- H. Kolsky

INSTRUCTION FORMAT

November 5, 1956

Instruction Control of LASL Computer

This memorandum will define several features of a proposed instruction control for the LASL Computer. Every attempt has been made to provide for an easy extension of the system to the input-output computer. It is hoped that essentially the same instruction format will apply to both machines, although it has not been determined that each machine will be able to execute every instruction.

Instruction Format

The general form of the instruction word is as follows:

Field No.	1	2	3	4	5	6
Field Size	1	12	6	21	12	12

The meanings assigned to the various fields are as follows:

Field No.	1.	Unassigned; for programmer uses
	2.	Operation Code
	3.	Address Interpreters
	4.	Word Address
	5.	Tag Address #1
	6.	Tag Address #2

Field #1 is a bit reserved for the use of the programmer. It has no special machine function assigned to it.

Operation Code

The bits of the operation code field will define the operations which the machine is expected to perform. This field contains bits which define the various classes of instructions. The class of an instruction may indicate a variation in the basic format described above. Some of the formats may be reserved exclusively for LINK operations; others might be limited to the LASL machine. Some formats may be interpreted by both machines.

Other bits of the operation code may be used to indicate variations in the operation code itself, such as fixed point or floating point operations. The bits of the operation code should be used in a manner which is most advantageous to the machine decodin of any particular operation.

Address Interpreters

Six bits have been allowed for specifying the manner in which the Tag addresses and the word address should be used. The modifications of the word address will be considered first. There will be three basic types of word addresses in this machine:

1. Immediate, (X) - the word address X of the instruction is used as data in the operation specified.

2. Direct, C (X) - the contents of the memory locations specified are used as data in the operation specified.

3. Indirect, C (C(X)) - the contents of the word at memory location X are used as the address of the word whose contents are to be used as data in the operation specified.

A fourth type of address will be a combination word and bit address. Since the word size is 64 bits, the low order 6 bits of the 21 bit address will designate the bit address in the word specified by the high order 15 bits of the 21 bit address field. This type of addressing will have to be implemented by indexing in order to obtain direct addressing of any bit in a one million word memory. It is expected that other formats will be available to cover complete bit addressing, but some sacrifice of flexibility might be made.

The Address Interpreter field will also control the use of the Tag fields in addition to the control of addresses described above. The control of the Tag fields as well as the Address fields will be effected through suitable decoding of the bits of the Address Interpreter field. There are several classes of use for the Tag fields.

1. Double Indexing - the two tags are used to index the Address through two indexing operations to produce an address which may be any one of the four types described above.

2. Two Address commands - at times, two address arithmetic instructions may very closely approximate the efficiency of three address instructions with reference to saving memory references for both instructions and data. In general, intermediate results may be handled more efficiently if some special commands are provided. To this end, three special uses of Tag field #2 are provided to aid in the handling of the intermediate results of a calculation.

   a) Tag field #1 is used as a normal Index Register address, but the address specified by Tag field #2 should receive the contents of the high-order (A) result register upon completion of the operation. This is sometimes known as post-storing.

b) Tag field #1 is used as a normal Index Register address, but the address specified by Tag field #2 should receive the contents of the data register (S). Thus, a number may be used in an operation and transferred to a second memory location during the completion of the operation.

c) Tag field #1 is used as a normal Index Register address, but the contents of the address specified by Tag field #2 are transferred to the high-order (A) result register prior to the start of the operation. The low-order result register (B) is cleared. At the completion of the operation, the contents of the high-order (A) result register are stored in the location specified by Tag field #2. This mode permits a two-address type operation with storage of the result in the location of the first operand.

3. Full Memory Tag Addresses - In case a very large amount of random access memory (say 1,000,000 words) is available on the machine, there will arise cases in which it is desired to use any memory location as an Index Register. To facilitate this operation, a tag larger than 12 bits is needed, and it is provided by combining the two Tag fields into one field of 24 bits. This single 24 bit field may be interpreted in three ways:

a) Direct Tag - the two Tag fields are combined into one 24 bit field. The low-order 21 bits of this field are to be used as a direct full address tag.

b) Indirect Tag - the two Tag fields are combined into one 24 bit field. The low-order 21 bits are to be used as an indirect address to specify the actual Tag address.

c) Immediate Tag - the two Tag fields are combined into one 24 bit field. The low-order 22 bits (21 bits and sign) are to be used as an immediate address to modify the word address.

4. Tag Word Address - A special word format is available for those cases which require multiple level indexing to a degree higher than that provided for in any given instruction format. In this case, a special "Tag Word" is provided. The location of the Tag Word is specified by the contents of Tag field #2. The format of the Tag Word is described later.

5. Indexing Order - If desired, an indexing operation may be applied to an address obtained by an Indirect reference to a memory location. There are two ways to use this feature:

   a) The Index Register specified by Tag field #2 will be used to index the address obtained by an Indirect reference through the effective address produced by the use of Tag field #1 as an Index Register address in the usual manner.

   b) The Index Register to be used in indexing the address obtained by an Indirect reference will be specified by Tag field #2 of the location in which the address is found. The use of Tag fields #1 and #2 in the instruction will be as defined previously.

6. Other Uses - The Address Interpreter field may also be used to indicate other meanings for the Tag and Word Address fields. For instance, Field Length, Character Size, Table Lookup and other such meanings might be specified if so desired.

A summary of the various modes of addressing that are provided by the Address Interpreter field will indicate wide range of address operations that are possible.

1. Double Indexing - Only one mode of double indexing is available. This mode uses two 12 bit tags in the normal manner. The address so produced may be any one of the four types specified.

2. Single Indexing - Two versions are available. One version with full 24 bit tags and another with 12 bit tags. If the second version is used, the remaining 12 bit Tag field will be interpreted variously, under control of the Address Interpreter field.

<u>Index Register Format</u>

Two formats are proposed for Index Registers (or Index Memory Locations) as follows:

a)

X	Compare (±) Address	Reset Quantity (±)	Address (±)
16	21 bits	21 bits	21 bits

This format contains three address-sized (20 bits) fields, each with sign, making each field 21 bits in length. The Address field will be used in the normal manner, wherein the effective address is produced by summing algebraically the Address field of the Index Register and the Word Address field of the instruction. This sum will be stored in the Address field of the Index Register, destroying the previous contents of that field.

The Compare Address field contains a singed number which is compared with the Address field after each indexing operation utilizing the specified Index Register. When equality is detected, the Address field will be restored with the contents of the Reset Quantity field instead of the effective address.

Special commands could be made available to compare the Address field with the Compare Address field and indicate Hi, Lo, or Equal conditions.

Bit X at the left end of the format will indicate whether the Index Register format is type a) as above or type b) as below.

b)

X	Count Increment (±)	Count (±)	Reset Quantity (±)	Address (±)
1b	10 bits	11 bits	21 bits	21 bits

This format like that of a) above with the exception of the Count field. The Count field is 10 bits long with another bit for sign. The Count Increment field is 9 bits long with one bit more for sign.

Each time the Address field is used in an index operation, the contents of the Count Increment field will be algebracially added to the contents of the Count field. If the result of the addition leaves a sero (plus or minus) in the Count field, the Address field will be reset with the contents of the Reset Quantity field. The Count field will remain unaltered.

Multiple Tag Words

A special word format is available for those cases which require multiple level indexing to a degree higher than that provided for in the normal instruction format. In this case a special "Tag Word", the use of which is specified in the instruction, will implement the multiple level indexing operations. The word will have a format as follows:

A	Tag	Tag	Tag	Tag	Tag
4	12	12	12	12	12

Each 12 bit Tag field can be interpreted as a tag only. The "A" field will specify the first (counting from the left) of the five Tag fields to be used. All of the tags to the right of the first one specified will be used in the normal manner. If any of the Tag fields are blank, indexing will not take place for that field.

The last two Tag fields may be combined to form one 24 bit Tag field. This form will be indicated by the "A" field. If this form is indicated by the "A" field, a maximum of four Tag fields are available.

No provision has been made for using more than one Tag Word successively. Any indexing operations of higher than is possible with one of these Tag Words will have to be extended with another instruction which specifies the next Tag Word to be used.

This proposal has the advantage of conserving memory access time (for instructions) in those cases requiring a high degree of multiple level indexing.

STRETCH Product Planning Group

J. E. Griffith

B. L. Sarahan

D. W. Sweeney

JEG:mms