System Software
CAT-204

Design By:
Prof. Pawandeep Sharma
A.P
Chandigarh University-Gharuan

Syllabus

UNIT-I

Introduction to System Software: Machine Structure, evolution of operating system, machine language.

Assembler: Elements of Assembly Language Programming, General design procedure, design of a Two Pass Assemblers, A Single Pass Assemblers Design.

Table Processing: Searching & Sorting.

Syllabus

UNIT-II

Macro and Macro Processors: Macro instructions, Features of a macro Facility: macro Instruction arguments, Conditional macro expansion, Macro calls within macros, Macro instruction defining macros, Advanced Macro Facilities, Implementation of simple macro processor, Two-pass algorithm, Implementation of macro calls within macros, Implementation within an assembler.

Linkers – Translated linked and load time addresses, relocation and linking concepts, Design of a linker, self relocating programs.

Syllabus

UNIT-III

Loaders: Loader scheme, absolute loaders, Subroutine linkages, Relocating loaders, Direct linking loaders, binders, linking loaders, overlays, Dynamic Binders, Design of an Absolute Loader, Design of a Direct-Linking Loader. Compilers: Phases of Compiler Construction, Symbol Table, Top-down and bottom-up Parsing, Operator-Precedence Parsing, LR Parsers, Code Generation and Code Optimization, Memory management, Design & other issues.

Lecture Outline

- **Overview of Software**
- **System Programming**
- **Relation of Programming Languages**

An Overview of Software

- **Computer programs:** sequences of instructions for the computer
- **Documentation:** describes program functions
- Software – system software and application software

Application Software

- **Application software:** helps users solve particular problems
- In most cases, application software resides on the computer’s hard disk
- Application software can also be stored on CDs, DVDs, and flash or keychain storage devices

Systems Software

- Systems software
  - Controls operations of computer hardware
  - Supports application programs' problem-solving capabilities
- Types of systems software
  - Operating systems
  - Utility programs

System Programming

- Design and Implementation of system software.
- System Software: a variety of programs supporting the operation of a computer.
- Typical system programs: OS, Complier, Assembler (Linker, Loader, Macro Processors), Text Editor, Debugger, …,
  - Their functions and relations among them?
- Why specific and important:
  - rely on hardware (computer architecture)
  - need related low level languages such as assembly language
  - Free users from knowing the detail of machines.

Relations?

- Programming Languages:
  - Machine Language (ML)→ Assembly Language (AL)→ High-Level Programming Language (HL)
    - ML: machine code, i.e, binary code, e.g., 01100110
    - AL: mnemonic instructions, e.g., STO (Store)
    - HL: statements, e.g. if … Then … else …
  - Programs in HL → Object Code (in AL or ML)
    - (Compiler)
  - Programs in AL → Object Code (in ML)
    - (Assembler)

Relations?

- Type and modify a program in HL or AL (by an editor)
- Translate it into object code in ML (by a compiler or assembler)
- Load it into memory for execution (by a linker and/or loader)
- Execute it on CPU (by an OS)
- Detect errors in the code (by a debugger)
- All the processes and programs are run under and controlled by an OS such as Windows or Linux.
- IDE (Integrated Development Environment)

System Software and Machine Architecture

- One main characteristic in which system software differs from application software: **machine dependence**
  - Assembler: instruction format, addressing modes
  - Compiler: registers (number, type), machine instructions
  - OS: all of the resources of a computing system.
- Of course, **some aspects** of system software are machine-independent.
  - General design and logic of a assembler
  - Code optimization in a compiler
  - Linking of independently assembled subprograms.

Conclusion

- Every software is a group of programs and related documentation.
- Typical system programs include OS, Complier, Assembler (Linker, Loader, Macro Processors), Text Editor, Debugger etc.
- Almost every System software is machine dependent however, few aspects of system software are machine independent also.

References

BOOKS:-

- Computer Fundamentals, P. K. Sinha.
- System Programming, Dhamdhere

Web Link:-

- http://nptel.ac.in/courses/106106092/2
- https://www.youtube.com/watch?v=VG9VopzV_T0
- http://whatis.techtarget.com/definition/system-software
- http://searchdatacenter.techtarget.com/definition/assembler
- http://www.icse.s5.com/notes/m2.html

Queries???

Thank You

UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

System Software
CAT-204

Design By:
Prof. Pawandeep Sharma
A.P
Chandigarh University-Gharuan

Syllabus

UNIT-I

Introduction to System Software: Machine Structure, evolution of operating system, machine language.

Assembler: Elements of Assembly Language Programming, General design procedure, design of a Two Pass Assemblers, A Single Pass Assemblers Design.

Table Processing: Searching & Sorting.

Syllabus

UNIT-II

Macro and Macro Processors: Macro instructions, Features of a macro Facility: macro Instruction arguments, Conditional macro expansion, Macro calls within macros, Macro instruction defining macros, Advanced Macro Facilities, Implementation of simple macro processor, Two-pass algorithm, Implementation of macro calls within macros, Implementation within an assembler.

Linkers – Translated linked and load time addresses, relocation and linking concepts, Design of a linker, self relocating programs.

Syllabus

UNIT-III

Loaders: Loader scheme, absolute loaders, Subroutine linkages, Relocating loaders, Direct linking loaders, binders, linking loaders, overlays, Dynamic Binders, Design of an Absolute Loader, Design of a Direct-Linking Loader. Compilers: Phases of Compiler Construction, Symbol Table, Top-down and bottom-up Parsing, Operator-Precedence Parsing, LR Parsers, Code Generation and Code Optimization, Memory management, Design & other issues.

General Machine Structure

History of IBM System/360/370

- IBM System/360 was announced in April 1964.
- System/360 was the first major product line designed for both business and scientific need.
- IBM System/370 was announced in June 1970.
- System/370 had an execution time 3 to 5 times faster than system/360 model 50 and 60.

The General machine structure

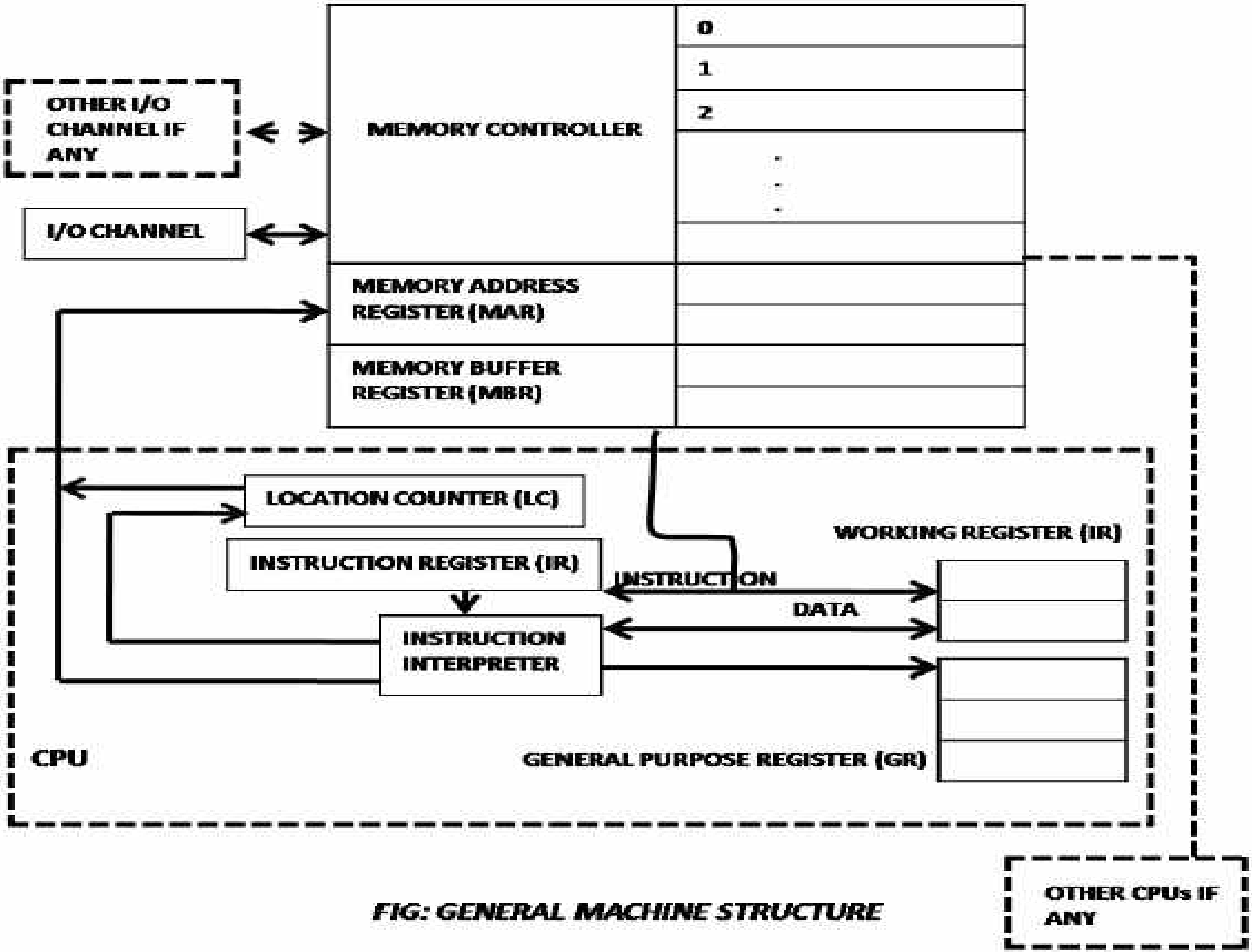


FIG: GENERAL MACHINE STRUCTURE

Explanation of GMS

- Instruction interpreter :
  - It is a group of electrical circuits that performs the purpose of the instructions fetched from the memory.
  - It is like a decoder that decodes the type of the instruction.
- Location counters (LC) :
  - It is also known as program counter or instruction counter which holds the location of the current instructions being executed
- Instruction registers (IR) :
  - It contains a copy of the current instruction that is executed.
- Working registers (WR) :
  - WR are memory devices that serves as „SCRATCH PAD‟S‟ for instruction interpreter.
  - WR are general purpose registers.

Explanation of GMS

- General purpose register :
  - GPR‟S are used by the programmer as storage locations and for special functions.
- Memory address registers (MAR):
  - It contains the address of memory location that is to be read from or stored into.
- Memory buffer register (MBR)
  - It contains a copy of the designated memory location specified by the MAR, after read or write.

Explanation of GMS

- Memory controller
  - It is hardware that transfers data between the MBR and the core memory location the address of which in the MAR
- I/O channels
  - It may be through of as separate computers which interpret special instructions for inputting and outputting information from the memory.

More concepts in machine structure (same for IBM 360)

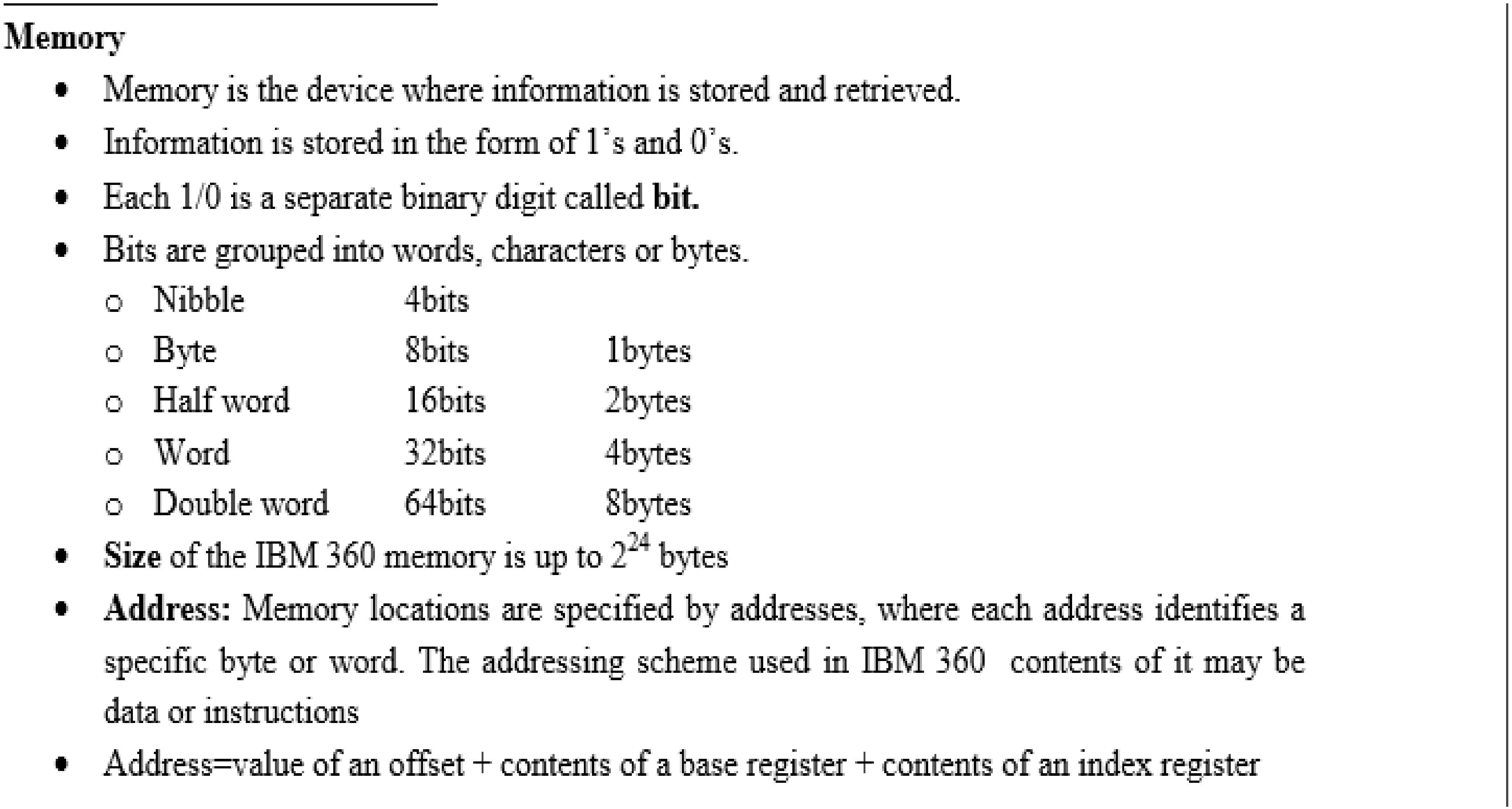
- Memory
- Registers
- Data Formats
- Instructions
- Instruction Formats
- Instruction Sets

UNIVERSITY INSTITUTE of COMPUTING

Asia's Fastest Growing University

CU CHANDIGARH UNIVERSITY

More concepts in machine structure (same for IBM 360)



Memory

- Memory is the device where information is stored and retrieved.
- Information is stored in the form of 1's and 0's.
- Each 1/0 is a separate binary digit called **bit.**
- Bits are grouped into words, characters or bytes.

Nibble	4bits	
Byte	8bits	1bytes
Half word	16bits	2bytes
Word	32bits	4bytes
Double word	64bits	8bytes

- **Size** of the IBM 360 memory is up to 2^{24} bytes
- **Address:** Memory locations are specified by addresses, where each address identifies a specific byte or word. The addressing scheme used in IBM 360 contents of it may be data or instructions
- Address=value of an offset + contents of a base register + contents of an index register

More concepts in machine structure (same for IBM 360)

- Registers

Types of register are :

Name of the register	number	size
• General purpose registers	16	32bits each
• Floating point registers	4	64bits each
• Program status word	1	64bits

- General purpose registers
- IBM 360 machine has 16 general purpose registers denoted by R0 to R15.
- The general purpose register may be used for various arithmetic and logical operations .

More concepts in machine structure (same for IBM 360)

- Floating Point registers
- They are same as general purpose registers with a difference that they are specifically used to handle floating point data.
- Program status word(PSW)
- It contains the value of the location counter, protection information and interrupts status which is 64bits in length.

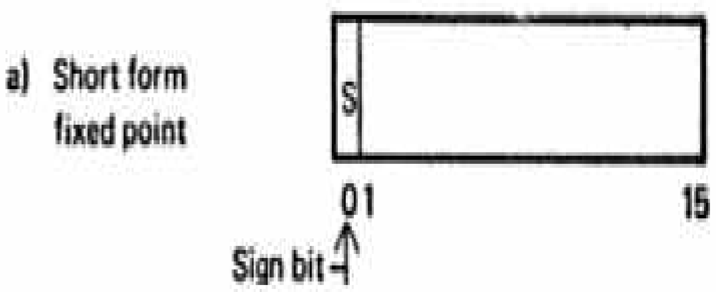
More concepts in machine structure (same for IBM 360)

Data Formats

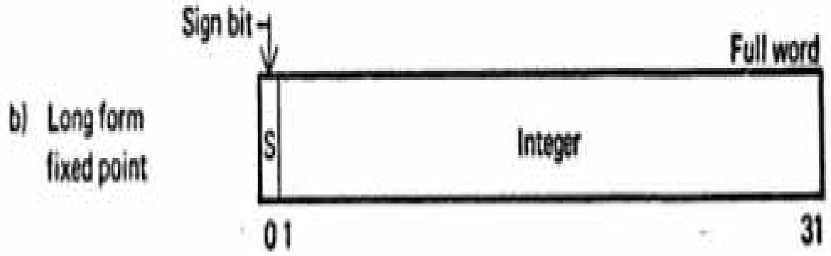
- Short form fixed point numbers
- Long form fixed point numbers
- Short form floating point number
- Long form floating point numbers

More concepts in machine structure (same for IBM 360)

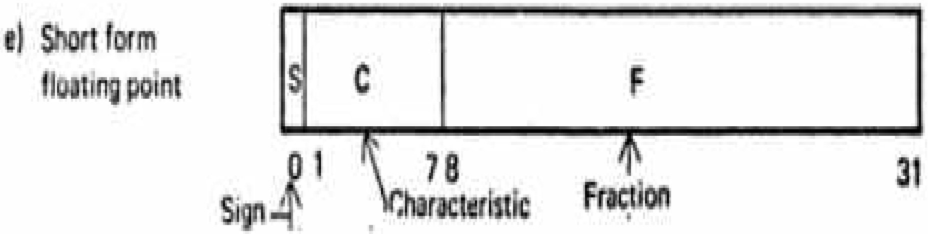
.Short form fixed point numbers



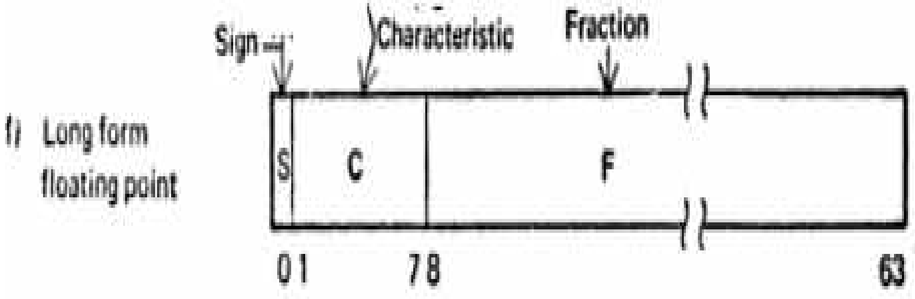
. Long form fixed point numbers



. Short form floating point number



Long form floating point numbers



More concepts in machine structure (same for IBM 360)

Instructions

- An instruction includes an opcode, specifying the operation to be performed, such as “add contents of memory to register” and zero or more operands, which may specify registers, memory locations or literal data.
- The operand may have addressing modes determining their meaning or may be in fixed fields.
- The size or length of an instruction varies from 2 bytes to 6 bytes depending on the instruction formats.
- Instructions are of different types they are
  - Arithmetic instructions
  - Control or transfer instructions
  - Special interrupt instruction

More concepts in machine structure (same for IBM 360)

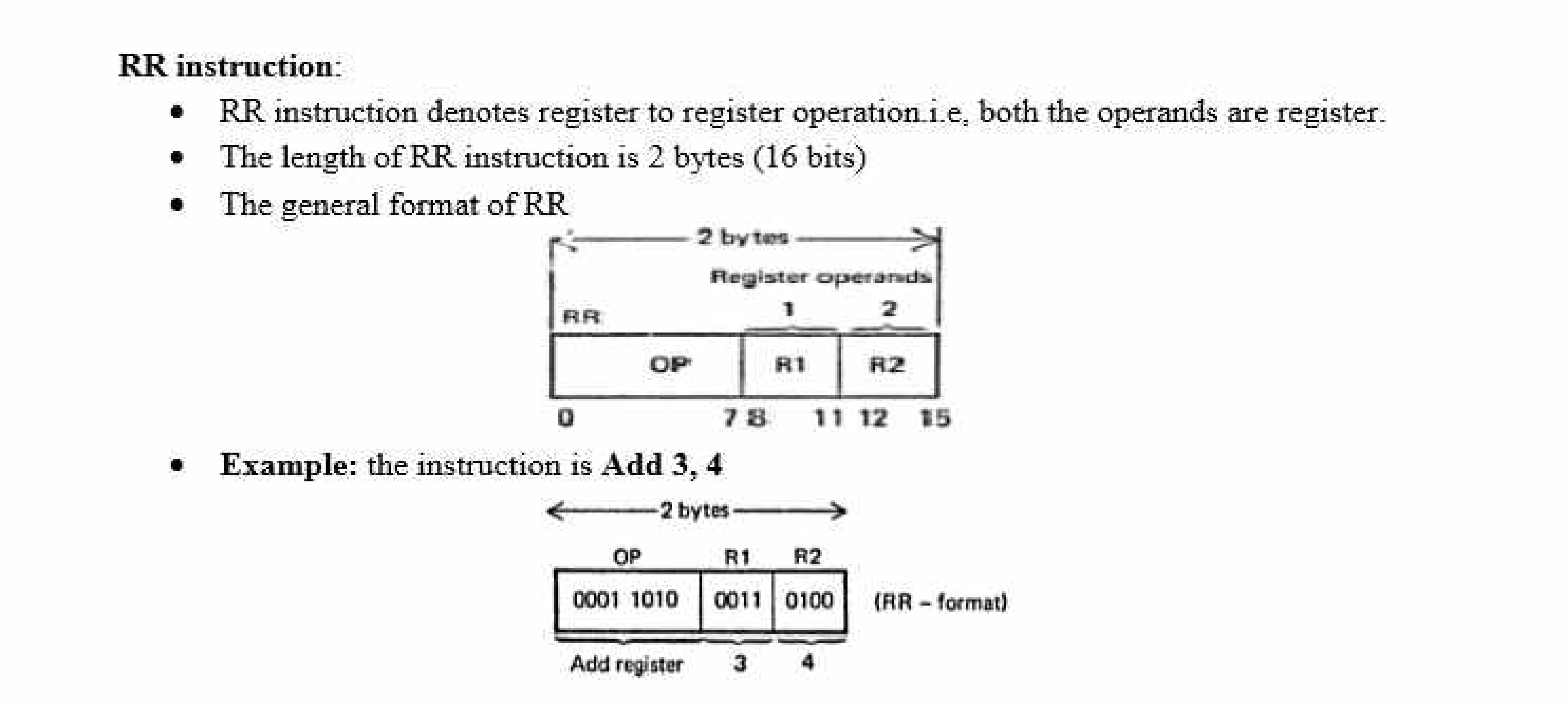
- The different types of operands are :
- Register operands : Refer to data stored in one of the 16 general purpose registers, which are addressed by 4 bit field in the instruction.
- Storage operands : Refers to data stored in core memory. The length of the operand depends upon the specific data type.
- Immediate operands : These are single byte of data and are stored as part of the instruction.

More concepts in machine structure (same for IBM 360)

- Instruction formats

- The different instruction formats are :
- RR format
- RX format
- RS format
- SI format
- SS format

More concepts in machine structure (same for IBM 360)



RR instruction:

- RR instruction denotes register to register operation.i.e, both the operands are register.
- The length of RR instruction is 2 bytes (16 bits)
- The general format of RR

2 bytes

RR

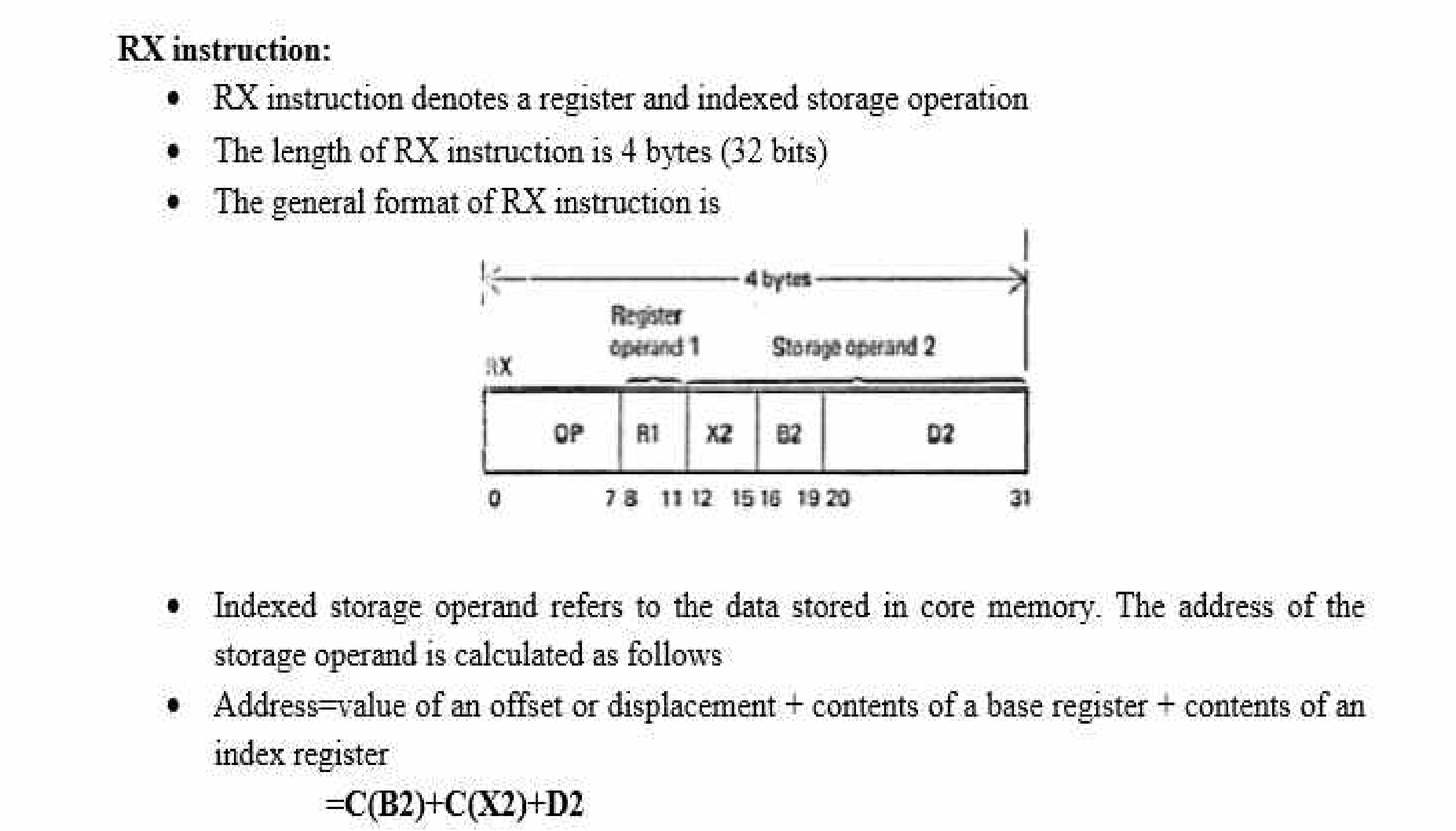
	Register operands 1	 2
OP	R1	R2
0 7	8 11	12 15

- **Example:** the instruction is **Add 3, 4**

2 bytes

OP	R1	R2	
0001 1010	0011	0100	(RR – format)
Add register	3	4	

More concepts in machine structure (same for IBM 360)



RX instruction:

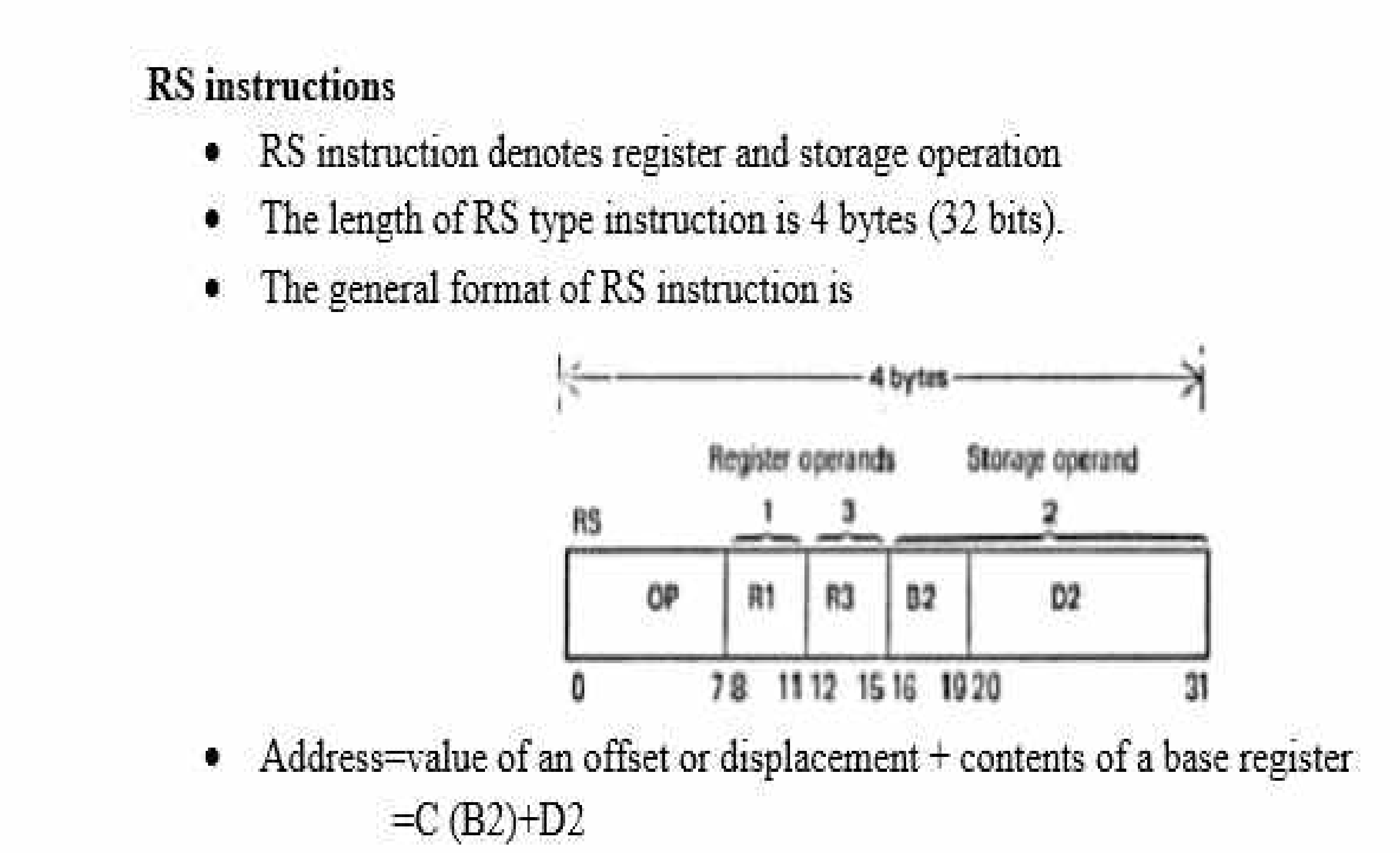
- RX instruction denotes a register and indexed storage operation
- The length of RX instruction is 4 bytes (32 bits)
- The general format of RX instruction is

4 bytes
Register operand 1
Storage operand 2
RX
OP
R1
X2
B2
D2
0
7 8
11 12
15 16
19 20
31

- Indexed storage operand refers to the data stored in core memory. The address of the storage operand is calculated as follows
- Address=value of an offset or displacement + contents of a base register + contents of an index register

=C(B2)+C(X2)+D2

More concepts in machine structure (same for IBM 360)



RS instructions

- RS instruction denotes register and storage operation
- The length of RS type instruction is 4 bytes (32 bits).
- The general format of RS instruction is

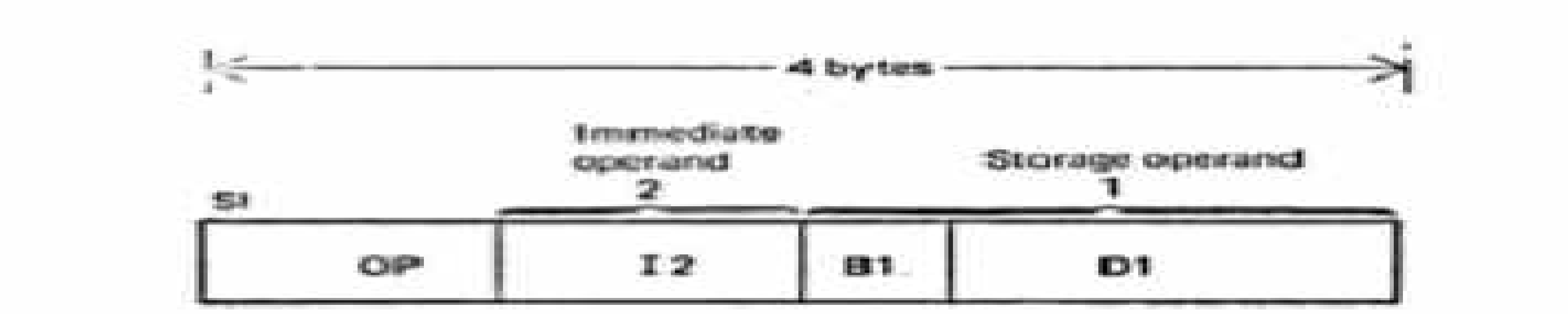
4 bytes
Register operands
Storage operand
RS
1
3
2
OP
R1
R3
B2
D2
0
7 8
11 12
15 16
19 20
31

- Address=value of an offset or displacement + contents of a base register

  =C (B2)+D2

More concepts in machine structure (same for IBM 360)

- **SI instruction**
- SI instruction denotes storage and immediate operand operation Immediate operands are single byte of data and stored as part of the instruction
- The length of SI type instruction is 4 bytes (32 bits).
- The general format of SI instruction is :

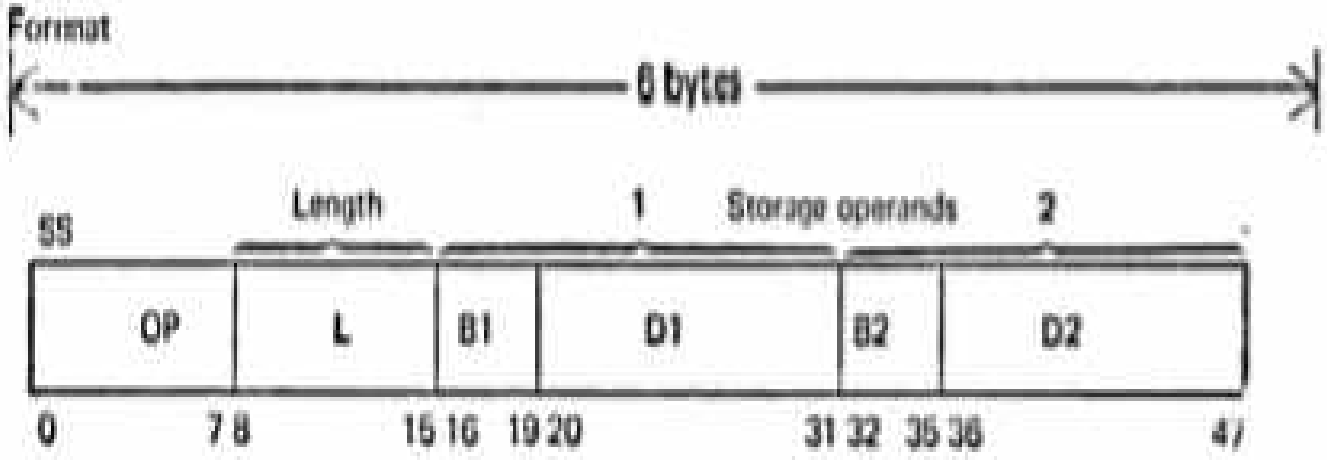


- Address=value of an offset or displacement + contents of a base register =C (B1) +D1

More concepts in machine structure (same for IBM 360)

SS instruction

- SS instruction denotes a storage to storage operation
- The length of SS instruction is 6 bytes(48 bits)
- The general format is



More concepts in machine structure (same for IBM 360)

- **Instruction set**

The various categories of instructions are:

1. load-store registers instructions

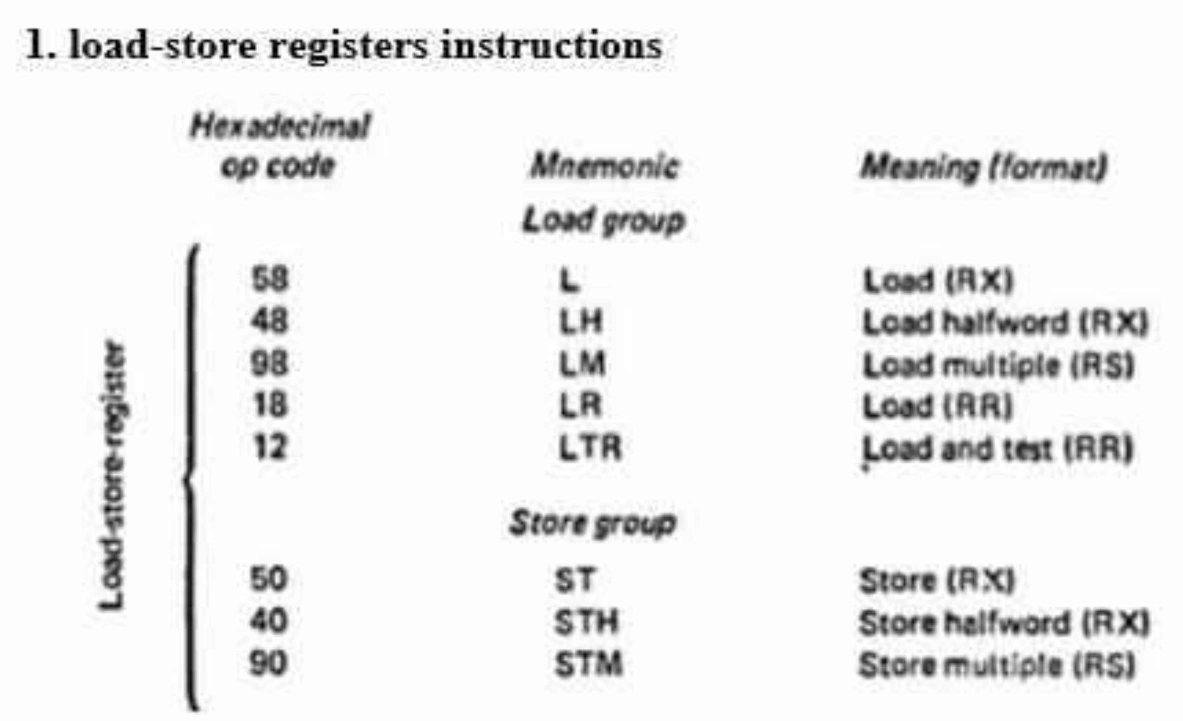
2. Fixed point arithmetic

3. Logical instructions

4. Transfer instructions

5. Miscellaneous instructions

More concepts in machine structure (same for IBM 360)

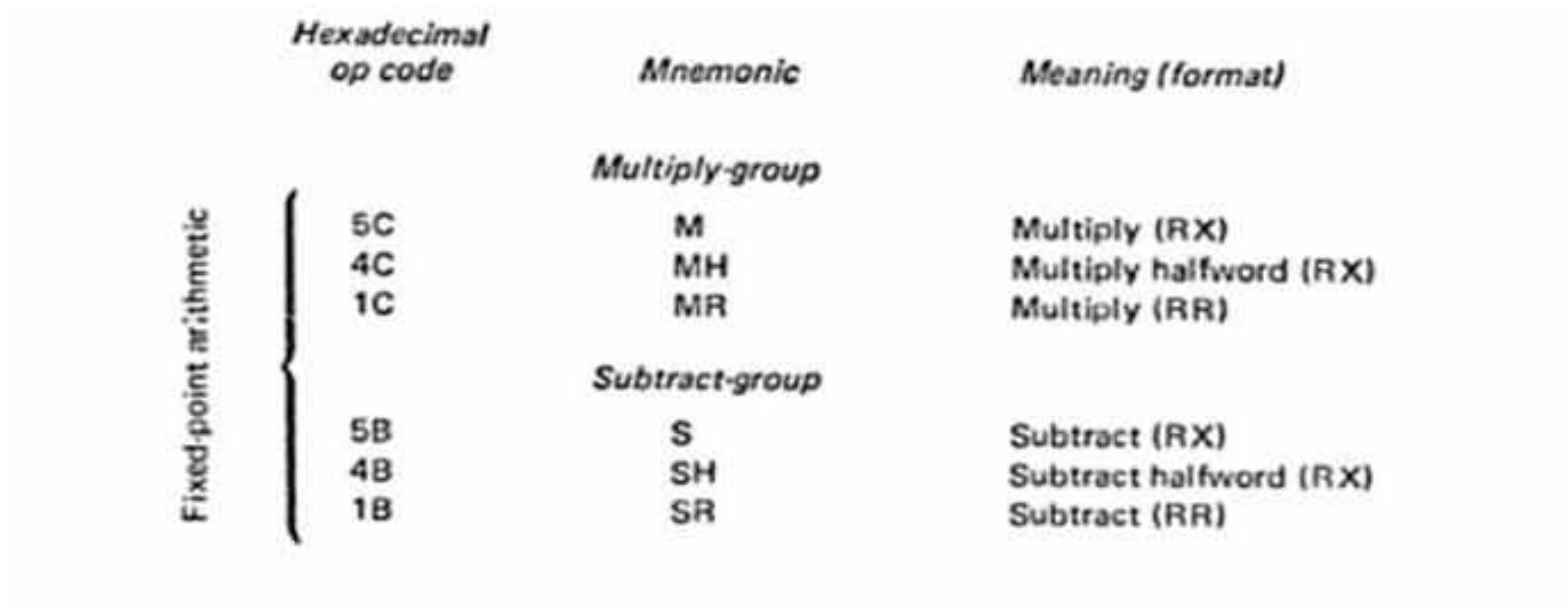


1. load-store registers instructions

	Hexadecimal op code	*Mnemonic*	*Meaning (format)*
		Load group	
Load-store-register	58	L	Load (RX)
	48	LH	Load halfword (RX)
	98	LM	Load multiple (RS)
	18	LR	Load (RR)
	12	LTR	Load and test (RR)
		Store group	
	50	ST	Store (RX)
	40	STH	Store halfword (RX)
	90	STM	Store multiple (RS)

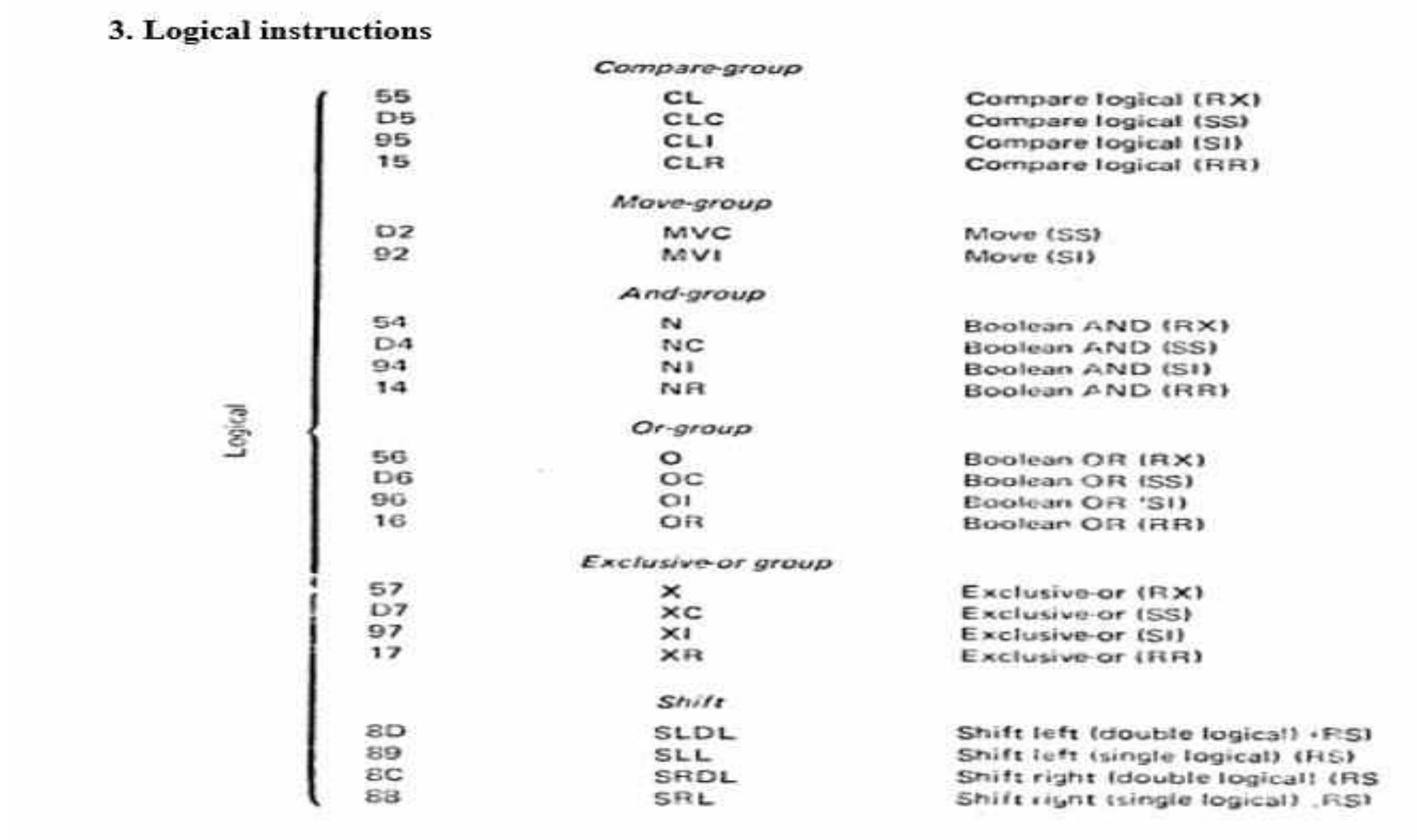
More concepts in machine structure (same for IBM 360)

2. Fixed Point Arithmetic



	Hexadecimal op code	*Mnemonic*	*Meaning (format)*
Fixed-point arithmetic		*Multiply-group*	
	5C	M	Multiply (RX)
	4C	MH	Multiply halfword (RX)
	1C	MR	Multiply (RR)
		Subtract-group	
	5B	S	Subtract (RX)
	4B	SH	Subtract halfword (RX)
	1B	SR	Subtract (RR)

More concepts in machine structure (same for IBM 360)

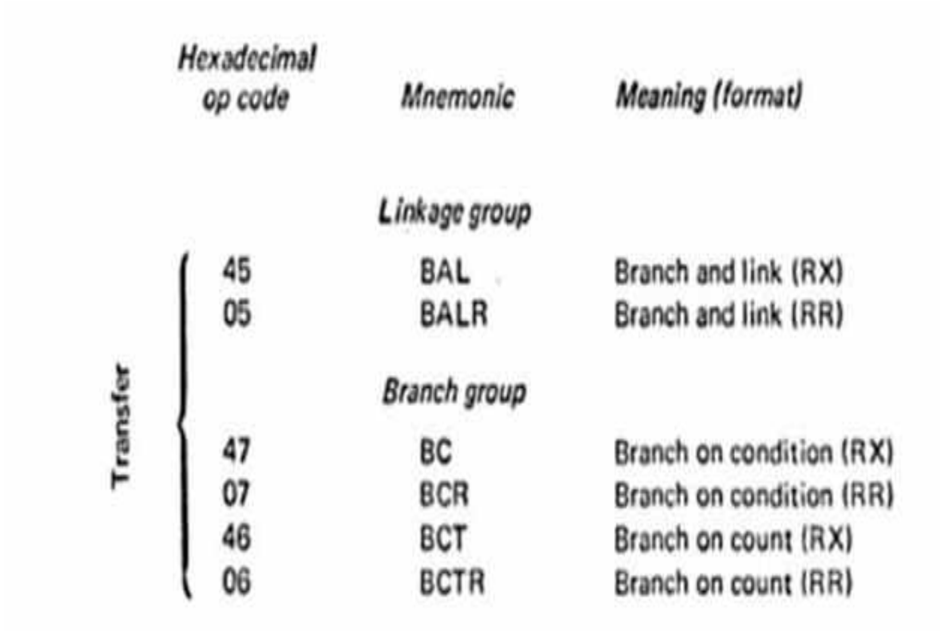


3. Logical instructions

Logical		*Compare-group*	
	55	CL	Compare logical (RX)
	D5	CLC	Compare logical (SS)
	95	CLI	Compare logical (SI)
	15	CLR	Compare logical (RR)
		Move-group	
	D2	MVC	Move (SS)
	92	MVI	Move (SI)
		And-group	
	54	N	Boolean AND (RX)
	D4	NC	Boolean AND (SS)
	94	NI	Boolean AND (SI)
	14	NR	Boolean AND (RR)
		Or-group	
	56	O	Boolean OR (RX)
	D6	OC	Boolean OR (SS)
	96	OI	Boolean OR (SI)
	16	OR	Boolean OR (RR)
		Exclusive-or group	
	57	X	Exclusive-or (RX)
	D7	XC	Exclusive-or (SS)
	97	XI	Exclusive-or (SI)
	17	XR	Exclusive-or (RR)
		Shift	
	8D	SLDL	Shift left (double logical) (RS)
	89	SLL	Shift left (single logical) (RS)
	8C	SRDL	Shift right (double logical) (RS
	88	SRL	Shift right (single logical) (RS)

More concepts in machine structure (same for IBM 360)

4. Transfer instructions



	Hexadecimal op code	Mnemonic	Meaning (format)
Transfer		*Linkage group*	
	45	BAL	Branch and link (RX)
	05	BALR	Branch and link (RR)
		Branch group	
	47	BC	Branch on condition (RX)
	07	BCR	Branch on condition (RR)
	46	BCT	Branch on count (RX)
	06	BCTR	Branch on count (RR)

References

Book:-

•System Programming, John J. Donovan, Chapter 2, Page No.21.
•System Programming, Dhamdhere

Queries???

Thank You