Basic Computer Organization and design-Part 1

BASIC COMPUTER ORGANIZATION AND DESIGN

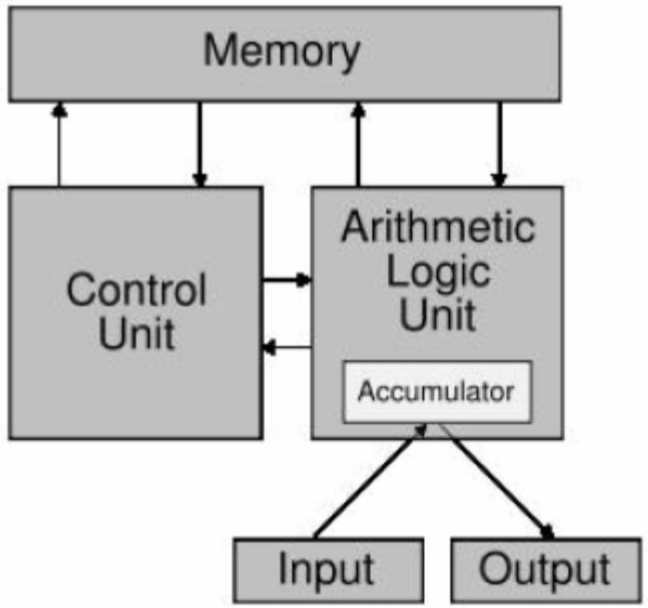
- Instruction Codes
- Computer Registers
- Computer Instructions
- Timing and Control
- Instruction Cycle
- Memory Reference Instructions
- Input-Output and Interrupt
- Complete Computer Description
- Design of Basic Computer
- Design of Accumulator Logic

Organization of a computer

- The organization of a computer is defined by its internal registers, the timing and ctrl structure, and the set of instructions that it uses.
- The internal org of a digital comp is defined by the seq of micro operations it performs on data stored in its registers
- A program is a set of instructions that specify the operations, operands and the seq by which processing has to occur.

Von Neumann architecture

- The von Neumann architecture is a computer design model that uses a processing unit and a single separate storage structure to hold both instructions and data. It is named after mathematician and early computer scientist John von Neumann.


Memory
Control Unit
Arithmetic Logic Unit
Accumulator
Input
Output

Stored program concept

- Instructions and data together are stored in computer memory. The computer reads each instruction from memory and places it in control register.
- The control then interprets the binary code of the instruction and proceeds to execute it by using a sequence of microoperations.
- The ability to store and execute instructions, the stored program concept, is the most imp property of a general purpose computer.

INSTRUCTION FORMATS

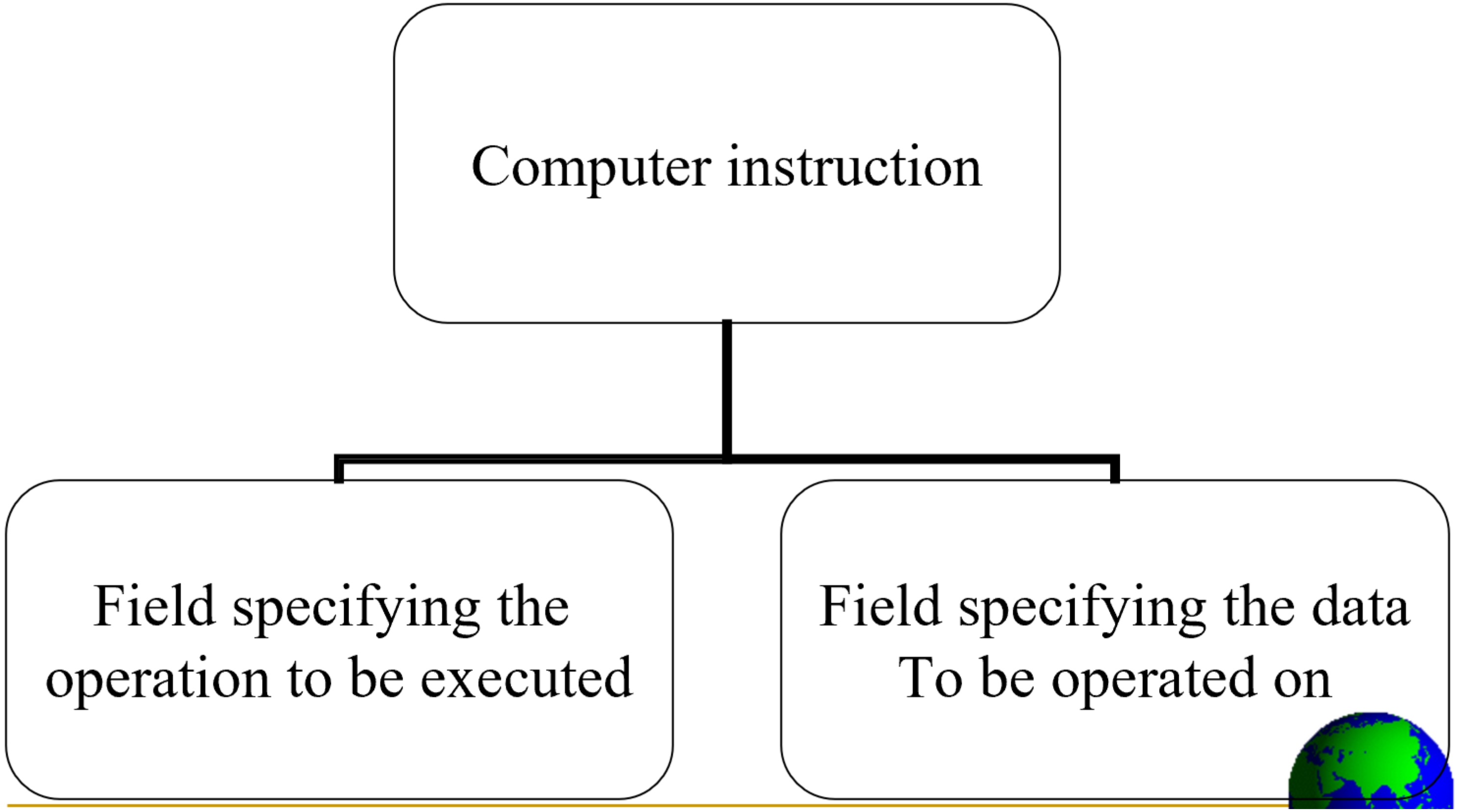
In selecting the instruction format(s) the following factors should be considered.

1. The number of instructions to be represented.
2. The addressability and addressing modes.
3. The ease of decoding.
4. Type of instruction field (fixed or variable)
5. The cost of hardware required to decode and execute instructions.

OP-CODE	ADDRESS(ES)

0-, 1-, 2- or 3-addressable instruction formats

Program statements and computer instructions



Basic definitions

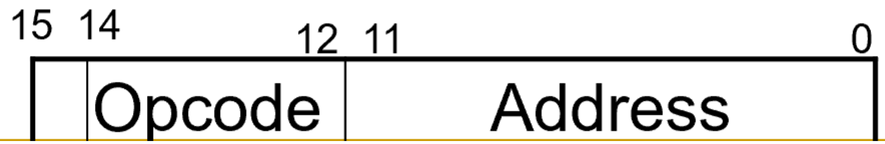
- Instruction code
  - It is group of bits that instruct the computer to perform the specific task.
  - The operation code is a group of bits that define operations such as add, subtract etc.
  - The total number of bits for the operation code of an instruction depends upon the total number of operations available in the computer.
  - n bits for 2^n operations

Computer operation and micro operation

- For every operation code, the control issues a seq of micro operations needed for the hardware implementation of a specified operation.
- Op code is also called macro operation.

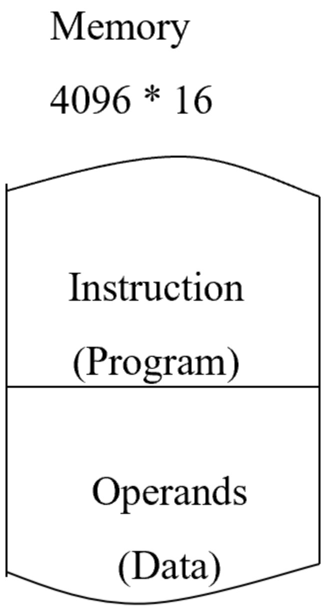
Instruction code format

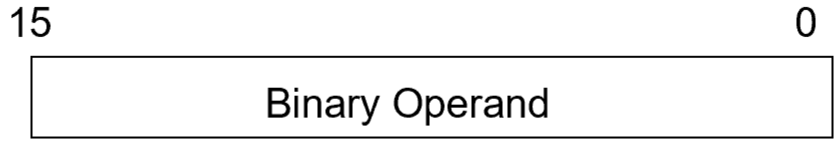
- The instruction code specifies operation, registers or the memory words where the operands are to be found and the registers or the memory words where the result is to be stored.
- Instruction code-16 bit memory word
- Operand address -12 bit
- Op code-3 bits
- Direct/ Indirect address



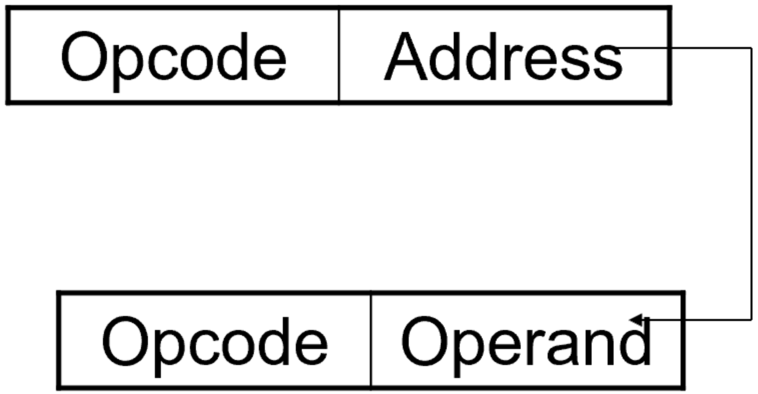
Stored program organization

- The ctrl reads the 16 bit instruction from the program portion of the memory. It uses the 12 bit address part of the instruction to read a 16 bit operand from the data portion of the memory.

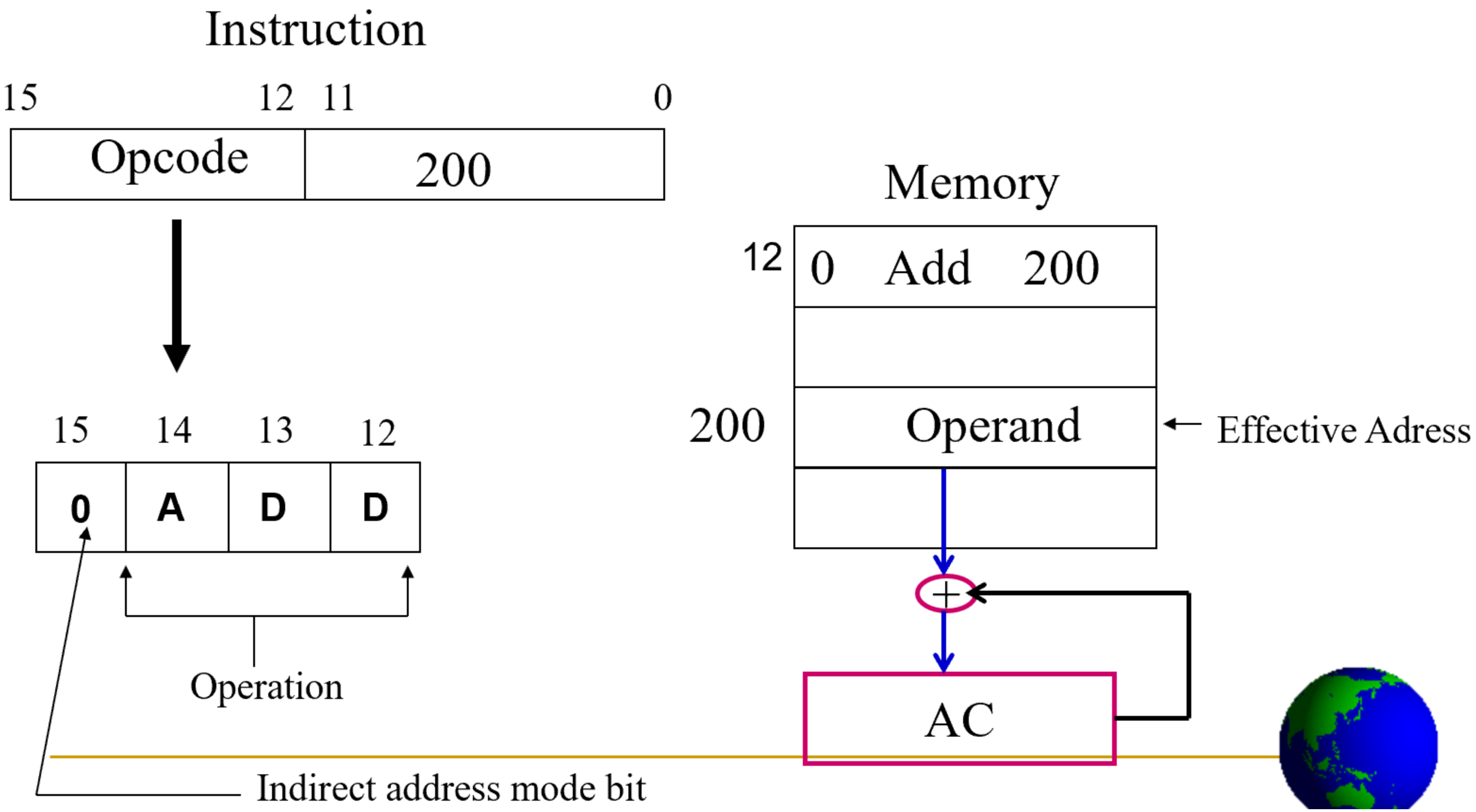




Immediate Mode

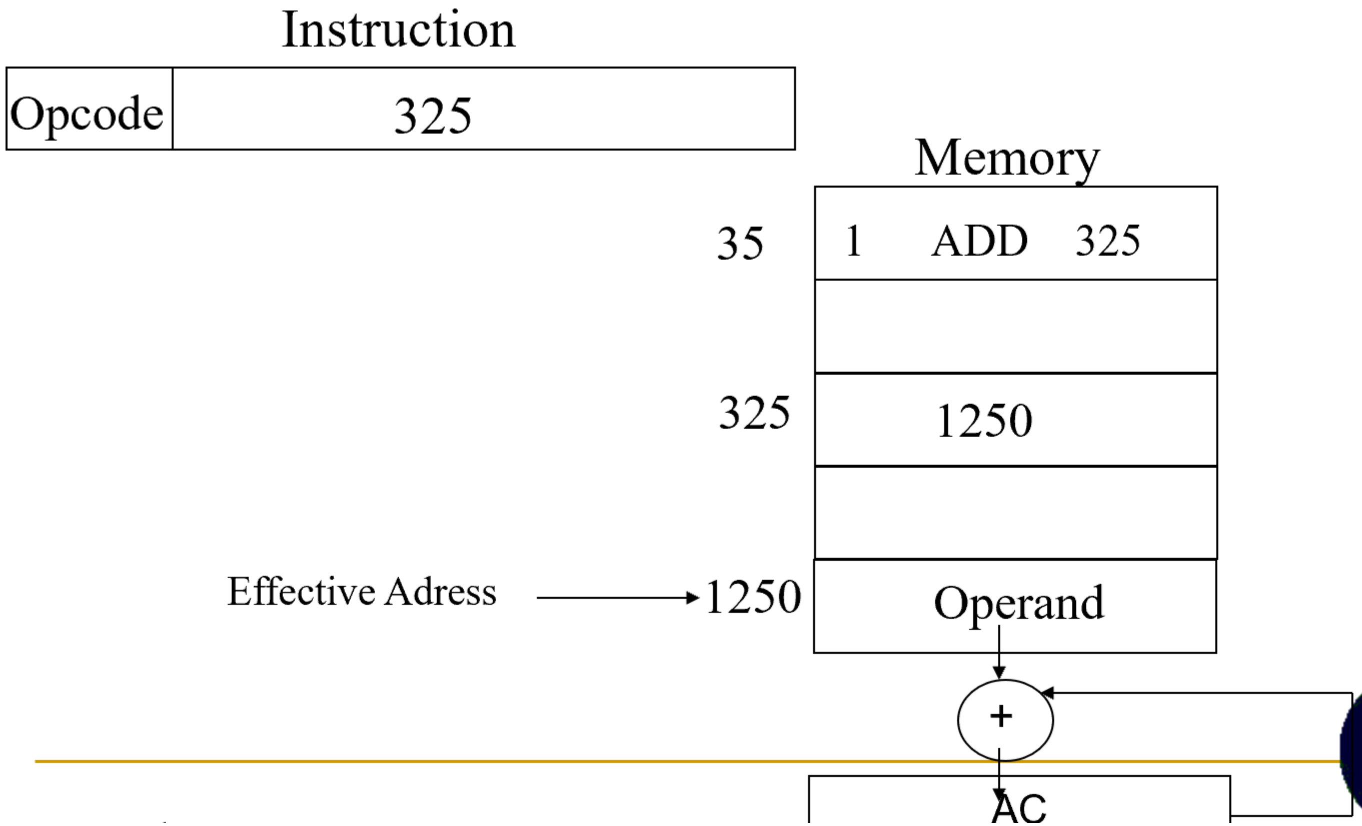


Direct Addresses



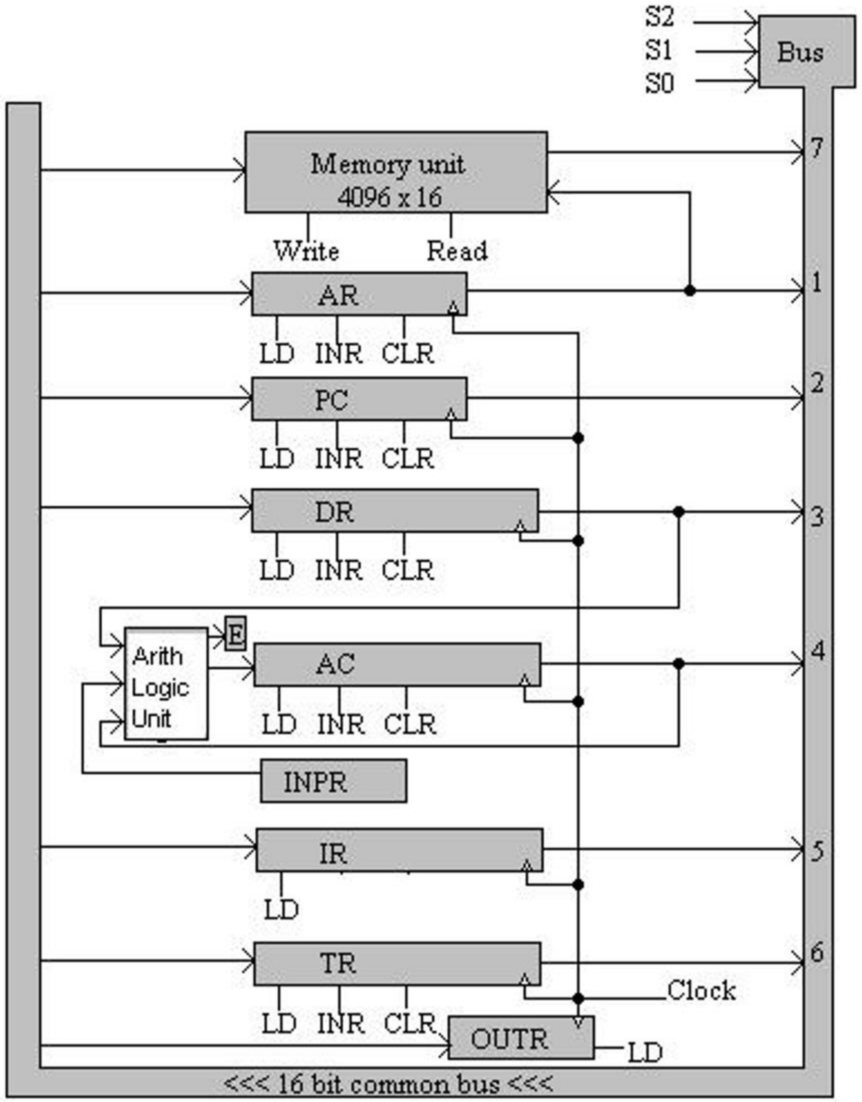
Indirect Addressing Mode

Memory cell pointed to by address field contains the address of (pointer to) the operan



Computer registers

- Computer needs processor registers for manipulating data and a register for holding a memory address.
- There is also a need for a register in the control unit to store the instruction code after its read from memory.
- The basic computer has 8 registers, a memory unit and a control unit.
- Paths must be provided to transfer information from one register to another and between memory and registers.
- The outputs of 7 registers and memory are connected to the common bus
- Specific output is selected by S_o,S_1,S_2.



S2
S1
S0
Bus
Memory unit
4096 x 16
Write
Read
7
AR
LD INR CLR
1
PC
LD INR CLR
2
DR
LD INR CLR
3
E
Arith
Logic
Unit
AC
LD INR CLR
4
INPR
IR
LD
5
TR
LD INR CLR
6
Clock
OUTR
LD
<<< 16 bit common bus <<<

Bus

- In computer architecture, a **bus** is a subsystem that transfers data between computer components inside a computer or between computers.
- Buses can be parallel buses, which carry data words in parallel on multiple wires, or serial buses, which carry data in bit-serial form.

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
- The outputs of 7 registers and memory are connected to the common bus.
- The specific output selected for the bus lines at any given time is determined from the binary value of the selection variables S_2,S_1 and S_0

COMMON BUS SYSTEM

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

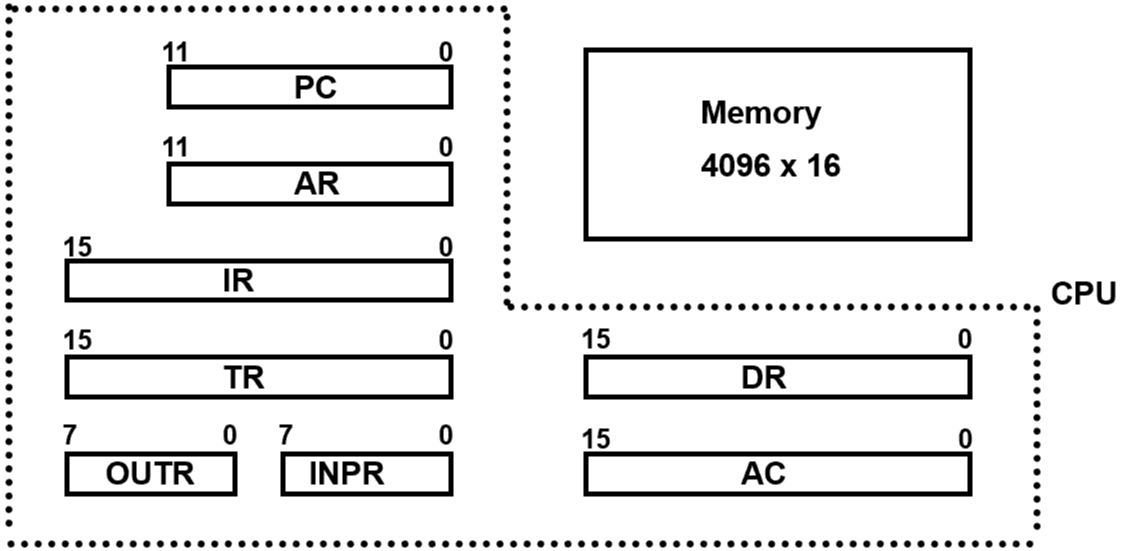
- The lines from the common bus are connected to the inputs of each register and the data inputs of the memory.
- The particular register whose LD (Load) input is enabled receives the data from the bus.
- The memory receives the contents of the bus when its write input is activated. The memory places its 16 bit output onto the bus when the read input is activated and $S_2S_1S_0 = 111$

Common bus cont

- The 12-bit registers, AR and PC, have 0’s loaded onto the bus in the high order 4 bit positions
- When AR and PC receive info from the bus, only the 12 least significant buts are transferred into the register.

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

Instruction set design

- The most significant and complex task in designing a computer is framing its instruction set. A computer architect has to consider the following aspects before finalizing the instruction set:-
  - Programming convenience: *number of instructions- with more number of instructions, appropriate operations are carried out by respective instructions but the control unit becomes quite complex.*
  - Powerful addressing- *all possible modes of addressing are present but the control unit design becomes complex*.
  - Number of general purpose registers- *data movement and processing is faster but the cost of CPU hardware increases*

Instruction set design cont…

- Target market segment- *scientific computer has more floating point arithmetic*
- System performance- *if a program has less number of instructions, the performance is enhanced since time spent by the CPU in instruction fetching is reduced. A single instruction must be able to perform several micro operations, which reduces the program size but increase the control unit complexity and instruction execution time.*

CPU organization

- The selection of an instruction set for a computer depends on the manner in which the CPU is organized. There are 3 different CPU organizations with certain specific instructions:-
  - Accumulator based CPU
  - Registers based CPU
  - Stack based CPU

Accumulator based CPU

- It’s a simple CPU, in which the accumulator contains an operand for the instruction. Similarly the instruction leave the result in the accumulator. The contents of the accumulator participate in the arithmetic operations such as addition, subtraction etc.

Example

- (A+B) – (C+D)

  LOAD A

  ADD B

  STORE T

  LOAD C

  ADD D

  SUB T

  STORE X

Registers based CPU

- In this type of CPU, multiple registers are used as accumulator. Such a CPU has a general purpose register (GPR) organization. The use of registers result in short programs with limited instructions.

Example

- (A+B) – (C+D)

```
LOAD R1, A
ADD R1, B
LOAD R2,C
ADD R2, D
SUB R1,R2
SUB T
STORE R1,X
```

Advantages and disadvantages

- The register based CPU results in shorter program size.
- The program for accumulator based CPU requires a memory location for storing partial result. Hence additional memory accesses are needed during program execution.
- Thus increase in the number of registers increases the CPU efficiency, but care should be taken to avoid unnecessary usage of registers. Hence compilers need to be more efficient in this aspect.

Stack based CPU

- The stack is a push down list with LIFO access mechanism. It is present either inside the CPU or a portion of the memory can be used as a stack.

Example

- (A+B) – (C+D)

STATEMENT	STACK CONTENTS
PUSH A	A
PUSH B	A,B
ADD	A+B
PUSH C	(A+B),C
PUSH D	(A+B),C,D
ADD	(A+B),(C+D)
SUB	(A+B)-(C+D)
POP X	EMPTY

Advantages and disadvantages

- Easy programming/ high compiler efficiency
- Highly suited for block structured languages
- Instructions don't have address field; short instructions
- Additional hardware circuitry needed for stack implementation
- Increased program size

Accumulator

- Computers that have a single processor register assign to it the name accumulator AC
- Operations such as *clear AC*, *complement AC* etc don't need an operand from memory. For these type of instructions, the second part of the instruction code is not needed and used to specify other operations for the comp

Accumulator cont

- The 16 inputs of AC come from an adder and logic circuit. The circuit has 3 sets of inputs.
  - Outputs of the AC. They are used for microoperations like complement AC, shift AC
  - From the DR. The inputs from DR and AC are used for arithmetic and logic microoperations such as add *DR to AC*, *AND DR to AC.* The result is transferred to AC and the end carry-out is transferred to flip flop E
  - Third set of 8 bits come from INPR

Program counter

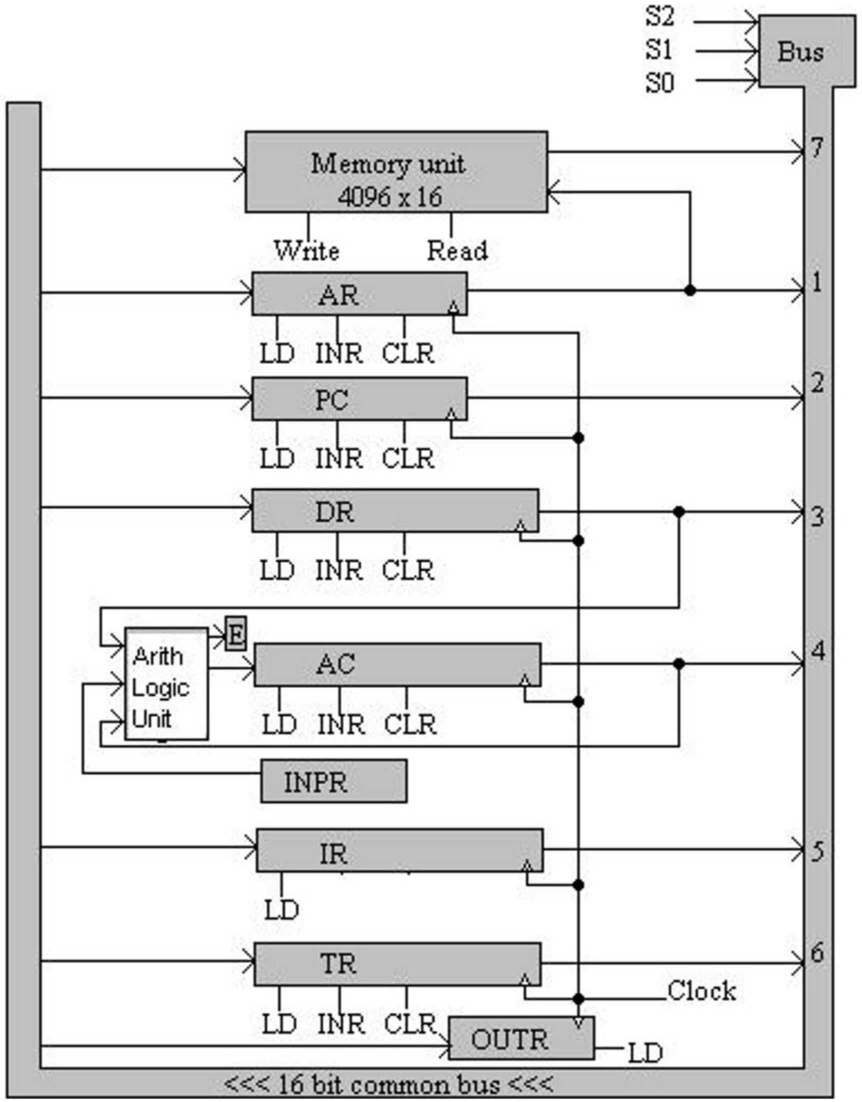
- Stores address of next instruction to execute
- Must be incremented after each instruction
- May be changed by function call or jump
- Controls flow of program execution.
- To read an instruction, the content of PC is taken as the address for memory and a memory read cycle is initiated.

Instruction register

- The instruction register contains the currently executing instruction
- Holds instruction while its being decoded
- Opcode field provides input to ctrl sys indicating operation to perform
- Contains addresses of operands to be used in operation
- Contains destination address of result
- Contains info about addressing modes to be used

Arithmetic logic unit

- Performs arithmetic and logical functions
- Add, subtract, multiply, divide, complement, shift etc
- Function performed is determined by the ctrl signals received
- Will have input and output latches to hold operands and results


S2
S1
S0
Bus
Memory unit
4096 x 16
Write
Read
7
AR
LD INR CLR
1
PC
LD INR CLR
2
DR
LD INR CLR
3
E
Arith
Logic
Unit
AC
LD INR CLR
4
INPR
IR
LD
5
TR
LD INR CLR
6
Clock
OUTR
LD
<<< 16 bit common bus <<<

Address register

- Holds address of the location in memory to be accessed
  - This may be the address of the next instruction to be fetched
  - May be address of the operand to be read from memory
  - May be the address of the info to be written to memory
- A single register is used so address bus is not required.

Memory buffer register

- Hold values to be transferred between main memory and the CPU
- Data and instructions read from the memory
- Values to be written to the memory
- Most modern machines are capable of transferring more than a single word

INPR and OUTR

- INPR and OUTR have 8 bits each and communicate with the 8 least significant buts in the bus.
- INPR provides information to the bus but OUTR can only receive information from the bus.
- INPR receives a character from an input device which is then transferred to AC
- OUTR receives a character from AC and delivers it to an output device. There is no transfer from OUTR to any other registers.

Registers

- The 16 lines of the common bus receive information from 6 registers and the memory unit.
- The bus lines are connected to the inputs of 6 registers and the memory
- Five registers have three control inputs: LD (Load), INR (increment) and CLR (clear)
- Two registers have only a LD input.

Number of bits in each register

- IR
- TR
- DR
- AC

16-bit Register

- PC
- AR

12-bit Register

- INPR
- OUTR

8-bit Register

Memory unit

- Used to store programs and data
- Volatile
- Usually uses DRAM
  - Slower than static RAM
  - Must be refreshed
  - Requires fewer transistors to implement
- Most memory is byte addressable
- Can be organized to access a full word or even multiple words per access
- Cache memory is a distinct memory positioned between the CPU and Main memory
  - Faster
  - Smaller
  - More expensive

Computer instructions

- The basic computer has three instruction code formats

15	14–12	11–0
I	Opcode	Address

Memory reference instruction

15	14		12	11–0
0	1	1	1	Register Operation

Register reference instruction

15	14		12	11–0
1	1	1	1	I/O Operation

Input-output instruction

Computer instructions cont

- A register reference instruction specifies an operation or a test of the AC register. An operand from memory is not needed, therefore the other 12 bits are used to specify the operation or test to be executed.
- An input output instruction also doesn’t need a reference to memory, the 12 bits are used to specify the input-output operation or test performed.
- The total number of instructions is 25

Computer Instruction

- Memory-reference instruction
  - Opcode = 000 ~ 110

    I=0 : 0xxx ~ 6xxx, I=1: 8xxx ~Exxx

I=0 : Direct,
I=1 : Indirect

15	14 12	11 0
I	Opcode	Address

- Register-reference instruction
  - 7xxx (7800 ~ 7001) : CLA, CMA,

15	14	12	11 0
0	1	1 1	Register Operation

- Input-Output instruction
  - Fxxx(F800 ~ F040) : INP, OUT, ION, SKI, ….

15	14	12	11 0
1	1	1 1	I/O Operation

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 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
CMS	7200		Complement AC
CME m	7100	e	Comp
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		Interrup
IOF	F040		Inter

Instructions are read from memory as words

- Instructions can be formatted to fit in one or more memory words.
- An instruction may contain
  - An opcode + data (immediate operand)
  - An opcode + the address of data (direct addressing)
  - An opcode + an address where the address of the data is found (indirect addressing)
  - Data only (location has no instructions)
  - An opcode only (register-reference or input/output instruction)

Instruction Set Completeness

- Arithmetic, Logical, and shift : CMA, INC
  - Provide computational capabilities for processing the data
- Moving information to and from memory and AC : STA, LDA
  - The bulk of the binary info is stored in memory but all the computations are done in the processor registers. Therefore information is to be moved between these units
- Program control and instructions that check the status conditions: BUN, BSA, ISZ
  - Used to change the sequence in which the program is executed
- Input/Output : INP, OUT
  - Needed for communication between the comp and the user.

Timing and Control

- Clock pulses
  - A master clock generator controls the timing for all registers in the basic computer
  - The clock pulses are applied to all flip flops and registers in system
  - The clock pulses do not change the state of a register unless the register is enabled by a control signal
  - The control signals are generated in the control unit
  - The control signals provide control inputs for the multiplexers in the common bus, control inputs in processor registers, and microoperations for the accumulator

Control organization

- Two major types of control organization
  - Hardwired Control :
    - The control logic is implemented with gates, flip flops, decoders, and other digital circuits
    - + Fast operation, - Wiring change(if the design has to be modified)
  - Microprogrammed Control :
    - The control information is stored in a control memory, and the control memory is programmed to initiate the required sequence of microoperations
    - + Any required change can be done by updating the microprogram in control memory, - Slow operation

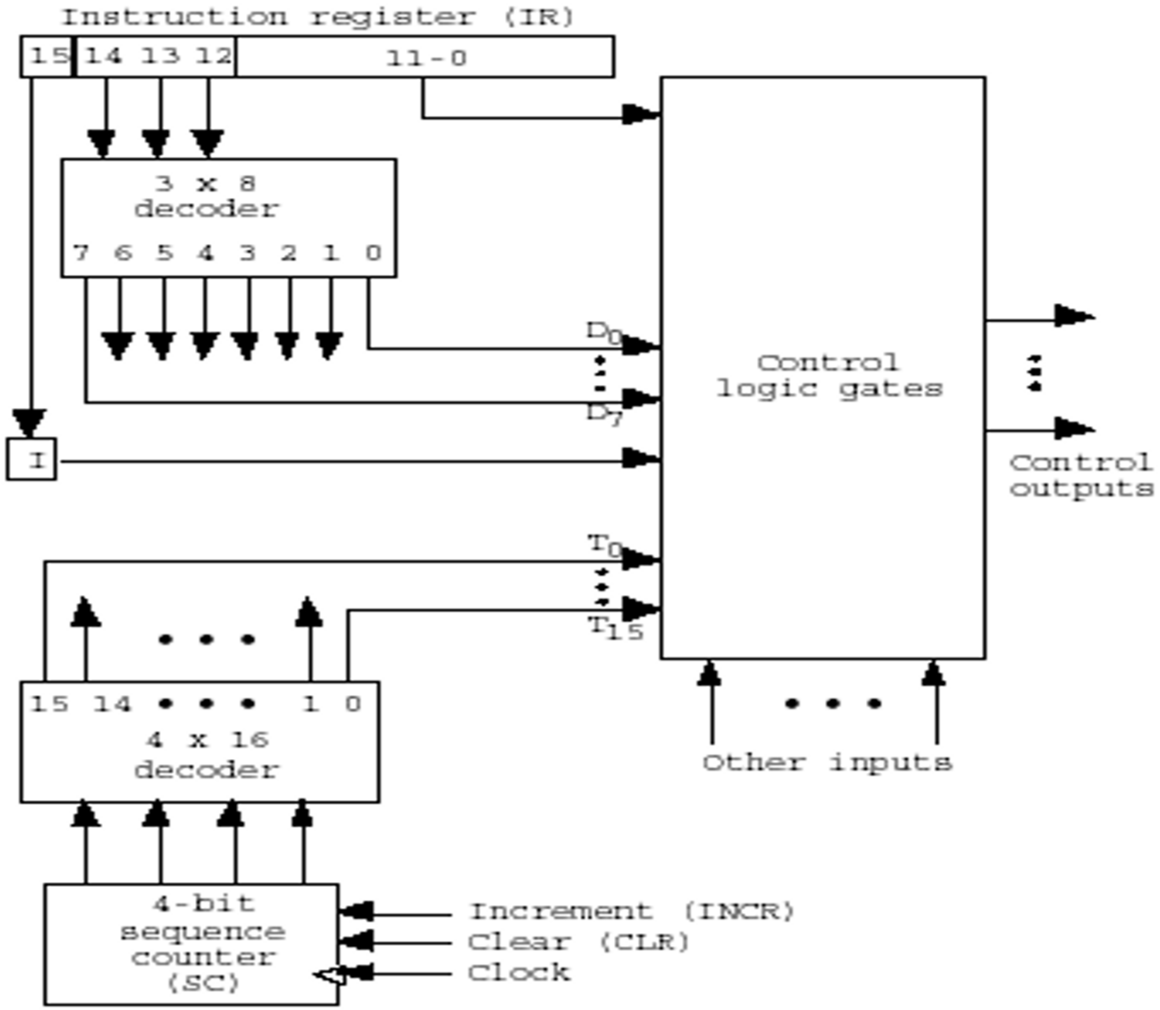


Figure: Control unit hardware.

Sequence Counter

- Inputs are increment (INR) and clear (CLR)
- Most of the time, the counter is incremented to provide the sequence of timing signals out of the 4 * 16 decoder.
- When the counter is cleared to 0, next active timing signal is T_0
- If SC is not cleared, the timing signals will continue with T_5, T_6 upto T_{15} and back to T_0
- Example
  - SC incremented to provide T_0, T_1, T_2, T_3, and T_4
  - At time T_4, SC is cleared to 0 if D_3 is active
  - Written as: D_3T_4: SC $\leftarrow$ 0

Timing Diagram