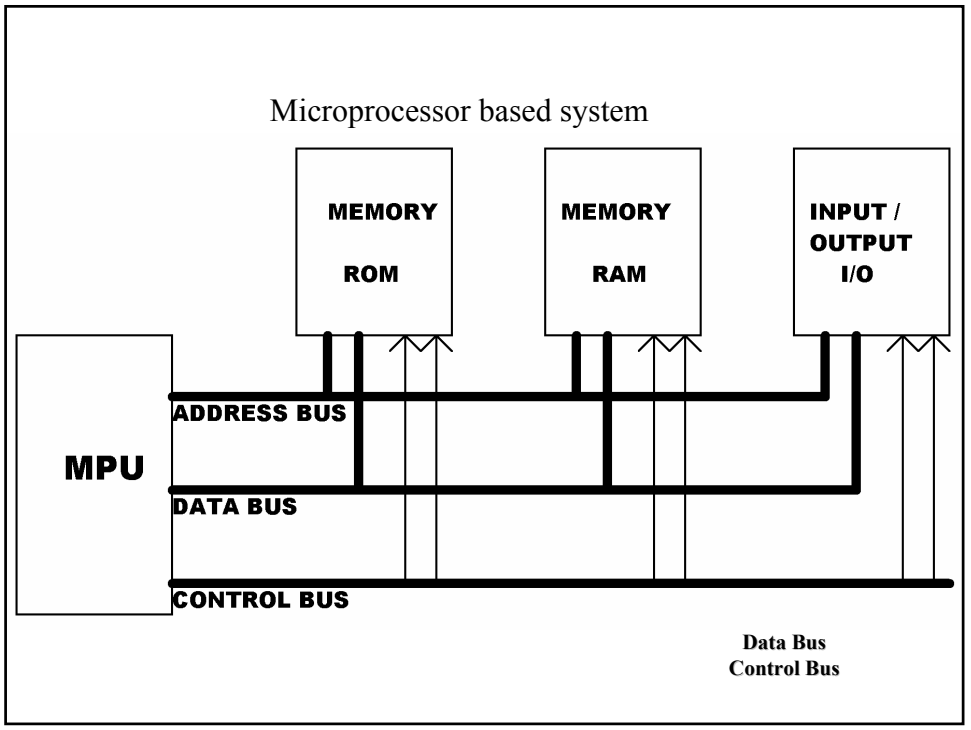
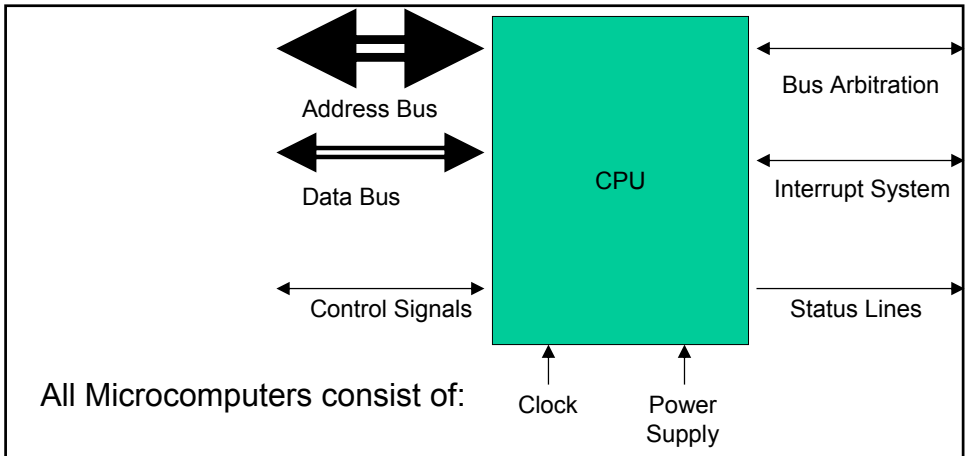




All Microcomputers consist of:

1. Microprocessor Unit (MPU)
2. Program Memory (ROM)
3. Data Memory (RAM)
4. Input / Output System
5. Bus System

(and Software)

The microprocessor is a single chip which is capable of processing data and controlling all of the components which make up the microcomputer system. It contains all of the circuits needed to create the 'brain of the microcomputer'.

1. Temporary storage locations in the form of a number of registers which can hold binary values (information), representing program instruction or data.

2. The Arithmetic Logic unit (ALU). This part of the MPU performs both arithmetic and logical operations

3. Timing and Control Circuits : that keep all of the other parts of system (memory & I/O) working together in the right time sequence.

A Bus is a common communications pathway used to carry information between the various elements of a computer system The term BUS refers to a group of wires or conduction tracks on a printed circuit board (PCB) though which binary information is transferred from one part of the microcomputer to another

There are three main bus groups

1. ADDRESS BUS

2. DATA BUS

3. CONTROL BUS

Data Bus:
The Data Bus carries the data which is transferred throughout the system. (bi-directional)
Address Bus:
An address is a binary number that identifies a specific memory storage location or I/O port involved in a data transfer

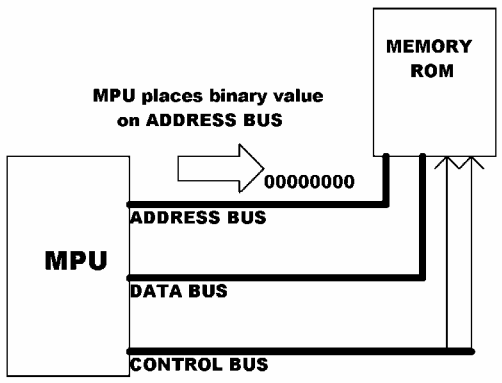
The Address Bus is used to transmit the address of the location to the memory or the I/O port.
The Address Bus is unidirectional (one way): addresses are always issued by the MPU.

The Control Bus: is another group of signals whose functions are to provide synchronization (timing control) between the MPU and the other system components. Control signals are unidirectional, and are mainly outputs from the MPU. Example Control signals
RD read signal asserted to read data into MPU
WR write signal asserted to write data from MPU

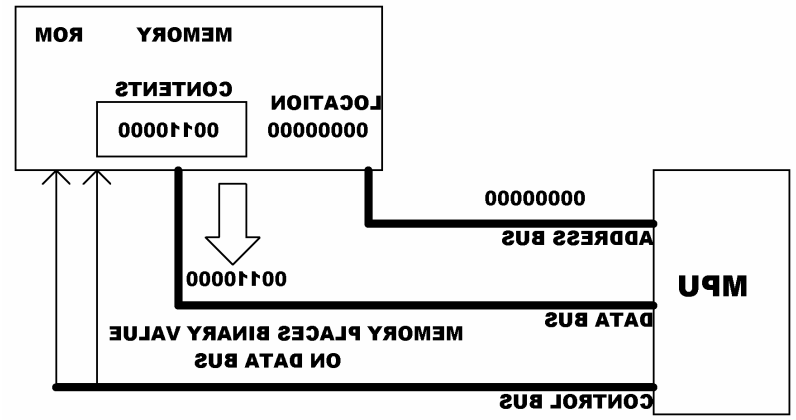
A Simplified view of MPU program execution

A MPU program is stored in memory as a sequence of binary values.
These binary values represent instructions. Each instruction has an unique binary value.
The MPU operation is controlled by reading and executing each instruction one at a time.

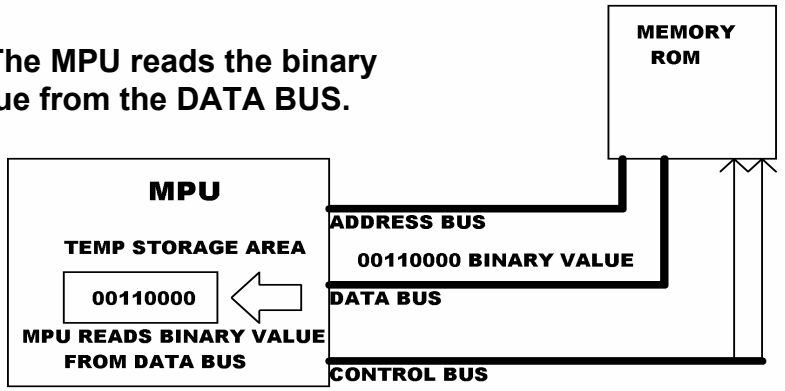
1. The MPU places an address on the ADDRESS BUS.
The address is a binary value which represents a unique memory location.



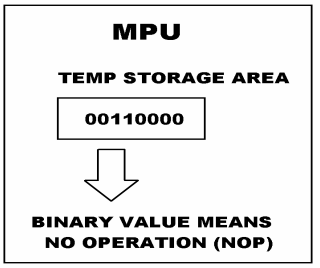
2. The Memory places the contents of the addressed location onto the DATA BUS. The contents is a binary value.



3. The MPU reads the binary value from the DATA BUS.



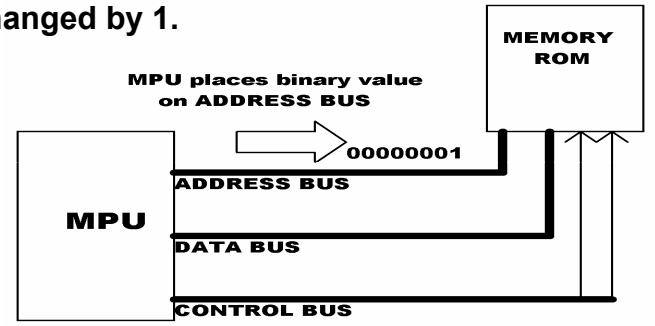
4. The MPU interprets what operation the BINARY Value represents and completes (executes) this operat



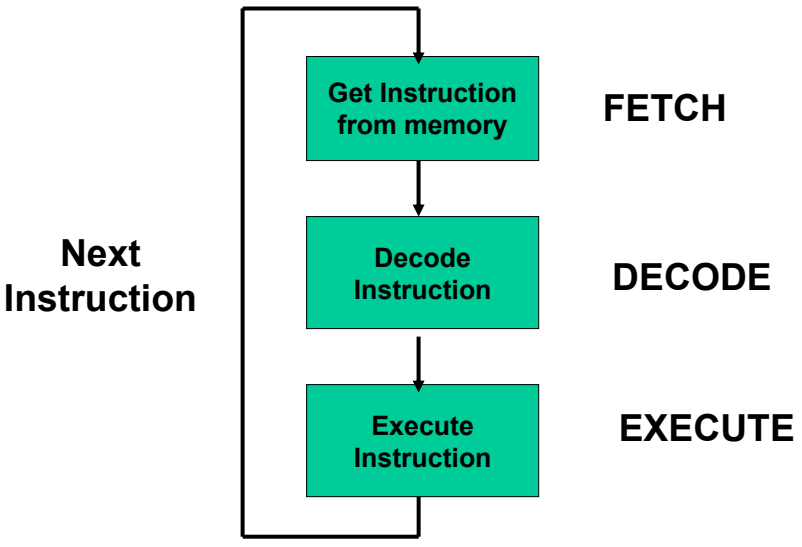
5. The MPU has completed its first instruction of the program stored as binary values in the memory.

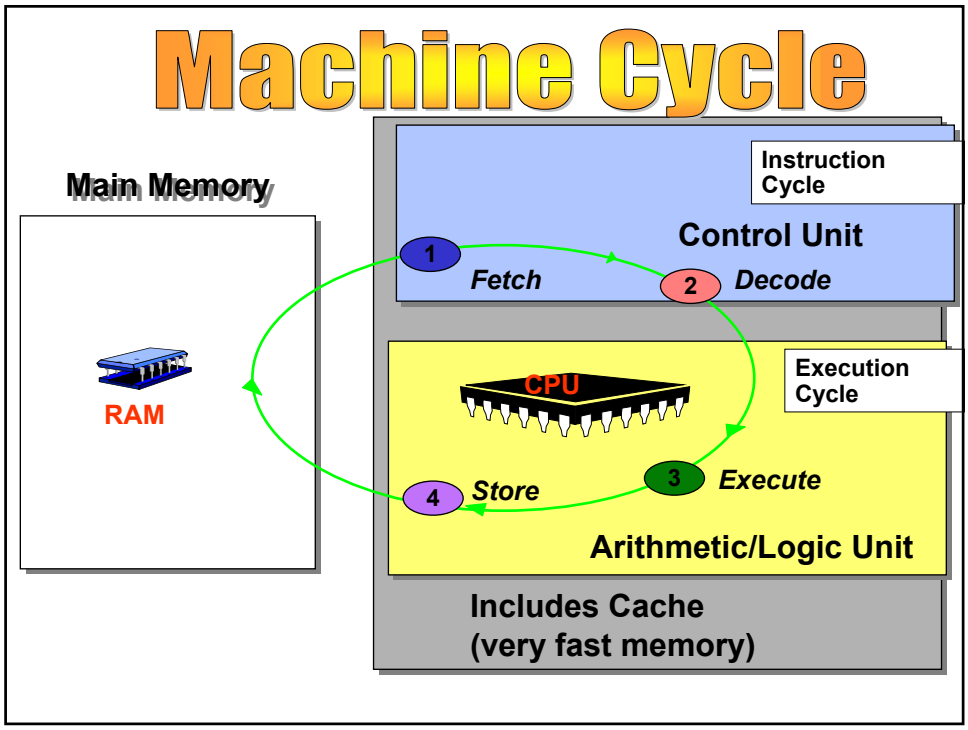
The MPU needs to be told what to do next. The next instruction is in the next memory location. Stages 1. through 4. are repeated,

Note. At Stage 1. The MPU places a new binary value onto the ADDRESS BUS. This is the memory location of the next instruction. The binary value on the ADDRESS BUS has only changed by 1.



All common microprocessor follows this sequence

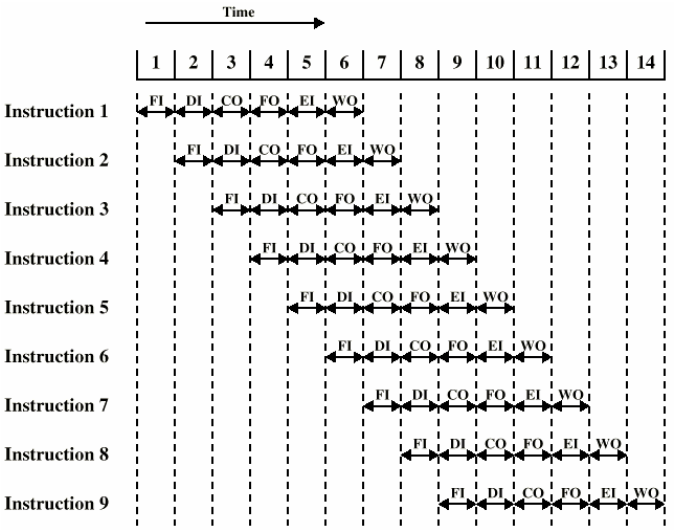




CPU Activities

- CPU is in charge of the fetch-execute cycle to execute all program instructions
  - This is not quite true, I/O operations are initiated by the CPU but the actual sequence of operations is often left up to I/O modules such as I/O channels or I/O processors
- **Fetch-execute cycle:**
  - Fetch instruction – CPU reads next program instruction from memory
  - Interpret (decode) instruction – control unit determines what machine actions are required including what data to fetch
  - Fetch data – data is fetched from memory or I/O
  - Process data – ALU operation or data movement
  - Write data – if needed, result stored in memory or I/O

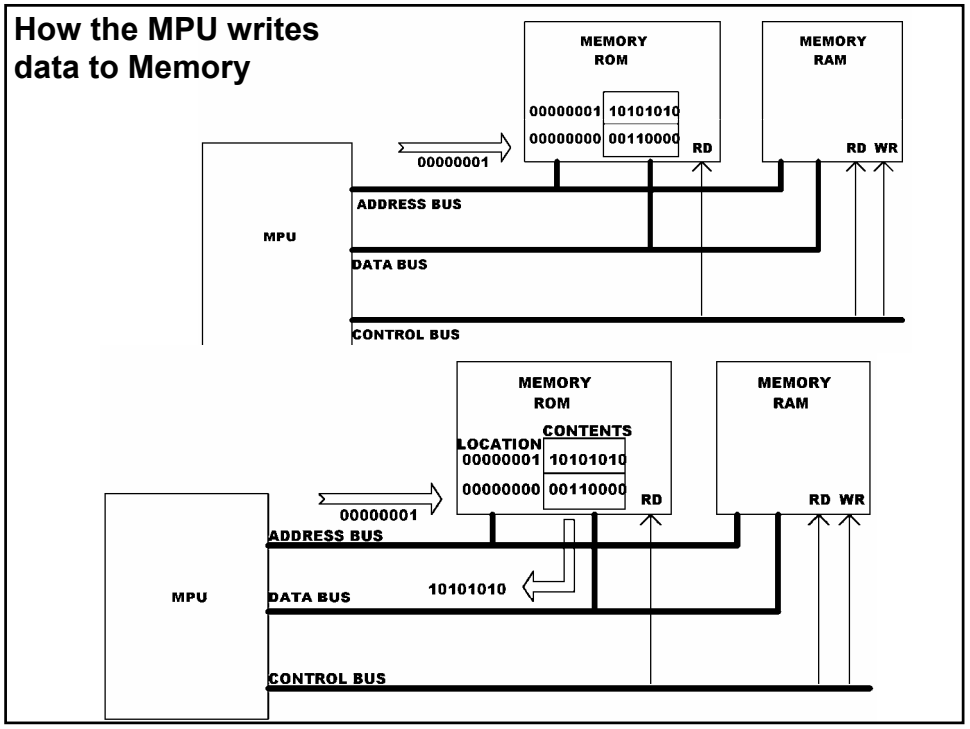
Pipeline Timing Diagram

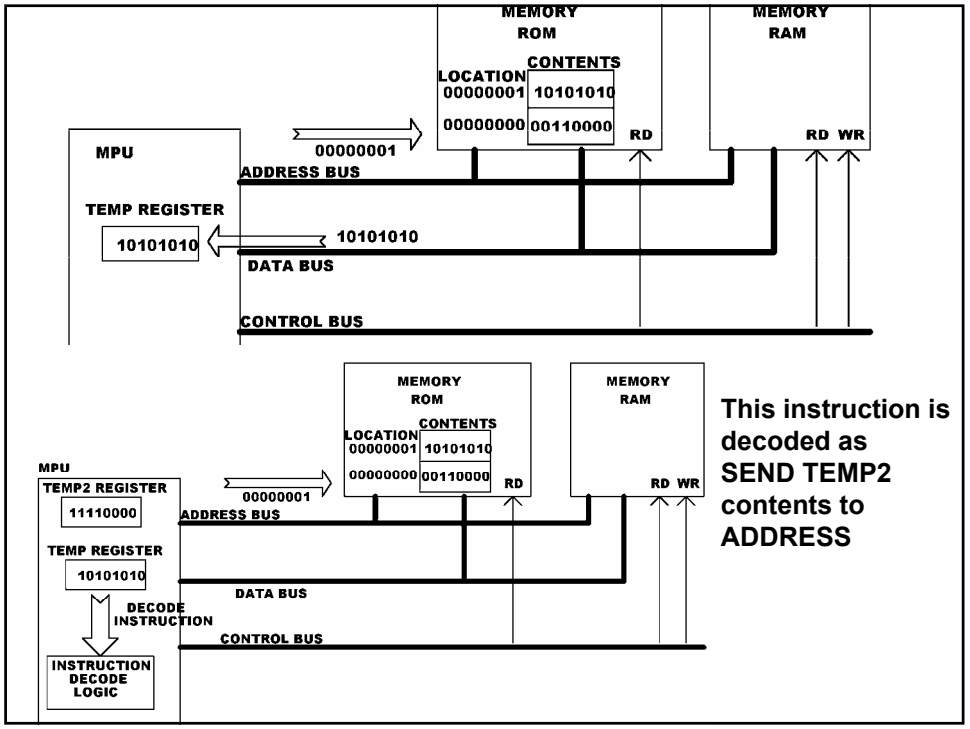


Pipeline Stages

- Different architectures use different number of stages
  - smaller pipelines have less complexity
  - larger pipelines are generally faster
  - here, we decompose the fetch-execute cycle into 6 stages:
    - FI – fetch instruction
    - DI – decode instruction
    - CO – calculate operand addresses
    - FO – fetch operands (from registers)
    - EI – execute instruction (includes computing branch results and all ALU operations)
    - WO – write or store result in registers (or memory)
  - the design of the pipeline stages has a major impact on the architecture's performance

How the MPU writes data to Memory



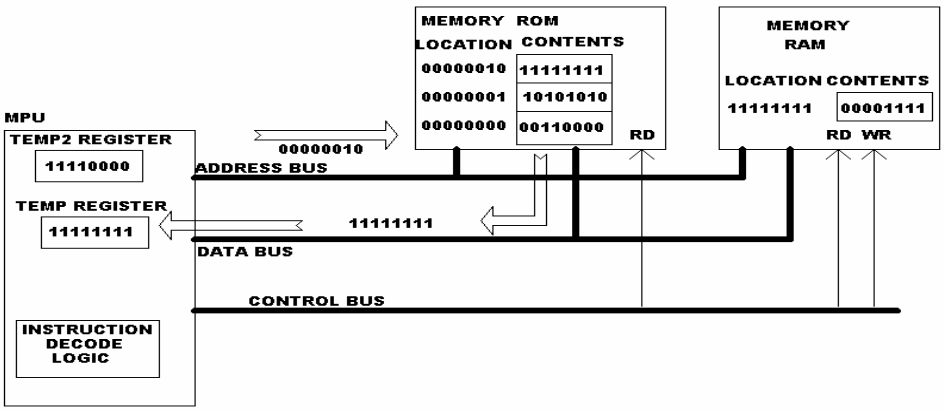


This instruction is decoded as SEND TEMP2 contents to ADDRESS

The MPU instruction decode logic now need to execute this instruction
SEND TEMP2 to an ADDRESS

TO WHAT ADDRESS …

This particular MPU instruction decodes as saying that the DESTINATION ADDRESS will be in the next memory location. The MPU then fetches the binary value from the next location, as this will be the ADDRESS where TEMP2 contents will be SENT.

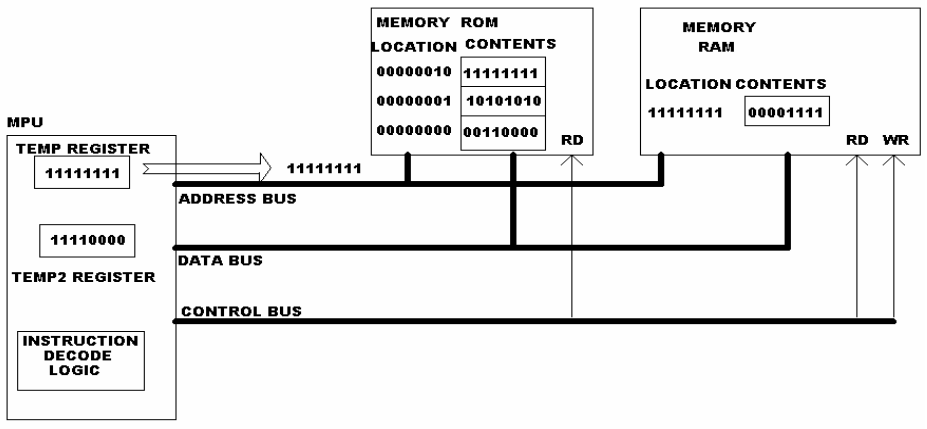


The Destination ADDRESS data has been fetched from memory and stored in a TEMP register.
WHAT HAPPENS NEXT ?

The TEMP register holds the DESTINATION ADDRESS where we wish the SEND our DATA (Binary Value) which is stored in TEMP2 register.

We do not try to decode the contents of TEMP because it is not an instruction, it is data associated with a specific operation, it is known as an OPERAND

WHAT HAPPENS NEXT ..
Remember we want TEMP2 contents to be sent to the MEMORY ADDRESS which is currently held in TEMP.



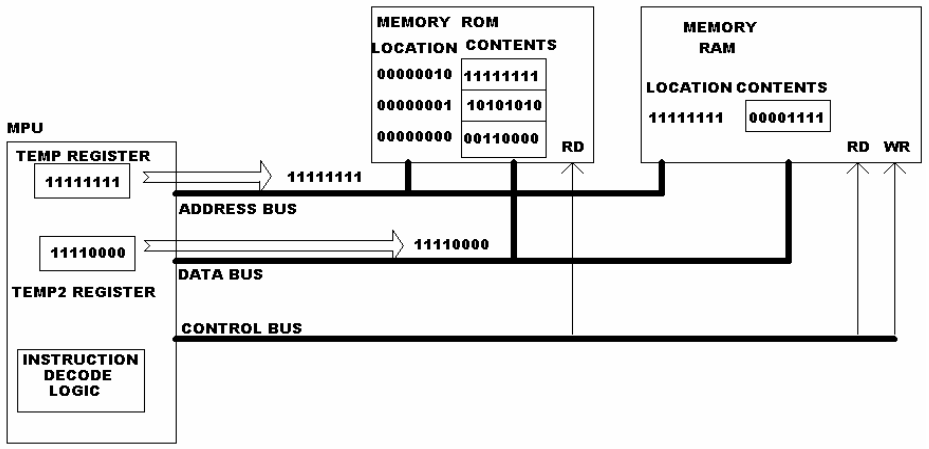
The contents of TEMP register are placed on the ADDRESS BUS.

The MEMORY WRITE part of our instruction has started. In this phase, since we are sending DATA to MEMORY or WRITING TO MEMORY we must be accessing a MEMORY store which allows us to do so.

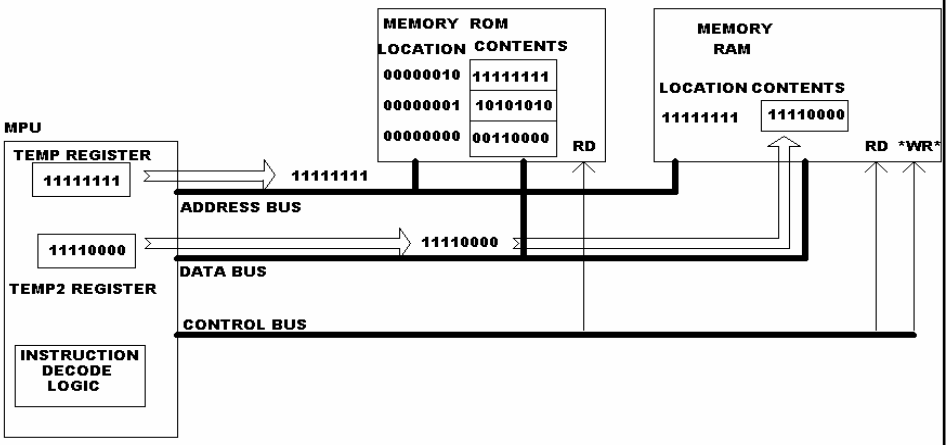
ROM is read only memory, where our program is stored, we cannot write to it.

The RAM random access memory ,(read/write memory) is where our DATA will be written.

Notice the control signal lines on each of the memory types.



The ADDRESS is a memory location within the RAM, the TEMP2 register places its contents onto the DATA BUS.



The RAM reads the DATA BUS value and stores it at the location which the ADDRESS value is accessing. NOTICE THE WR signal is ACTIVE

The instruction has now completed executing The result of this operation is that the contents of TEMP2 register have been WRITTEN to a RAM Memory location 11111111.

Prior to the DATA WRITE the contents of RAM location was 00001111. After the MEMORY WRITE RAM Memory location now contains 11110000 the previous DATA value has been overwritten.

A Quick Review :

The MPU fetches instructions one at a time from memory The instruction is interpreted by MPU and then this operation is executed. This called the FETCH DECODE EXECUTE CYCLE. The program, sequence of instructions are held in consecutive memory locations.

We have seen that not every binary value read from memory is an instruction. Some of the values can be DATA associated with an instruction.

The instructions are known as Operation Codes OPCODES.

The Data and Registers associated with a specific OPCODE are called OPERANDS.

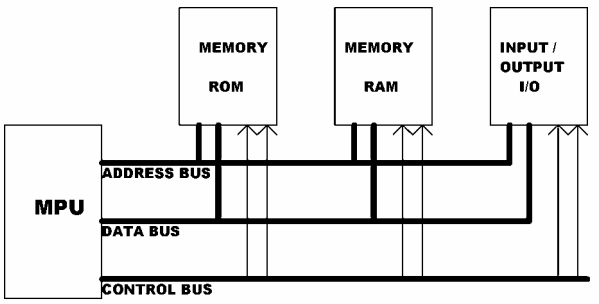
The format of these OPCODES and OPERANDS are directly related to the internal architecture of the MPU.

The question is what does the MPU do next, we have seen that a single instruction can cause several BUS accesses. But we have completed the current instruction. The MPU needs to get the NEXT instruction from memory where our program resides.

Examples

- MOV BL,AL
- Opcode for MOV = 100010
- We'll encode AL so
  - D = 0 (AL source operand)
- W bit = 0 (8-bits)
- MOD = 11 (register mode)
- REG = 000 (code for AL)
- R/M = 011

MOV AL,BL = 10001000 11000011 = 88 C3h
ADD AX,[SI] = 00000011 00000100 = 03 04 h
ADD [BX],[DI] + 1234h, AX = 00000001 10000001 __ __ h
= 01 81 34 12 h



Bus Sizes: Buses are specified in width, number of bits. There are other parameters which can also used to characterise a Bus but for this presentation they are not relevant.

Address Bus :
The MPU Address bus size specifies how much memory and I/O locations may accessed.

Number of locations = 2^N where N is number of Address lines

The Address Bus may be specified using the following
16 Bits Wide, (A15 to A0), 16 Address Lines

In this instance $2^N = 2^{16} = 65536$ locations

The Address Range starts at 0 and ends at a maximum 65535.
Notice that the highest address location is one less than the Number of locations given by 2^N .

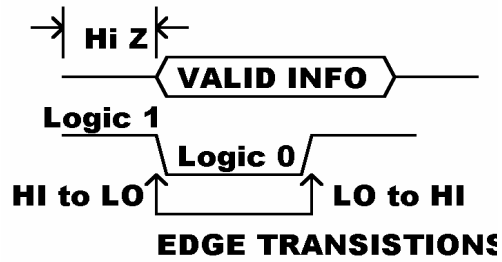
This is due to the number 0 being a valid memory location.

Addresses are general expressed in the Hexadecimal numbering system. Which will be covered later. This number system is base 16. So that a typical 16 bit address will use values such as 1234H or 1234_{16} or FFFFH or $FFFF_{16}$ more later.

The Data Bus is similarly characterised by its width, number of bits.
The width of the external Data bus generally gives a very good indication of the internal MPU architecture, particularly the registers size and the ALU. However, there are exceptions

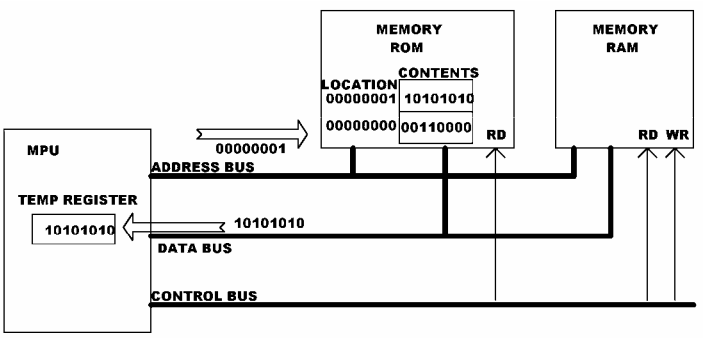
The Data Bus width limits the bandwidth of the data that can be transferred per bus operation.

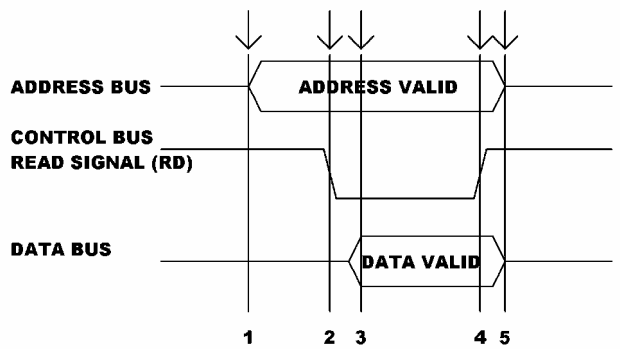
Data Bus 8 Bits , tends to represent a 8 bit internal architecture.



Hi Z means High Impedance or floating bus / signal.

VALID INFO means either a group of BUS signals or an individual signal is Stable at this time.



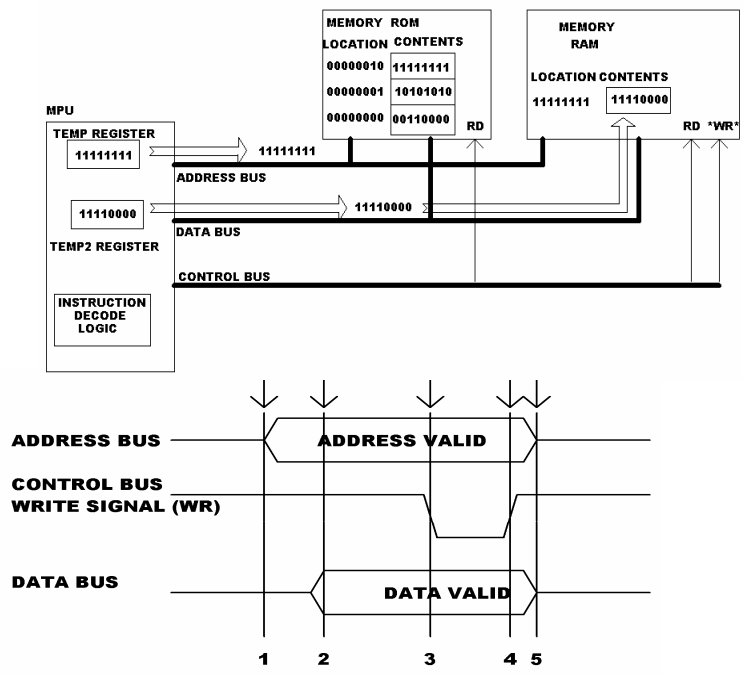


Simple Read Cycle

1. Address onto Address Bus
2. READ (RD) Control Signal Asserted by MPU
3. Data Placed from Memory onto the Data Bus
4. RD signal De-Asserted, Data into MPU temp register.
5. Read Cycle Ends

Write Cycle

1. Address placed on ADDRESS BUS
2. Data placed onto DATA BUS
3. Write (WR) Control Signal Asserted by MPU
4. Write (WR) Control Signal De-Asserted and Data Written into Memory
5. Write Cycles completes.



–CPU has four key parts that we will examine:

- **Control Unit**
- **Arithmetic & Logic Unit**
- **Registers**
- **Clock**

The MPU architectural
The MPU architecture can be viewed both externally or internally. (
The MPU architecture refers to both its Hardware and Software features

—**CONTROL UNIT** (CU): circuitry for **coordinating** machine's activities.

—**ARITHMETIC & LOGIC UNIT** (ALU): circuitry to perform data **manipulation** (**arithmetic** & **logic**).

—**Registers**: Temporary holding places. Holds information applicable to the _____

—**Clock**: Times all CPU ***operations***. (store, retrieve, compare, and mathematical operations)

The basic function as stated is to perform logic and maths operations.
The Accumulator and the Flags registers play an important role in the operation of the ALU.

The Accumulator is used in all ALU operations.

The Flags or Status Register is a group of bits representing the result of the ALU operation. The Flags register contains bits such as OV representing an overflow flag. If the result of an addition is greater than 255 then the OV is set to indicate this condition. More Details to come.
The type of functions performed by the ALUs of MPU include

Addition Shift Left Subtraction Shift Right

Logical AND Decrement Logical OR Increment Logical XOR