BULLETIN OF THE POLISH
ACADEMY OF SCIENCES
TECHNICAL SCIENCES
Vol. 34, No. 9–10, 1986

Decision Table Computer

by

Zdzisław PAWLAK

Presented by Z. PAWLAK on November 19, 1985

Summary. We propose in this note a new computer architecture based on the idea of the decision table. This kind of computers can be applied as a process control unit or as an ordinary stored program computers, using the non-Von Neuman architecture organization.

1. Introduction. Seeking for a new computer architectures is one of the most important and challenging tasks in contemporary computer science. This note contains a proposal of a new computer organization, based on the idea of decision table, which seems to be very well suited for a wide range of applications.

The Decision Table Computer (DTC) architecture is based on the decision tables theory developed by the author (see [3] and [4]) and we assume that the reader is familiar with these papers. General information on decision tables can be found in [5].

2. Decision tables (DT). An example of a decision table is shown in Table 1. The table presents a cement kiln control algorithm (see [7]).

Columns of the Table are labelled by attributes, called decisions and actions (decisions) attributes. In the example a, b, c and d are conditions whereas e and f are actions attributes. Each row in the table is called a decision rule, which specifies the set of actions to be performed if corresponding conditions are satisfied. Formal definition of a decision table and some other related concepts can be found in [3] and [4].

The above decision table can be represented in the form of a decision algorithm shown below:

$$(a: = 3)\ (d: \neq 3) \Rightarrow (e: = 2)\ (f: = 4)$$

$$(a: = 2) \Rightarrow (e: = 1)\ (f: = 4)$$

$$(c: = 2)\ (d: = 3) \Rightarrow (e: = 2)\ (f: = 3)$$

$$(c: = 3) \Rightarrow (e: = 2)\ (f: = 2).$$

The algorithm is selfexplanatory and needs no detail comments. The algorithm leads to the simplified decision table (Table 2).

TABLE 1

Rule number	a	b	c	d	e	f
1	3	2	2	2	2	4
2	3	2	2	1	2	4
3	2	2	2	1	1	4
4	2	2	2	2	1	4
5	3	2	2	3	2	3
6	3	3	2	3	2	3
7	4	3	2	3	2	3
8	4	3	3	3	2	2
9	4	4	3	3	2	2
10	4	4	3	2	2	2
11	4	3	3	2	2	2
12	4	2	3	2	2	2
13	3	3	2	2	2	4

TABLE 2

	a	b	c	d	e	f
1	3	×	×	$\neq 3$	2	4
2	2	×	×	×	1	4
3	×	×	2	3	2	3
4	×	×	3	×	2	2

The crosses represent "don't cares". Large class of control algorithms and programs (see [2]) can be represented in a form of a decision tables.

We recommend using the decision language as a high level programming language and to implement directly in hardware a computer architecture in which decision rules are basic instructions of the computer and decision algorithms are programs, which are directly performed by the computer.

In order to simplify the computer structure instead of decision rules as instructions, we can employ the set of simpler instructions, for example such as shown below:

IF $\alpha = n$, GO TO m

SET $\beta = n$

PERFORM $(\beta_1, \ldots, \beta_m)$

GO TO m

SET CONDITIONS.

Instruction — IF $\alpha = n$, GO TO m — is to mean that if the value

of the condition attribute α is equal to n, one shall go to the next instructions, e.g. to instruction number m. Of course instead of $(=)$ we can also use other predicate symbols, for example $(\neq, <, >)$ etc.

Instruction — SET $\beta = n$ — sets the value of action attribute β to the value n.

The meaning of remaining instructions is obvious.

Using above set of a simplified set of instructions, the above decision algorithm can be written as follows:

0: SET CONDITIONS
1: IF $a = 3$, GO TO 7
2: IF $d \neq 3$, GO TO 7
3: SET $e = 2$
4: SET $f = 4$
5: PERFORM (e,f)
6: GO TO 0
7: IF $a = 2$, GO TO 12
8: SET $e = 1$
9: SET $f = 4$
10: PERFORM (e,f)
11: GO TO 0
12: IF $c = 2$, GO TO 18
13: IF $d = 3$, GO TO 18
14: SET $e = 2$
15: SET $f = 3$
16: PERFORM (e,f)
17: GO TO 0
18: IF $c = 3$, GO TO 23
19: SET $e = 2$
20: SET $f = 2$
21: PERFORM (e,f)
22: GO TO 0
23: ERROR.

This kind of program can be obtained automatically from the decision algorithm by a compiler (which can be also implemented in hardware).

3. DTC architecture. The architecture of DTC is shown in Fig. 1. Decision Rules Memory contains decision algorithm (or a corresponding program).

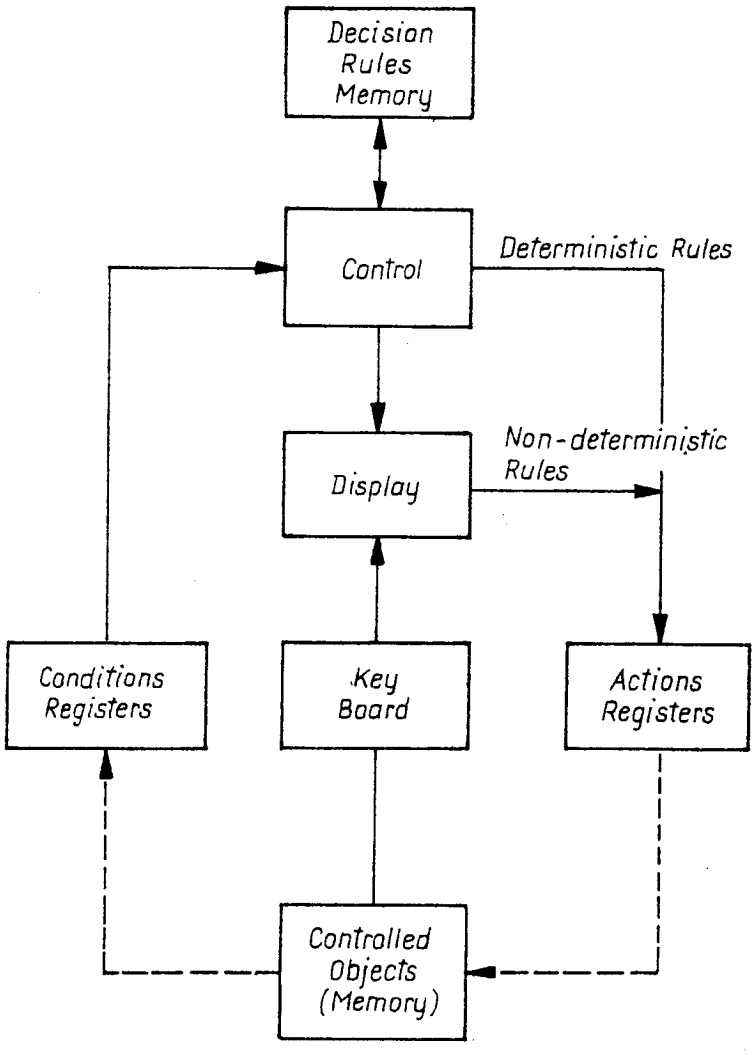

Fig. 1

Conditions Registers contain values of all conditions attributes, and Actions Registers contain values of actions attributes.

Content of Conditions registers are set (through a proper instruction) by the state of the controlled object (or the state of data memory if DTC is used not to control but for computation purposes).

Control by means of a stored program (or decision algorithm) checks the state of conditions registers and set proper values of Action Registers accordingly.

In the case of non-deterministic (or inconsistent) decision rules (see [3] and [4]), the actions performed are to be chosen by the user; so the possible set of actions is displayed on the display, and the user sets the proper actions in the Action Registers, through the Key Board accordingly to his decision.

The instruction — PERFORM — activates Actions Registers and the

proper actions are performed on the controlled object (or the memory).

Of course in the case of computational applications of DTC, proper instructions should be chosen (for example like that in [2]) in order to enable us to perform the computation required. High Speed parallel implementation of the idea given above is also possible, but we shall not discuss it here.

DEPARTMENT OF COMPLEX CONTROL SYSTEMS, POLISH ACADEMY OF SCIENCES, BALTYCKA 5, 44-100 GLIWICE

(ZAKŁAD SYSTEMÓW AUTOMATYKI KOMPLEKSOWEJ PAN)

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF NORTH CAROLINA, NC 28223, USA

REFERENCES

[1] A. Mrózek, *Information systems and control algorithms*, Bull. Pol. Ac.: Tech., **33** (1985), 195–204.

[2] A. Lew, *Decision tables for general purpose scientific programming*, Software — Practice and Experience, **13** (1983), 181–188.

[3] Z. Pawlak, *Decision tables and decision algorithms*, Bull. Pol. Ac.: Tech., **33** (1985), 487–494.

[4] Z. Pawlak, *Rough sets and decision tables*, Lecture Notes, Springer-Verlag (in press).

[5] S. Pollack, H. Hicks, W. Harrison, *Decision tables: theory and practice*, Wiley and Sons Inc., New York, 1971.

3. Павляк, **Компьютер таблиц принятия решений**

В настоящей статье предлагается архитектура компьютера, основанная на понятии таблицы принятия решений. Такие компьютеры могут применяться как блок управления процессом или как обыкновенные вычислительные машины с запоминаемой программой, используя не-Фон Неймана устройство архитектуры.