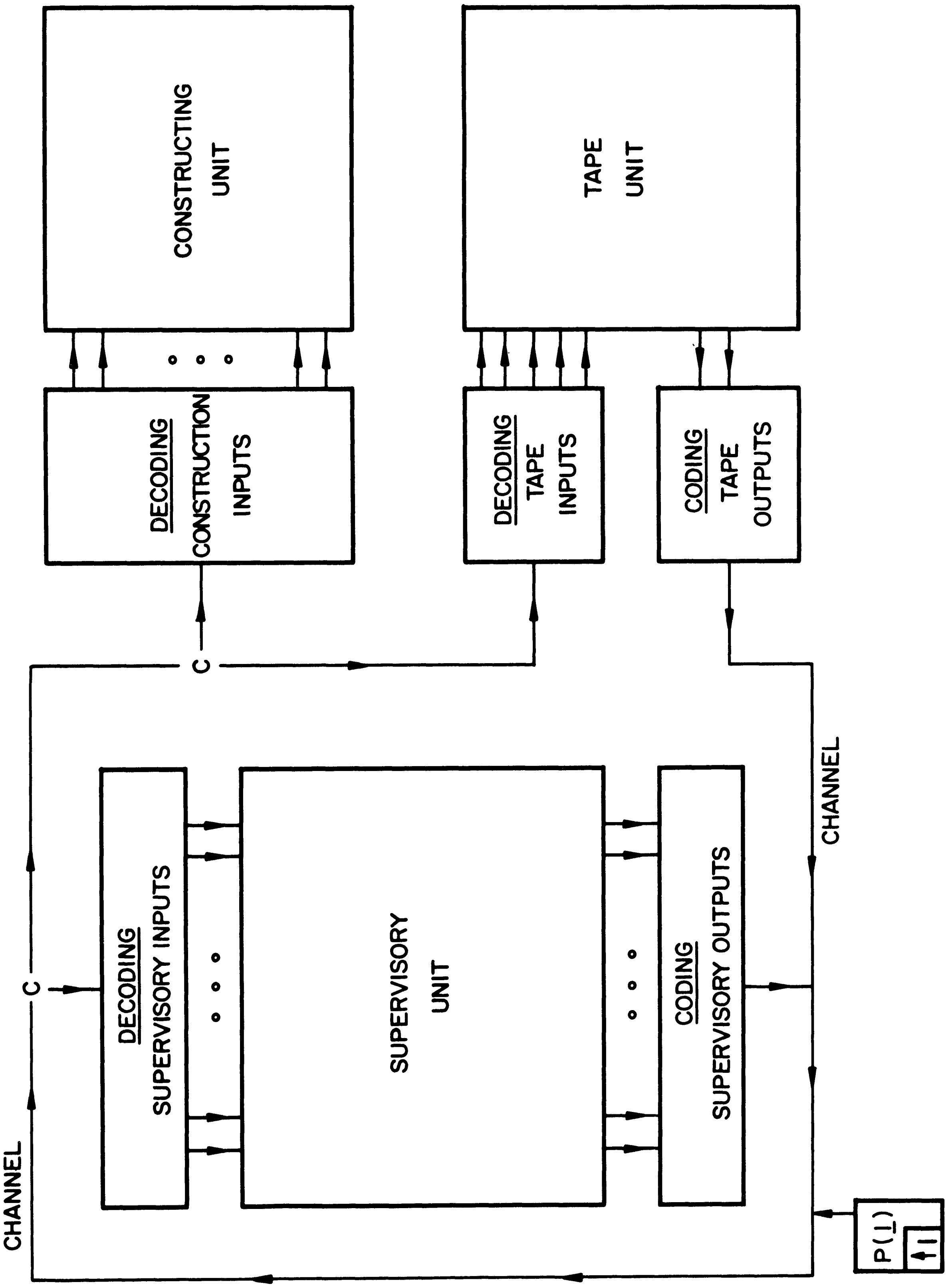


Figure 7.1.1 Schematic Diagram of a von Neumann Machine

In a general von Neumann machine there are four basic types of information transmission:

(1)	Supervisory unit to constructing unit
(2)	Supervisory unit to tape unit
(3)	Tape unit to supervisory unit
(4)	Supervisory unit to supervisory unit

The corresponding information that must be transmitted consists of:

(1) The construction inputs, $\underline{r}, \underline{d}, \underline{BU}$, etc.

(2) The tape inputs, $\underline{-}$, $\underline{e}$, $\underline{R}$, etc.

(3) The tape unit outputs, $\underline{X0}$, $\underline{X1}$.

(4) A supervisory input for each instruction in the given program for which the supervisory unit is being designed $\underline{1}, \underline{2}, \ldots, \underline{n}$.

Because of the requirements of (4), the design of the coded channel is, like the design of the supervisory unit, dependent on a particular program.

Each symbol (we will refer to the objects $\underline{r}$, $\underline{1}$, $\underline{X0}$, as symbols) in the list above must have a code for its representation in the coded channel. For the symbol X, π_X will denote the corresponding code. The total number of symbols required by (1)-(4) above will determine the coding length in the channel. In particular, since there are 23 symbols involved with (1)-(3), a coding length of ℓ will be used where

$$\ell = [\log_2 (n+23)] + 2.$$

($[p]$ denotes the greatest integer m such that $m \leq p$). Every code will have the form $1 i_2 i_3 \ldots i_{\ell-1} 1$, being bounded by a 1 at the beginning and end of the sequence. These codes must be chosen so that they are distinct for distinct symbols. Since recognizers will be used for decoding in the channel, this convention insures uniqueness of representation. To be precise, we will assume a fixed ordering $s_1, \ldots, s_{23}$ on the 23 symbols required for (1) - (3). Further, given a program $f_1, \ldots, f_n$ we will identify s_{23+i} with the input $\underline{i}$ required in (4). Then the code for input s_j in the coded channel is given by

$$\pi_{s_j} = 1\, b(j)\, 1$$

where $b(j)$ is the binary representation of j.

To transmit the symbol X from any unit in the von Neumann machine, π_X is pulsed into the channel using the pulser of Section 3.3. At the desired destination a recognizer, $R(\pi_X)$, will accept the sequence π_X and initiate the activity indicated by X. For example, the supervisory unit will indicate the read-tape command by pulsing the sequence $\pi_{\underline{R}}$ into the channel (the output of the supervisory unit is on the lower edge in Figure 7.1.1). As this sequence is transmitted along the coded channel it will be multiplexed at the top of the supervisory unit where

$\pi_{\underline{R}}$ will enter every input recognizer. Presumably it will not be accepted by any of these devices. Similarly, although $\pi_{\underline{R}}$ will feed every recognizer in the decoding block of the constructing unit, none of these will produce inputs for construction since the coding for all the construction commands will be distinct from that for $\underline{R}$. Finally, $\pi_{\underline{R}}$ will be recognized by $R(\pi_{\underline{R}})$ in the decoding block for the tape unit. After reading the tape, the output $\underline{X0}$ (for example) will pulse $P(\pi_{\underline{X0}})$ and the resulting seqaence will be recognized by $R(\pi_{\underline{X0}})$ somewhere in the decoding block for the supervisory unit. $\pi_{\underline{X0}}$ will continue to travel around the channel but again, by the uniqueness of coding, it will not evoke responses from any other recognizers.

To this point we have distinguished between a particular input, X and the associated code in the channel, π_X. In the sequel this distinction will not be maintained and notation such as P(X) or R(X) is certainly unambiguous.

The coding and decoding areas for the tape and constructing units are drawn in figures 7.2.1 and 7.2.2. These are simply banks of recognizers for decoding the appropriate inputs and, in the case of the tape unit, the coding area consists of a pair of pulsers for coding the outputs $\underline{X0}$ and $\underline{X1}$.

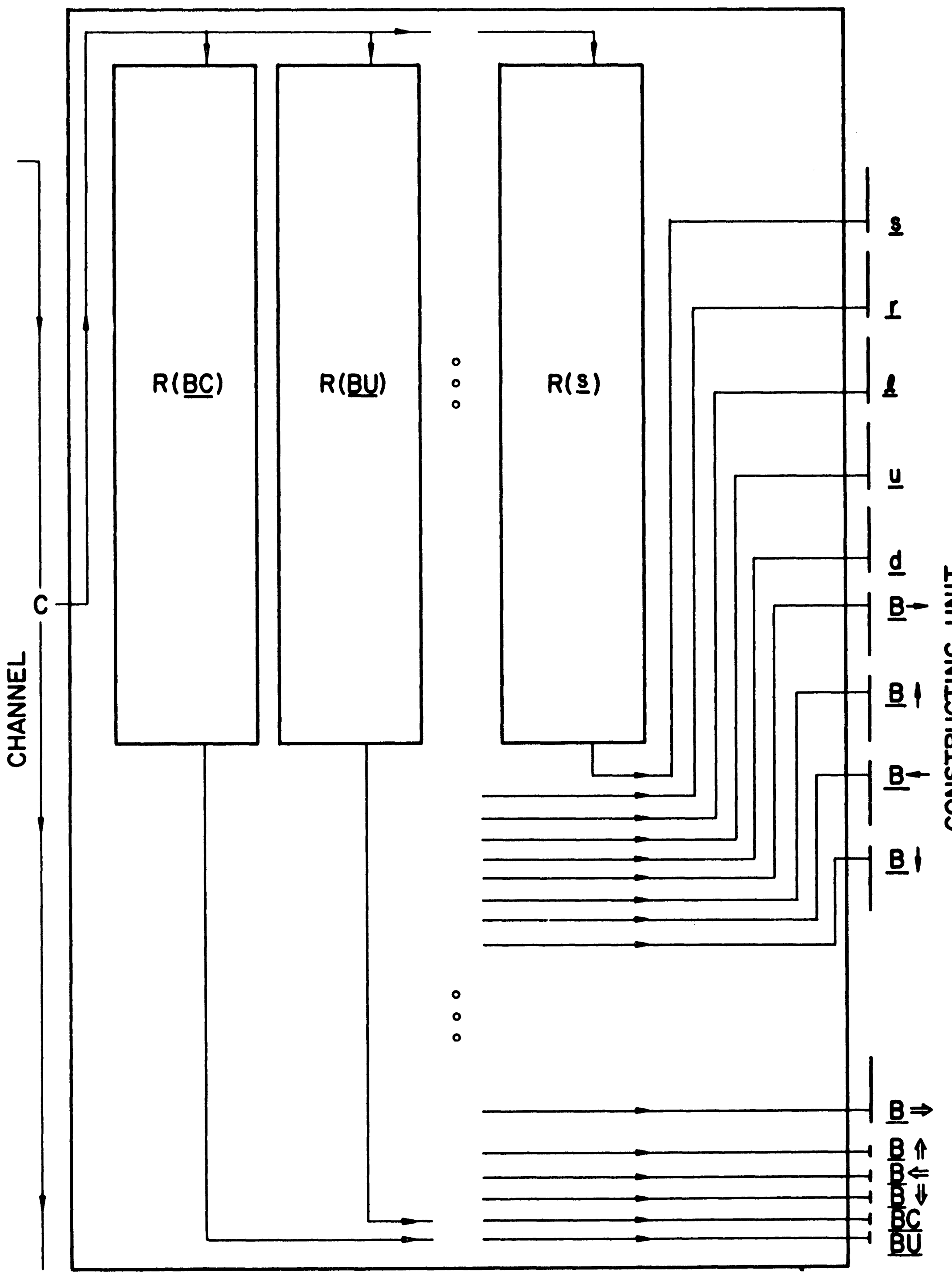


Figure 7.2.1 Decoding Areas for the Constructing Unit

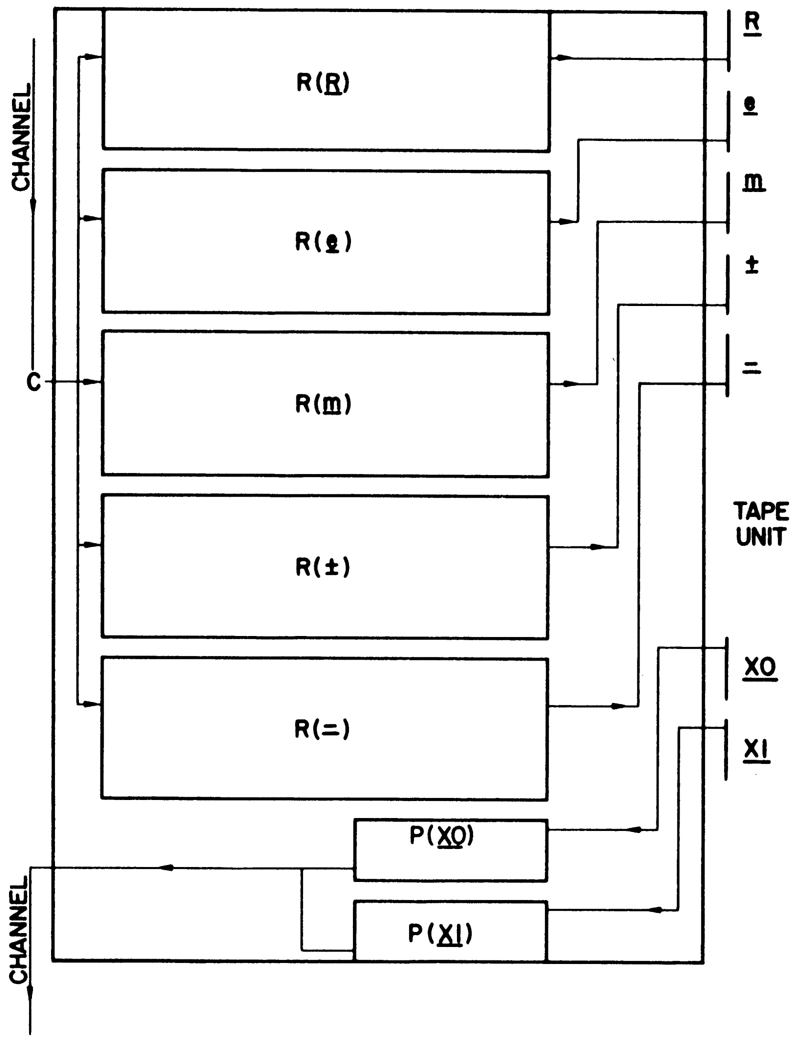


Figure 7.2.2 Coding and Decoding Areas for the Tape Unit

Although, in the representative diagram of 7.1.1, the decoding and coding areas are separated from the supervisory unit, in actual design they will be an integral part of the unit. In fact, except for a periodic pulser which is required for transfer instructions, the supervisory unit is almost entirely coding and decoding.

In Figure 7.1.1 the pulser in the lower left-hand corner is the start mechanism. The input ordinary transmission state of this pulser is excited. With a von Neumann machine as an initial configuration of the model, this excited transmission state will pulse the supervisory input $\underline{1}$ which in turn will initiate the activity of the first instruction.

7.3 Design of a Supervisory Unit

A program in the language of Section 4. The supervisory unit corresponding to the program $F = f_1, \ldots, f_n$ will consist of n segments, one for each instruction, as pictured in Figure 7.3.1. The execution of the i-th instruction, f_i, will be carried out by activity in C_i. In particular, if f_i is not a transfer instruction, the activity will be simply to cause f_i to be transmitted into the channel and to cause the activity to be passed on to the next control unit, C_{i+1}, after a delay sufficient to insure proper operations in the tape and constructing units. If f_i is the instruction tk then $\underline{R}$ is pulsed into the channel to initiate the read operation and the control organ C_i must, in effect, wait for the response $\underline{X0}$ or $\underline{X1}$ from the tape unit. If $\underline{X0}$ is received, control is transferred

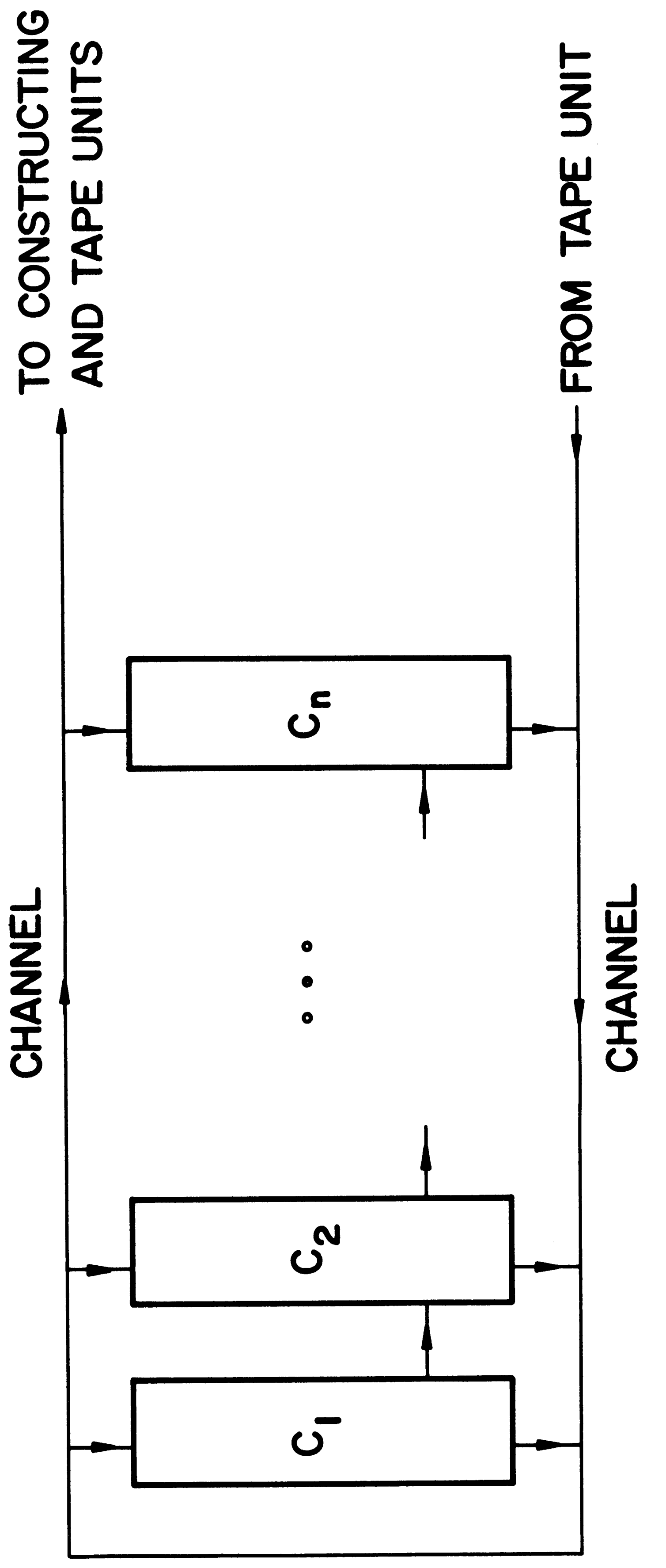


Figure 7.3.1 Organization of a Supervisory Unit

to C_{i+1} but if $\underline{X1}$ is received, $\underline{k}$ is transmitted into the channel and this results in initiating the activity of control organ C_k.

With this indication of how the control will operate we can describe the specific segments C_i. There are two cases depending on whether or not f_i is a transfer instruction. For the case when f_i is not a transfer instruction, the control unit is drawn in Figure 7.3.2. The three parts of this unit are the recognizer, $R(\underline{i})$, the pulser $P(\underline{f}_i)$ and a delay area. Activity in C_i can be brought about by either $\underline{i}$ entering from the channel to be recognized by $R(\underline{i})$ or by a pulse entering from the neighboring unit on the left. The latter would be normal sequencing and the former could occur either through the execution of a transfer instruction ti or with $i=1$ through the starting mechanism. The activity brought about in either of these two ways is simply to pulse the code for the instruction f_i into the channel and to pass activity to the next unit, C_{i+1}. As was indicated in Sections 5.5 and 6.7, the delay of 300 is ample to insure proper operation in the tape and constructing units.

When f_i is a transfer instruction a more complicated unit is required. In Figure 7.3.3, the same methods of stimulating the unit C_i are available. The activity, once the unit is in operation is first to initiate the reading of the tape by transmitting $\underline{R}$ into the channel. But also the periodic pulser is turned on. As a result of this the second

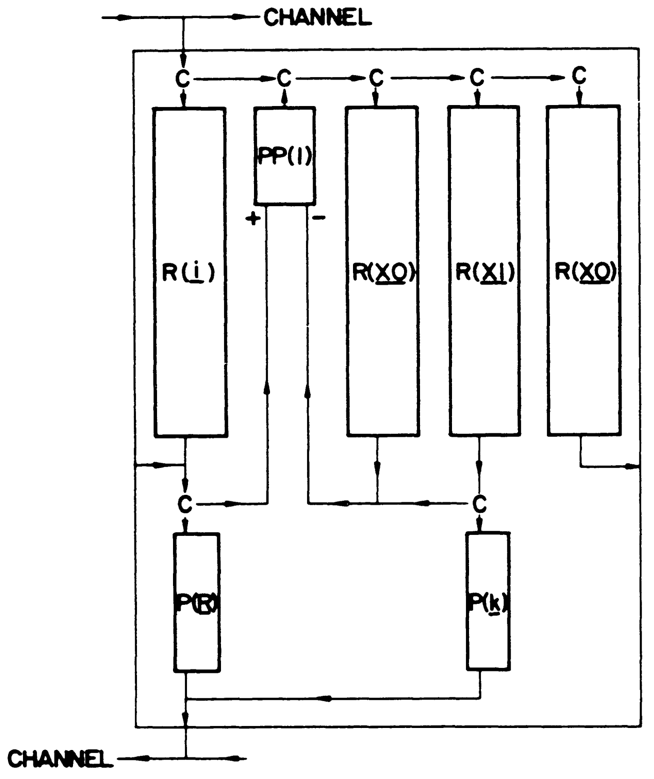


Figure 7.3.3 Control Unit C_i when f_i is the transfer instruction tk

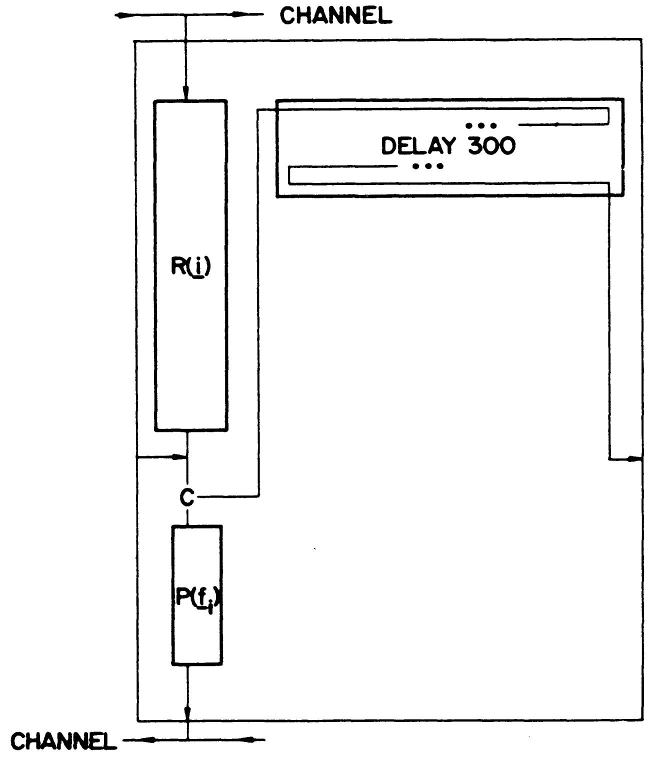


Figure 7.3.2 Control Unit C_i when f_i is not a transfer instruction

confluent state at the top of the Figure will act as an ordinary transmission state (with delay 2) and inputs from the channel will enter the three recognizers, two copies of R(X0) and R(X1). The return from the tape unit of either X0 or X1 will be accepted by one of the recognizers and will turn off the periodic pulser. In addition if X1 is returned, if the current tape square is 1, then transfer to C_k is executed by pulsing k into the channel (the code k in turn being recognized by R(k) in C_k). If X0 is received activity is passed to the next control unit. This is as it was done for non-transfer instructions except that no delay is necessary.

We have not yet considered the one special non-transfer instruction, namely, when $f_i = *$, the stop instruction. Since the design of the supervisory unit is modular in that each control organ C_i is the same size, the simplest accomodation of the stop instruction is to have a segment like that of the other non-transfer instructions with a confluent state at each input and with all other cells of the segment in the quiescent state. It is clear that this, although terrifically uneconomical in terms of size, accomplishes the required purpose.

The description of the method for obtaining a configuration from a program is not in the form of an algorithm. Various orderings of the twenty-three symbols would yield different von Neumann machines, distances between the major components have not been specified and finally we have not considered the coordinates of the resulting configuration at all.

Let us assume that some fixed ordering is chosen and that we have picked a coordinate δ_c which is the lower left-hand corner of the constructing area (Figure 5.2.1). The tape and constructing areas are respectively,

$$U_t = \{\delta + \langle 0, n\rangle \mid n = 0, 1, \ldots\}$$

$$U_c = \{\delta + \langle m, n\rangle \mid m, n = 0, 1, \ldots\}$$

With such choices the process described above can be made to yield a unique configuration v_F for any program F. Further, if F is a program for which construction is "legal" (the constructing arm must work from the top down) then with respect to the constructing area U_c and the tape area U_t, the configuration v_F will behave as dictated by the program for which it was designed.(12)

(12) This is a claim of which, it is hoped, the reader is reasonably convinced. If a formal proof were possible it would certainly be ridiculously complicated with available tools and notations of Cellular Automata Theory.

8. UNIVERSALITY IN THE VON NEUMANN MODEL

The computation universality of the von Neumann model (Proposition V) follows immediately from the algorithm of the previous section. We take the set T' of all tapes made up of states U and $\downarrow$ in the area U_t. Any Turing machine which computes a function from T' into T' is described by some program F in the Turing machine language of Section 4. Thus there is a configuration v_F, the result of applying the algorithm to F, which computes the same function.

The existence of a universal Turing machine does not however imply the existence of a universal computer in the von Neumann model. This is a result of the definition given in Section 1 where, in effect, the only coding allowed is translation. Thus if U is a program for a universal Turing machine (12), the configuration v_U is universal in a reasonable sense (involving coding of both program and data) but not in the sense defined.

There is also an interpretation of the concepts involved that make a single cell in state $T^0_{u,1}$ a universal computer for the von Neumann model. Let v'_F be the configuration obtained from the program F where the quiescent state U replaces the excited ordinary transmission state in the start mechanism (the lower left-hand corner of Figure 7.1.1) and let u be the configuration (exactly one excited ordinary transmission state)

(13) A specific universal Turing Machine program is given in C. Y. Lee, "Automata and Finite Automata", Bell System Monograph 3687, September, 1960.

such that $u(v'_F) = v_F$. v'_F is completely passive and thus satisfies the definition of tape. If F computes the function φ then $u(v'_F)$ computes φ_{δ_t}. Again, it is not unreasonable to say that u is a universal computer but still, the definition of Section 1 is not satisfied.

Since we are talking about tape functions on the set T' (a set of "linear" tapes effectively on $\{0,1\}$, actually on $\{U, \downarrow\}$), the requirement of our definition is that the tape for the universal computer also be in T'.

Let $f_1, f_2, \ldots, f_n, \ldots$ be an effective enumeration of the instructions of the language of Section 4. We define an instruction encoding by

$$\rho(f_n) = 1^{n+1} = \underbrace{11\ldots1}_{n+1 \text{ times}}.$$

Then ρ is extended to a program encoding by:

$$\rho(f_{i_1}, f_{i_2}, \ldots, f_{i_m}) = \rho(f_{i_1}) \frown 0 \frown \rho(f_{i_2}) \frown 0 \frown \ldots \frown 0 \frown \rho(f_{i_m}).$$

It is claimed that the transformation from a program F to a configuration v_F is effective and so also would be the transformation from the string $\rho(F)$ to the configuration v_F. Thus, accepting the thesis of Section 4,(14) there exists a program C of the language of that section which, operating

(14) There are two approaches possible at this point; either to write out in detail a program C that works or to appeal to a Turing-like thesis as we have chosen to do--this choice is obviously the only one to take.

on the tape $\rho(F)$, will construct (and start)[15] v_F at some location in the constructing area U_c. For a given program F, let δ be the coordinate of the cell X_o of the tape of v_F as constructed by C. Then $v_C(\rho(F))$ computes φ_δ when F computes the tape function φ and therefore v_C is a universal computer in the von Neumann model (Proposition VI).

By appealing to some results concerning self-describing machines[16] it is possible to conclude the existence of a self-reproducing von Neumann machine. In particular it is possible to modify the program C, described above, to obtain a program C' which first prints on its tape the encoding e(C') (its own description) and then operates as C would operate. Thus, embedded in the von Neumann model, $v_{C'}$ would construct a copy of $v_{C'}$ somewhere in its construction area--an instance of self-reproduction.

An alternate method for obtaining a self-reproducing automaton is closely related to von Neumann's original proposal. We can modify the program C to obtain another program C" which acts as follows.

(15) Requiring starting offers no difficulty because by arrangement, the cell to be excited is the last cell constructed before retracting the arm.

(16) See C. Y. Lee, "A Turing machine which prints its own code script" and the authors paper "The construction of a self-describing machine" both appearing in Mathematical Theory of Automata, Microwave Research Institute Symposia Series, Volume XII, Polytechnic Press, Brooklyn, 1963.

With the input tape $\rho(F)$, C'' first computes the location of the cell X_o in the configuration v_F that C would construct. Then C'' copies $\rho(F)$ starting with X_o and extending to the right. When this is complete the remainder of C'' acts just as C would with the input $\rho(F)$, i.e., constructs v_F. Since these operations are certainly effective, there is a program C'' which acts as described. Putting this into the cellular system via the algorithm of Section 7, $v_{C''}$ operating on the tape $\rho(C'')$ will construct $v_{C''}$ with tape $\rho(C'')$ which is a second instance of self-reproduction.

DISTRIBUTION LIST

(One copy unless otherwise noted)

Technical Library
Director Defense Res. & Eng.
Room 3C-128, The Pentagon
Washington, D.C. 20301

Defense Documentation Center 20
Cameron Station
Alexandria, Virginia 22314

Chief of Naval Research 2
Department of the Navy
Washington 25, D.C.
Attn: Code 437, Information
Systems Branch

Director, Naval Research Laboratory 6
Technical Information Officer
Washington 25, D.C.
Attention: Code 2000

Commanding Officer 10
Office of Naval Research
Navy 100, Fleet Post Office Box 39
New York, New York 09599

Commanding Officer
ONR Branch Office
207 West 24th Street
New York 11, New York

Office of Naval Research Branch
Office
495 Summer Street
Boston, Massachusetts 02110

Naval Ordnance Laboratory
White Oaks, Silver Spring 19
Maryland
Attn: Technical Library

David Taylor Model Basin
Washington,D.C. 20007
Attn: Code 042, Technical Library

Naval Electronics Laboratory
San Diego 52, California
Attn: Technical Library

Dr. Daniel Alpert, Director
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois

Air Force Cambridge Research Labs
Laurence C. Hanscom Field
Bedford, Massachusetts
Attn: Research Library, CRMXL R

U. S. Naval Weapons Laboratory
Dahlgren, Virginia 22448
Attn: G. H. Gleissner, Code K4
Asst. Dir. for Computation

National Bureau of Standards
Data Processing Systems Division
Room 239, Building 10
Washington 25, D.C.
Attn: A. K. Smilow

George C. Francis
Computing Laboratory, BRL
Aberdeen Proving Ground, Maryland

Office of Naval Research
Branch Office Chicago
230 N. Michigan Avenue
Chicago, Illinois 60601

Commanding Officer
ONR Branch Office
1030 E. Green Street
Pasadena, California

Commanding Officer
ONR Branch Office
1000 Geary Street
San Francisco 9, California

DISTRIBUTION LIST (Concluded)

The University of Michigan
Department of Philosophy
Attn: Professor A. W. Burks

National Physical Laboratory
Teddington, Middlesex, England
Attn: Dr. A. M. Uttley, Supt.
Autonomics Division

Commanding Officer
Harry Diamond Laboratories
Washington, D.C. 20438
Attn: Library

Commanding Officer and Director
U. S. Naval Training Device Center
Port Washington
Long Island, New York
Attn: Technical Library

Department of the Army
Office of the Chief of Research
and Development
Pentagon, Room 3D442
Washington 25, D.C.
Attn: Mr. L. H. Geiger

National Security Agency
Fort George G. Meade, Maryland
Attn: Librarian, C-332

Lincoln Laboratory
Massachusetts Institute of Technology
Lexington 73, Massachusetts
Attn: Library

Office of Naval Research
Washington 25, D.C.
Attn: Code 432

Kenneth Krohn
6001 Dunham Springs Road
Nashville, Tennessee

Mr. Laurence J. Fogel
General Dynamics/Astronautics
Division of General Dynamics Corp.
San Diego, California