**Basic Computer Operations**

**1. Basic Computer Organization**

All operations include:

* Instructions
* Data

eg.

2+3 contains:

* Addition (+)
* [2] and [3]

The microprocessor consists of transistors.

Transistors have two states:

OFF ON

volt: LOW HIGH

Binary 0 1

In Binary Format, multiple bits represent more combinations.

b indicates Binary Format.

eg.

11112 represents 15.

Written as 1111b.

In Hexadecimal Format, the letters A-F represent numbers.

h indicates Hexadecimal Format.

eg.

B56Ch represents 1011 0101 0110 1000b

We use Hex because it is shorter and so less mistake-prone.

Byte = 8 bits.

Nibble = 4 bits

Word = 32 bits

DWord = 64 bits

**Microprocessor Basic Structure**

A Microprocessor is an IC that contains 4 parts:

1. CPU

2. Memory

3. Input/Output Interface

4. System Bus

The CPU contains three modules.

1. Control Unit:

* Controls all unit module operations
* Made of Logic Circuits
* Uses a Clock Signal

2. Arithmetic Logic Unit:

* Performs Arithmetic and Logic operations

3. Register Array

* Fast Data Store and Retrieval
* Made of High Speed Internal Memory
* Can be specialized for Specific Functions

The Memory is a device that stores data.

1. Data is stored in bytes (8 bits)

2. Each byte is given a unique address

* If data is larger than 8 bits, it takes consecutive addresses.
* The Lower address is used for reference.

3. Memory Addresses are identified by the binary pattern on the address bus.

Memory size = 2n bytes [when n = 10, 210 = 1024 = 1KB]

The System Bus transfers information.

1. Data Bus

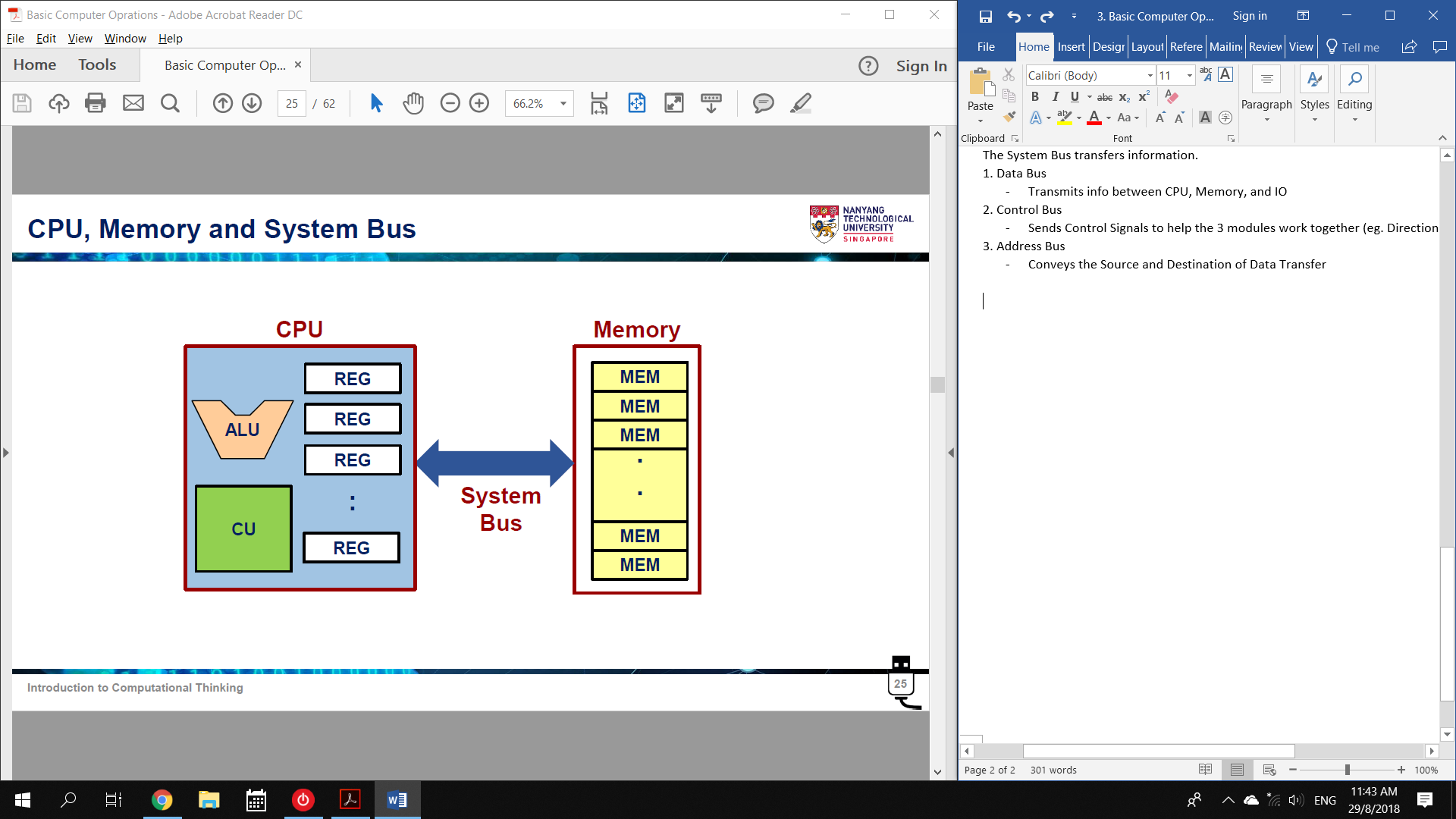
* Transmits info between CPU, Memory, and IO

2. Control Bus

* Sends Control Signals to help the 3 modules work together (eg. Direction, Access)

3. Address Bus

* Conveys the Source and Destination of Data Transfer



**Program Execution**

An operation is an instruction given by a user.

A Program is a series of operations. (software)

* All operations are in binary.
* Interpreted as commands by the CPU.

To run a Program, it needs 2 things:

1. Instructions

* Stored in Non-Volatile harddrive
* Copied to Memory

2. Data

* A file to be acted on

1. Instructions are **Fetched** from Memory to CPU Control Unit.

* The Machine Instructions have a specific binary pattern
* CPU is designed to recognize this Instruction Set

2. Instruction is **Decoded**

3. Instruction is **Executed**

**Basic CPU Operations**

Registers and Memory

A Register is a fast memory in the CPU.

* Labelled with A-B and 0-9

A Memory is a large data storage.

* Labelled with Hexadecimal

Basic CPU Operation:

* Num1 and Num2 are stored in the Memory
* Num 1 is loaded into REG A
* Num 2 is loaded into REG B
* ALU executes addition: REG A + REG B
* Result is loaded into REG C
* REG C is stored into memory

How does the CU find Instructions?

1. Program instructions and Data are loaded into Default Memory Locations

2. Clocking Signal is applied

3. CU uses clock’s Rising/Falling to retrieve Instruction from Default Memory

4. CU recognizes the Instruction Set and performs the action.

**Program Counter and Instruction Register (PC and IR)**

The CPU contains 2 special registers:

1. Program Counter

* Tells CU where to find Instructions in the Memory

2. Instruction Register

* Holds a Copy of the Instruction to be Decoded and Executed

Full Fetch-Decode-Execute

1. Program Counter (PC) gives the Instruction Address

2. CPU Fetches Instructions from Address

3. CPU Stores Instructions into Instruction Register (IR)

4. CU Decodes Machine Instruction in IR

5. CPU Executes Instruction: Load Operands and Operate

Typical Notebooks use x86

* Different CPU families have different Set Instructions

Tablets and Smartphones use ARM

eg.

3A means: Load REG A from Memory

**An Illustrative Example**

Start

Power on

Memory is Loaded with Program and Data

Program Counter’s content starts at 00h

03h 41

02h 3B

01h 40

00h 3A

Fetch 1

Content of 00h and 01h are fetched into IR

* Two addresses for a 16-bit instruction

PC increments to 02h

Decode 1

CU decodes the contents of IR:

3A 40

* where 3A means “Load REG A”
* where 40 means “Memory Location 40h”

Execute 1

CU executes instruction:

* “Load REG A with Memory Location 40h”

Fetch 2

Content of 02h and 03h are fetched into IR

PC increments to 04h

Decode 2

CU decodes IR:

3B 41

* where 3B means “Load REG B”
* where 41 means “Memory Location 41h”

Execute 2

[same as Execute 1]

… …

Decode 3 (Example of ALU)

CU decodes IR

1C AB

* where 1C means “add and save result in REG C”
* where AB means “REG A” and “REG B”

Execute 3

CU executes instruction:

* Use ALU to “Add REG A and REG B” and “Save result to REG C”

… …

Decode 4 (Example of Storage)

CU decodes IR

2C 43

* where 2C means “store content of REG C”
* where 43 means “Memory Location 43h”

Execute

“Store content of REG C in Memory 43h”