PDP-11/40 TECHNICAL MEMORANDUM # 12

TITLE:	PDP-11/40 Addressing Modes
AUTHOR(s):	Ad van de Goor
INDEX KEYS:	Addressing Modes Indexing List Addressing
DISTRIBUTION KEYS:	PDP-11/40 Group PDP-11 Co-ordinating Committee Bruce Delagi Larry McGowan Dave Knight Jim Bell Ron Brender Jack Richardson Jeff Scott
REVISION:	None
OBSOLETE:	None
DATE:	June 29, 1970

0.0 ABSTRACT

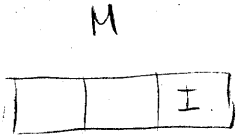
The auto decrement deferred mode "@-(R)" as implemented on the PDP-11/20 has been found to be of very little use. In all the PDP-11 programs the author has seen, it has been used only twice.

Three alternative addressing modes are discussed to replace the "@-(R)" mode. It is concluded that the adjusting index mode "(R)A" was the most promising.

The adjusting index mode "(R)A" is very similar to the regular index mode "A(R)" except for the computation of the effective address "EA".

For A(R), the EA is: EA= (R) + A

For (R)A, the EA is: EA= A + L * (R)
where L= length of data in bytes



0	0	OPR R
0	1	OPR +(R)
1	0	OPR -(R)
1	1	OPR X(R)

1.0 ALTERNATIVE ADDRESS MODES

The alternative modes to be discussed are listed below.

1.1 Post Indexing

Post indexing is very important in page oriented machines (i.e., machines where all of core-memory cannot be addressed in the basic instruction). The effective address computation in post indexing is as follows:

$$EA = (A) + (R)$$

In a machine where all of core memory can be addressed directly this is of very little use (only the Sigma 5/7 has it!).

PDP-11/20 programmers expressed the desire to be able to index a register with another register rather than a register with a constant, i.e., $EA = (R_i) + (R_j)$ rather than $EA = A + (R)$.

The need for this register-register indexing arises from the fact that, in re-entrant subroutine calls, parameters are pushed on the stack. A parameter passed to a subroutine could be the address of an array A. The PDP-11/20 code for the expression: $A[I] := 5$ in the body of such a subroutine is as shown below.

The following assumptions are made.

1. The array A is a 16-bit integer array
2. The quantity I is in register R_i
3. The array pointer is located "n" words below the stack pointer R6

```
MOV   n(R6), R0  / Move array address A to R0
MOV     Ri,  R1
ROL          R1  / Assume (C)=0
ADD     R0,  R1  / Form address of A I
MOV     #5,@ R1
```

When register-register indexing would be possible, the ADD instruction would not be necessary.

This can be accomplished by:

1) laying out the register block over core-memory and
2) using the suggested post indexing mode.

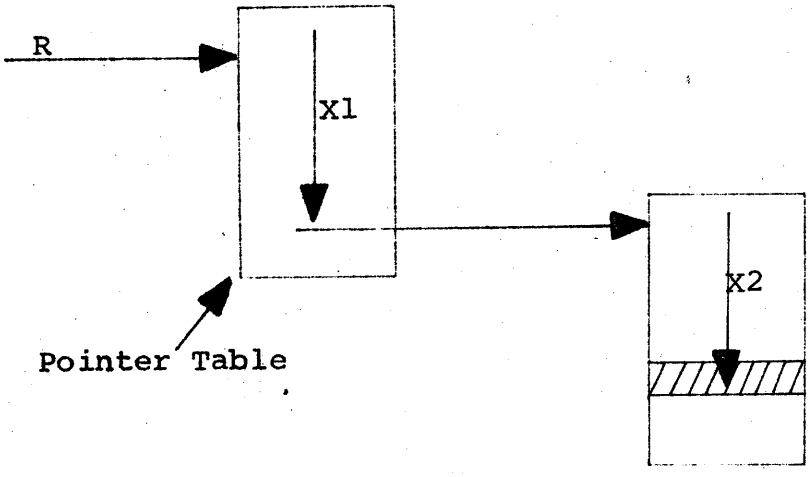
Indexing with two registers is accomplished as follows:

EA= (address of Ri) + (Rj)

1.2 List Addressing

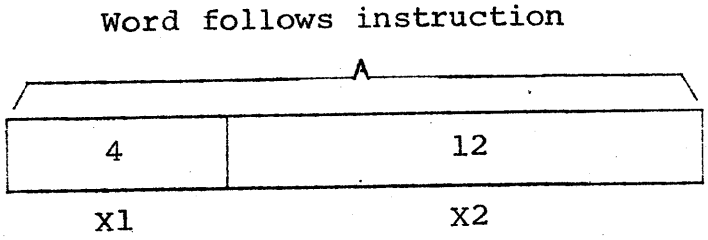
This addressing mode is used on an Algol oriented machine, the X8, made by ELECTROLOGICA in Holland. The addressing is done as follows:

EA= ((R) + X1) + X2



In the PDP-11/40, this instruction could be implemented as follows:

R= specifies the register as in @-(R)
X1 and X2 are specified, in a word following the instruction, as shown below.



This addressing mode is very useful in Algol compilers (and probably also in other compilers and data structures). It is used to address variables which, in Algol, can be on different lexicographical levels. X1 is used to specify the lexicographical level. X2 is used to specify the specific variable at that level. In Algol terminology; this mode solves the Algol display problem in hardware.

1.3 Adjusting Index Mode

This method of indexing is used on the Sigma 5/7 and the CSI computer. For computers which allow addressable multiple length data (like the PDP-11), this is a very important indexing mode. The handling of arrays becomes much easier because the index register adjusts itself according to the length of the data.

The adjusting index mode will be symbolically represented as "(R)A" as compared with "A(R)" for regular indexing. The effective address for the (R)A mode is computed as follows:

EA= A + L * (R); where L= length of data type in bytes

EXAMPLE:

Assume the following program loop:

```
For I:= 1 step 1 until 100 do
Begin
    A[I] := I;
    B[I] := ----
    ----
    End;
```

Assume A to be a 16-bit integer array and assume in the examples below that the integer I is stored in register Ri.

PDP-11/20 Method

```
MOV   Ri, Rj
ROL       Rj      /assume (c) bit= 0
MOV   Ri, A(Rj)
```

Adjusting Index Method

```
MOV   Ri, (Ri)A
```

This saves 2 instructions and the use of an extra register. It should be noted that in the current PDP-11/20 A(R) mode, the register is used as a base register, primarily.

2.0 RECOMMENDATION

The adjusting index mode "(R)A" is the most promising because of its expected high frequency of use. Indexing into arrays or tables can then be done without adjusting the index quantity "(R)" according to the data size and no extra registers are needed to compute the adjusted index quantity.

Comparing the proposed indexing mode "(R)A" with the current @-(R) mode, it has to be noted that this new mode is much more useful. Also, the @-(R) mode is hardly ever used and as such should be replaced with the (R)A mode.