Identification

Data Representation
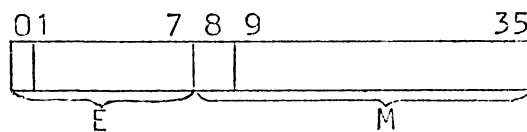R. M. Graham and M. D. McIlroy

Purpose

The actual machine representations of the PL/I data is
implementation dependent. These representations are specified
in this section.
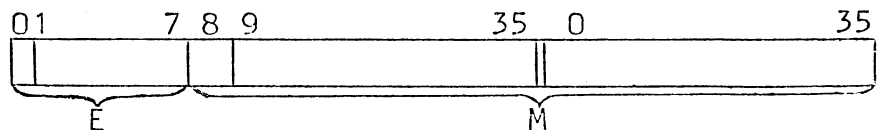
Arithmetic Data

1) Real Binary Floating-Point

    i) For precision   27, the datum is stored as a 645
        single-word precision (binary) floating-point
        number, i.e.,

        where E is a signed integral exponent and M is a signed
        fractional mantissa. Both M and E are in 2's
        complement form.

    ii) For 28   precision   63, the datum is stored as a
         645 double-word precision (binary) floating point
         number, i.e.,

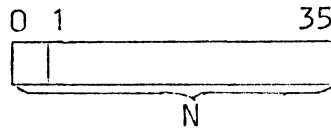   which is always located at an even/odd pair of
         addresses.

2) Real Decimal Floating-Point

    i) For precision   8, the datum is stored as a 645
        single-word precision (binary) floating-point
        number.

ii) For $9 \leq$ precision ≤ 18, the datum is stored as a 645 double-word precision (binary) floating-point number.
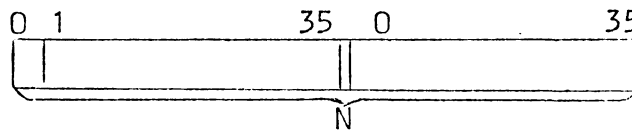
3) Real Binary Fixed-Point

i) For precision ≤ 35, the datum is stored as a 645 single-word precision (binary) fixed-point number, i.e.,.

where N is a signed integer in 2's complement form.

ii) For $36 \leq$ precision ≤ 71, the datum is stored as a 645 double-word precision fixed-point (binary) number, i.e.,

which are always located at an even/odd pair of addresses.

4) Real Decimal Fixed-Point

i) For precision ≤ 10, the datum is stored as a 645 single-word precision (binary) fixed-point number.

ii) For $11 \leq$ precision ≤ 21, the datum is stored as a 645 double-word precision fixed (binary) number.

5) Scaling of Fixed Point

Fixed point data with scale factors different from zero, will be represented as integers of the appropriate precision and the necessary scaling instructions will be compiled.

5) Complex

Complex data will be stored as pairs of the corresponding real data in their appropriate representation. The part is first and the imaginary part is second.

String Data

1) Bit String

A bit string may begin at any bit in a word and extends
into as many consecutive words as are required for its
length. All words, except possibly the first and last,
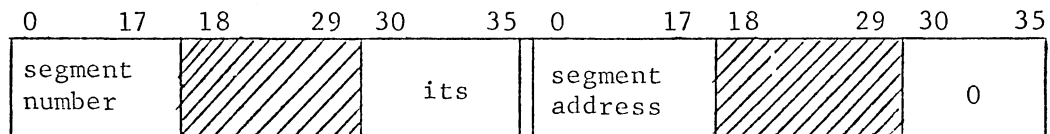contain 36 bits of the string.

2) Character String

Individual characters are coded in 7 bits, as specified
in BC.2.01, right justified in 9-bit bytes with leading
zeroes. A character string may begin at any one of the
four 9-bit bytes in a word and extends into as many
consecutive words as are required for its length. All
words, except possibly the first and last, contain four
characters of the string.

Program Control Data

1) Pointer

i) "Absolute" Pointer

A pointer datum is stored as an _its_ pair, i.e.,

0 17	18 29	30 35	0 17	18 29	30 35
segment number	////////	its	segment address	////////	0

which is always located at an even/odd pair of
addresses. Since this type of pointer contains
a segment number it is process dependent.

ii) "Relative" Pointer

A relative pointer is an 18 bit integer, stored
in the left half of a full word. A relative
pointer is relative to some absolute pointer and
is process independent.

2) Label and Procedure Entry

A label is represented by a block of six words. The
first two are an _its_ pair which specifies the program
point. The second two words are an _its_ pair which
identifies the generation of automatic storage appropriate

to that program point. This is in fact the value of the
stack pointer (base pair sb←sp) at the time the label was
defined. The last two words are special error-checking
information which is yet to be specified. Procedure entry
points are represented exactly like labels.

3)   Area

An area is represented by a 1-dimensional array. An
n-dimensional array of areas is represented by an
n+1-dimensional array, of which the last dimension
represents a single area.

4)   File

A file name is represented by an _its_ pair which specifies
the data origin of the file control structure for that
file.

Aggregates

1)   Array

An array is stored contiguously in row major order
(right-most subscript varying most rapidly). An array
of structures is a repetition of structures. If the
elementary data item of an array occupies more than
one word, then the elementary data items always begin
on _even_ word boundaries.

2)   Structures

The elements of a structure are stored contiguously
in the order of their declaration. If an element of
a structure is itself an aggregate, then all the
members of that aggregate taken together constitute
the storage for that element. If the origin of a
structure requires an even word boundary (i.e., the
structure is an elementary data item which occupies
more than one word), then the origins of _all_ containing
structures are also even. A maximal structure of
character or bit class (but not mixed) always begins
at a word boundary. For any two consecutive items in
a packed structure, which are not character or bit
class, the second item starts at the next boundary
natural to that element type. In an aligned structure
every item starts on a word boundary.