The convention used here and in the following paragraphs is that a capital letter indicates a field and the corresponding lower-case letter the content of that field: thus *p* is the number of the page stored in the page-number field P, and the sub-field D3 of the address field contains the decimal digit d3.

3.5 *Instruction format*

3.5.1 General description: System 25 uses 2-address instructions of the form

F, A, B

where F is the function code and A,B are operand addresses. The interpretation of the contents of the address fields depends on the contents of other fields stored in the instruction word, as follows:

LA, LB	give the lengths of the operands
AC, BC	are address markers
IA, IB	concern indexing
EIX	concerns extended indexing
IDA, IDB	concern direct/indirect addressing

These are explained in the succeeding paragraphs.

An instruction occupies 10 bytes, which we number conventionally 0 to 9 from the left. The A,B addresses, in binary coded decimal, are held in bytes 1 to 4 and 6 to 9, respectively, with the format described in Section 3.4. The decimal digits are coded in bits 4 to 1 of the respective bytes, with bit 4 as the most significant bit. They must be in the range 0 to 9; a value in the range 10 to 15 will cause a Program Check.

The detailed layout of the instruction is shown in the diagram below, where the columns are the bytes and the rows the bits within the bytes. Bit 8 of each byte is always zero and bit 6, which normally contains the inverse of bit 7, is ignored.

Byte	0	1	2	3	4	5	6	7	8	9
Bit 8	0	0	0	0	0	0	0	0	0	0
7	F3	F2	F1	F0	AC	IA1	IA0	IB1	IB0	BC
6	*	*	*	*	*	*	*	*	*	*
5	$\overline{IDA}$	$\overline{PA0}$	$\overline{PA1}$	$\overline{PA2}$	$\overline{F4}$	$\overline{IDB}$	$\overline{PB0}$	$\overline{PB1}$	$\overline{PB2}$	$\overline{EIX}$
4										
3	LA	A3	A2	A1	A0	LB	B3	B2	B1	B0
2										
1										

3.5.2 F, the Function: The function field F comprises 5 bits F0 to F4 located as follows:

F3 to F0 in bit 7 of each of bytes 0 to 3 respectively
F4, inverted, in bit 5 of byte 4

3.5.3 A and B, the Addresses (cf. Section 3.4): An address is specified by the combination of the P field (bits 5 of bytes 1-3 or 6-8) and four decimal digits (binary coded in bits 4-1 of bytes 1-4 or 6-9). The P field gives the page number in the range 0-7, stored in inverse form, and the decimal digits the adress of a byte within a page.

For the A address for example the P field is stored as follows:

bit 5 byte 1	$\overline{PA0}$	1	0	1	0	1	0	1	0
2	$\overline{PA1}$	1	1	0	0	1	1	0	0
3	$\overline{PA2}$	1	1	1	1	0	0	0	0
page number		0	1	2	3	4	5	6	7

and similarly for B.

In the decimal address, A3 is the most significant digit (i.e. the thousands digit) and A0 the least (i.e. the units digit).

3.5.4 LA and LB, the Operand Lengths: These fields are normally used to define the lengths of the operands in bytes; their contents can have values in the range 0 to 9, with the value 0 indicating a length of 10 bytes.

For some instructions the two fields are used together to give a single operand length in the range 1 to 100 bytes; in this case LA is the tens digit and LB the units, with 00 indicating a length of 100 bytes.

3.5.5 AC and BC, the Address Markers: These are used to indicate whether the address is relative to Common base or to a partition base. If AC is set to 1, the A address is in Common; if to 0, it is in the partition store and therefore relative to the current partition base.

3.5.6 IA, IB and $\overline{EIX}$: Indexing: The IA, IB fields are used to specify whether or not the relevant addresses are to be indexed and if so, which index registers are to be used. For the A address the interpretation is as follows:

IA1	IA0	Index Register
0	0	No indexing
0	1	11P to 14P
1	0	21P to 24P
1	1	31P to 34P

and similarly for B. The location of the index registers is given in Section 3.2.3.

The $\overline{EIX}$ field concerns *extended indexing*, explained in Section 3.6.1 below. This is a 1-bit field, stored in inverse form in bit 5 of byte 9. If this bit is set to 0, extended indexing is specified.

3.5.7 $\overline{IDA}$ and $\overline{IDB}$: Indirect Addressing: These fields are used to indicate whether the contents of an address field (possibly indexed) are to be used as the operand