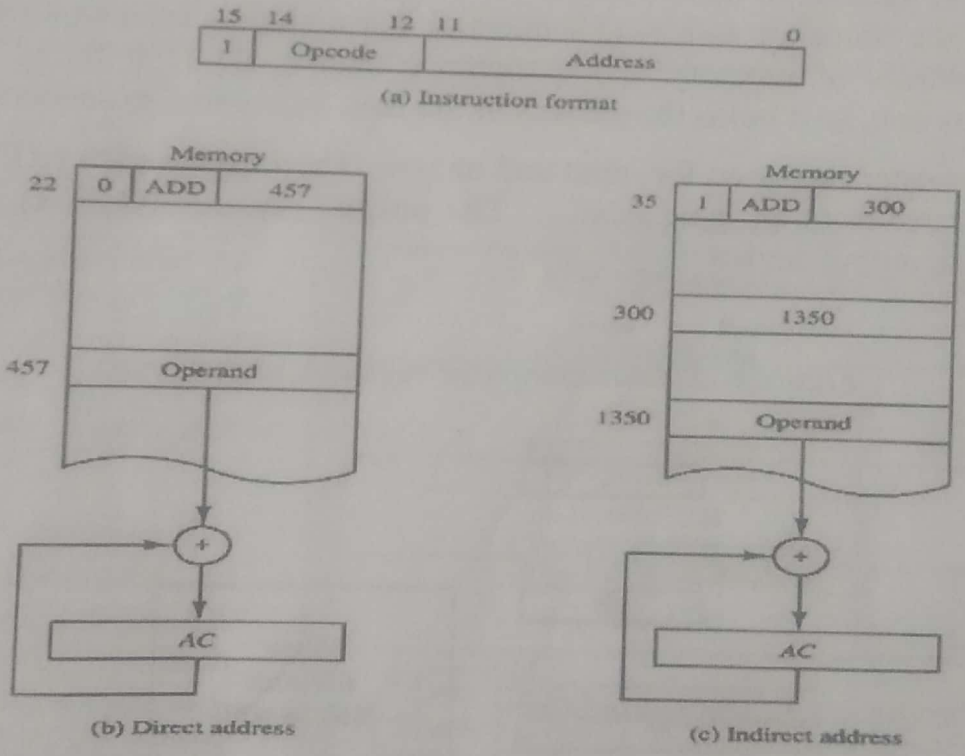
The instruction code format is shown in fig.1.2.a. which consists of 3-bit operation code, a 12-bit address, and an address mode bit designated by I. The mode bit is 0 for a direct and 1 for an indirect address.

A direct address instruction is shown in fig. 1.2.b. It is placed in address 22 in memory. The I bit is 0, so the instruction is a direct address instruction. The opcode specifies an ADD instruction, and the address part is the binary equivalent of 457. The control finds the operand in memory at address 457 and adds it to the content of AC.

The instruction in address 35 shown in fig.1.2.c.The a mode bit is I = 1. Therefore, it is an indirect address instruction. The address part is the binary equivalent of 300. The control goes to address 300 to find the address of the operand. The address of the operand is 1350. The operand found in address 1350 is added to the content of AC.

Figure 1.2 Demonstration of direct and indirect address



1.2 COMPUTER REGISTERS

Consider the memory unit with a capacity of 4096 words and each word contains 16 bits. 12 bits of an instruction word is used to specify the address of an operand and 3 bits for the operation part of the instruction and 1 bit to specify a direct or indirect address. The **data register (DR)** holds the operand read from memory. The **accumulator (AC)** register is a general-purpose processing register. The instruction read from memory is placed in the **instruction register (IR)**. The **temporary register (TR)** is used for holding temporary data during the processing.

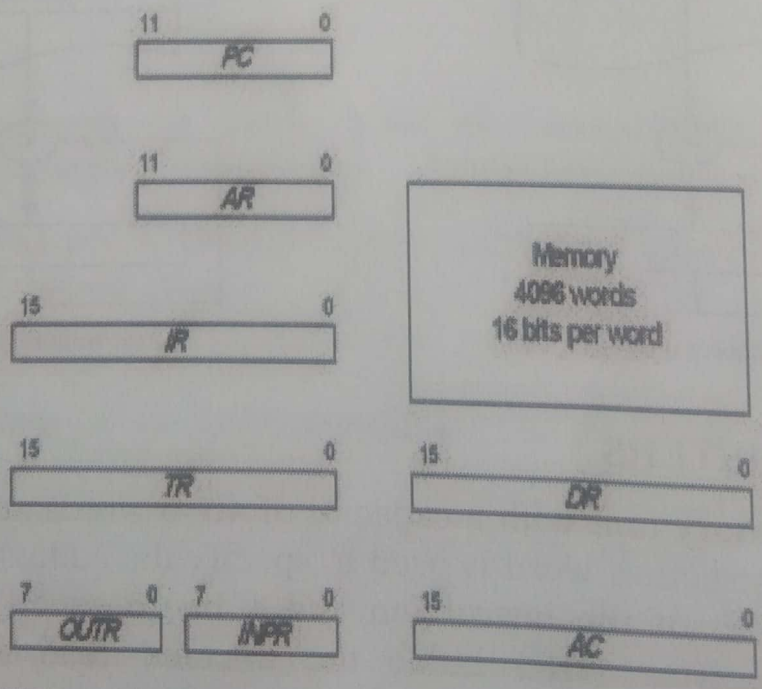
Table 1-1 List of registers for the Basic Computer

Register symbol	Number of bits	Register Name	Function
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 out put character

The memory **address register (AR)** has 12 bits because it is the width of a memory address, The**program counter (PC)** also has 12 bits and it holds the address of the next instruction to be read from memory after the current instruction is executed. The PC is used to count the sequence and read sequential instructions. To read an instruction PC holds the address of memory and a memory read cycle is initiated. PC is then incremented by one, so it holds the address of the next instruction in sequence.

Two registers are used for input and output. The **input register (INPR)** receives an 8-bit character from an input device. The **output register (OUTR)** holds an 8-bit character for an output device.

Figure 1. 3 Basic computer registers and memory



Common Bus System

The basic computer has eight registers, a memory unit, and a control unit. Paths are provided to transfer information from one register to another and between memory and register. The number of wires are connected between the outputs of each register and the inputs of the other registers.

For transferring the information in a system with many registers an efficient scheme is used in the common bus system.

The connection of the registers and memory of the basic computer to a common bus system is shown in fig.1.4.

The outputs of seven registers AR, PC, DR, AC, IR, TR and memory are connected to the common bus. The specific output that is selected for the bus lines are found from the binary value of the selection variables S2,S1and S0. The number with each output gives the decimal equivalent of the required binary selection. For example, the number with the output of DR is 3. The 16-bit outputs of DR are placed on the bus lines when S2S1S0 = 011. The particular register whose LD (Load) input is enabled receives the data from the bus during the next clock pulse transition. The memory receives the contents of the bus when its write input is activated. The memory places its 16-bit output onto the bus when the read input is activated and S2S1S0 = 111.

The 4 registers, DR, AC, IR, and TR, have 16 bits each. The registers, AR and PC have 12 bits each since they hold a memory address. When the contents of AR or PC are applied to the 16-bit common bus, the four most significant bits are set to 0's. When AR or PC receivesthe information from the bus, only the 12 least significant bits are transferred into the register.

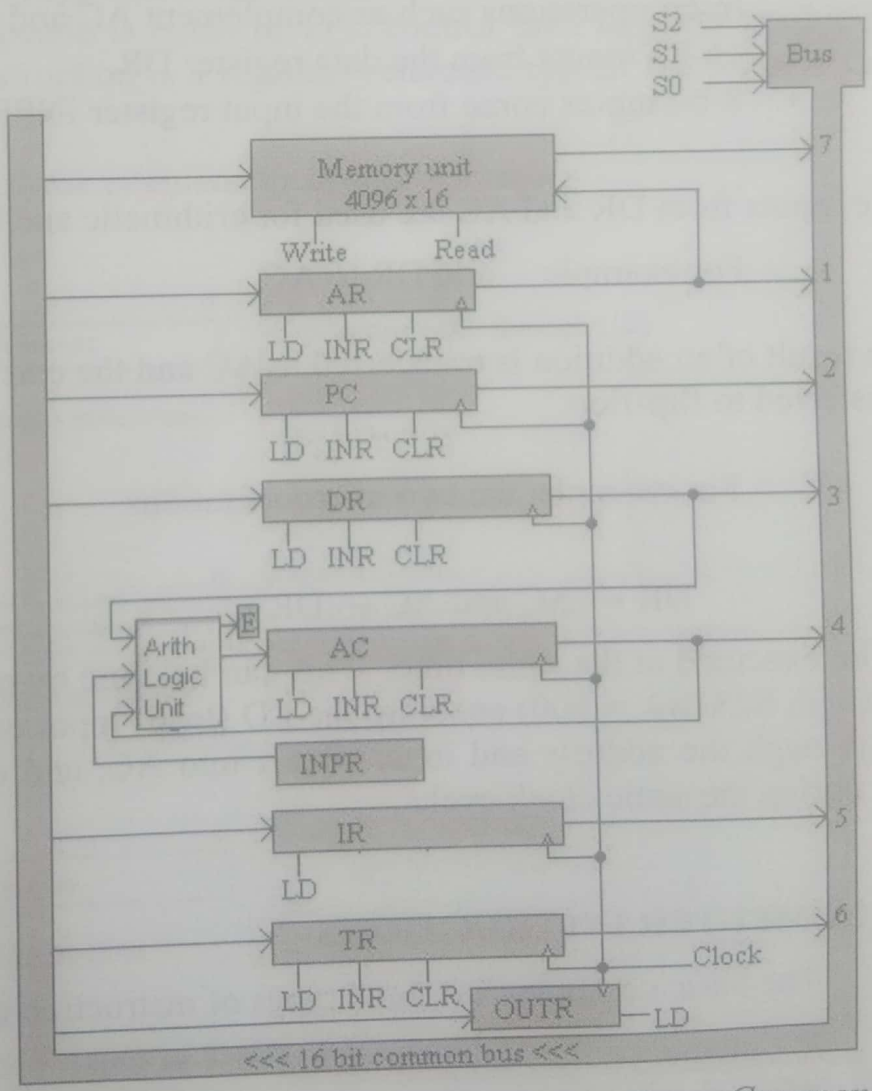


Figure 1.4 Basic Computer Registers connected to a Common bus

The input register INPR and the output register OUTR have 8 bits each. INPR provides theinformation to the bus. OUTR can only receive information from the bus. The INPR receives a character from an input device and transfer to AC. OUTR receives a character from AC and send it to an output device.

The 16 lines of the common bus receive information from six registers and the memory unit. The bus lines are connected to the inputs of six registers and the memory. Five registers have three control inputs :LD (load), INR (increment), and CLR (clear).

The input data and output of the memory are connected to the common bus. The memory address is connected to AR. Therefore, AR is used to specify a memory address. During a write operation the content of any register can be stored in the memory as input data . Similarly during the read operation any register can receive the data from memory.

The 16 outputs of Adder and logic circuit are the inputs of an AC. It has three sets of inputs. They are

- 16-bit inputs are from the outputs of AC. They are used for the microoperations such as complement AC and shift AC.
- 16-bit inputs from the data register DR.
- 8-bit inputs come from the input register INPR.

The inputs from DR and AC are used for arithmetic and logic microoperations.

For example add DR to AC

The result of an addition is transferred to AC and the end carry-out of the addition is transferred to flip-flop.

For example, the two microoperations

$$DR \leftarrow AC \text{ and } AC \leftarrow DR$$

can be executed at the same time. This can be done by placing the content of AC on the bus (with S2S1S0 = 100) enabling the LD (load) input of DR, transferring the content of DR through the address and logic circuit into AC, and enabling the LD (load) input of AC, during the same clock cycle.

1.3 COMPUTER INSTRUCTIONS

The basic computer has three types of instruction code formats. They are

- memory-reference instruction
- register reference instruction
- input-output instruction

Each format has 16 bits. The operation code (opcode) of the instruction contains 3 bits and the remaining 13 bits depends on the operation code.

Memory-reference instruction uses 12 bits to specify an address and 1 bit to specify the addressing mode I. I is equal to 0 for direct address and to 1 for indirect address

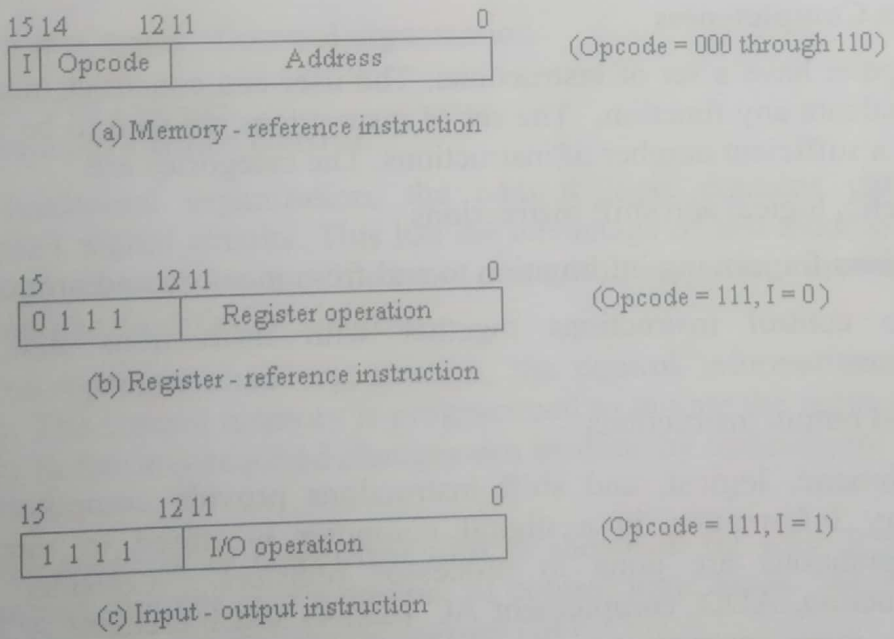
Register reference instruction has the operation code 111 with a 0 in the leftmost bit (bit 15) of the instruction. It specifies an operation on or test of the AC register. An operand from memory is not needed. So the remaining 12 bits are used to specify the operation or test to be executed.

Input-output instruction has the operation code 111 with a 1 in the leftmost bit of the instruction. It needs a reference to memory. The remaining 12 bits are used to specify the type of input-output operation or test to be executed.

The control unit recognizes the type of instruction. It is found by the 4 bits in positions 12 through 15. If the three opcode bits in positions 12 through 14 are not equal to 111, the

instruction is a memory-reference type and the bit in position 15 is taken as the addressing mode I. If the 3-bit opcode is equal to 111, control then inspects the bit in position 15. If this bit is 0, the instruction is a register- reference type.If the bit is 1, the instruction is an input-output type.

Figure 1.5 Basic computer Instruction formats



The instructions for the computer are listed in Table 1-2.

Table 1-2 Basic Computer Instruction

Symbol	Hex Code I = 0	Hex Code I = 1	Description
AND	0xxx	8xxx	AND memory word to AC
ADD	1xxx	9xxx	Add memory word to AC
LDA	2xxx	Axxx	Load AC from memory
STA	3xxx	Bxxx	Store content of AC into 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 instr. if AC is positive
SNA	7008		Skip next instr. if AC is negative
SZA	7004		Skip next instr. if AC is zero
SZE	7002		Skip next instr. if E is zero
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

Instruction Set Completeness

A computer have a set of instructions. The user can construct machine language programs to evaluate any function. The set of instructions are said to be complete if the computer have a sufficient number of instructions. The categories are

1. Arithmetic, logical and shift instructions
2. Instructions for moving information to and from memory and processor register
3. Program control instructions together with instructions that check status conditions\
4. Input and output instructions

- Arithmetic, logical, and shift instructions provide computations. All the binary information in a digital computer is stored in memory. All the computations are done in processor registers. Consider an arithmetic instruction, ADD, complement AC (CMA) and increment AC (INC). With these three instructions we can add and subtract binary numbers when negative numbers are in signed-2s complement representation. The circulate instructions, CIR and CIL, can be used for arithmetic shifts as well as any other type of shifts desired. Multiplication and division can be performed using addition, subtraction, and shifting. There are 3 logic operations : AND,

complement AC(CMA) and clear AC (CLA). The AND and complement provide a NAND operation

- The information must be moved between memory and registers. Moving information from memory to AC is done by load AC (LDA) instruction. Storing information from AC into memory is done with the store AC(STA) instruction.
- Program control instructions such as branch instructions are used to change the sequence in which the program is executed. Branch instructions are used for decision making. The branch instructions are BUN, BSA, ISZ.
- Input and output instructions are needed for communication between the computer and the user. Programs and data must be transferred into memory and results of computations must be transferred back to the user. The input(INP) and output(OUT) instructions are used to transfer information between computer and external devices.

1.4 TIMING AND CONTROL

The master clock generator controls the timing of all the registers in the basic computer. The clock pulses are applied to all flip-flops and registers in the system including the flip-flops and registers in the control unit. The clock pulses changes the state of a register only when the register is enabled by a control signal. The control signals are generated in the control unit. They are the control inputs for the multiplexers, processor registers, and the microoperations for the accumulator.

There are two types of control organization.

- **hardwired control**
- **microprogrammed control**.

In the hardwired organization, the control logic contains gates, flip-flops, decoders, and other digital circuits. This has the advantage of fast mode of operation. In this if the design has to be modified or changed the wiring among the various components should be modified or changed.

In the microprogrammed organization, the control information is stored in a control memory. The control memory is programmed to initiate the required sequence of microoperations. In this any required changes can be done by updating the microprogram in control memory.

The block Diagram of the**control unit** is shown in fig 1.6. It consists of two decoders, a sequence counter, and a number of control logic gates. An instruction read from memory is placed in the instruction register (IR)

The instruction register is divided into three parts the I bit, the operation code and bits 0 through 11. The operation code is; bits12 through 14 are decoded with a 3 x 8 decoder. The 8 outputs of the decoder are denoted by D_0 through D_7. The subscripted decimal number corresponds to the binary value of the corresponding operation code.