CS101 Introduction to Computing
Lec-4
Understanding the
Harder Side of Hardware
(CPU & Memory)

Monday, March 15, 2021

A Brief Clarification

- Some of you seem to be confused as to what data is – whether electrons have memory…
- **Data** with respect to our course is the ***information*** stored within the memory of a system (Memory will be explained later. For the time being it is just a storage space.)
- *One bit of data means a logical **0** or **1*** (for our understanding)
- If we say that the output of a logic gate is ***1***, it means it is providing (not necessarily storing) a bit of data whose state is ***1***.
- If you measure the voltage at its output it will be around 5V (or 3V). Likewise, if it is ***0***, then the voltage will be 0V indicating that the data is ***0***.
- So if a conducting line is somehow held at 5V (or 3V) we can say that it is carrying a bit ***1*** or its state is logic ***1***. (Else if it is 0V, then bit ***0*** logic ***0***)
- So deep down it means that if a bit ***0*** is being stored in the memory, then that particular point in the memory where we are storing the bit, is somehow being held at 0V. If the data stored is ***1***, then the voltage at that point is somehow held at 5V (or 3V).
- Conceptually if I wish to store ***01001000*** (1 byte) then I would need the memory to hold the following voltages at 8 different points within as:

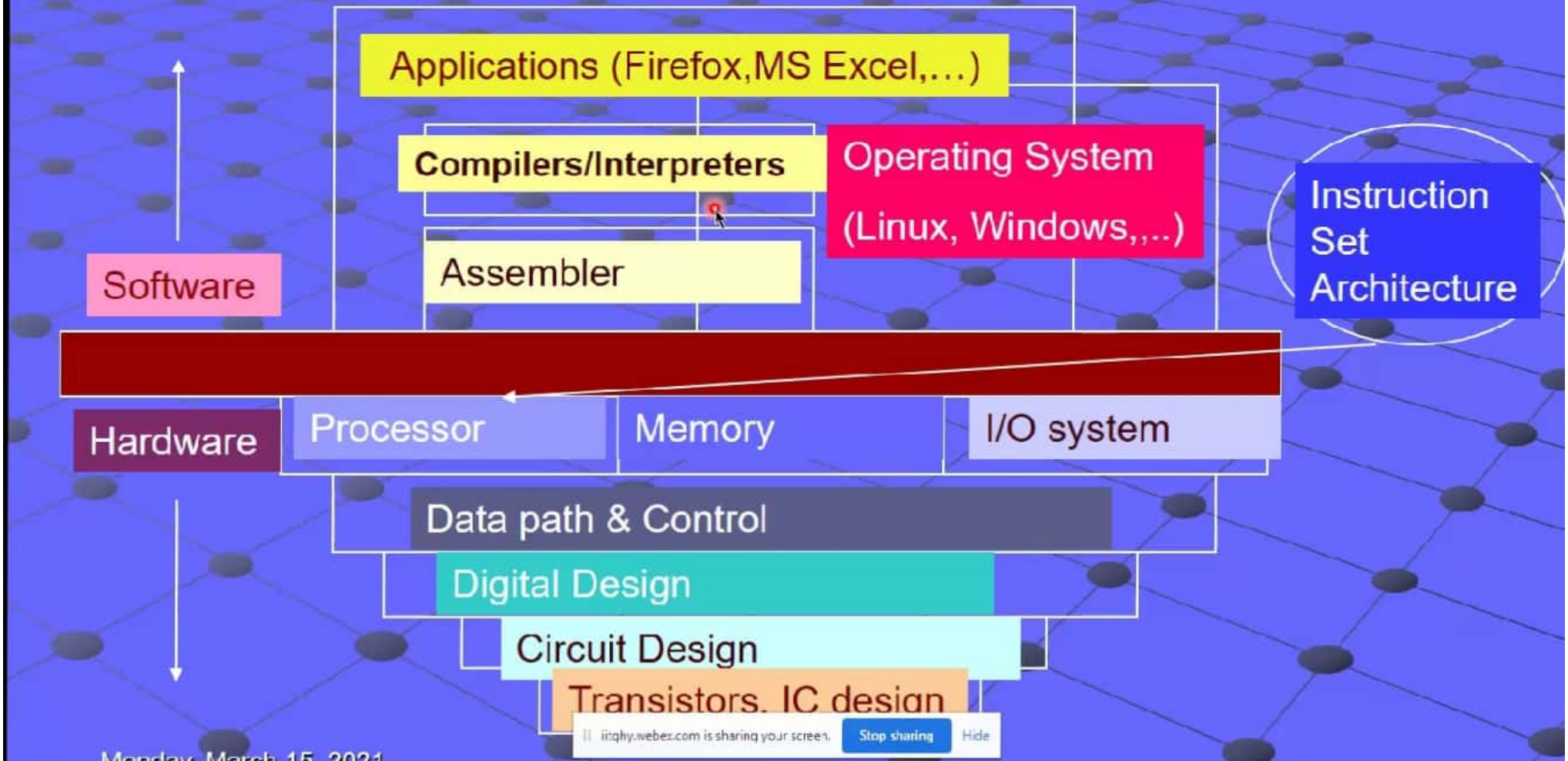
0V	5V	0V	0V	5V	0V	0V	0V	0V

- The above 8 blocks storing 8 bits is often referred to as an 8-bit ***Register***

Coming soon: More on Memories!

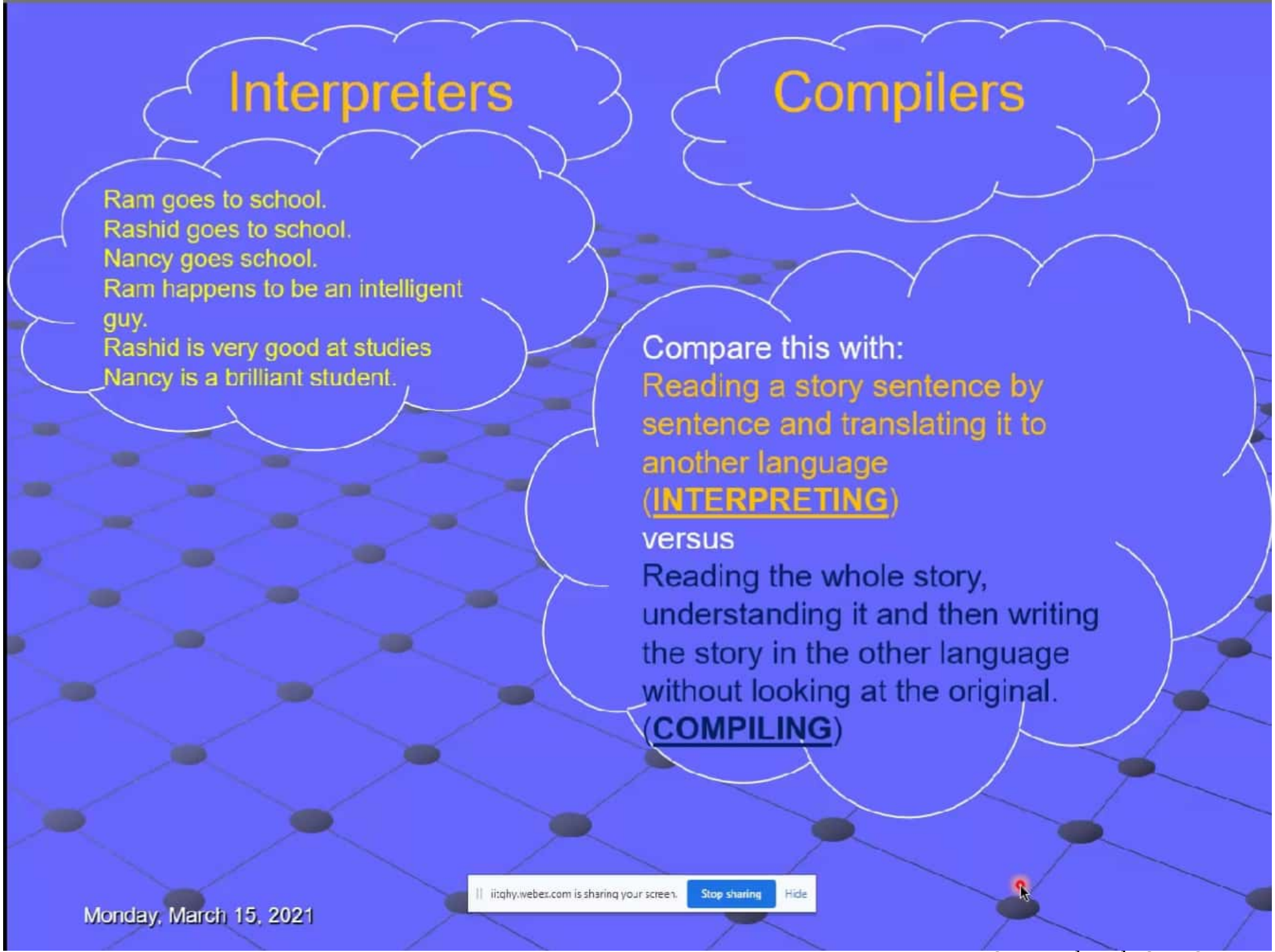
Monday, March 15, 2021

Computer Architecture = Instruction Set Architecture + Computer System Organization



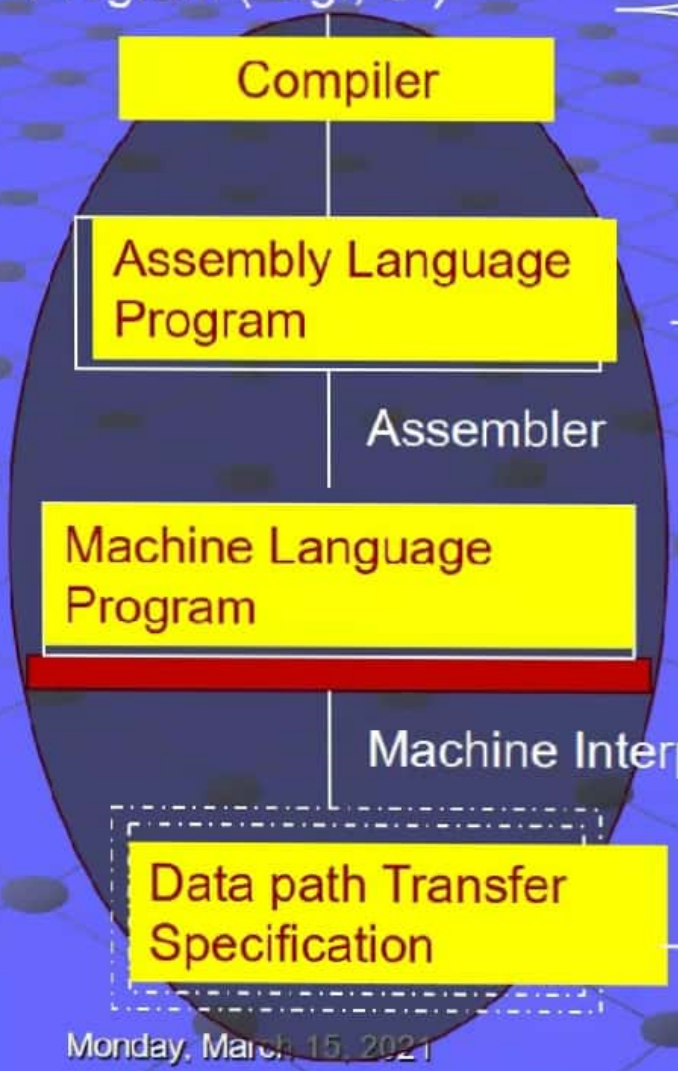
Levels Of Abstraction

- **Application Level** - All kinds of application programs written in High Level Languages
- **Compilation Level** - Each high level program is compiled into machine code
- **Operating System Level** - Set of services for managing resources of a computer
- **Instruction Set Architecture Level** - The design of this level involves specification of
  a) Memory space
  b) Processor registers
  c) Instruction formats
  .....
- **Microarchitecture Level** - Implementation and Organization of hardware components including the Control Unit
- **Logic & Circuit Design Level** - Design of Components in specific technology
- **Device and Technology Level** - Manufacturing , packing etc.



Interpreters	Compilers
Translates the source program statement by statement into machine code.	Scans the entire program and then translates the whole of it into machine code at once.
Since only one statement is being dealt with at a time, an interpreter takes very less time to analyze the source code. But, the overall execution time of the entire program is effectively large - i.e. execution is slow.	A compiler takes a lot of time to analyze the source code. However, the overall time taken to execute the process is much faster.
An interpreter does not generate an intermediary code. So it does not need much of memory – in short it is efficient when it comes to use of memory.	A compiler always generates an intermediary object code. Later it may need to link many aspects of the intermediary code. Hence more memory is needed.
Keeps translating the program continuously line by line till the first error is confronted. If any error is detected, it stops working and hence debugging is easier.	A compiler generates the error message only after it scans the complete program and hence debugging is relatively harder.
E.g. Python (Find more examples)	E.g. C (Find more examples)

Levels of Abstraction




```
x = 3;
y = 4;
z = x + y;
```

```
MOV A, 03H
MOV B, 04H
ADD B
MOV C, A
```

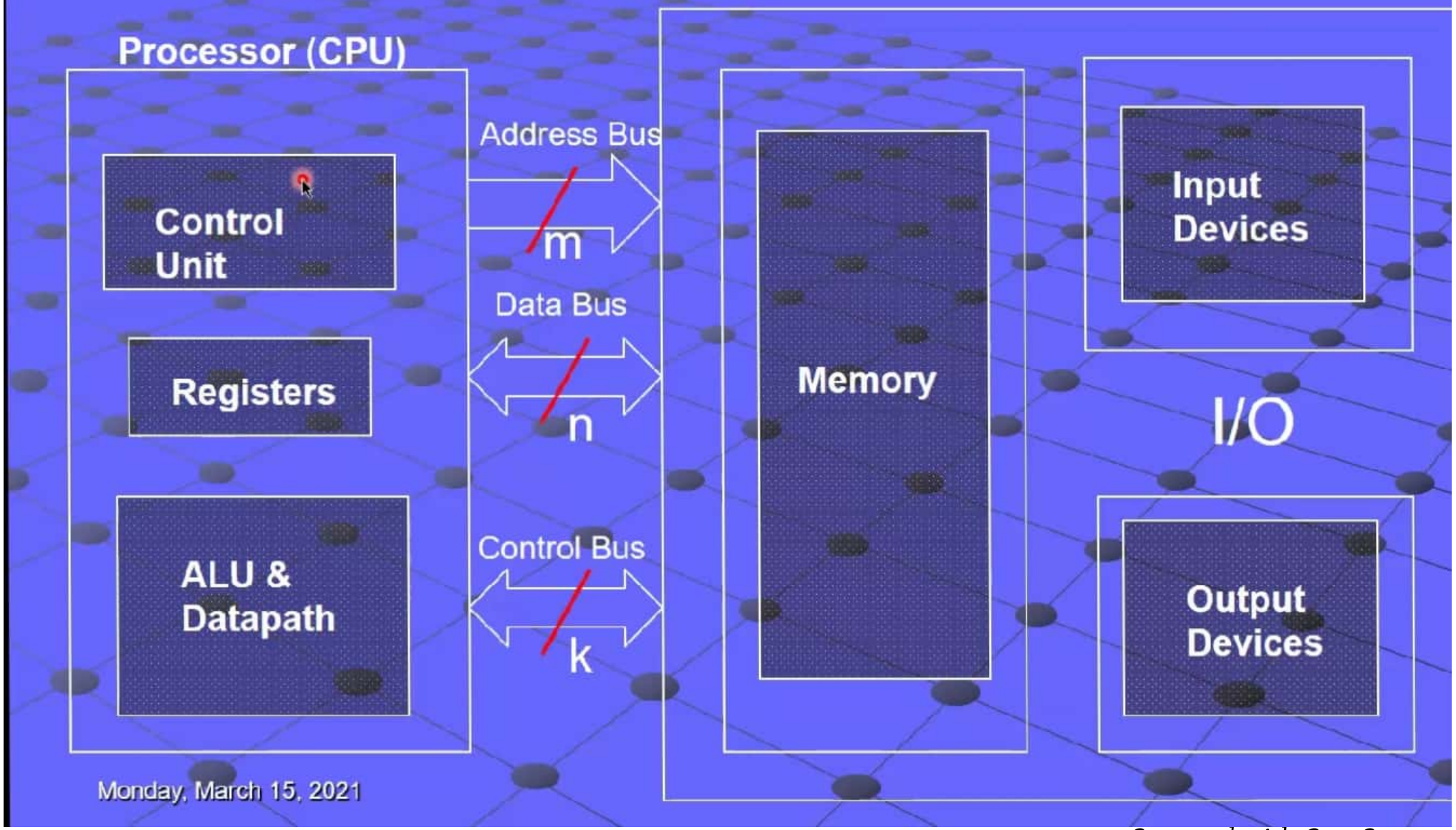
```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

IR ← Imem[PC]; PC ← PC + 1

NB: IR is the Instruction Register and PC is the Program Counter, Both these are within the CPU

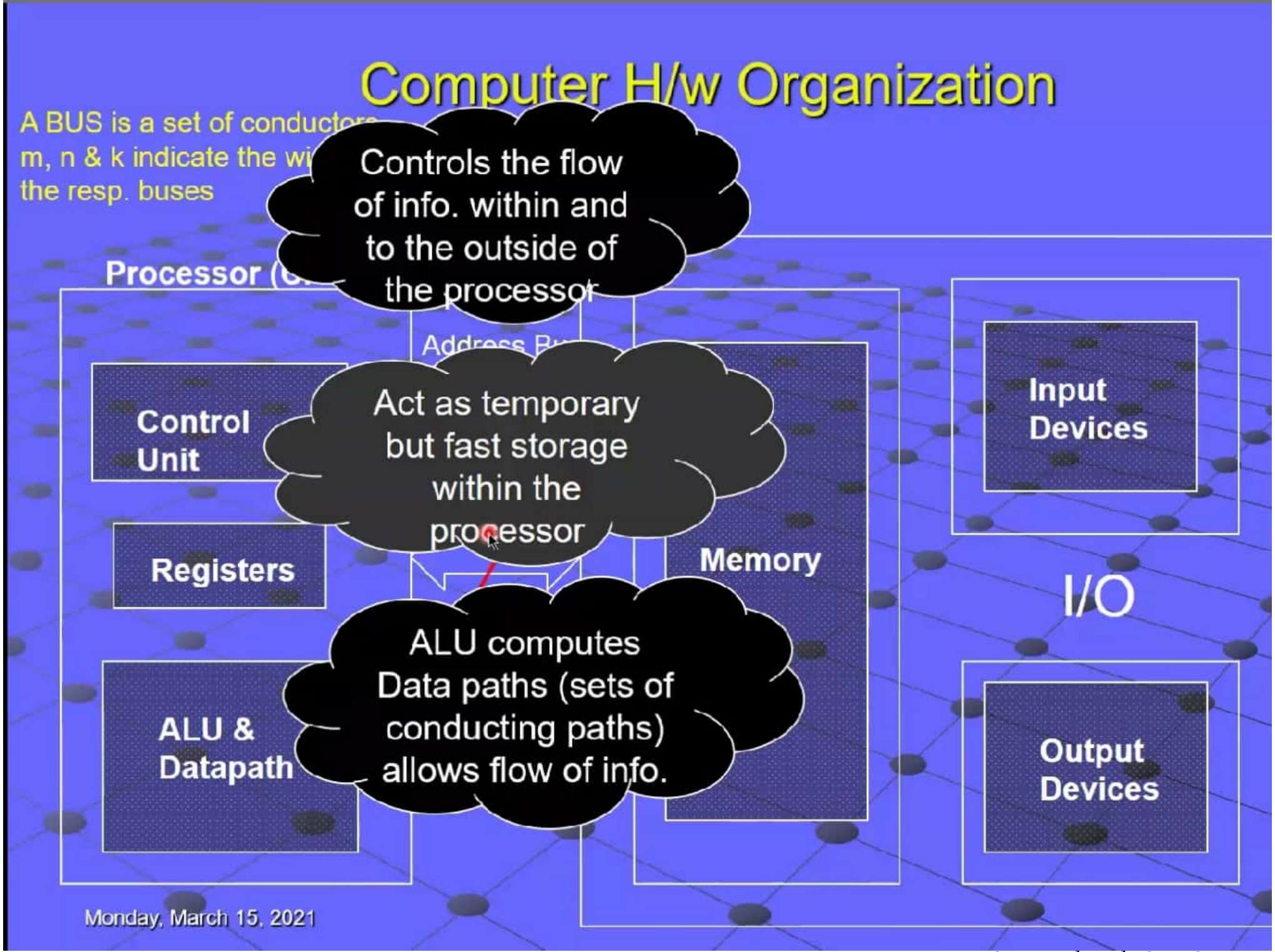
Computer H/w Organization

A BUS is a set of conductors.
m, n & k indicate the width of the resp. buses



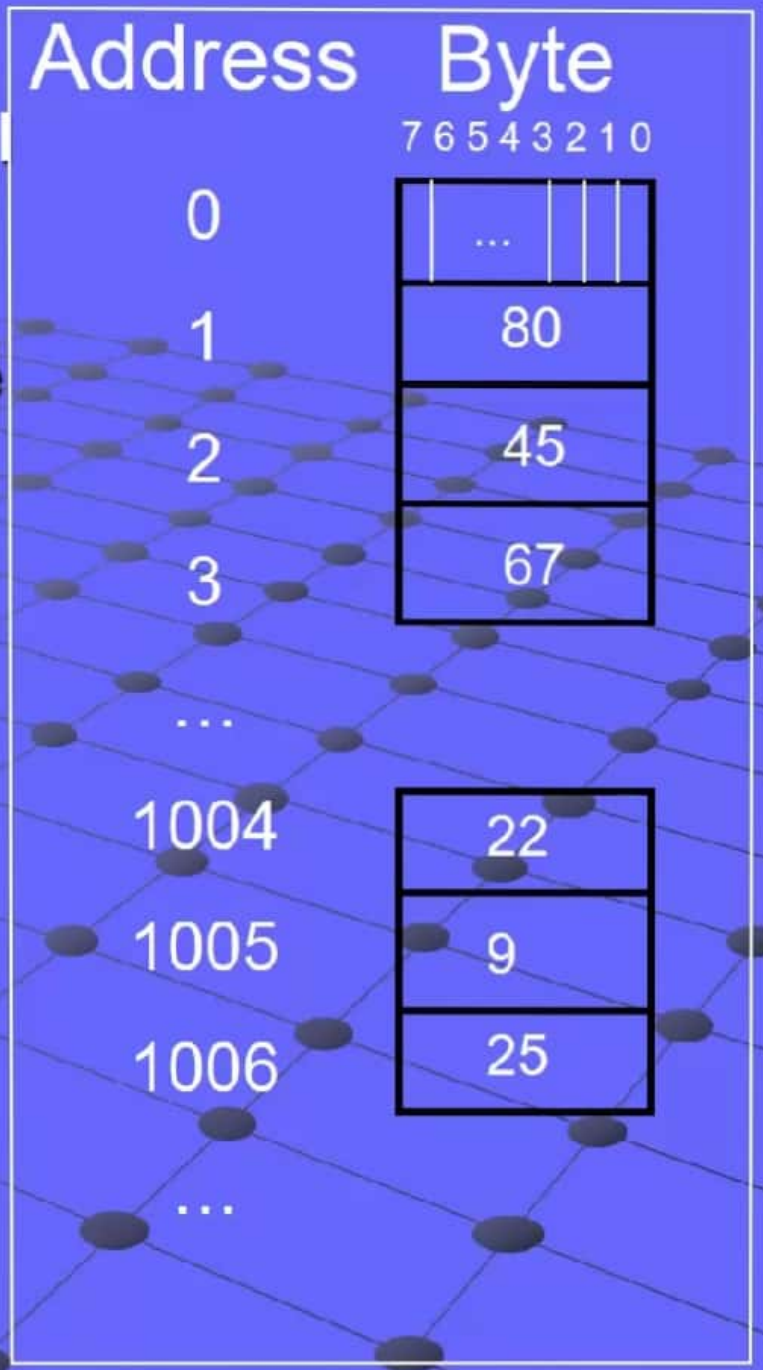
Monday, March 15, 2021

Computer H/w Organization



Memory Memorabilia

- Stores data and facilitates its retrieval at a later stage
- Conceptually, a computer memory is simply a collection of locations where information can be stored as bits
- Most often, memory **is byte-addressable**
- This means the memory is divided into *bytes* (8-bit quantities) each identified by a **unique address**
- Generally, bytes are addressed sequentially, beginning with the address 0.



Memory Memorabilia

- Stores data and facilitates it
at a later stage
- Conceptually, a computer m
simply a collection of locations where
information can be stored as bi
- Most often, memory
addressable
- This means the m
into *bytes* (8-bi
identified by a
- Generally, byte
sequentially, be
address 0.

ddress Byte

7 6 5 4 3 2 1 0

0

1

...

80

5

Do you give each cell in your body a name or a roll no.? You give this "whole" guy a unique Roll No. In this case a set of 8 bits (one byte) taken together is given a unique address. When you address it, all these 8 bits (1 byte) are accessed together just as when I call out a roll no. the whole of the addressed person responds!

☺

Size of Memory

- Generally specified as:

  No. of words x No. of bits bit/word

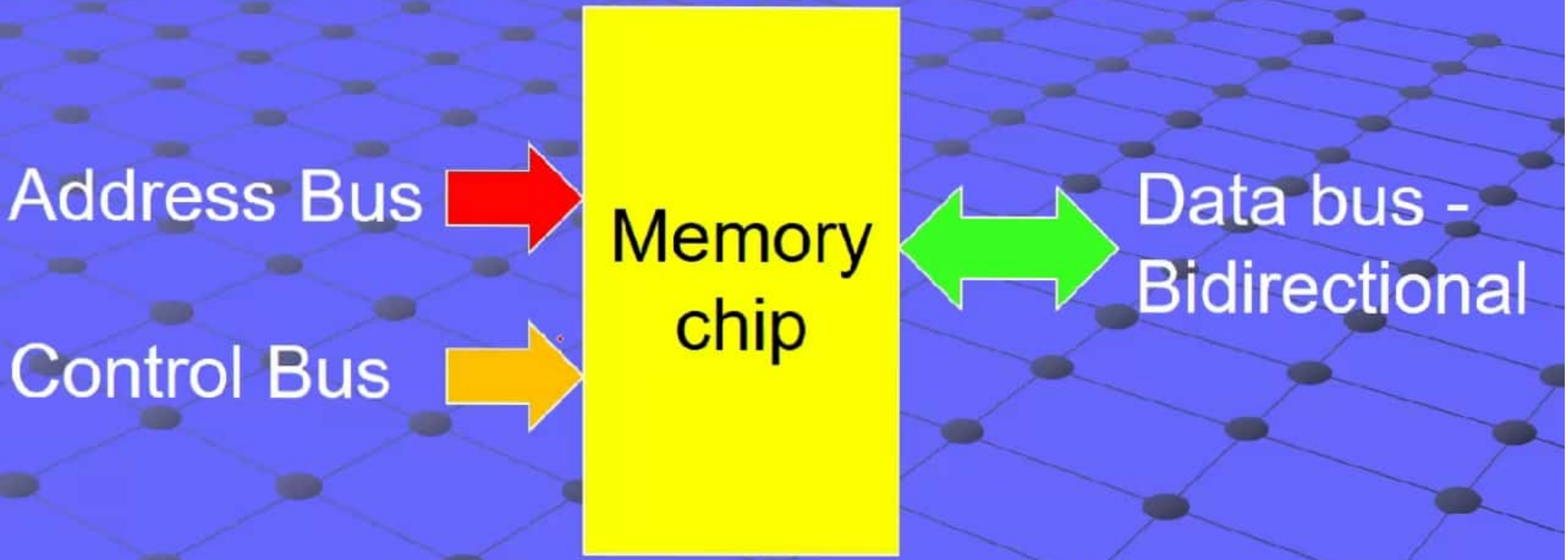
E.g. 1024X1 = 1Kbit → There are 1024 words each of 1 bit length

1Kb

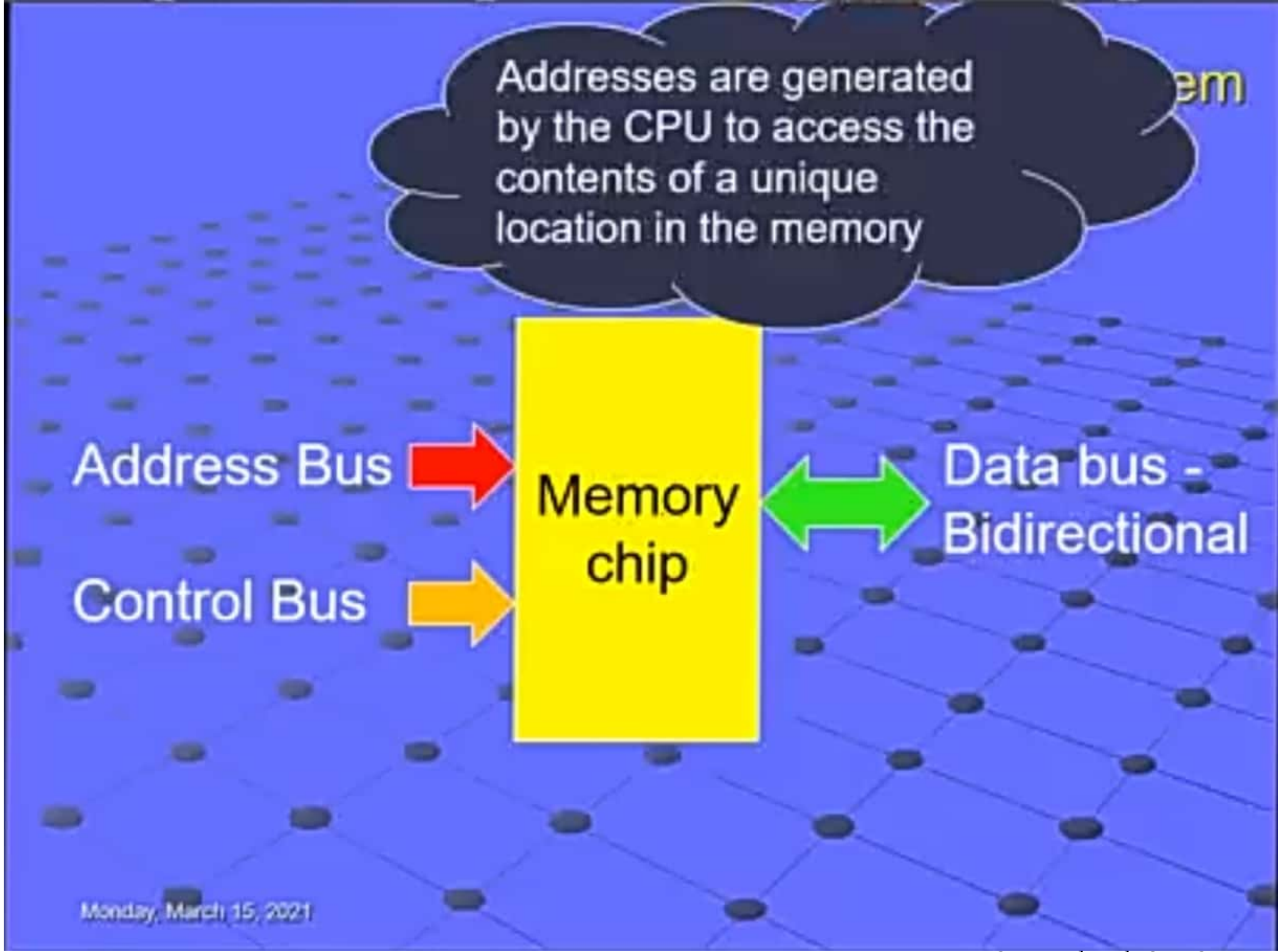
2048X8=2KByte → There are 2048 words each 8 bits in length

2KB

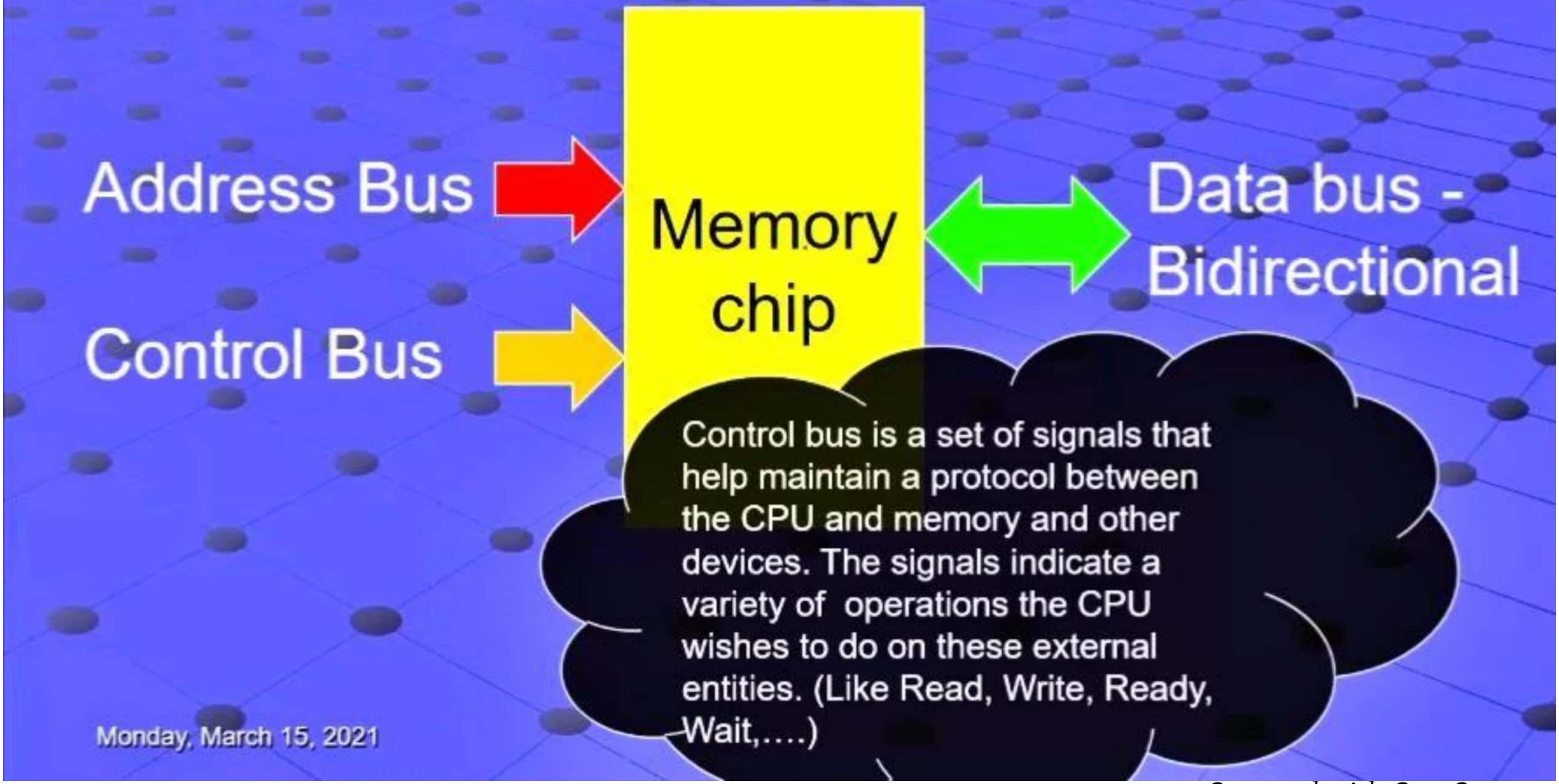
Memory Subsystem



Monday, March 15, 2021



Memory Subsystem



Monday, March 15, 2021

Execution of some hypothetical Instructions

Address	Assembly Instructions	Comments
7H	ADD A, B	;Add content of register A with that of B and store it in A i.e. [A] ← [A] + [B]
8H	STA [C]	; Store contents of register A to the memory location whose address is contained in register C.

Address	Machine Language	Assembly Instructions
7H	34	ADD A, B
8H	57	STA [C]