CPT_S 260 Intro to Computer Architecture
Lecture 22

Digital Design I
March 2, 2022

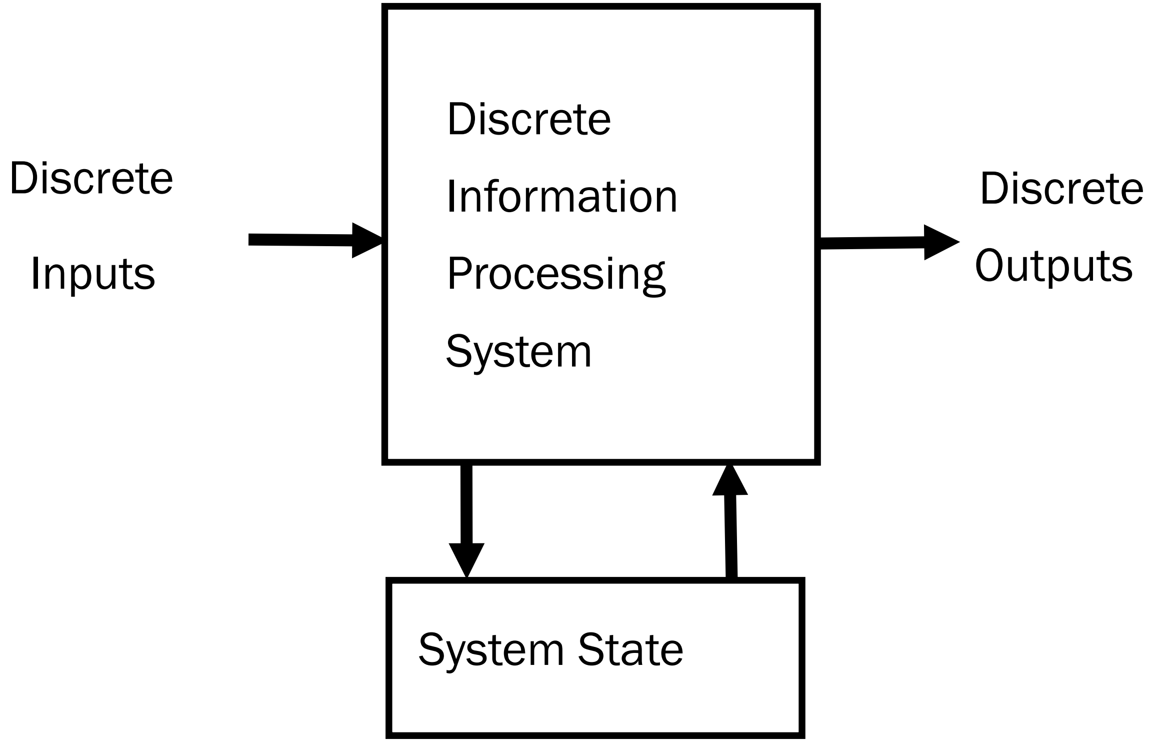
Ganapati Bhat
School of Electrical Engineering and Computer Science
Washington State University

Topics Covered

- **Performance of the computer**
  - Instruction Count
  - Clock Cycle Time
  - Clock Cycle Instruction (CPI)
- **Instruction Set Architecture (MIPS)**
- **We will now construct the datapath and control unit (CU) that connects the previous two topics together**
  - Sequential processor that implements all MIPS instructions

Digital System

- **Takes a set of discrete information <u>inputs</u> and discrete internal information <u>(system state)</u> and generates a set of discrete information <u>outputs</u>.**



Types of Digital Systems

- **No state present**
  - Combinational Logic System
  - Output = Function(Input)
- **State present**
  - State updated at discrete times
    - » Synchronous Sequential System
  - State updated at any time
    - » Asynchronous Sequential System
  - State = Function (State, Input)
  - Output = Function (State) or Function(State, Input)

Logic Design Basics

- **Information encoded in binary**
  - Low voltage = 0, High voltage = 1
  - One wire per bit
  - Multi-bit data encoded on multi-wire buses
- **Combinational element**
  - Operate on data
  - Output is a function of input
- **State (sequential) elements**
  - Store information

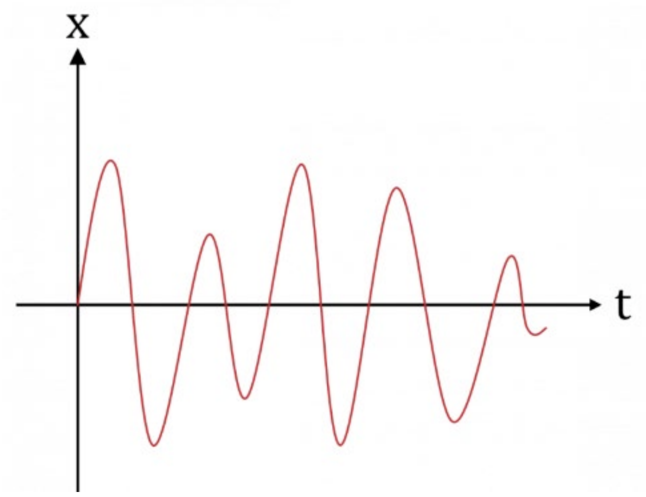
Signal

- **An information variable represented by physical quantity**
- **For digital systems, the variable takes on discrete values.**
- **Two level, or binary values are the most prevalent values in digital systems.**
- **Binary values are represented abstractly by:**
  - digits 0 and 1
  - words (symbols) False (F) and True (T)
  - words (symbols) Low (L) and High (H)
  - and words On and Off
- **Binary values are represented by values or ranges of values of physical quantities**

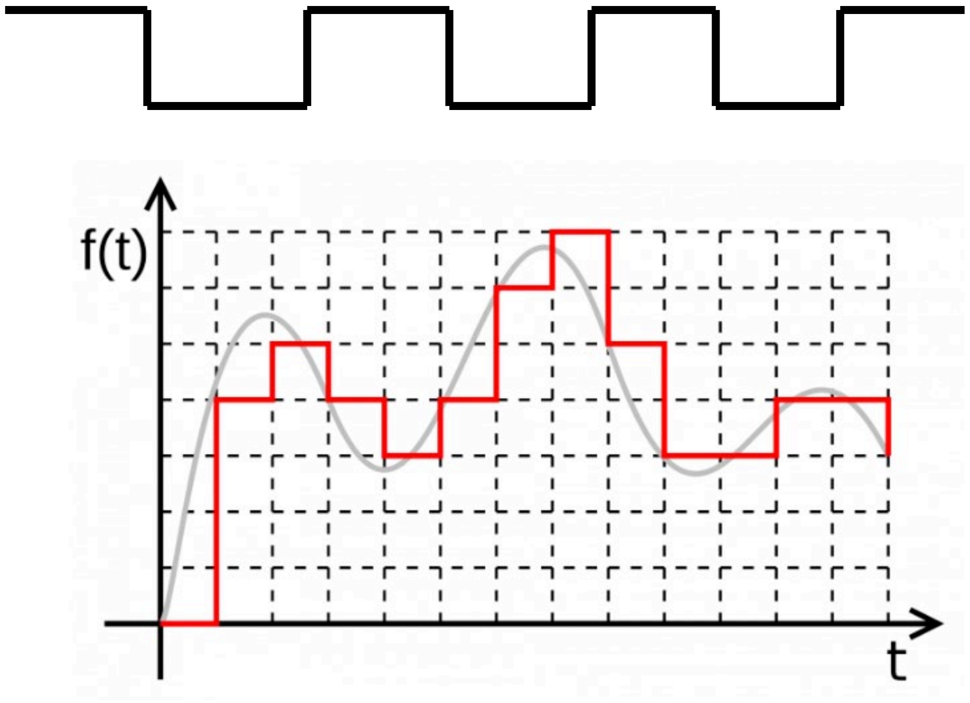
Signal Examples

- **Analog signals are continuous in nature, whereas digital signals are discrete**

Analog Signal



Digital Signal



Digital Signal

Basic Logic Gates

- **NOT, AND, and OR Gates**
- **NAND and NOR Gates**
- **Exclusive-OR (XOR) Gate**

Truth Tables

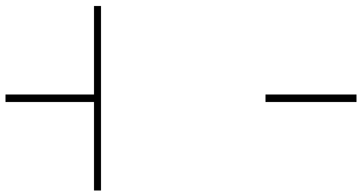
- *Truth table* – a tabular listing of the values of a function for all possible combinations of values on its arguments
- Example: Truth tables for the basic logic operations:

AND		
X	Y	$Z = X \cdot Y$
0	0	0
0	1	0
1	0	0
1	1	1

OR		
X	Y	$Z = X + Y$
0	0	0
0	1	1
1	0	1
1	1	1

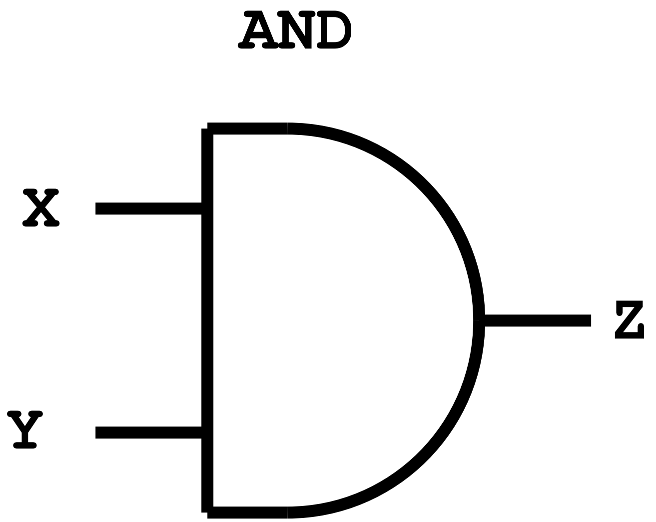
NOT	
X	$Z = \overline{X}$
0	1
1	0

NOT Gate -- Inverter



X	Y
0	1
1	0

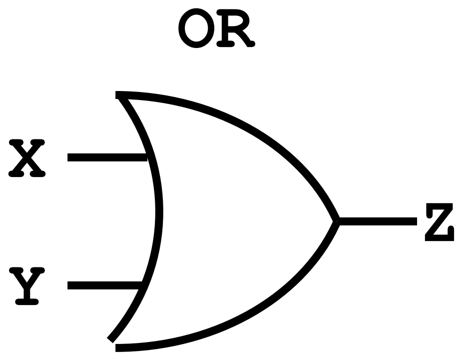
AND Gate



Z = X & Y

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

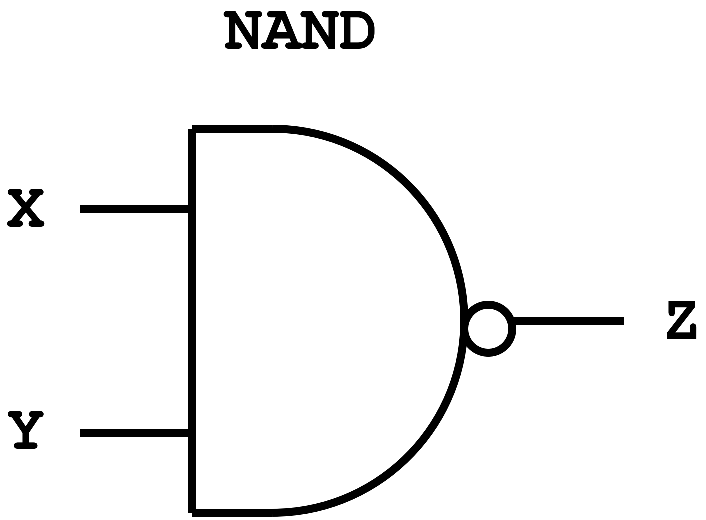
OR Gate



Z = X | Y

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

NAND Gate

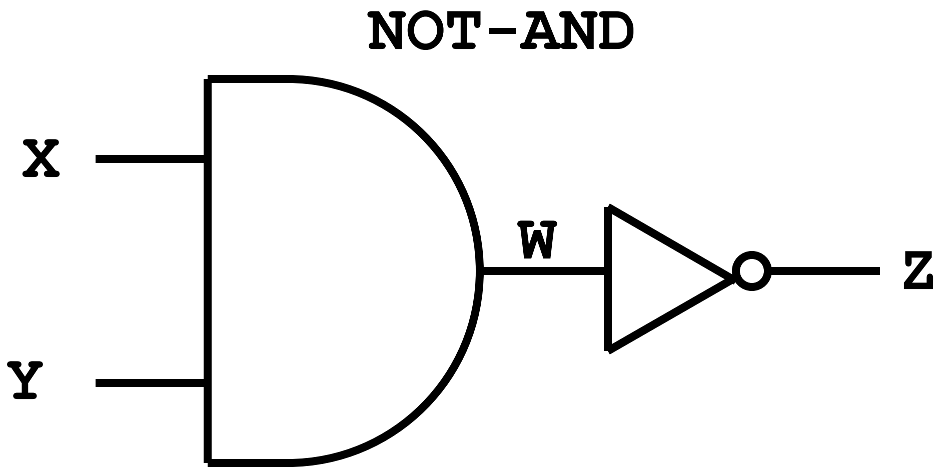



```
Z = ~(X & Y)

nand(Z,X,Y)
```

X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

NAND Gate

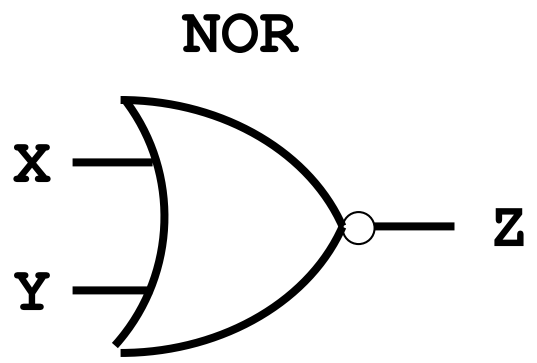


W = X & Y

X	Y	W	Z
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Z = ~W = ~(X & Y)

NOR Gate

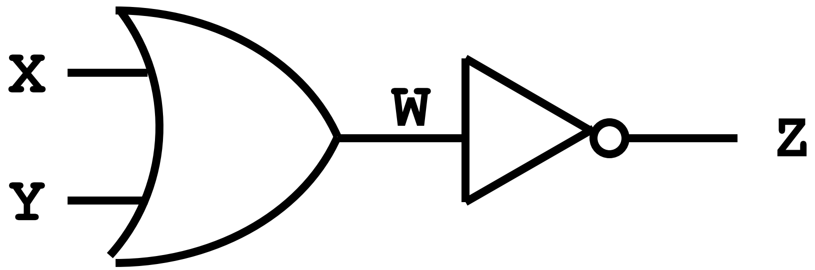


`Z = ~(X | Y)`

`nor(Z,X,Y)`

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

NOR Gate



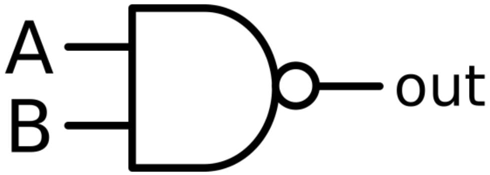
W = X | Y

Z = ~W = ~(X | Y)

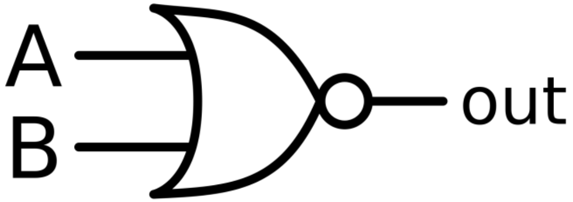
X	Y	W	Z
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

NAND and NOR

- **NAND : NOT of AND : A nand B = $\overline{A \cdot B}$**
- **NOR : NOT of OR : A nor B = $\overline{A + B}$**
- **NAND and NOR are *universal gates*, i.e., they can be used to construct any complex logical function**



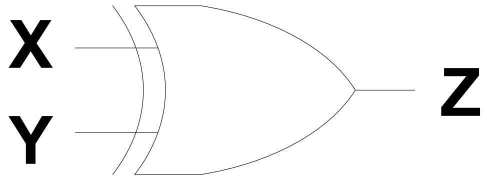
NAND



NOR

Exclusive-OR Gate

XOR



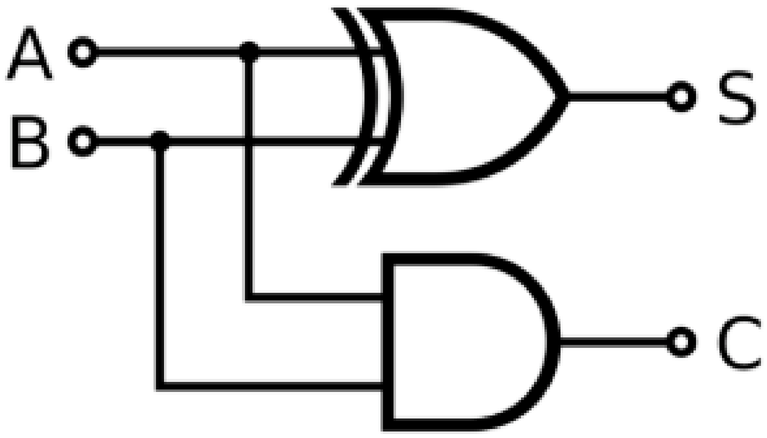
`Z = X ^ Y`

`xor(Z,X,Y)`

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

Stateless (Combinational) Digital Circuits

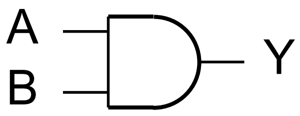
- **This Circuit does not have an internal state. The signals simply flow left to right, from inputs to outputs.**



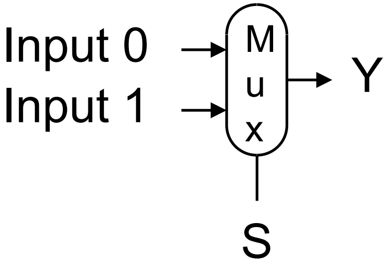
- **S=!(A|B)**
- **C=A*B**

Combinational Elements

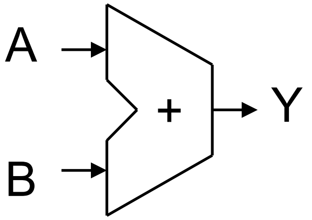
- AND Gate
  - Y = A & B



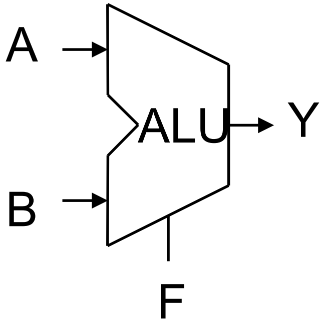
- Multiplexer
  - Y = S ? I1 : I0



- Adder
  - Y = A + B

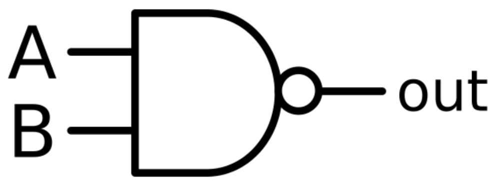


- Arithmetic/Logic Unit
  - Y = F(A, B)

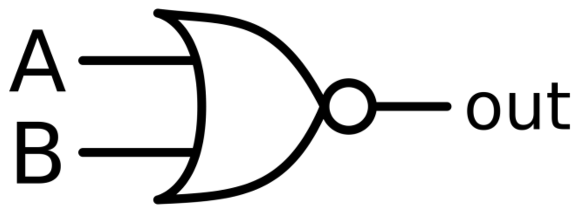


NAND and NOR

- **NAND : NOT of AND : A nand B = $\overline{A \cdot B}$**
- **NOR : NOT of OR : A nor B = $\overline{A + B}$**
- **NAND and NOR are *universal gates*, i.e., they can be used to construct any complex logical function**



NAND



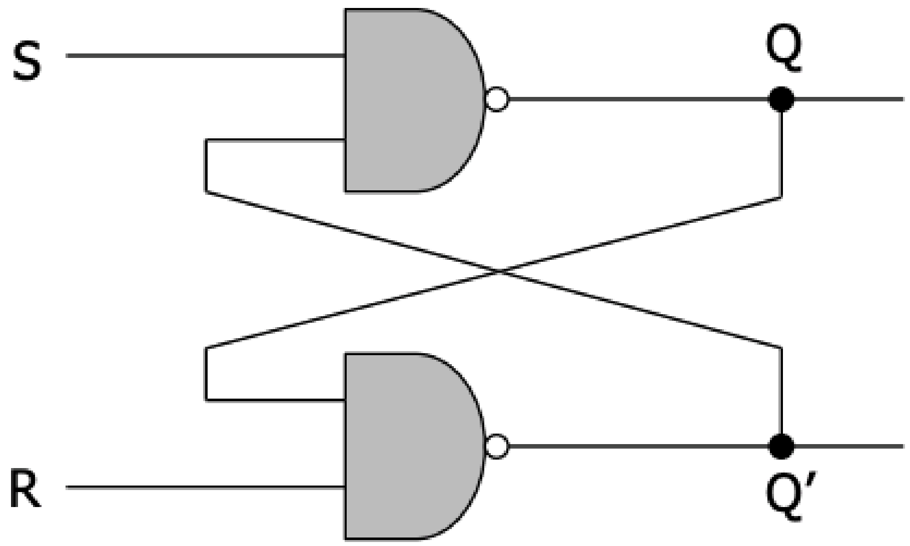
NOR

Types of Digital Systems

- **No state present**
  - – Combinational Logic System
  - – Output = Function(Input)
- **State present**
  - – State updated at discrete times
    - » Synchronous Sequential System
  - – State updated at any time
    - » Asynchronous Sequential System
  - – State = Function (State, Input)
  - – Output = Function (State) or Function(State, Input)

Stateful (Sequential) Digital Circuits

- This circuit utilizes one of the wire loops from the right to the left providing feedback
- "flip flop" building blocks to preserve state.

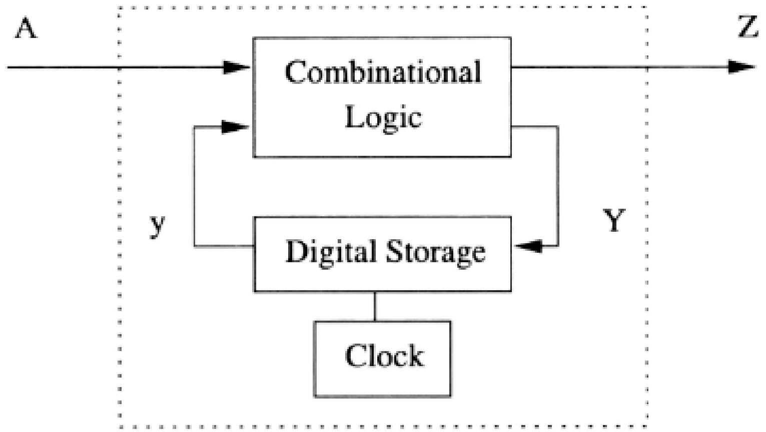


R-S Latch

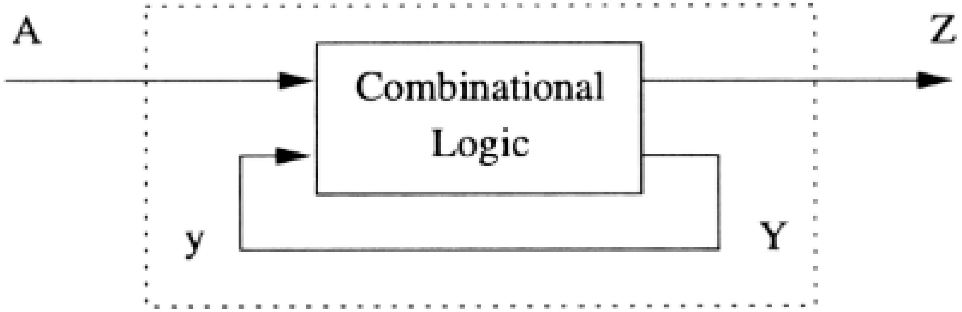
Input		Output
R	S	Q
1	1	Q_{prev}
1	0	1
0	1	0
0	0	Invalid

Synchronous vs Asynchronous

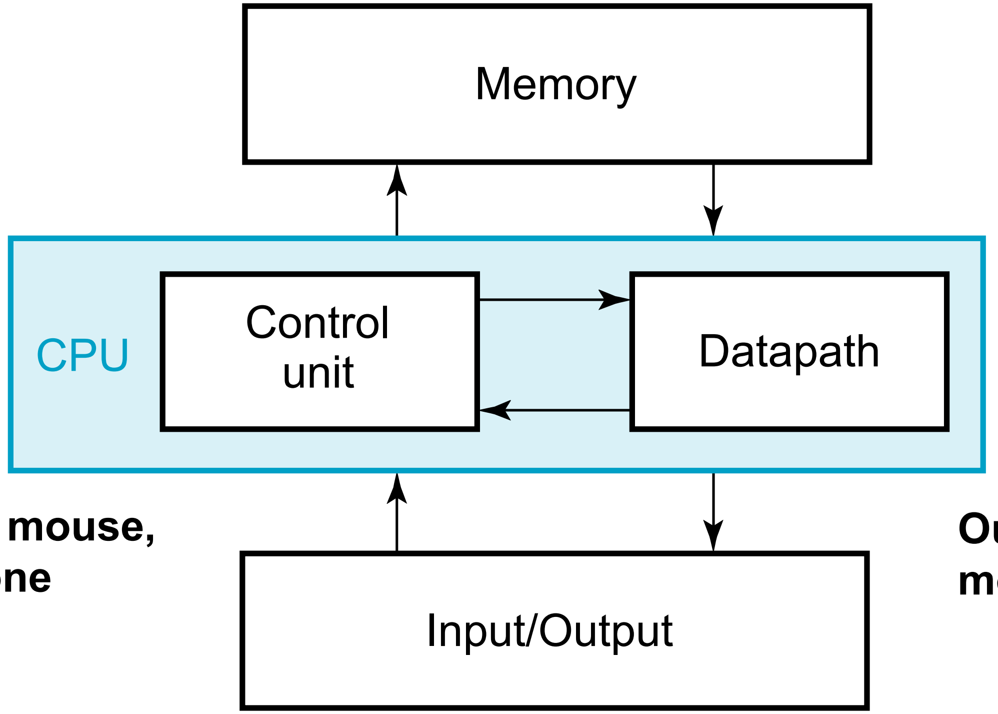
- Synchronizing periodic clock connected to the clock inputs of all the memory elements of the circuit, to synchronize all internal changes of state.



- *asynchronous* if it does not employ a periodic clock signal C to synchronize its internal changes of state.
- Therefore the state changes occur in direct response to signal changes on primary (data) input lines,



A Digital Computer Example



Inputs: Keyboard, mouse, modem, microphone

Outputs: CRT, LCD, modem, speakers

We will look into this model in more details later in this course …

Checkpoint – Truth tables

- S=$(A + B) \cdot (\bar{A} + \bar{B})$

- C=$A \cdot B$