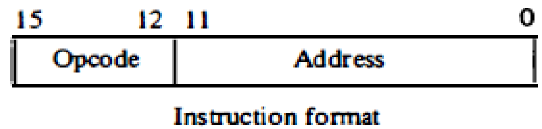
Content

- GENERAL-PURPOSE COMPUTER
- DATA PATH AND CONTROL
  - REGISTERS
  - BASIC COMPUTER INSTRUCTIONS
  - COMMON BUS SYSTEM
  - MEMORY

General-purpose computer



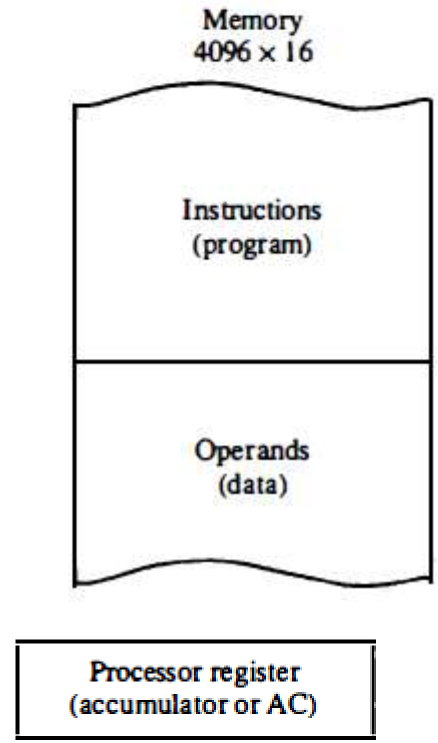


Figure 1 General purpose computer [1]

- An instruction code is a group of bits that instruct the computer to perform a specific operation. It is usually divided into parts.
- Instructions are stored in one section of memory and data in another.
- The memory has 4096 words
  - $4096 = 2^{12}$, so it takes 12 bits to specify an Address to select a word in memory
- Each word is 16 bits long
- 16-bit memory word, we have available four bits for the operation code specify one out of 16 possible operations, and 12 bits to specify address of an operand.
- Computers that have a single-processor register called accumulator and label it AC.
- The operation is performed with the memory operand and the content of AC.

Central Processing Unit (CPU)

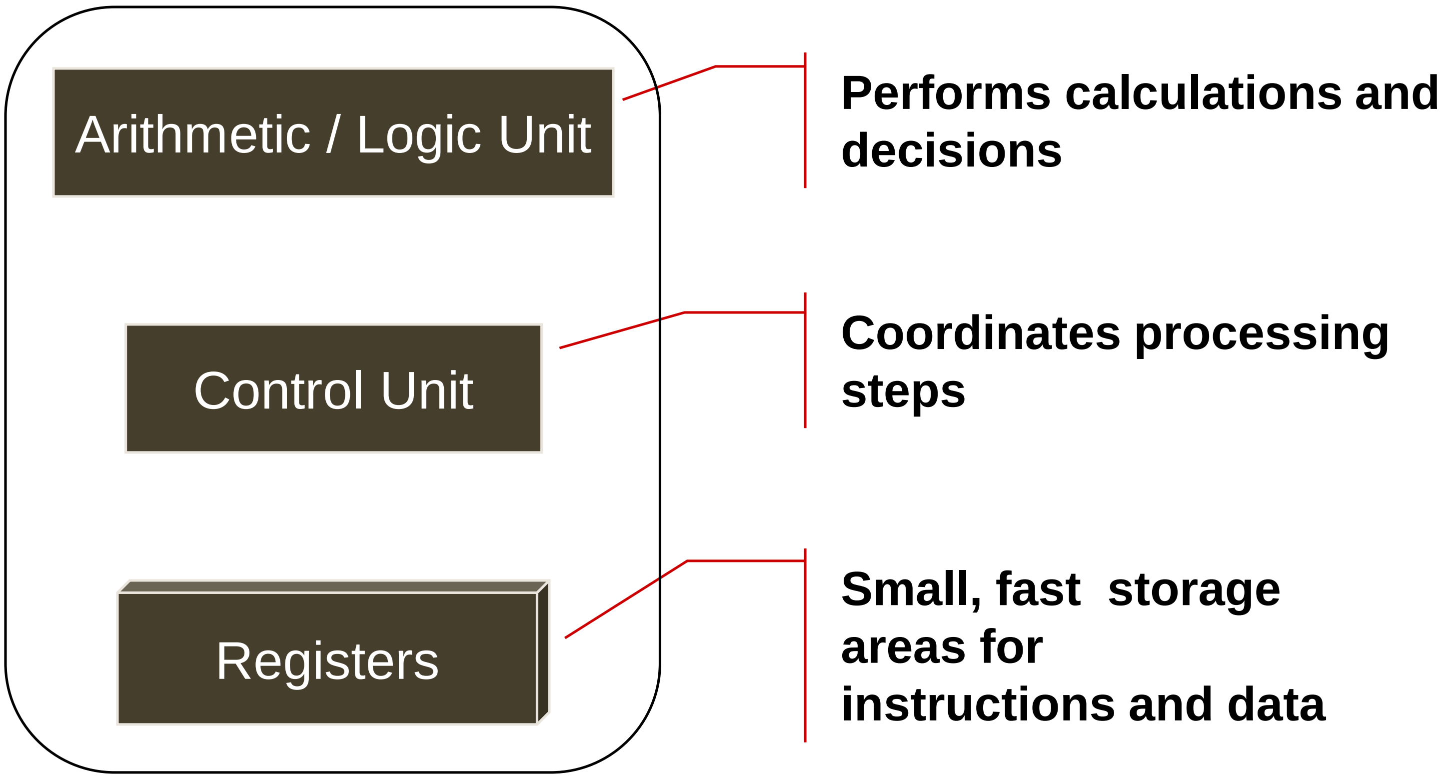


Figure 2 Central processing unit

Data Path and Control

- The CPU can be divided into a data section and a control section. The data section, which is also called the datapath.

- **Datapath**: Components of the processor that perform arithmetic operations and holds data. The registers, the ALU, and the interconnecting bus are collectively referred to as the datapath. The datapath is capable of performing certain operations on data items.

- **Control section** :Component of the processor which is basically the control unit, that issues control signals to the datapath. It commands the datapath, memory, I/O devices according to the instructions of the memory.

Registers

- Registers are very fast computer memory which are used to execute programs and operations efficiently. there are several different classes of CPU registers which works in coordination with the computer memory to run operations efficiently.[1]
- **Accumulator:** This is the most frequently used register used to store data taken from memory.
- **Memory Address Registers (MAR):** It holds the address of the location to be accessed from memory. MAR and MDR (Memory Data Register) together facilitate the communication of the CPU and the main memory.
- **Memory Data Registers (MDR):** It contains data to be written into or to be read out from the addressed location.
- **General Purpose Registers: (GPRs):** These are numbered as R0, R1, R2….Rn, and used to store temporary data during any ongoing operation. Its content can be accessed by assembly programming.
- **Instruction Register (IR):**
  It is the register which holds the instruction which is currently been executed.

Registers

- Program Counter (PC): is used to keep the track of execution of the program. It contains the memory address of the next instruction to be fetched. PC points to the address of the next instruction to be fetched from the main memory when the previous instruction has been successfully completed.

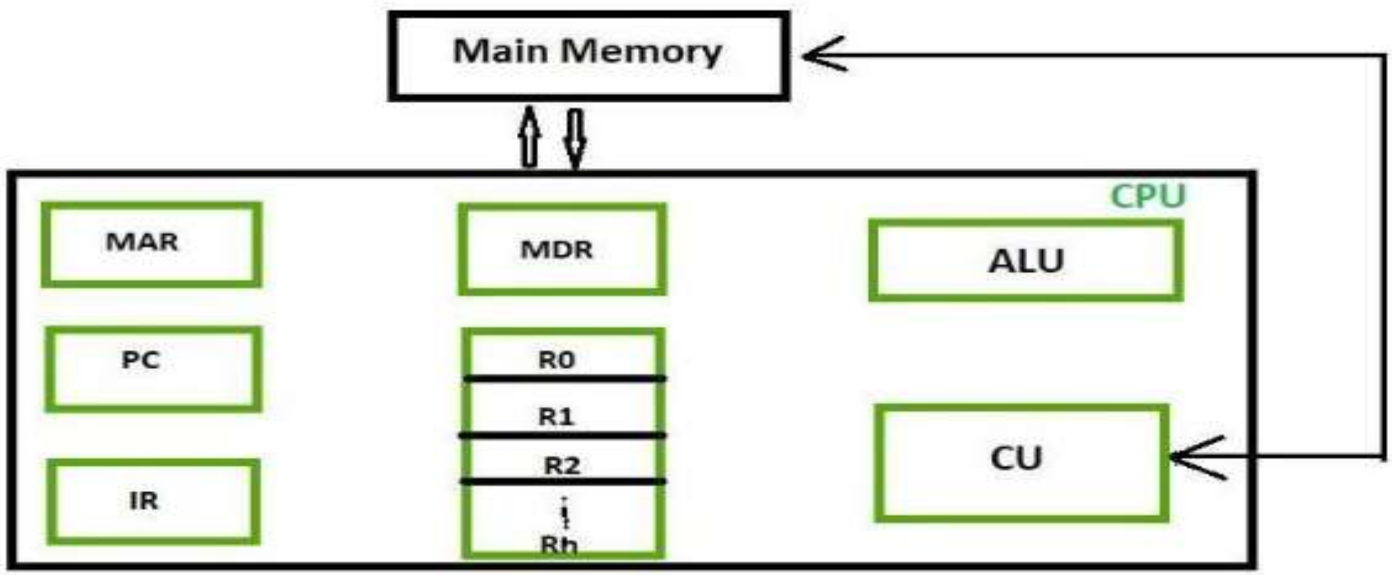


Figure 3 processor registers

PROCESSOR REGISTERS

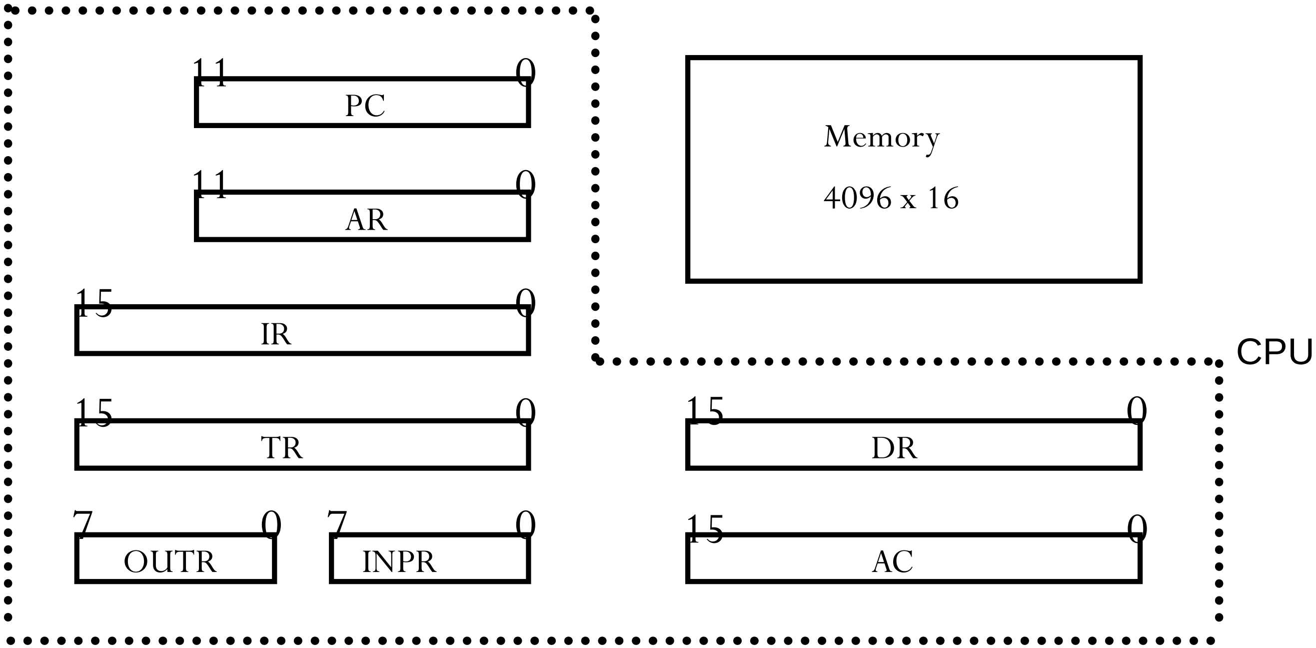
- A processor has many registers to hold instructions, addresses, data, etc
- The processor has a register, the *Program Counter* (PC) that holds the memory address of the next instruction to get
  - Since the memory in the Basic Computer only has 4096 locations, the PC only needs 12 bits
- In a direct or indirect addressing, the processor needs to keep track of what locations in memory it is addressing: The *Address Register* (AR) is used for this
  - The AR is a 12 bit register in the Basic Computer
- When an operand is found, using either direct or indirect addressing, it is placed in the *Data Register* (DR). The processor then uses this value as data for its operation
- The Basic Computer has a single *general purpose register* – the *Accumulator* (AC)

PROCESSOR REGISTERS

- The significance of a general purpose register is that it can be referred to in instructions
  - e.g. load AC with the contents of a specific memory location; store the contents of AC into a specified memory location
- Often a processor will need a scratch register to store intermediate results or other temporary data; in the Basic Computer this is the *Temporary Register* (TR)
- The Basic Computer uses a very simple model of input/output (I/O) operations
  - Input devices are considered to send 8 bits of character data to the processor
  - The processor can send 8 bits of character data to output devices
- The *Input Register* (INPR) holds an 8 bit character gotten from an input device
- The *Output Register* (OUTR) holds an 8 bit character to be send to an output device

BASIC COMPUTER REGISTERS

Registers in the Basic Computer



List of BC Registers

DR	16	Data Register	Holds memory operand
AR	12	Address Register	Holds address for memory
AC	16	Accumulator	Processor register
IR	16	Instruction Register	Holds instruction code
PC	12	Program Counter	Holds address of instruction
TR	16	Temporary Register	Holds temporary data
INPR	8	Input Register	Holds input character
OUTR	8	Output Register	Holds output character

Figure 4 Basic computer registers [1]

GENERAL REGISTER ORGANIZATION

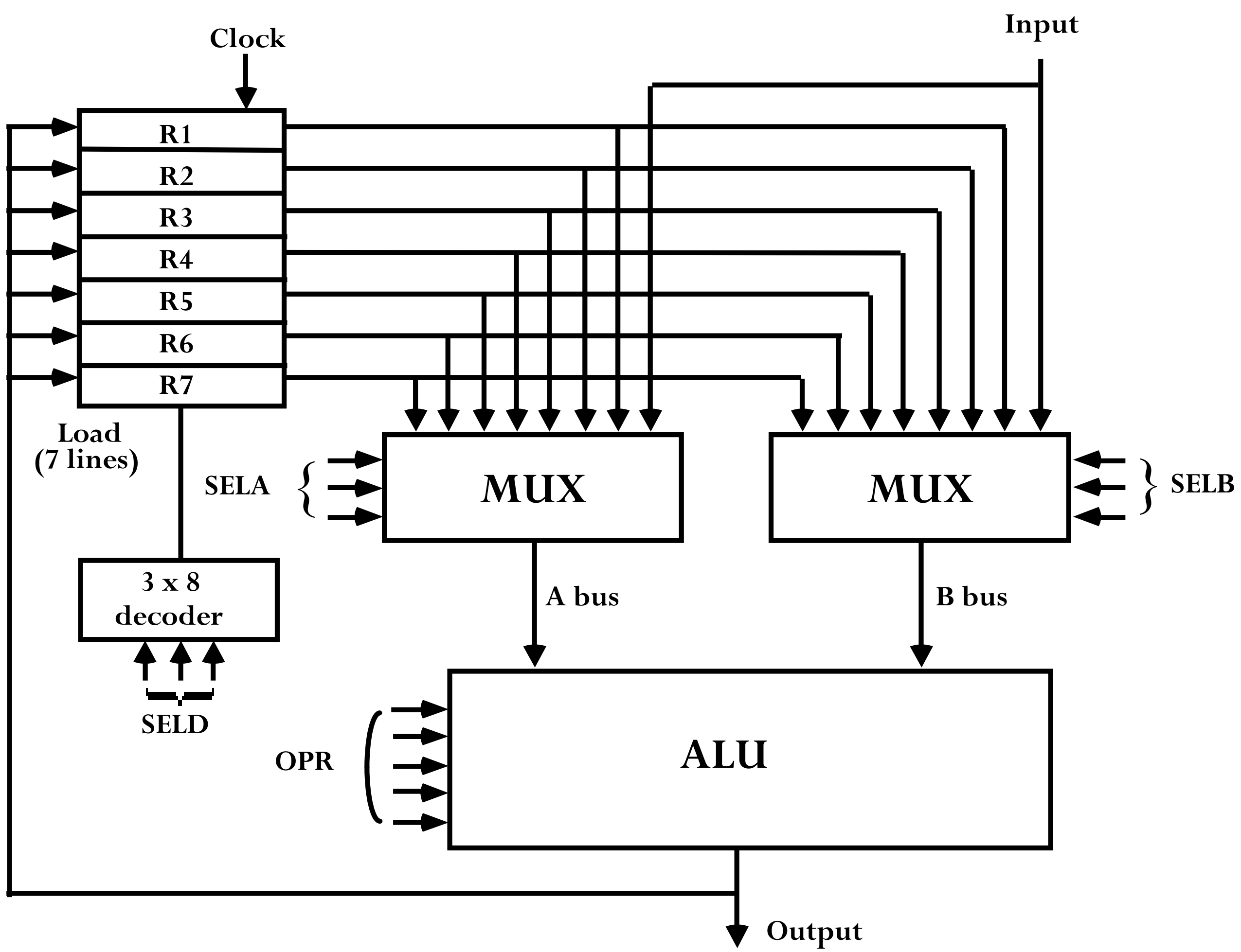


Figure 5. General Register Organization [1]

INSTRUCTIONS

- Program
  - A sequence of (machine) instructions
- (Machine) Instruction
  - A group of bits that tell the computer to *perform a specific operation* (a sequence of micro-operation)
- The instructions of a program, along with any needed data are stored in memory
- The CPU reads the next instruction from memory
- It is placed in an *Instruction Register* (IR)
- Control circuitry in control unit then translates the instruction into the sequence of microoperations necessary to implement it

INSTRUCTION FORMAT

- A computer instruction is often divided into two parts
  - An *opcode* (Operation Code) that specifies the operation for that instruction
  - An *address* that specifies the registers and/or locations in memory to use for that operation
- In the Basic Computer, since the memory contains 4096 ($= 2^{12}$) words, we needs 12 bit to specify which memory address this instruction will use
- In the Basic Computer, bit 15 of the instruction specifies the *addressing mode* (0: direct addressing, 1: indirect addressing)
- Since the memory words, and hence the instructions, are 16 bits long, that leaves 3 bits for the instruction's opcode

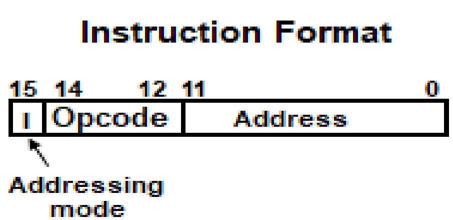


Figure 6. Instruction format [1]

ADDRESSING MODES

- The address field of an instruction can represent either
  - Direct address: the address in memory of the data to use (the address of the operand), or
  - Indirect address: the address in memory of the address in memory of the data to use

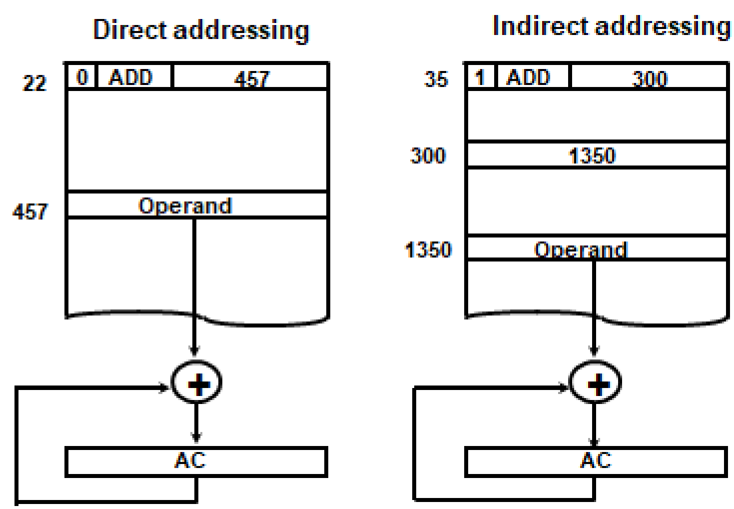


Figure 7. Addressing modes [1]

- Effective Address (EA)
  - The address, that can be directly used without modification to access an operand for a computation-type instruction, or as the target address for a branch-type instruction

BASIC COMPUTER INSTRUCTIONS

- Basic Computer Instruction Format

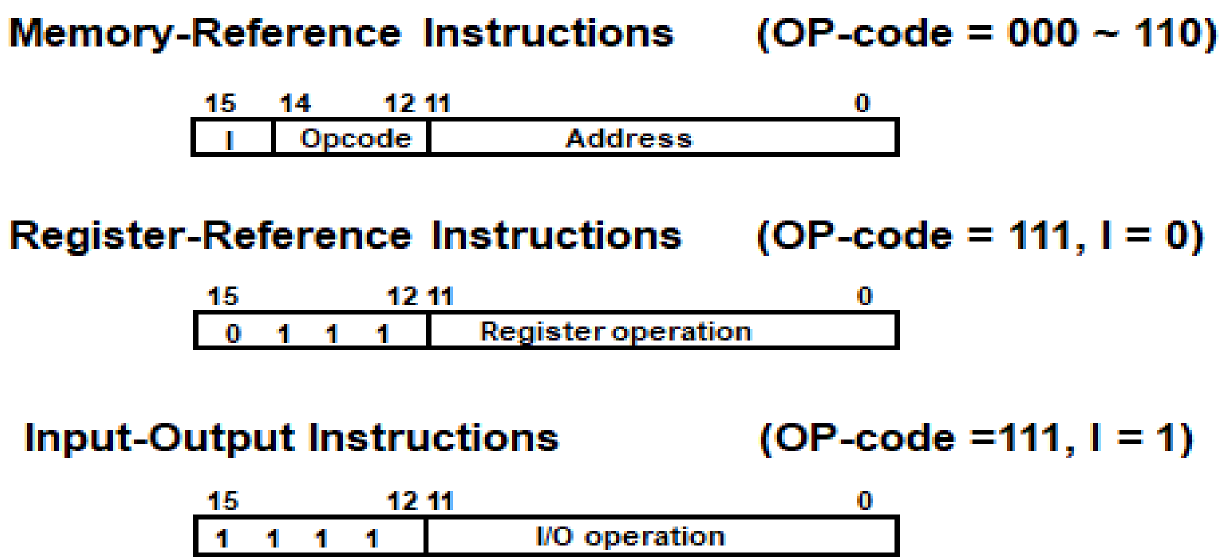


Figure 8. Basic computer instructions [1]

BASIC COMPUTER INSTRUCTIONS

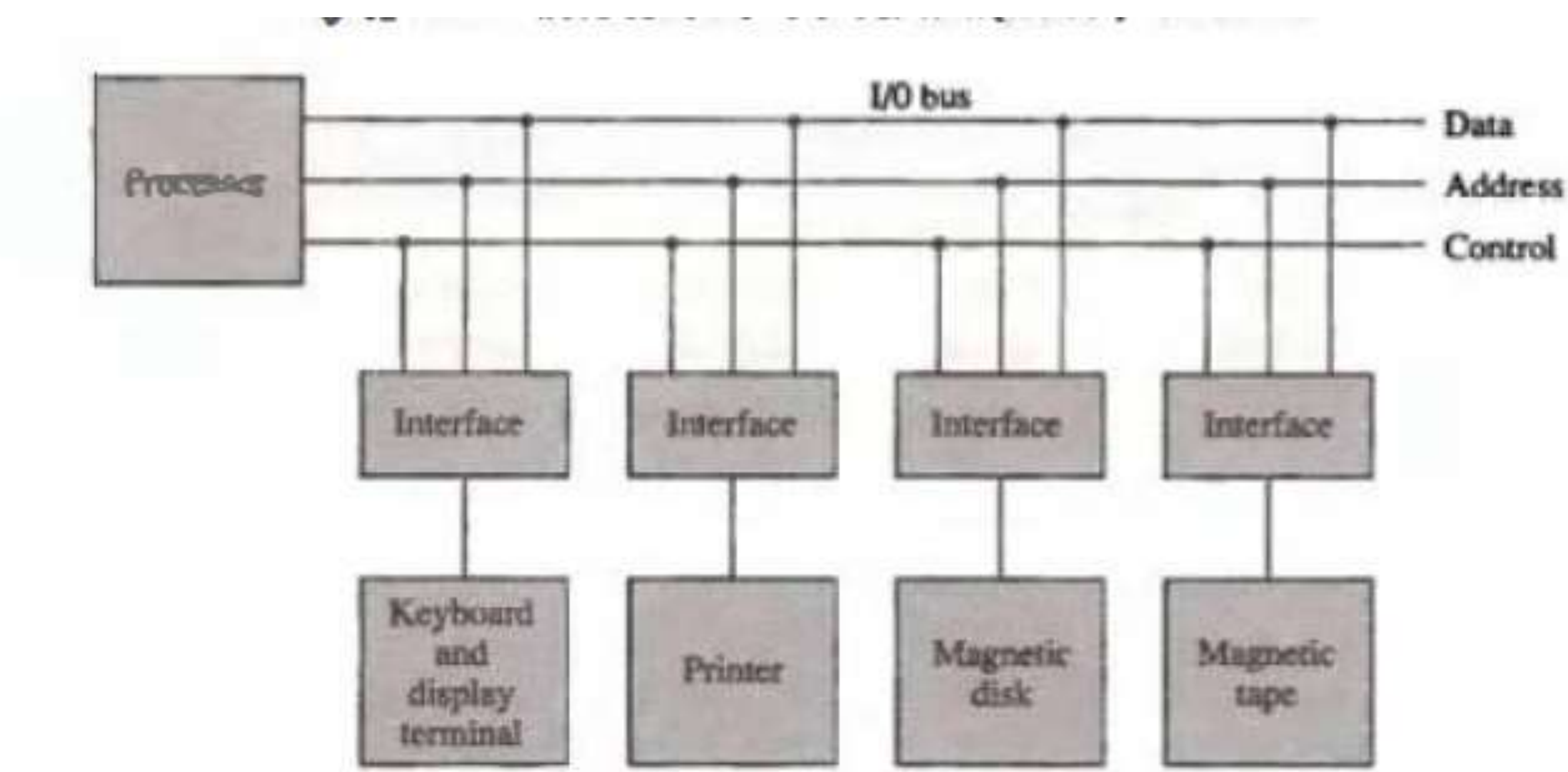
TABLE 5-2 Basic Computer Instructions

Symbol	Hexadecimal code $I = 0$	Hexadecimal code $I = 1$	Description
AND	0xxx	8xxx	AND memory word to AC
ADD	1xxx	9xxx	Add memory word to AC
LDA	2xxx	Axxx	Load memory word to AC
STA	3xxx	Bxxx	Store content of AC in memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and skip if zero
CLA	7800		Clear AC
CLE	7400		Clear E
CMA	7200		Complement AC
CME	7100		Complement E
CIR	7080		Circulate right AC and E
CIL	7040		Circulate left AC and E
INC	7020		Increment AC
SPA	7010		Skip next instruction if AC positive
SNA	7008		Skip next instruction if AC negative
SZA	7004		Skip next instruction if AC zero
SZE	7002		Skip next instruction if E is 0
HLT	7001		Halt computer
INP	F800		Input character to AC
OUT	F400		Output character from AC
SKI	F200		Skip on input flag
SKO	F100		Skip on output flag
ION	F080		Interrupt on
IOF	F040		Interrupt off

Figure 9. Basic computer instructions [1]

COMMON BUS SYSTEM

- The registers in the Basic Computer are connected using a bus
- This gives a savings in circuitry over complete connections between registers.[3]
- Data path means that how the data can flow through the Buses. Bus is the interconnections through which voltage signal will be flowing. A bus is a high-speed internal connection. Buses are used to send control signals and data between the processor and other components.
- **Address Bus**: carries memory addresses from the processor to other components such as primary storage and input/output devices. The address bus is **unidirectional**.
- **Data Bus**: carries the data between the processor and other components. The data bus is **bidirectional**.
- **Control Bus**: carries control signals from the processor to other components. The control bus also carries the clock's pulses. The control bus is **unidirectional.**



Control bus address bus and data bus [1]

COMMON BUS SYSTEM

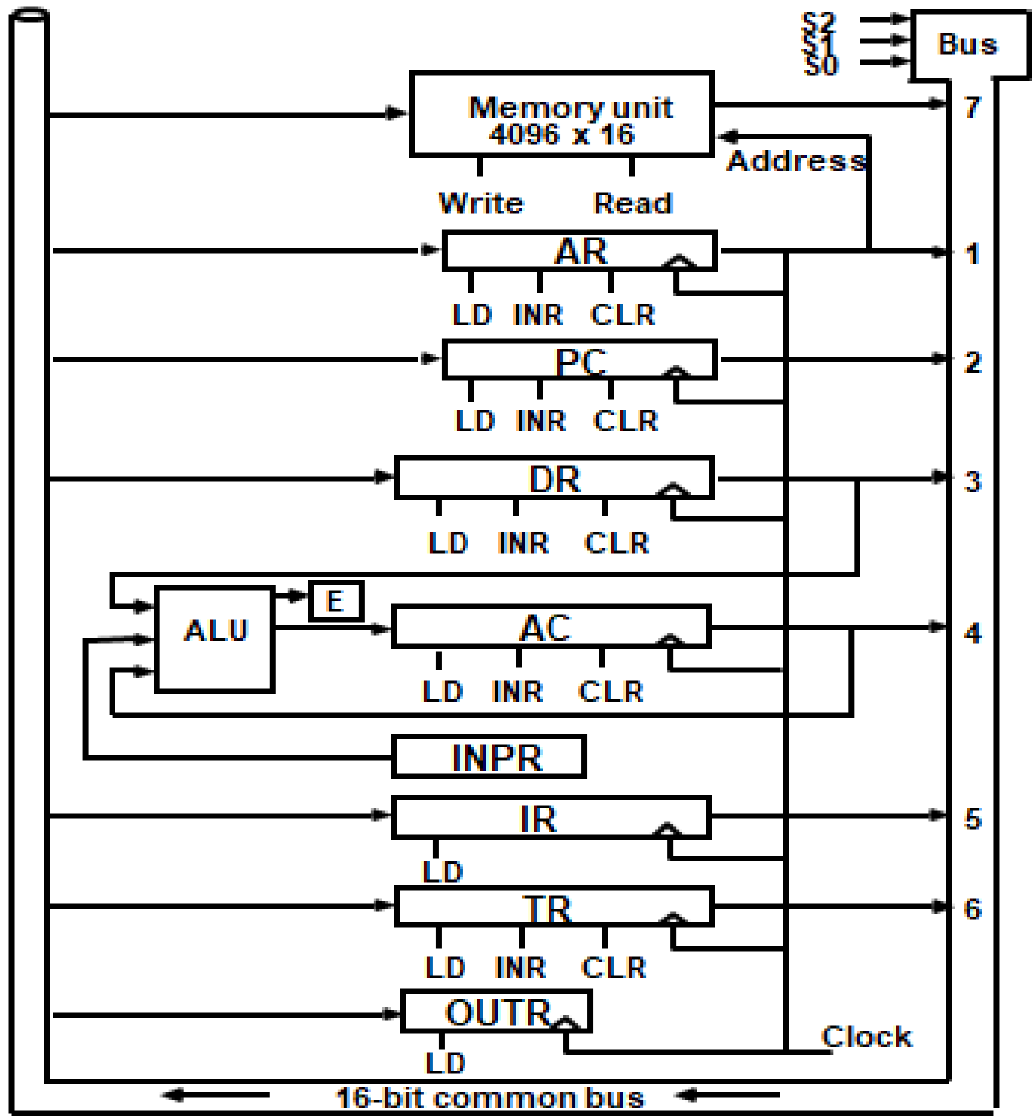


Figure 10. Common bus system [1]

COMMON BUS SYSTEM

- Three control lines, S_2, S_1, and S_0 control which register the bus selects as its input

S_2	S_1	S_0	Register
0	0	0	x
0	0	1	AR
0	1	0	PC
0	1	1	DR
1	0	0	AC
1	0	1	IR
1	1	0	TR
1	1	1	Memory

- Either one of the registers will have its load signal activated, or the memory will have its read signal activated
  - Will determine where the data from the bus gets loaded
- The 12-bit registers, AR and PC, have 0's loaded onto the bus in the high order 4 bit positions
- When the 8-bit register OUTR is loaded from the bus, the data comes from the low order 8 bits on the bus

Memory

Divided into:

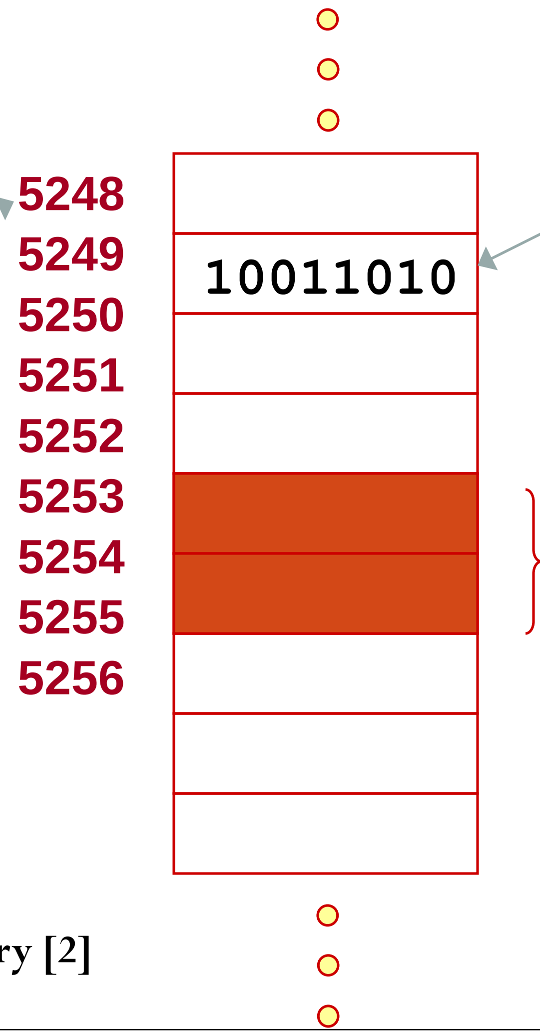
- **Bits** 0 or 1
- **Bytes** Groups of 8 bits
  A byte is the smallest unit of storage. (Can hold one text character)
- **Words** Groups of bits/bytes (8, 16, 32, 64-bits)

Storage is usually too large to be expressed in **bytes** or **words**. Instead we use:

- **Kilobyte (KB)** = 1024 bytes (2^{10} bytes)
- **Megabyte (MB)** = 1024 x 1024 bytes or one million bytes (2^{20} bytes)
- **Gigabyte (GB)** = 1024 x 1024 x 1024 bytes or one trillion bytes (2^{30} bytes)
- **Terabyte (TB)** = 1024 x 1024 x 1024 x 1024 bytes one quadrillion bytes (2^{40} bytes)

Main Memory

Each memory cell has a numeric **address**, which uniquely identifies its location



Each memory cell stores a set number of bits (some computers use 8 bits/one **byte**, others use **words**)

A **word** is stored in consecutive memory bytes.

Figure 11. Main memory [2]

CPU and Memory

- CPU can interact with main memory in two ways [2][3]:
  - It can **write** a byte/word to a given memory location.
    - The previous bits that were in that location are destroyed
    - The new bits are saved for future use.
  - It can **read** a byte/word from a given memory location.
    - The CPU copies the bits stored at that location and stores them in a CPU register
    - The contents of the memory location are NOT changed.

Main Memory Characteristics

- Very closely connected to the CPU.
- Contents are quickly and easily changed.
- Holds the programs and data that the processor is actively working with.
- Interacts with the processor millions of times per second.
- Nothing permanent is kept in main memory.
- Each computer has a specific word size
  - Word sizes vary from computer to computer.
  - Word size is an even multiple of a bytes.
- Each word within memory can hold either
  - data or
  - program instructions

References

- [1]. M. Morris Mano, Computer System Architecture, Prentice Hall of India Pvt Ltd, 3rd Edition (updated) , 30 June 2017.
- [2]. Tanenbaum & Austin, Structured Computer Organization, 6th Edition, Pearson Education
- [3]. William Stallings, Computer Organization and Architecture– Designing for Performance, Ninth Edition, Pearson Education, 2013.