Session-03

Instruction Formats of IBM-360

Instruction formats of IBM 360

- Instruction length is variable.
- Some instructions cause no reference to main storage; others cause one or more references to main storage.
- To conserve storage space and save time in instruction execution, instruction length is variable and can be one, two, or three half words.
- Instructions specify the operation to be done and the location of data. Data may be located in main storage, registers, or a combination of the two.
- Instruction length is related to the number of storage addresses necessary for the operation. As a result, instructions will be of different lengths depending on the location of data.

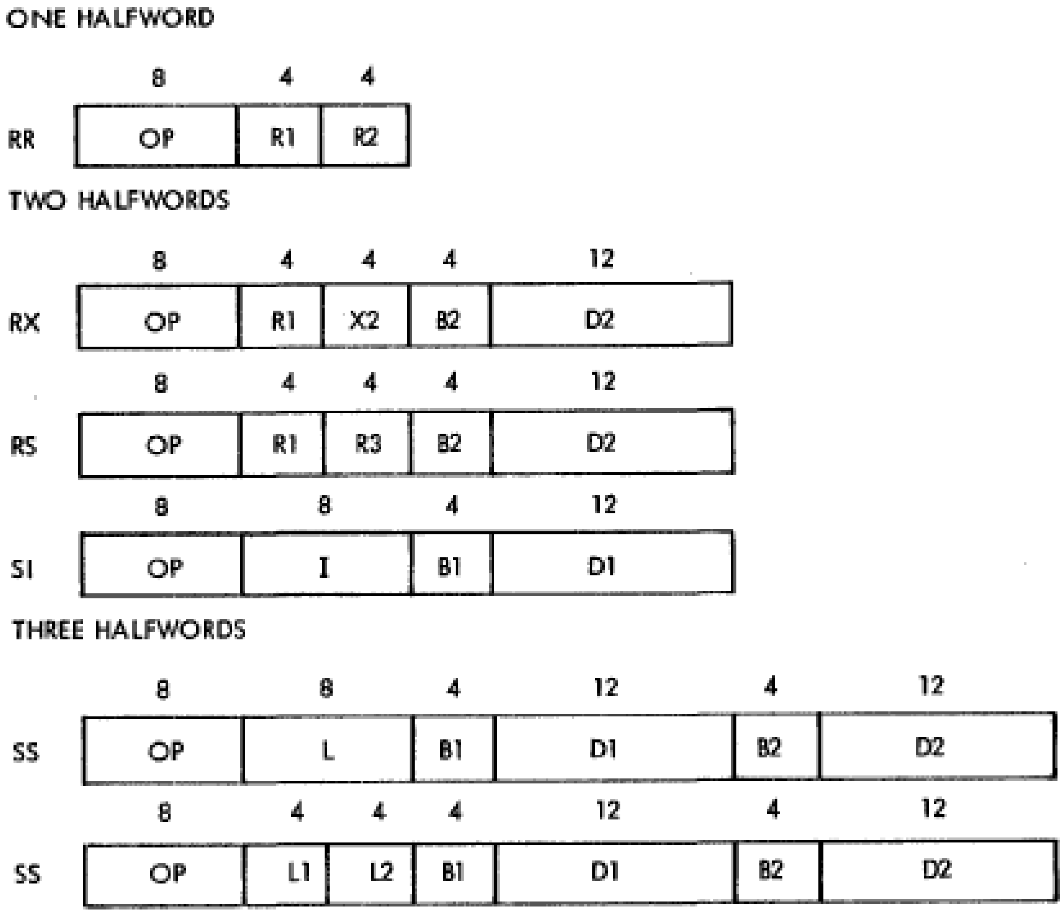


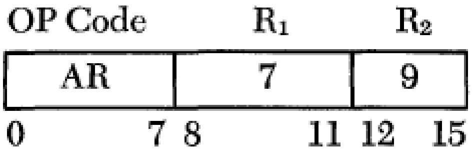
Fig 2. Instruction Formats

- **Figure 2** shows the five basic instruction formats. The format codes are:
  - RR denotes a register-to-register operation;

- RX, a register... to-indexed storage operation;
- RS, a register-to-storage operation;
- SI, a storage and: immediate operand operation; and
- SS, a storage-to-storage operation.

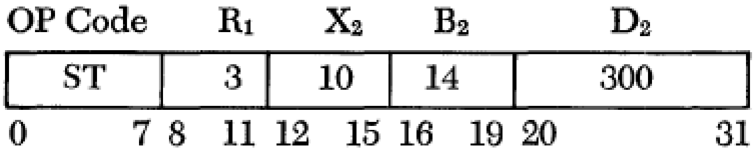
- R1 specifies the address of the register containing the first operand. The second operand location, if any, is defined differently for each format.
- In the RR format, the R2 field specifies the address of the general register containing the second operand.
- In the RX format, the contents of the general registers specified by the X2 and B2 fields are added to the contents of the D2 field to form an address designating the storage location of the second operand.
- In most cases the results replace the first operand except for the Store instruction, and then Convert to Decimal instruction, where the result replaces the second operand. The contents of all registers and storage locations participating in the addressing or execution part of an operation remain unchanged, except for the storing of the final result.
- In the following examples of the instruction formats, the operands are expressed as decimal numbers, and the operation codes are expressed in the symbolic assembly language.

- **RR Format**



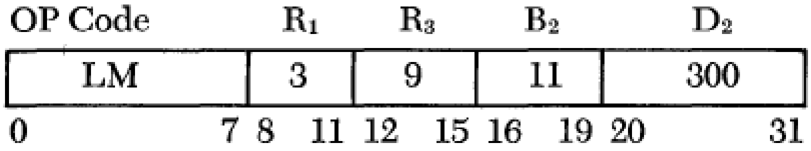
  - Execution of this Add instruction adds the contents of general register 9 to the contents of general register 7, and the sum is placed in general register 7.

- **RX Format**



  - Execution of this Store instruction stores the contents of general register 3 at a main storage location addressed by the sum of 300 and the low-order 24 bits of general registers 14 and 10.

- **RS Format**

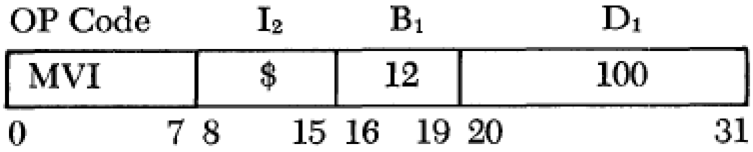


  - This Load Multiple instruction causes the set of general registers starting with the register specified by R1 and ending with the register specified by R3 to be loaded from the locations designated by the second operand address.
  - The storage area from which the contents of the general registers are obtained starts at the location designated by the second operand address and continues

through as many words as needed. The general registers are loaded in the ascending order of their addresses, starting with the register specified by R1 and continuing up to and including the register specified by R3

- It was pointed out earlier that the storing and restoration of registers is a relatively simple matter. There is also a multiple store instruction that provides for the storing of the registers, while this multiple load instruction provides for their restoration.

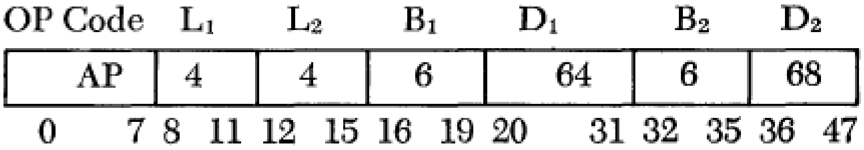
- **SI Format**



- With this Move Immediate instruction in the example shown, a dollar sign ($) is to be placed in location 2100, leaving locations 2101-2104 unchanged. Let Z represent a four-bit zone. Assume that:

Register 12 contains	00	00	20	00	
Location 2100-2104 (before)	Z0	ZI	Z2	Z3	Z0
Locations 2100-2104 (after)	$	ZI	Z2	Z3	Z0

- **SS Format**



- With this Add Decimal instruction, the second operand is added to the first operand, and the sum is placed in the first operand location. If necessary, high order zeros are supplied for either operand. Note that in the register-to-register (RR) instruction example, the addition is on fixed-length binary fields. The decimal arithmetic instruction in the
- SS format operates on data in the packed format with two decimal digits placed in one eight-bit byte. The length of the fields is specified explicitly in the instruction rather than implied in the operation code.

- In each format (RR, RX, RS, SI, or SS) the first byte contains the operation code in the binary code, which is the actual machine language. In binary, the length and format of an instruction are specified by the first two bits of the operation code.

BIT POSITION	**INSTRUCTION LENGTH**	**INSTRUCTION FORMAT**
00	One halfword	RR
01	Two halfwords	RX
10	Two halfwords	RS or SI
11	Three halfwords	SS

- During instruction decoding, the processing unit examines these first two bits of the operation code and determines how many bytes to fetch for this instruction. These bit configurations are part of the machine instruction, so that when, for example, we specify an Add register-to-register instruction, we are not concerned with specifying the instruction length. We have seen that for fixed-length instructions the length of the operand is implicit in the instruction, and for variable-length operands the length is specified in the instruction. We have also seen that the length of the instruction itself is part of the operation code.

Reference:

System Programming by Johan Donovan