DRUM STORE ADDRESSING SYSTEMS AND THEIR INFLUENCE ON EFFECTIVE SPEED OF OPERATION OF AN ELECTRONIC COMPUTER

By

J. BAŃKOWSKI

Department of Electronical Constructions, Polytechnical University, Warszawa

(Received June 3, 1963)

Presented by Prof. Dr. L. KOZMA

Word position in a drum store is fully determined by two numbers, namely, track number and a number which determines word position on a track. As far as effective speed of operation is concerned only the addressing system on one track is of interest, because in most computers words on all tracks are addressed in the same way. In the following our considerations will be limited only to distribution of words on one track.

Let us say, that there is N cells on one track, full cycle of executing one instruction is τ word durations[1] and that instruction nr. i + 1 comes under the read-write head Δ_i word durations after instruction nr. i had departed.[2] Then, if obeying instruction nr. i needs a word from drum store, there is $\Delta_i—\tau + 1$ addresses which possess the property for the following instruction to be taken after Δ_i word durations. If any other address is used in the instruction nr. 1 then the following instruction would be taken after $N + \Delta_i$ word durations. When a program was not optimalized an average effective time of obeying one instruction is:

$$T_{ai} = \frac{(\Delta_i - \tau + 1)}{N} \Delta_i + (N + \Delta_i) \frac{(N + \tau - \Delta_i - 1)}{N} = \frac{1}{N} (\Delta_i^2 - \Delta_i \tau +$$
$$+ \Delta_i + N^2 + N\tau - N\Delta_i - N + N\Delta_i - \Delta_i^2 + \Delta_i \tau - \Delta_i) = N + \tau - 1 \qquad (1)$$

If an instruction needs no word from the drum store and $\Delta_i \geqslant \tau$ then the next instruction is taken after Δ_i word durations. When the probability of such instruction is p, and all Δ_i's are equal, then an average time of obeying one instruction is:

$$T_a = p\Delta + (1—p) \cdot (N + \tau—1) \qquad (2)$$

The value of T_a increases for larger values of Δ. It can easily be shown that minimal values of T_a occur when all Δ_i's are equal. In reality, if all

[1] Instructions that are executed in time other than that are disregarded.
[2] We assume one-address, sequential computer.

Δ's are not equal value $\frac{1}{N}\sum_{1}^{N}\Delta_1$ instead of Δ in the equality (2) should be substituted and, of course, $\frac{1}{N}\sum_{i}^{N}\Delta_i > \tau$.

If all Δ's are equal a set of x's may be chosen that there be a congruence:

$$\Delta \cdot a \equiv x \pmod{N} \tag{3}$$

where $x = 0, 1, 2, \ldots N\text{-}1$ is word number on a track and $a = 0, 1, 2, \ldots N-1$ is address of this word. This method of addressing will be called linear addressing with constant distance Δ. For practical use it is necessary to have one, and only one address, for each cell. This cannot obviously be achieved for all values of Δ. This condition may be written as

$$kN \neq \Delta \mathrm{a} \tag{4}$$

for all a and natural k.

Equality

$$(N, \Delta) = 1 \tag{5}$$

(i.e. N and Δ are mutually primes)

is the necessary and sufficient condition for fulfilling unequality (4). If N and Δ are divisible by any $r > 1$ then

$$k \cdot \frac{N}{r} \cdot r \neq a \cdot r \cdot \frac{\Delta}{r} \tag{6}$$

and

$$k \cdot \frac{N}{r} \neq \frac{\Delta}{r} \cdot a \tag{7}$$

which is not true for $= s\frac{N}{r}$ where $s = 1, 2, \ldots r-1$, so this condition is necessary; it is also sufficient because the smallest number divisible by two mutually primes is their product, so the smallest value which does not satisfy unequality (4) is $a = N$ a number not belonging to the set of a.

In case of non-optimalized programmes it is obviously most suitable to use linear addressing with distance $\Delta \geqslant \tau$ where Δ is the smallest mutually prime with respect to N.

Most programmes for electronic computers utilizing drum store as working store are time optimalized. Now we shall consider the case of fully optimalized program. The program is considered as fully optimalized when it is executed in the shortest time possible and data distribution is given a priori. Such a situation is typical for computation programmes and not for standard subroutines in which coefficients are often optimally distributed. A fully optimalized program is, in most cases, about two times faster than a non-optimalized one for the same problem.

Time optimalization is achieved by putting the instruction nr. i in such a cell for it to be obeyed in Δ word durations. If the preceding instruction of the program was not at the preceding address these instructions should be joined by means of a jump instruction, but instruction nr. i must not come under reading head sooner than τ word durations had passed after receiving a jump command. Effective time of execution of an instruction is measured after receiving a jump instruction (if any such instruction is necessary) up to the moment when following instruction is taken.

If $\tau \leqslant \Delta \leqslant 2\tau - 1$ then there is

$\Delta - \tau + 1$	cells from which an instruction is executed in Δ word durations
$\Delta - \tau + 1$	$+ \tau$ word durations
1	$\Delta + \tau + 1$ word durations
1	$\Delta + \tau + 2$ word durations
....	....
$2\tau - \Delta - 1$	$N + \Delta$ word durations

Average time of executing one instruction in such a program is

$$T_{1a} = \frac{1}{N}\Big[\Delta(\Delta - \tau + 1) + (\Delta - \tau + 1)(\Delta + \tau) + (N + \Delta)(2\tau - \Delta - 1) + + \sum_{k=1}^{N-\Delta-1} (\Delta + T + k)\Big] =$$

$$= \frac{1}{N}\Big[\Delta^2 + \Delta(\tau - N + 1) - \tau^2 + \tau + 2N\tau - N + + \frac{1}{2}(N - \Delta - 1)(N + \Delta + 2\tau)\Big] =$$

$$= \frac{1}{2N}[\Delta^2 - \Delta(2N - 1) + N^2 + 6N\tau - 3N - 2\tau^2] \qquad (8)$$

If $2\tau - 1 \leqslant \Delta$ then there is

$\Delta - \tau + 1$	cells from which an instruction is executed in Δ word durations
τ	$\Delta + \tau$ word durations
1	$\Delta + \tau + 1$ word durations
1	$\Delta + \tau + 1$ word durations
....	....
....	....

Average time of executing one instruction in such a program in this case is

$$T_{2a} = \frac{1}{N}\left[(\Delta - \tau + 1)\Delta + \tau(\Delta + \tau) + \sum_{K=1}^{N-\Delta=1}(\Delta + \tau + k)\right] =$$

$$= \frac{1}{2N}[2\Delta^2 + 2\Delta + 2\tau^2 + (N - \Delta - 1)(N + \Delta + 2\tau)] =$$

$$= \frac{1}{2N}[\Delta^2 - \Delta(2\tau - 1) + N^2 + 2N\tau - N + 2\tau^2 - 2\tau]. \qquad (9)$$

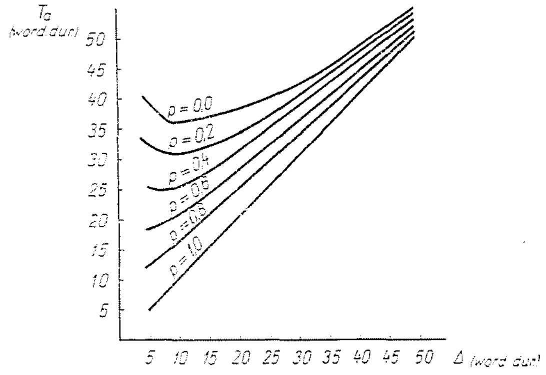

Fig. 1

Function T_{1a} has a minimum for $\Delta = \frac{2N-1}{2}$ and in the interval $\tau \leqslant \Delta \leq 2\tau—1$ monotonically decreases while function T_{2a} has a minimum for $\Delta = :\frac{2\tau - 1}{2}$ and for $\Delta > 2\tau—1$ monotonically increases.

For

$$T_{1a}(2\tau—1) = T_{2a}(2\tau—1) \qquad (10)$$

then the most suitable distance of linear addressing for fully optimalized programmes is $\Delta = 2\tau—1$ if only instructions using drum store are concerned. Taking the others under consideration, this leads to the formula

$$T_a(\Delta) = p\Delta + (1—p) \ \{T_{1a}(\Delta) \cdot [1(\tau)—1/2\tau—1)] + T_{2a}(\Delta) \cdot 1/2\tau—1)\} \qquad (11)$$

where $1(x)$ is Heaviside unit function of argument x. On the Fig. 1 functions $T_a(\Delta)$ are plotted for a few values of p when $N = 64$ and $\tau = 5$.

From the diagram it can easily be seen that up to $p = 0.4$ optimal distance of addressing is $\Delta = 9 = 2\tau—1$. For $0 \leqslant p \leqslant 0.2$ (in practical programmes there is rarely more than 20% instructions not using drum store) such addressing system gives effective speed of operation greater by x% and

$9.9 \leqslant x \leqslant 12$	for $\varDelta = 5$
$3.5 \leqslant x \leqslant 4.1$	for $\varDelta = 7$
$0.6 \leqslant x \leqslant 1.9$	for $\varDelta = 11$
$1.2 \leqslant x \leqslant 3.9$	for $\varDelta = 13$

The largest value of p at which $\varDelta = 9$ is still an optimal distance is $p_9 = 0.464$ correspondingly $p_7 = 0.473$ and $p_5 \geqslant 0.477$; these values of p in most cases do not occur in practical programmes.

The most practical programmes involve programming loops, so that some instructions cannot be put optimally or it is uneconomical to do so (time optimalized program is split to pieces and suitably distributed in the drum store and, besides, described optimalization technique needs one jump instruction for each instruction to be optimalized).

This fact affects our considerations only a little, because only interpretation of p must be changed. In this case p is the probability of an instruction not using drum store from among all instructions of the program, but the instructions not optimalized, obeyed in $N + \tau—1$ word durations.

Summary

In the paper various drum store addressing systems with special attention to linear addressing systems with constant distance between successive addresses are described. Conditions are also given which must be fulfilled to make an addressing system realizable and mathematical formulae for finding the most effective value of distance for optimalized and non-optimalized programmes.

Mgr. eng. Jacek Bańkowski, ul. Karolkowa 58, Warszawa, Poland
Katedra Budowy Maszyn Matematycznych Politechnika Warszawska.