Wetware as a Bridge between Computer Engineering and Biology

D. Mange

Logic Systems Laboratory
The Swiss Federal Institute of Technology
CH 1015 LAUSANNE (Switzerland)

Phone : (+41 21) 693 26 39
Fax: (+41 21) 693 37 05
e-mail: daniel_mange@lslqm.epfl.ch

Abstract

Computer engineers, and particularly hardware specialists, observed these last years two fields in rapid development: molecular biology and field-programmable gate array (FPGA) integrated circuits; these circuits are characterized by a regular structure (2-dimension array of identical cells) and programmable capabilities (each cell is programmed separately). It is therefore tempting to use the fundamental biological process of cell development (i.e. the division of a mother cell into two daughter cells and the specialization of each cell depending on its environment) for programming FPGA digital circuits: this is the main goal of this paper.

Starting with a very simple example of digital circuit (a 4-state counter), it is shown that this circuit, as any logic system, can be realized under the form of an irregular 2-dimension cellular automaton. Each cell of this automaton is then programmed by a 8-bit instruction or gene, and the complete information of the network of cells is the genome of the automaton.

Describing the genome by a linear string of instructions or genogram (genome program) allows us finally to use the fundamental process of cell development: the division of the mother cell into two daughter cells is realized by a copying algorithm, while the specialization of the cell is obtained by direct interpretation of the genogram, which is written in a high level language.

This methodology could provide very complex integrated circuits with living creature proprieties: robustness, selfdiagnosis and selfrepair.

1. Introduction

We observed these last years two fields in rapid development which constitute the bases of our present project:

- **molecular biology** which is characterized by information processing at the dna level, on a 4 letter alphabet; each living creature can be decomposed in cells, and in each cell takes place a genome with the complete "blueprint" of the creature, written in the DNA alphabet;
- **field-programmable gate array (FPGA) integrated circuits** which constitute the new hardware technology of digital machines, characterized by a regular structure (2-dimension array of identical cells) and programmable capabilities.

Our project is to build a bridge from computer engineering to biology that we call **wetware**. The final goal of this project is to try to use the fundamental process of cell development, i.e. the division of a mother cell into two daughter cells and the specialization of each cell depending on its environment, for programming FPGA digital circuits. Even if the results reported in this paper are beyond the present possibilities of silicon technology, the important point is the following: **we propose a complete methodology for designing very complex integrated circuits with living creature properties, in particular healing properties.**

As the aim of the conference is to bring together scientists from different fields in the search for common rules and algorithms underlying living systems, we think this is also an opportunity for engineers to show how they could introduce life-like mecanisms in artificial creatures.

2. Historical background

In order to construct a self-reproducing automaton simpler than those of von Neumann (von Neumann 1966) and Codd (Codd 1968), Langton (Langton 1984), followed by Byl (Byl 1989), adopted more liberal criteria. He dropped the condition that the self-reproducing unit must be capable of universal construction. To rule out trivial cases of "reproduction", Langton specified that the **self** (in self-reproduction) should be taken seriously and that the construction of a copy should be actively directed by the configuration itself, rather than being merely a consequence of the transition rules. The criterion employed by Langton requires that the information embedded in the cellular automaton should be both translated (i.e. used as instructions to be executed) and transcribed (i.e. copied as uninterpreted data). Each cell of Langton's automaton produces a 3-bit state and depends on a 5 neighbour environment (the cell itself and its North, East, South and West neighbours); the combinational part of the cell depends finally on 15 binary variables, and the transition table exhibits 131 actual states, a very small subset of the 2^{15} possible states.

In the same spirit, but with a completely different methodology, H. C. Morris (Morris 1988) used the concept of typogenetics first introduced par Hofstadter (Hofstadter 1979) to reproduce in a one dimension environment strings of characters analogous to those of DNA (A, C, G, T).

In both environments (Langton's 2-dimension and Morris' 1-dimension), the initial configurations "do nothing but propagate" (Morris 1988). The final goal of our research is to get more interesting automata, enabling them to "behave" in some senses.

3. Realizing logic systems with irregular 2-dimension cellular automata

Let us have a look at a very simple example to illustrate the different steps of our methodology and to prove that our project is feasible. We start with a classic sequential circuit : a 4-state reversible counter represented by a state graph (Fig. 1). Depending on the value of the input x, zero or one, the counter is counting up or down.

It is well known that any combinational or sequential logic system can be represented by one or several binary decision diagrams (BDD) (Lee 1959) (Akers 1978) (Bryant 1986) (Mange 1992). In our example, two such diagrams represent the two boolean functions Q_1^+ and Q_0^+ describing the next state of the counter (Fig. 2). For the sake of simplicity, we have admitted the two following constraints in this diagram:

1) at each level, the same boolean variable is tested (x, Q_1 or Q_0);
2) there is no branch crossing.

We are therefore ready to realize the BDDs by means of a 2-dimension cellular automaton (Fig. 3): each cell of such an automaton, i.e. each diamond in the BDDs, will realize the test of one logic variable (x, Q_0 or Q_1).

For our own purposes, we have developed a simple prototype of a FPGA cell, called MUXTREE (Fig. 4). This cell is based on a one-variable multiplexer with eight programmable inputs, a horizontal and a vertical bus for long distance connections and a flip-flop for clocked sequential behaviour. Eight bits are sufficient to completely control the MUXTREE cell:

- 2 bits ENABLE $_{1:0}$ for controlling the horizontal and vertical buses;
- 3 bits RIGHT $_{2:0}$ for choosing 1 out of 8 inputs of the right multiplexer branch;
- 3 bits LEFT $_{2:0}$ for choosing 1 out of 8 inputs of the left multiplexer branch.

Using a rectangular network composed of 12 MUXTREE cells (Fig. 5), we are able to realize the two BDDs Q_1^+ and Q_0^+ by means of a 2-dimension irregular cellular automaton. Programming each cell by a 8 bit instruction or **gene** (3 digits in octal code), we obtain the complete description of the automaton, i.e. its **genome** (the dashes in Fig. 5 are don't care conditions).

4. Copying and interpreting the genome program

If we introduce binary coordinates (A and B: horizontal coordinates, C and D: vertical coordinates), it is finally possible to describe the genome itself by a binary decision tree (BDT); depending on the values of A, B, C and D variables, the tree gives as outputs the particular values of the genome for a given cell, i.e. the gene of the cell (Fig. 6). We shall call **genotree** the particular binary decision tree describing a given genome.

Using a very simple high level language called NANOPASCAL, which constitutes the minimal subset of PASCAL, we are able to translate the genotree under the linear form of a sequence of instructions: the genome program or **genogram**. We are finally faced with our last challenge:

- copying the genogram from the mother cell into two daughter cells, and so on, until the whole circuit is completely programmed;

- interpreting the genogram in each cell, in order to calculate the proper gene of the cell, depending on its environment (i.e. its coordinates).

It is worth noting that Langton's criterion for self-reproduction automata (Langton 1984) requires also that the information embedded in the automaton should be both interpreted (i.e. used as instructions to be executed) and transcribed (i.e copied as uninterpreted data).

We propose a first solution for copying the NANOPASCAL genogram (Fig. 5). This program takes place in the mother cell, with coordinates A,B = 00 and C,D = 10. At each clock pulse (times t1...t5), one instruction of the mother cell program is written in the two daughter cells which are its immediate neighbours; and so one for each cell.

Finally, at each clock pulse, each instruction of the program written in every cell is directly executed by a very simple NANOPASCAL interpreter (Fig. 7); this interpreter first calculates the actual coordinates of the cell (A, B, C and D), based on the already known coordinates of the neighbours, and afterwards calculates the gene of the cell depending on its actual coordinates values. The only coordinates which must be provided to the whole system are those for the mother cell, i.e. in our example A,B = 10 and C,D = 10 (Fig. 7).

5. Conclusions

At the moment, we are not able to guess all the consequences and repercussions of such a project; but we know that this project is feasible; we have built and tested two prototypes showing:

- the transformation of a sequential system into an irregular cellular automaton using MUXTREE cells;
- the realization of a possible division/specialization process by means of a NANOPASCAL interpreter;
- the shift mode of the healing process; in this case, it is possible to reprogram the genogram into the cellular space only by changing the border conditions. The first logical coordinates (A,B = 00, C,D = 10) will be given to the cell located at the right and above the faulty cell. The genogram is therefore reprogrammed beyond the area containing the faulty cell. In the example depicted in Figs. 5 and 7, replacing the border condition A,B = 10 by A,B = 11 results in shifting the whole pattern one cell to the right.

Finally, we hope that this bridge between computer engineering and molecular biology could perhaps bring something new to biology itself: particularly the very important part of DNA (90%) which is actually not decoded and which perhaps supports control functions instead of the well known data functions (producing the proteins depending on the DNA language decoding).

Our final conclusion will be extracted from Gilbert's book "Developmental Biology" (Gilbert 1991) (p. 400):

"So far in our discussion of transcriptional control, we have looked at genes whose products are the result of differentiation rather than its cause. The production of globin, for instance, is but the last stage in the differentiation of a red blood cell. Are there cases where transcription is able to cause the cell to become a blood cell rather than a bone cell? Can genes acting early in development direct the cell's fate? In 1940, C.H. Waddington, one of the foremost theorists in developmental biology, predicted the existence of **switch** genes. These genes, he said, would act like the

binary switches in a train yard that cause the trains to follow one path and not the other. Such genes would be responsible for somehow activating a battery of genes for one developmental pathway rather than for an alternative pathway.

Recently, a group of genes has been discovered that have the properties of such **switch** genes. The cells wherein these genes are expressed have the potential to become either of two cell types, and the presence or absence of the gene product determines which lineage the cell generates. In these studies, one sees the convergence of embryology and molecular genetics."

References

Akers, S. B. 1978. Binary Decision Diagrams, *IEEE Transactions on Computers*, vol. C-27, No. 6, pp. 509-516.

Bryant, R.E. 1986. Graph-Based Algorithms for Boolean Function Manipulation, *IEEE Transactions on Computers*, vol. C-35, No. 8, pp. 677-691.

Byl, J. 1989. Self-Reproduction in Small Cellular Automata, *Physica D34*, pp. 295-299.

Codd, E.F. 1968. Cellular Automata, Academic Press, New York.

Gilbert, S. F. 1991. Developmental Biology, 3th edition, Sinauer Associates, Sunderland (Mass.).

Hofstadter, D. 1979. Godel, Escher, Bach, Basic Books, New York.

Langton, C. 1984. Self-Reproduction in Cellular Automata, *Physica 10D*, pp. 135-144.

Lee, C.Y. 1959. Representation of Switching Circuits by Binary-Decision Programs, *The Bell System Technical Journal*, vol. 38, No. 4, pp. 985-999.

Mange, D. 1992. Microprogrammed Systems: an Introduction to Firmware Theory, Chapman & Hall, London.

Morris, H.C. 1988. Typogenetics: a Logic for Artificial Life. In: Artificial Life, Addison-Wesley, Redwood City.

Von Neumann, J. 1966. The Theory of Self-Reproducing Automata, A. W. Burks, ed., Univ. Illinois Press, Urbana.

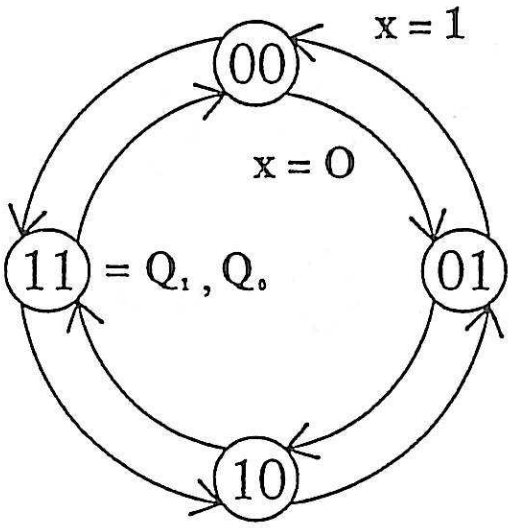

Fig.1. State graph

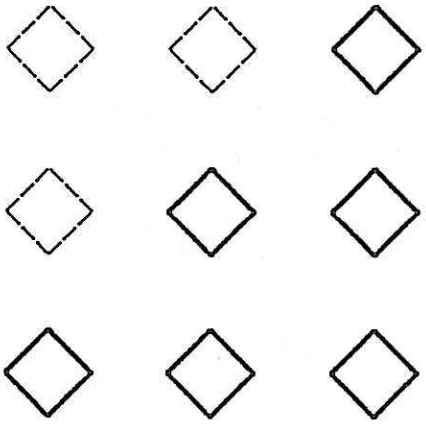

Fig.3. 2-D cellular automaton

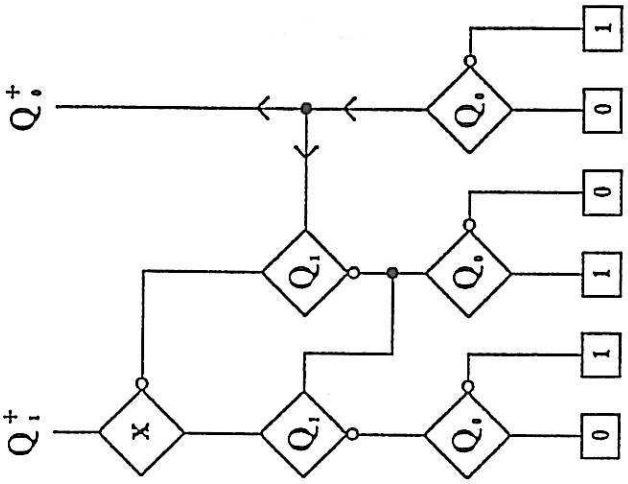

Fig.2. Binary decision diagrams (BDD)

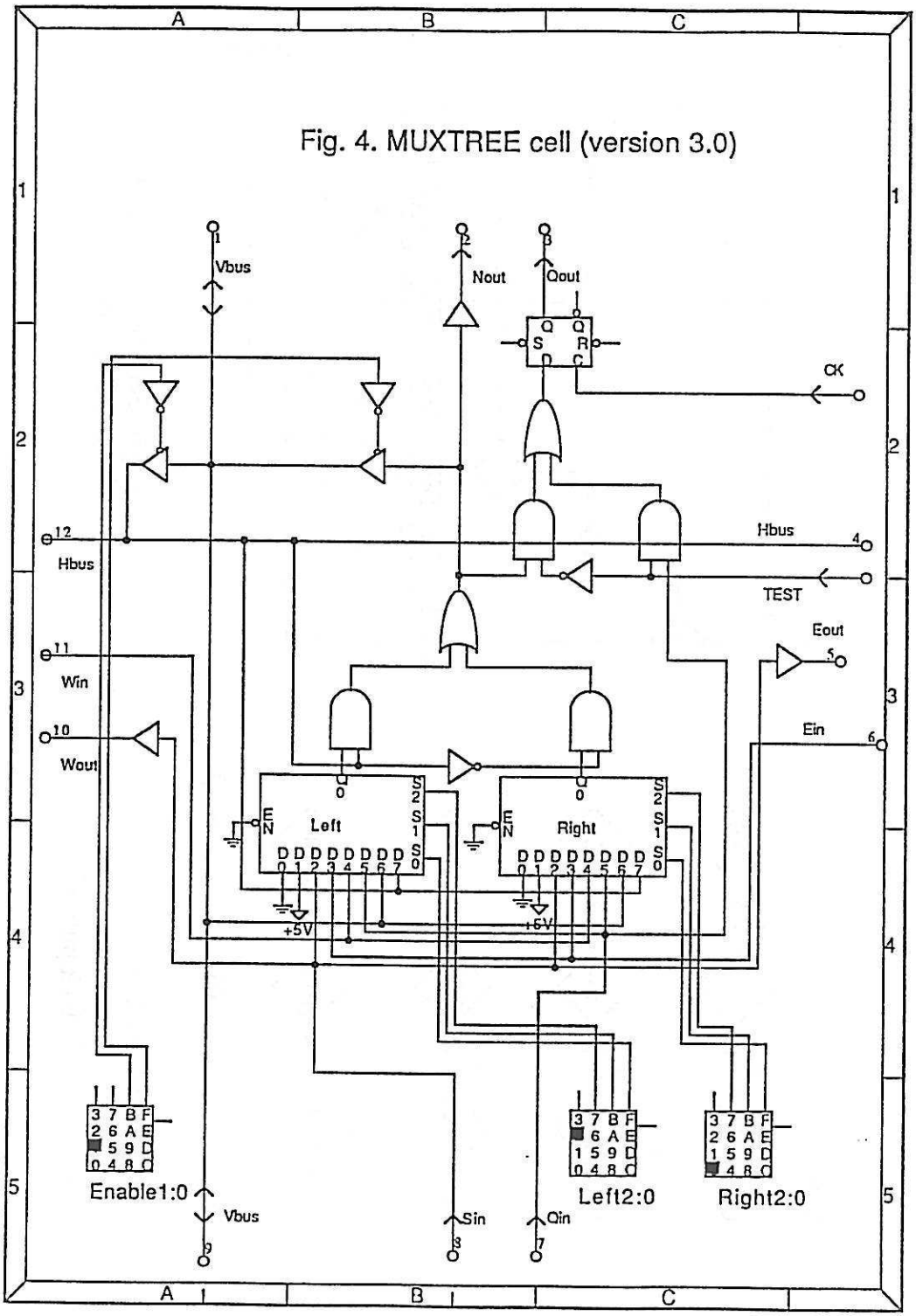

Fig. 4. MUXTREE cell (version 3.0)

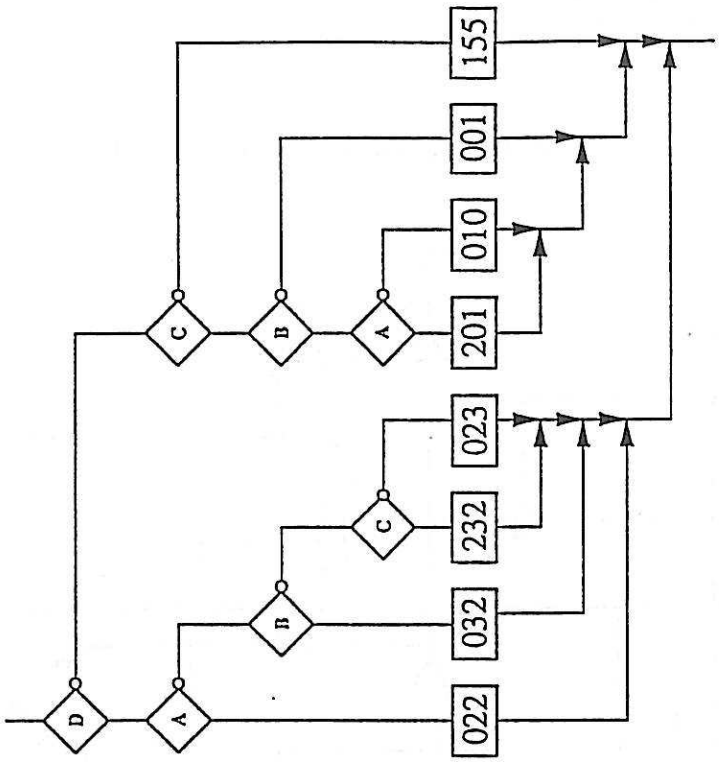

Fig.6. Genome tree (genotree) of the 4-state counter

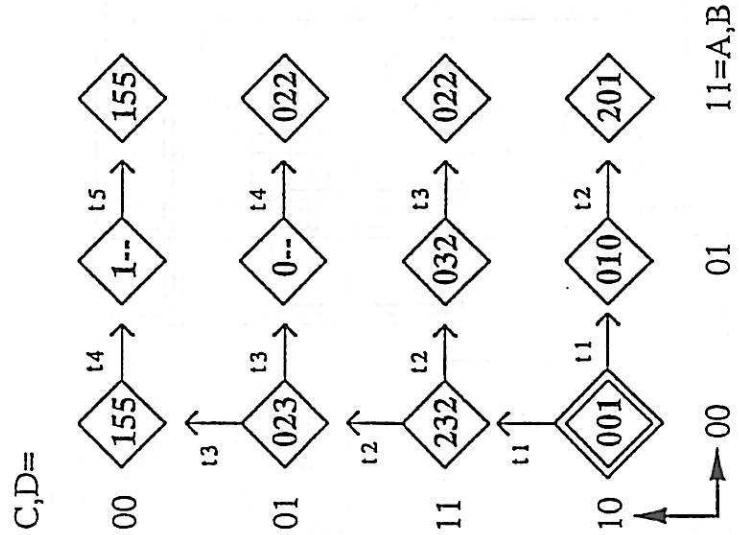

Fig.5. Network of 12 MUXTREE cells ($t_1 \ldots t_5$: 5 clock pulses)

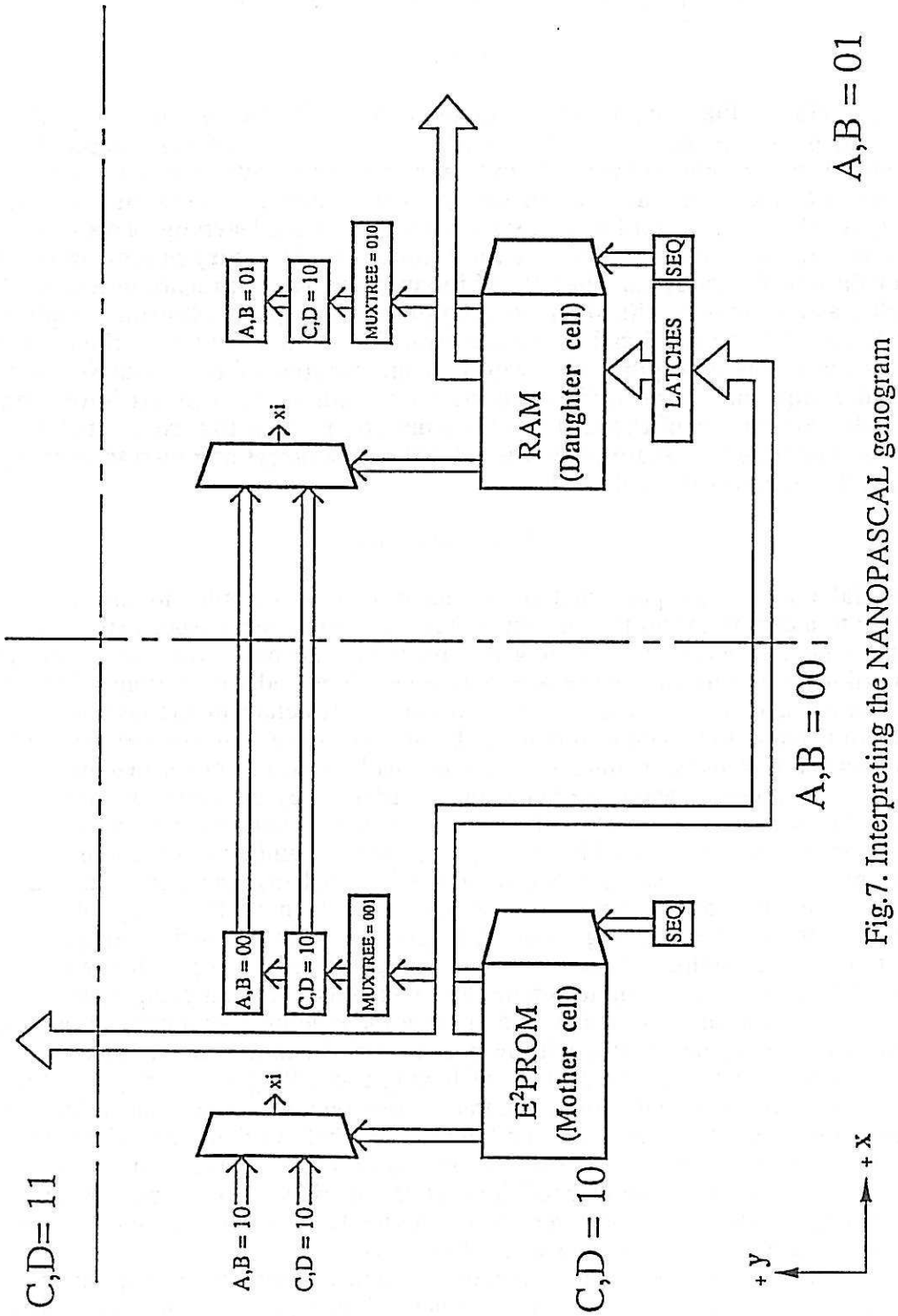

Fig.7. Interpreting the NANOPASCAL genogram