

THE CORRECTION OF THE „NON-PERFORMING” DIVISION ALGORITHM

by

FARKAS GHEORGHE (Cluj)

The „non-performing” division algorithm is described in [1]. The interest of this algorithm relies in the fact, that, depending on the structure of the arithmetic unit, it can be faster than the non-restoring one.

We introduce the following notations :

- D — dividend
- I — divisor
- Q — quotient
- Q' — the corrected quotient
- q_i — the i -th bit of the quotient
- R_i — the i -th rest
- S_{i+1} — the $i + 1$ — th content of the adder
- Sg — sign
- $\equiv$ — equivalence (logic function)
- $\bar{I}$ — the two's complement of the I

The forming rules of the quotient and of the rest are presented in table Nr. 1

Table Nr. 1

Nr. case	$Sg D \equiv Sg I$	S_{i+1}	$Sg D \equiv Sg S_{i+1}$	q_{i+1}	R_{i+1}
1	1	$2R_i - I$	1	1	S_{i+1}
2	1	$2R_i - I$	0	0	$2R_i$
3	0	$2R_i + I$	1	0	S_{i+1}
4	0	$2R_i + I$	0	1	$2R_i$

After finishing the division cycle corresponding with table Nr. 1, the following correction is proposed in [1]:

if $Q < 0$, then

$$Q' = Q + 2^{-n}; \quad (1)$$

The algorithm has the property to hold the sign of the rest equal with the sign of the dividend. From this property it results, that the condition

$$D < 0 \quad \text{and} \quad R = 0 \quad (2)$$

is not realizable. Consequently, the exact division (rest equal to zero) for $D < 0$, will not be correctly realized.

The aim of this paper consists in the elaboration of a correction method, which averts this inexactitude.

We consider two cases:

$$\left. \begin{aligned} 1) \quad & \text{Sg}D = -; \\ & \text{Sg}D \equiv \text{Sg}I = 1; \\ & S_{i+1} = 2R_i - I = 0; \\ & \text{Sg}D \equiv \text{Sg}S_{i+1} = 0; \end{aligned} \right\} \quad (3)$$

From table Nr. 1 it results the case Nr. 2:

$$q_{i+1} = 0 \quad (4)$$

$$R_{i+1} = 2R_i. \quad (5)$$

From the condition $S_{i+1} = 0$ it results that

$$2R_i = I. \quad (6)$$

From (5) and (6) it results:

$$R_{i+1} = I. \quad (7)$$

In reality, the condition (3) assures a succesfull operation with the rest equal to zero, and the quotient bit equal to one. The application of the algorithm gives us the inexact results (4) and (7).

We continue the division, the following step being:

$$S_{i+2} = 2R_{i+1} - I = 2I - I = I; \quad (8)$$

From the table Nr. 1 it results the case Nr. 1

$$q_{i+2} = 1; \quad (9)$$

$$R_{i+2} = S_{i+2} = I. \quad (10)$$

Consequently, for the discussed case, if the division is exact, it results that

$$R_{i+1} = R_{i+2} = \dots = R_n = I; \quad (11)$$

$$q_{i+2} = q_{i+3} = \dots = q_n = 1. \quad (12)$$

The quotient will have an inexact form:

$$Q = q_0 q_1 q_2 \dots q_i 0 111 \dots 111; \quad (13)$$

The correct quotient should be

$$Q' = q_0 q_1 q_2 \dots q_i 100 \dots 00 \quad (14)$$

From (11), (13) and (14) it results the correction rule for the quotient and the rest in the considered case:

$$Q' = Q + 2^{-n}; \quad (15)$$

$$R'_n = 0; \quad (16)$$

$$\left. \begin{aligned} 2) \quad & \text{Sg}D = -; \\ & \text{Sg}D \equiv \text{Sg}I = 0; \\ & S_{i+1} = 2R_i + I = 0; \\ & \text{Sg}D \equiv \text{Sg}S_{i+1} = 0. \end{aligned} \right\} \quad (17)$$

From table Nr. 1 it results the case Nr. 4

$$q_{i+1} = 1; \quad (18)$$

$$R_{i+1} = 2R_i. \quad (19)$$

From the condition $S_{i+1} = 0$, it results that

$$2R_i = -I = \bar{I}. \quad (20)$$

From (19) and (20)

$$R_{i+1} = \bar{I}. \quad (21)$$

We continue the division, the next step being:

$$S_{i+2} = 2R_{i+1} + I = 2\bar{I} + I = \bar{I}. \quad (22)$$

From table Nr. 1 we obtain the case Nr. 3:

$$q_{i+2} = 0; \quad (23)$$

$$R_{i+2} = S_{i+2} = \bar{I}. \quad (24)$$

For this case it results, for the exact division:

$$R_{i+1} = R_{i+2} = \dots = R_n = \bar{I} \quad (25)$$

$$q_{i+2} = q_{i+3} = \dots = q_n = 0. \quad (26)$$

Consequently, we obtained the correct form of the quotient; without correction:

$$Q = q_0 q_1 q_2 \dots q_i 100 \dots 00. \quad (27)$$

The rest must be corrected in this case, too:

$$R'_n = 0. \quad (28)$$

The correcting rule, resulting from this paper is presented in table Nr. 2.

Table Nr. 2

SgD	$SgD \equiv SgI$	$R_n \pm I \equiv 0$	correction for Q	correction for R
+	1	—	—	—
+	0	—	$Q' = Q + 2^{-n}$	—
-	1	0	—	—
-	1	1	$Q' = Q + 2^{-n}$	$R'_n = 0$
-	0	0	$Q' = Q + 2^{-n}$	—
-	0	1	—	$R'_n = 0$

The logic schema, realizing the correction corresponding with table Nr. 2, is presented in fig. Nr. 1.

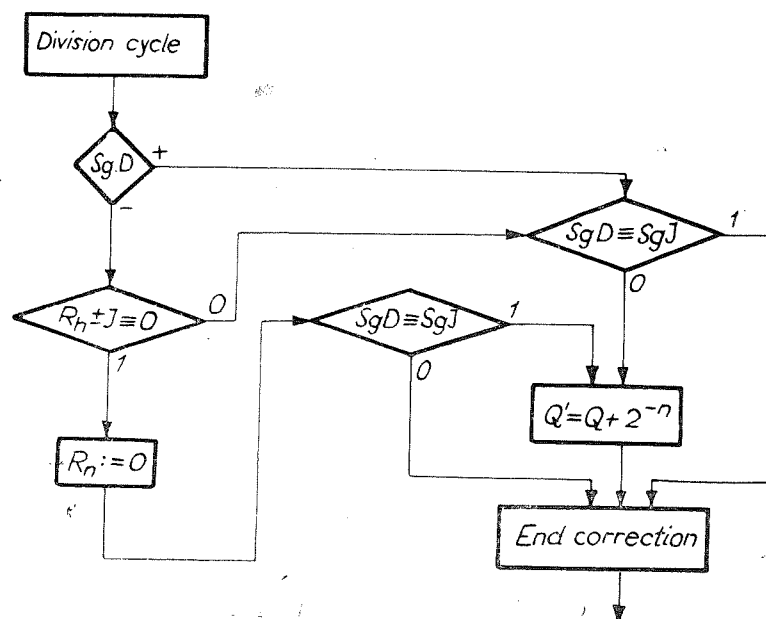


Fig. 1

This work is elaborated at the „Institut de Calcul” of the Cluj Branch of the Academy of the Socialist Republic Rumania.

Received 1.III.1968

BIBLIOGRAPHY

1. IVAN FLORES, The logic of computer arithmetic. Prentice-Hall. Inc. 1963.