

Computer Systems Organization

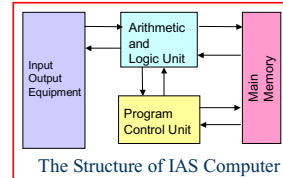


School of Computer Science GS1CSA

1

The IAS (von Neumann) Machine

Stored Program concept
Main memory storing programs and data
ALU operating on binary data
Control unit interpreting instructions from memory and executing
Input and output equipment operated by control unit



1946 ~ 1952
John von Neumann
Princeton
Institute for Advanced Studies

Almost all of today's computers have the same general structure as the IAS - referred to as **von Neumann machines**.

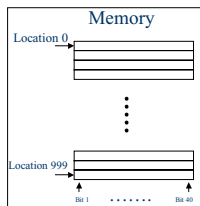
The Structure of IAS Computer



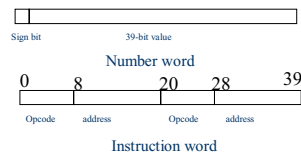
School of Computer Science GS1CSA

2

The IAS Machine: Memory



IAS Memory
- 1000 storage locations (words), each 40 bits
- Both data and instruction are stored in the memory

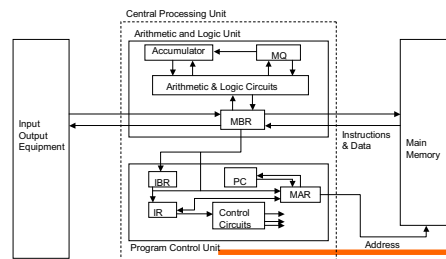


School of Computer Science GS1CSA

3

The IAS Machine: Control Unit

The control unit operates the machine by fetching instructions from memory and executing them **ONE** at a time.



School of Computer Science GS1CSA

4

The IAS Machine: Instruction Cycle

The IAS operates by repetitively performing an *instruction cycle*.

Two sub-cycles:

- During the *fetch cycle*, the opcode of the NEXT instruction is loaded in to the IR and the address portion is loaded into the MAR
- Once the opcode is in the IR, the *execute cycle* is performed. Control circuitry interprets the opcode and executes the instruction by sending out appropriate control signals to cause data to be moved or an operation to be performed by the ALU.



School of Computer Science GS1CSA

5

Von Neumann Architecture

- ♣ Data and instructions are stored in a single read-write memory
- ♣ The content of this memory are addressable by location, without regard to the type of data contained in it.
- ♣ Execution occurs in a sequential fashion (unless explicitly modified) from one instruction to the next.



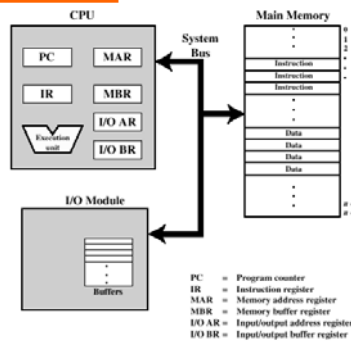
School of Computer Science GS1CSA

6

Computer Components

- A computer consists of a set of modules of three basic types:

- They communicate with each other
- Needs connection paths for connecting the modules

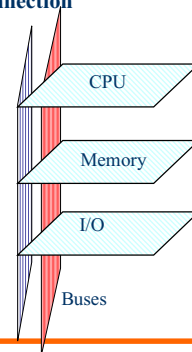


School of Computer Science GS1CSA

7

Bus Interconnection

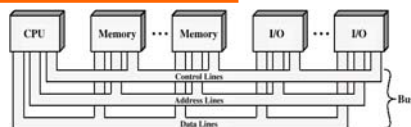
- A bus is a communication path that connects two or more devices
- A bus is a shared transmission medium.
- A signal transmitted by one device is available for reception by all other devices attached to the bus
- If two devices transmit during the same period, their signal will overlap and become garbled.
- Only one device can successfully transmit at any one time.



School of Computer Science GS1CSA

8

Bus Interconnection



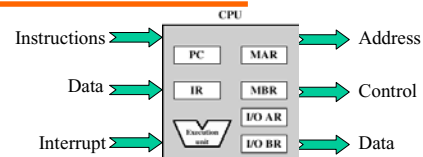
- Data Lines (Data Bus, DB)
- Address Lines (Address Bus, AB)
- Control Lines (Control Bus, CB): Control the access to and the use of DB and AB. (remember AB and DB shared by all devices)
- CB send out both command and timing signals
 - Command: specify the type of operation (R/W)
 - Timing: Indicate the validity of data on DB and AB
- Typical control lines include: Memory R/W, I/O R/W, Bus request, Bus grant, Interrupt request, Interrupt grant, Transfer ACK



School of Computer Science GS1CSA

9

Computer Components: CPU module



- Read instructions and data
- Write out data after processing
- Use control signal to control the overall operation of the computer
- Receive interrupt signal



School of Computer Science GS1CSA

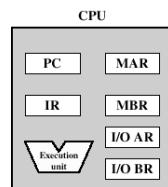
10

Computer Components: CPU module

- Registers
 - Permanent storage locations within the CPU
 - Each used for a particular, defined purpose
- Accumulator
 - General purpose register
- Registers in the Control Unit
 - PC - program counter register
 - IR - Instruction register
 - MAR - memory address register
 - MBR - memory buffer register
 - I/O AR - I/O address register
 - I/O BR - I/O buffer register
 - Status register

Move data

Manipulate data



School of Computer Science GS1CSA

11

Fetch-Execute Cycle

- Basis for programmable processors
- Allows processor to move through program steps
- Automatically Implemented by processor hardware
- Note: At some level, every programmable processor implements a fetch-execute cycle.



School of Computer Science GS1CSA

12

13

14

19

16

12

18

Memory System Architecture

- Primary memory is a collection of independent storage units. Each unit stores a single multi-bit value.
- The number of bits in a storage unit is a constant for all storage units in the memory system, and this constant is called the memory width.
- Addresses are used to access the storage units in the memory system. Each storage unit has a unique address.



School of Computer Science G51CSA

19

Memory System Architecture

A graphical representation of a memory with 128 storage units and width 8.

The memory is called 128 x 8 memory.

Address	Memory Content
1	1 0 1 0 1 1 1 0
2	1 1 1 1 0 1 0 1
3	1 1 0 1 1 0 0 0
•	•
•	•
•	•
128	1 0 1 0 1 1 1 1



School of Computer Science G51CSA

20

Memory System Architecture

Manufacturers produce a number of different types of memory devices having a variety of technologies.

The technology affects not only the operating characteristics, such as power consumption, size, and speed, but also the manufacturing cost.

Thus in the selection of memory chips for a particular application, designers must weigh the trade-offs between cost and performance.



School of Computer Science G51CSA

21

Memory System Architecture

Read-only memory

Read-only memories (ROMs) are memory devices that the CPU can read but cannot write.

Many ROMs are factory programmed and there is no way to alter their contents (the term programming here means writing values into a ROM). These devices are denser and cheaper to manufacture than other type of ROM.

Programmable ROMs (PROMs): This type of ROM can be programmed by using special high current device to destroy (burn) the fuse that were manufactured into the device. The result of burning a PROM is that certain bits are always 0 and the rest are always 1. These values cannot be altered once written.

Erasable PROMs (EPROMs): This type of ROM is alterable, although not during ordinary use. A technician can program an EPROM off line, later completely erase its contents by using ultraviolet light, and then reprogram it.



School of Computer Science G51CSA

22

Memory System Architecture

Read/Write Memory

Read/Write memories refer to memory devices can be read from and write into with equal ease. Two main types of read/write memory devices are **static** random access memories (SRAMs) and **dynamic** random access memories (DRAMs).

SRAMs: In SRAMs, the individual memory contents, once written, do not need to be further addressed or manipulated to hold their values. These devices are composed of flip-flops that use a small current to maintain their contents. SRAMs are used mostly in CPU registers and other high speed storage devices. Some computers use them for **cache** and main memory. SRAMs are currently the fastest and most expensive of semiconductor memory circuit.

DRAMs: These are semiconductor memory devices in which the stored data will not remain permanently stored, even with power applied, unless the data are periodically rewritten into the memory. The latter operation is called the *refresh* operation. Although much cheaper than SRAMs, DRAMs are also slower and used mostly for main memory



School of Computer Science G51CSA

23

Memory System Architecture

General Memory Operation

Although each type of memory is different in its internal operation, there are certain basic operating principles that are the same for all memory systems. Every memory system requires several different types of input and output lines to perform the following function:

Select the address in memory that is being accessed for READ or WRITE operation

Select either READ or WRITE operation to be performed

Supply the input data to be stored in memory during write operation

Hold the output data coming from memory during a read operation

Enable (or disable) the memory so that it will (or will not) respond to the address inputs and read/write command.



School of Computer Science G51CSA

24

The Fetch-Store Paradigm

- └ Access paradigm used by memory
- └ Two operations
 - – Fetch the value from a specified location
 - – Store a value into a specified location
- └ Two operations also called
 - – Read
 - – Write
- └ We will discuss implementation of fetch / store later

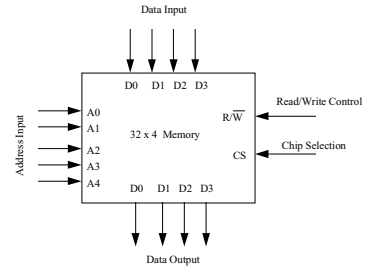


School of Computer Science GS1CSA

25

Memory System Architecture

General Memory Operation



School of Computer Science GS1CSA

26

Physical Memory Addresses

- └ Each word of memory is assigned a unique number known as a *physical memory address*
- └ Programmer imagines physical memory to be an array of words
- └ Note: entire word must be transferred

Physical Memory And Word Size

- └ Bits of physical memory are divided into blocks of N bits each
- └ Terminology
 - – Group of N bits is called a *word*
 - – N is known as the *width* of a word or the *word size*

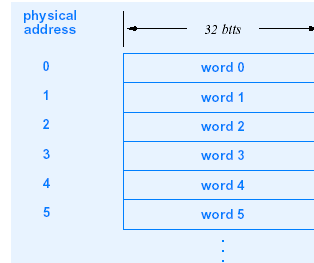


School of Computer Science GS1CSA

27

Physical Memory Addresses and Word Size

- └ Word size = 32 bits



Physical memory is organized into words, where a word is equal to the memory transfer size (number of bits can be transferred simultaneously). Each read or write operation applies to an entire word.

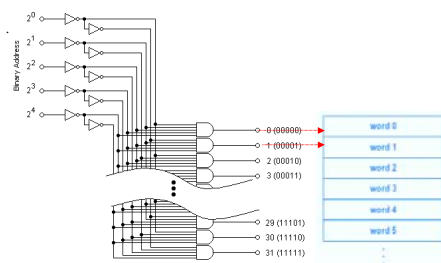


School of Computer Science GS1CSA

28

Memory System Architecture

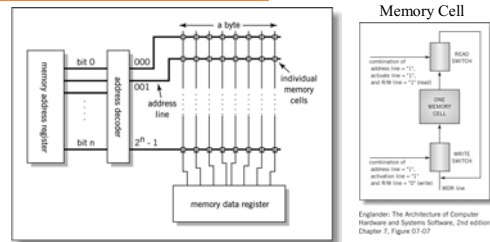
Address Decoder



School of Computer Science GS1CSA

29

Computer Components: Memory module



Englander: The Architecture of Computer Hardware and Systems Software, 2nd edition Chapter 7, Figure 07-04

The relationship between the MDR, the MAR, and memory

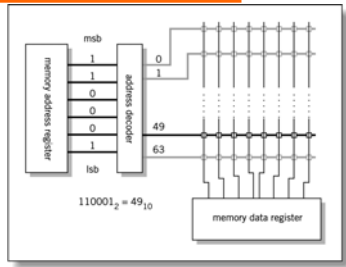
Address Decoder



School of Computer Science GS1CSA

30

Computer Components: Memory module



Englander: The Architecture of Computer Hardware and Systems Software, 2nd edition Chapter 7, Figure 07-05

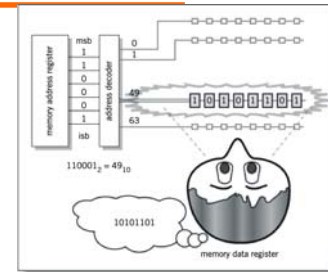
MAR-MDR example



School of Computer Science GS1CSA

31

Computer Components: Memory module



Englander: The Architecture of Computer Hardware and Systems Software, 2nd edition Chapter 7, Figure 07-06



School of Computer Science GS1CSA

32

Computer Components: Memory module

Memory Address Space and Memory Map

⚡ The total amount of memory contained in any system is limited by the size of the address bus.

Example: A 6502 processor has 16-bit address bus, what is the maximum amount of memory which a system can utilise?

⚡ The number of unique addresses of is known as *address space*

⚡ Note: word addressing allows larger memory than byte addressing



School of Computer Science GS1CSA

33

Computer Components: Memory module

Memory Map

The microcomputer designer has to allocate this address space among the RAM, ROM, and I/O devices that are to be part of the system.

The manner in which the total address space is apportioned among these devices depends, to certain extent, on characteristics of the processor.

A **memory map** is a simple diagram which identifies the size and location of any memory block in the total address space.



School of Computer Science GS1CSA

34

Computer Components: Memory module

FFFF	ROM 1K
FC00	Not Used
B0FF	I/O 256
B000	Not Used
07FF	RAM 2K
0000	

A typical memory map for a microprocessor system with 16-bit address bus



School of Computer Science GS1CSA

35

Computer Components: Memory module

IBM PC Main Memory Map

Main memory, also called conventional memory, refers to the storage locations that the CPU can reference during an ordinary memory-read or memory-write bus cycle without special hardware.

The amount of main memory a PC could directly address to is 1MB (2²⁰ bytes).

The PC architects divided the address space of conventional memory into a number of blocks, which they allocated for various software components.

They allocated the largest block, with addresses ranging from 0K to 640K, to program memory, which they implemented with DRAM chips.

They reserved the remaining block, with address ranging from 640K and 1024K, for ROM BIOS and other system components. The following table summarises the allocation of addresses within conventional memory of a PC.



School of Computer Science GS1CSA

36

Computer Components: Memory module

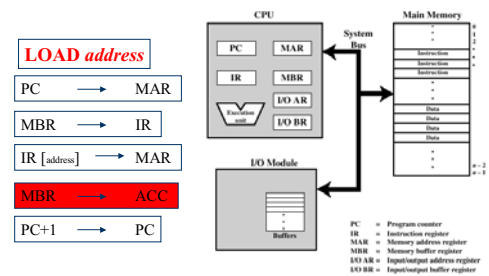
Address	PC Usage
960K - 1024K	ROM BIOS
880K - 960K	Unused
848K - 880K	LIM data area
816K - 848K	LIM data area
800K - 816K	Hard disk ROM
784K - 800K	Unused
768K - 784K	EGA ROM
752K - 768K	Unused
736K - 752K	CGA
720K - 736K	Unused
704K - 720K	MDA
640K - 704K	EGA or VGA



School of Computer Science G51CSA

37

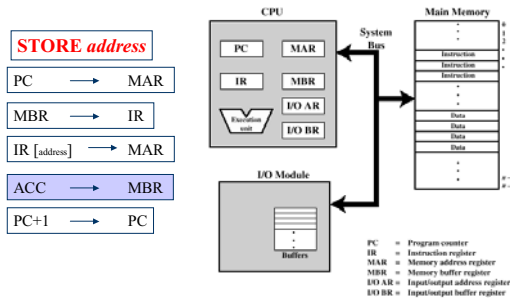
The Fetch-Execute Instruction Cycle



School of Computer Science G51CSA

38

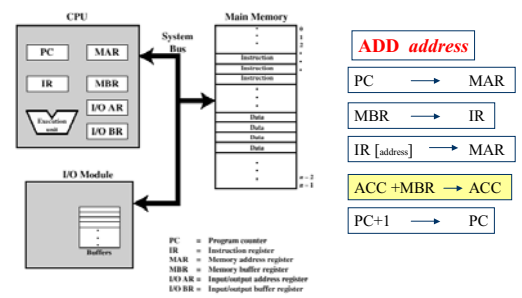
The Fetch-Execute Instruction Cycle



School of Computer Science G51CSA

39

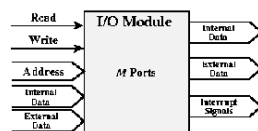
The Fetch-Execute Instruction Cycle



School of Computer Science G51CSA

40

Computer Components: I/O module



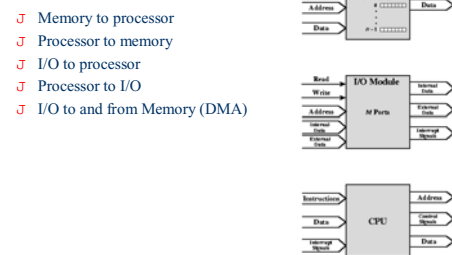
- From an internal (to the computer system) point of view, I/O is functionally similar to memory
- There two operations read and write
- I/O module may control more than one devices.
- The interfaces to each external devices is referred to as port, and each port is given a unique address
- External data path for input output data with external devices
- I/O module may be able to send interrupt signals to the CPU



School of Computer Science G51CSA

41

Computer Components: Data Exchange

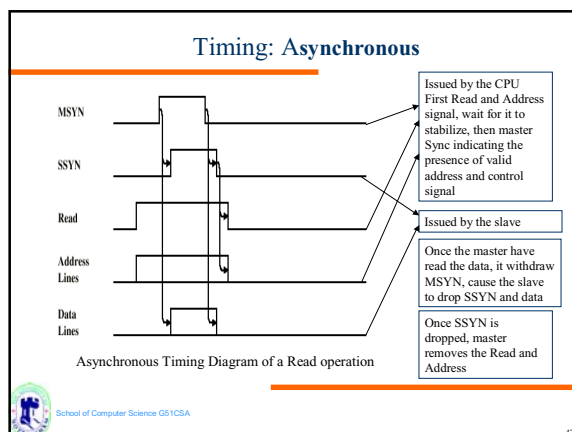
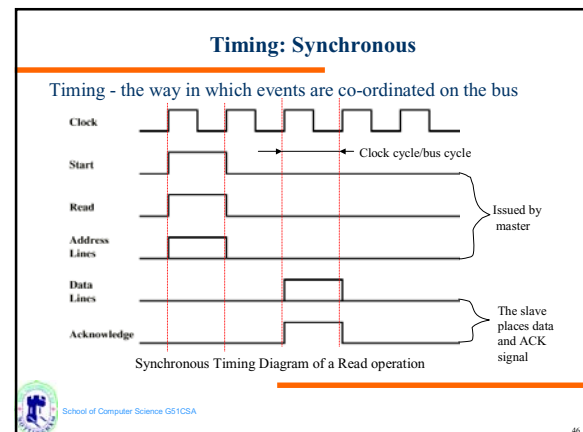
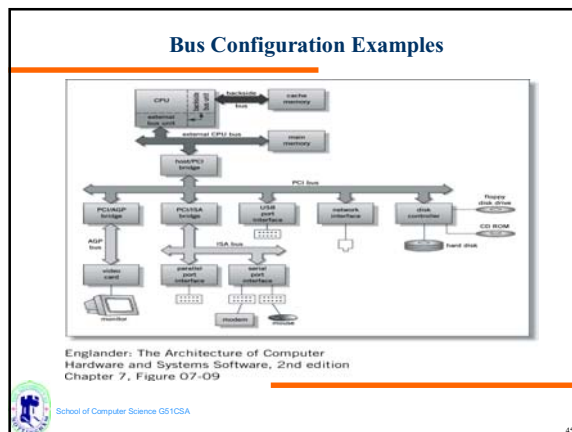
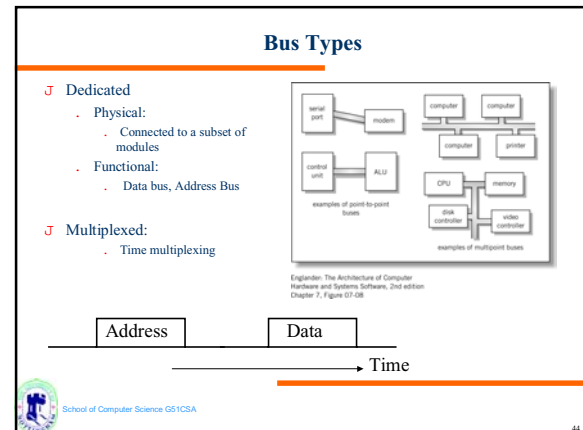
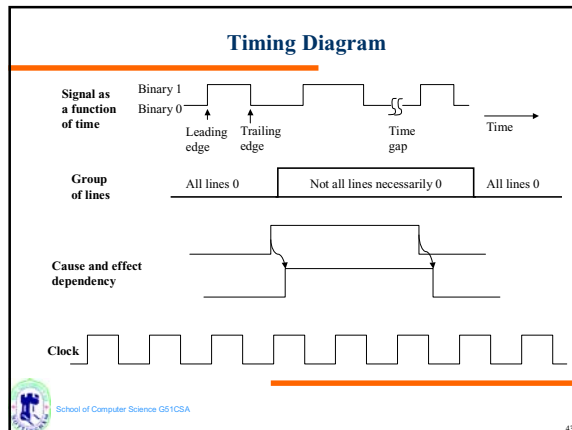


- Memory to processor
- Processor to memory
- I/O to processor
- Processor to I/O
- I/O to and from Memory (DMA)



School of Computer Science G51CSA

42



- ### Bus Arbitration
- b More than one module (e.g. CPU and DMA controller) may need control of the bus
 - b Only one module may control bus at one time
 - b Needs some form of arbitration
- 1 Centralised Arbitration
 - 1 Single hardware device controlling bus access (Bus Controller or Arbiter)
 - 1 May be a separate module or part of CPU or separate
 - 1 Distributed Arbitration
 - 1 Each module may claim the bus
 - 1 Control logic on all modules
- One device is designated as master, which may initiate a data transfer with some other device (slave)
- School of Computer Science GS1CSA

