**PUE 3141: ADVANCED DIGITAL SYSTEMS DESIGN**

**LECTURE 1A**

**BY DR. MUTUGI KIRUKI**

**Topics:**

i) Overview of digital logic design methodology
ii) Review of combinational logic designs
iii) Review of sequential logic designs with flip flops (FF)

**References:**


i) Elijah, M. (2009) *Foundations of Digital Electronics*. University of Nairobi Press
ii) Rafiquzzaman, M. (2005) *Fundamentals of Digital Logic & Microcomputer Design* ($5^{th}$ Edition). Wiley.


## 1. Overview of Digital Logic Design Methodology

### Design Levels for Digital Systems

i) Device Level: Utilizes transistors to design logic gates
ii) Logic Level: Design technique in which chips containing logic gates such as AND, OR, and NOT are used to design a digital components such as the ALU.
iii) Systems Level: Type of design in which CPU, memory, and I/O chips are interfaced to build a computer

### General digital logic design methodology

Traditionally, there are 3 approaches:

i) Top-down design methodology
   - Start with larger block of design & then work downwards to sub-block levels.
ii) Bottom-up design methodology
   - Start with the sub-blocks and use them as the building blocks for the larger system blocks
iii) Hybrid
   - A combination of bottom-up and top-down is used.

### Digital Design Methodology (FPGAs, ASICs …)

- Design & Functional Specification
  - ✓ State diagrams
  - ✓ Timing diagrams
  - ✓ Hardware Description Languages
- Design Partition & Entry
  - ✓ System design partitioned into functional units & individual components

- ✓ Behavioral descriptions / modelling
- Simulation & Functional Verification
  - ✓ Test bench development
  - ✓ Model verification
- Design Integration and Verification
  - ✓ Individual components verified independently & integrated
- Synthesis & post-synthesis
  - ✓ Synthesize the design from the behavior description
  - ✓ Synthesis produces netlist to configure target FPGA
  - ✓ Post-synthesis: design validation & timing verification
- Placement & Routing
  - ✓ For FPGAs, this involves placing functions into particular Configurable Logic Blocks (CLBs) and slices that are interconnected via switch matrix

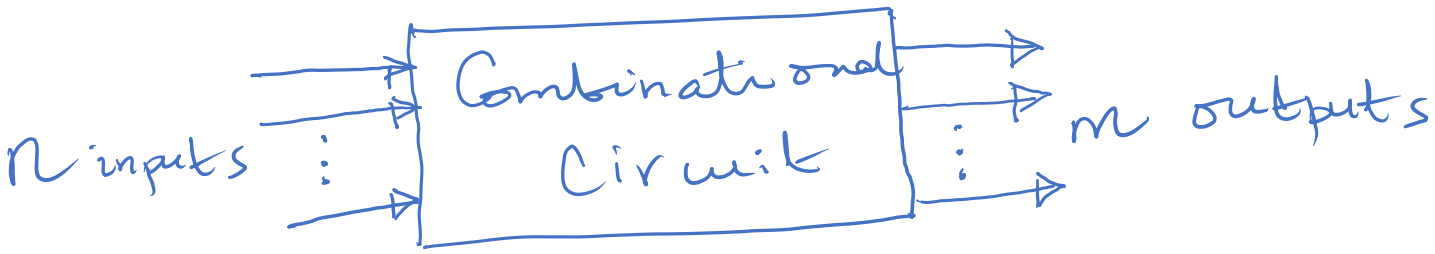
## 2. Review of Combinational Logic Designs

Digital logic circuits can be broadly classified into 2 types:

i) Combinational Logic
ii) Sequential Logic

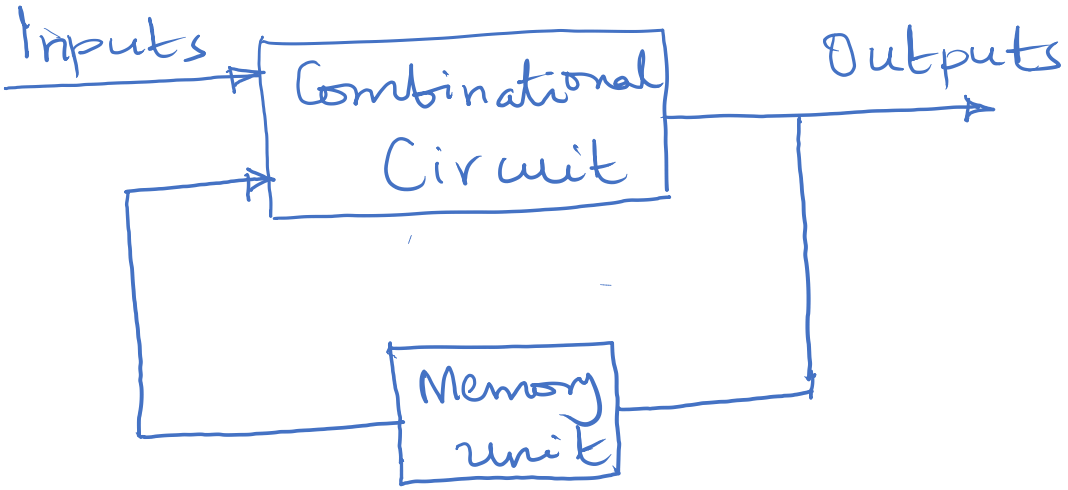
### Combinational Circuit

- Consists of logic gates whose outputs at any time are determined by logic operations on the input values i.e. outputs depend on the current input values.
- Perform an operation that can be specified logically by a set of Boolean equations
- Are memoryless
- Consist of input variables, output variables, logic gates, and interconnections



**Sequential Circuit**

- Can be regarded as combinational circuits that incorporate memory i.e. employ elements that store bit values
- Their outputs are functions of the inputs and the bit values in the storage elements. Hence, outputs of sequential circuit depend not only on the current input values, but also on past inputs.
- The behavior of the circuit must be specified by a sequence in time of inputs and internal stored bit values




**Combinational Logic Design**

- Typical combinational logic modules include: binary adders, subtractors, comparators, decoders, encoders, multiplexers, and demultiplexers.
- Analysis and design of logic circuits is based on Boolean algebra.

**Boolean algebra**

- Enables logic statements to be represented as mathematical functions.
- An integral part of digital electronic circuit design
- 3 basic logic operations: **OR** (**+**); **AND** (**.**); **NOT** ($\bar{A}$) / A'
- 8 basic theorems of Boolean logic:

| | | | |
|---|---|---|---|
| i) | $1 + A = 1$ | v) | $0.A = 0$ |
| ii) | $0 + A = A$ | vi) | $1.A = A$ |
| iii) | $A + A = A$ | vii) | $A.A = A$ |
| iv) | $A + \bar{A} = 1$ | viii) | $A.\bar{A} = 0$ |

- 3 Boolean algebra laws:

i) Commutative Laws:
   a) A + B = B + A
   b) AB = BA

iii) Associative Laws:
   a) A + (B+C) = (A+B) + C
   b) A(BC) = (AB)C

ii) Distributive Law:
   a) A (B+C) = AB + AC

- Simplification of logic functions / Boolean expressions

ii) DeMorgan's Theorem

i) Karnaugh Maps

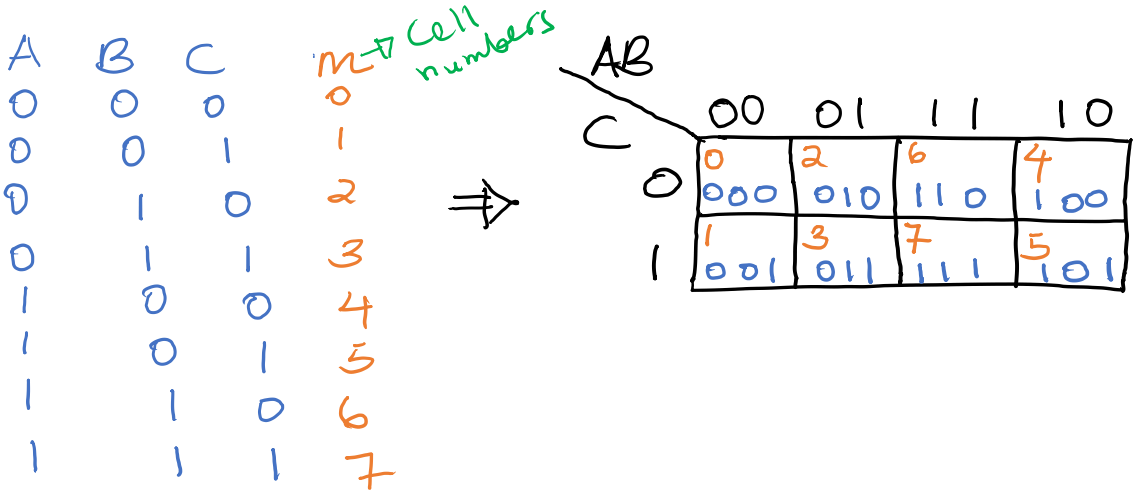
***DeMorgan's Theorem** **: If A, B and C are logic variables:

i) $\overline{A+B+C} = \bar{A}.\bar{B}.\bar{C}$

ii) $\overline{ABC} = \bar{A} + \bar{B} + \bar{C}$

***Karnaugh Maps (K-map) ****

- Well suited for the simplification of logic expressions with up to 4 logic variables
- Consider a 3-variable (ABC) K-map:




***Examples and assignments to be given in physical class***