KONERU LAKSHMAIAH EDUCATION FOUNDATION
(Deemed to be University estd, u/s, 3 of the UGC Act, 1956)
(NAAC Accredited "A++" Grade University)
Green Fields, Guntur District, A.P., India – 522502
Department of Basic Engineering Science - II

I B.Tech. II Semester – CSE / AI & DS / ECE / EEE / CS & IT / IOT

A.Y.2024-25 - EVEN SEMESTER

Digital Design and Computer Architecture (23EC1202)

CO – 1: Combinational Digital Logic Circuits

Session 4: Adders & Subtractors

1. Course Description (Description of the subject):

The course on "Digital Design and Computer Architecture" provides a comprehensive exploration of the foundational principles in digital design process and computer organization. Students explore the concepts of combinational and sequential circuits, memory circuits. The curriculum extends to the Basic computer architecture concepts, memory hierarchies, and input/output fundamentals, fostering a deep understanding of computer organization. Through practical projects and simulations, students develop the skills to design and implement digital circuits. Graduates emerge with a robust skill set, ready to embark on careers in hardware design, computer architecture, and related fields, equipped to contribute to the ever-evolving landscape of digital technology.

2. Aim of the Course:

The course aims to equip students with the knowledge and skills related to:

i. Proficiency in designing and optimizing Combinational and Sequential Circuits using Boolean algebra and programmable logic devices with a solid foundation in digital design.

ii. Skill development using hands-on experience in designing digital circuits which includes latches, flip-flops, and counters in combination with memory, registers, and timing and sequence control modules using hardware & modeling tools.

iii. Explore the architecture of modern computers, including the organization and structure of central processing units, memory systems, and input/output interfaces.

iv. Bridge theoretical concepts with real-world applications by examining case studies and examples of digital design and computer architecture in modern computing systems.

Overall, the aim of the course is to prepare the student well-equipped to apply their knowledge to the design and analysis of digital systems and computer architectures, preparing them for careers in areas such as hardware design, computer engineering, and embedded systems development.

3. Instructional Objectives (Course Objectives):

The course objectives for "Digital Design and Computer Architecture" typically include:

i. To Understand and apply foundational concepts in digital design which results in proficiency over designing and analyzing combinational and sequential logic circuits.

ii. To Gain hands-on experience with industry-standard simulation and modeling tools, for verifying and testing digital designs.

iii. To analyze the architecture of a computer system, including the organization and operation of the CPU, memory hierarchy, and input/output subsystems.

iv. To apply digital design and computer architecture principles to solve real-world engineering problems and challenges by reinforcing theoretical knowledge with hands-on experience.

4. Learning Outcomes (Course Outcomes):

i. Able to build the combinational and programmable digital logic circuits using logic gates and optimization methods.

ii. Able to construct the sequential and memory circuits using flip-flops, demonstrating a comprehensive understanding of the principles governing clocked sequential logic.

iii. Able to organize computer architecture and instructions sequence through a grasp of the foundational principles that govern the organization and functioning of a computer system.

iv. Capable of modeling Memory Architecture and I/O Organization modules proficiently.

v. Able to develop and analyze the computer architecture modules using basic combinational, sequential and memory logics.

5. Module Description (CO - 1 Description):

The module covers essential topics in digital electronics, starting with Boolean algebra and progressing to the representation and optimization techniques of digital logic using SOP/POS forms. Students will delve into the design of key components such as adders, subtractors, multiplexers, de-multiplexers, decoders, and encoders. The module introduces the concept of reversible gates, exploring their unique properties. Additionally, students will gain insights into Programmable Logic Devices (PLDs) like PROM, PAL, and PLA, understanding their design principles. The implementation of Complex Programmable Logic Devices (CPLDs) with macrocells and Field-Programmable Gate Arrays (FPGAs) featuring Configurable Logic Blocks (CLBs) and Look-Up Tables (LUTs) will be covered. Practical applications of these digital logic modules in various scenarios will be emphasized, providing students with a comprehensive understanding of digital electronics and its real-world applications.

6. Session Introduction:

In digital circuits, a half adder is a basic component used to add two single-bit binary numbers, producing a sum and carry output. It has two input bits representing the numbers to be added and generates a sum bit (S) and a carry bit (C). However, a half adder does not account for any carry-in from previous addition operations, limiting its use to single-bit addition. Conversely, a full adder is an enhanced version capable of adding two single-bit binary numbers along with an additional carry-in from a previous stage of addition. It produces a sum bit (S) and a carry-out bit (Cout), facilitating multi-bit addition by handling carry propagation between stages.

7. Session Description:

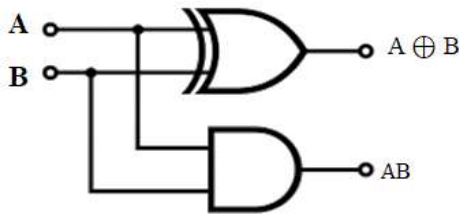
7.1 Half Adder:

A half adder is a combinational logic circuit that performs the arithmetic addition of two bits. Such a circuit thus has two inputs that represent the two bits to be added and two outputs, with one producing the SUM output and the other producing the CARRY.

Truth Table:

A	B	SUM	CARRY
0	0	0	0
0	I	I	0
I	0	I	0
I	I	0	I

Logic Diagram:



Output Expressions: Sum = A'B+AB'=A ⊕ B Carry = AB

The Boolean expressions for the SUM and CARRY outputs are given by the equations

$$S = A'B + AB' = A \oplus \mathrm{B}$$

$$C = A \cdot B$$

7.2 Full Adder:

A full adder circuit is an arithmetic circuit block that can be used to add three bits to produce a SUM and a CARRY output. Two of the input variables X and Y represent the two significant bits to be added and the third input Z represents the carry from the previous lower significant position.

A	B	C	S	CY
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

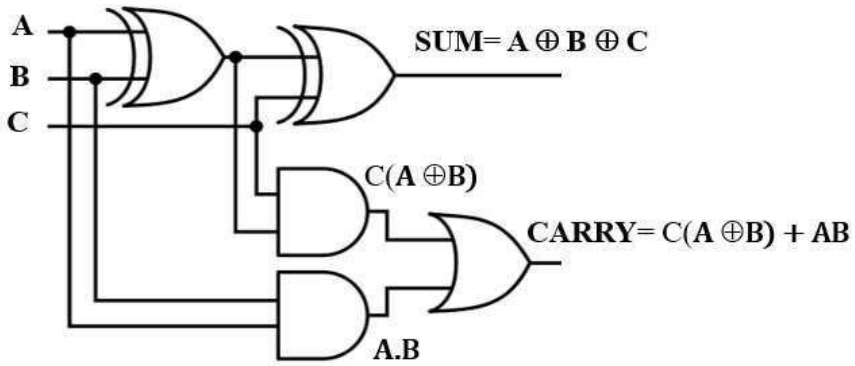
$$S = \sum m(1, 2, 4, 7)$$

$$CY = \sum m(3, 5, 6, 7)$$

$$S = A'B'C + A'BC' + AB'C' + ABCC = AB + BC + AC$$

$$S = A'B'C + A'BC' + AB'C' + ABC = C'(A'B + AB') + C(AB + A'B') = A \oplus \mathrm{B} \oplus \mathrm{C}$$

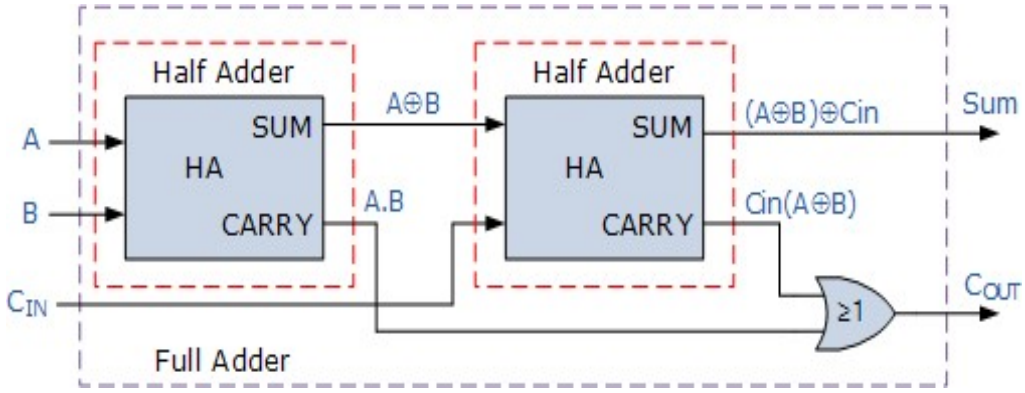
$$CARRY = A'BC + AB'C + ABC' + ABC = C(A'B + AB') + AB(C + C')$$
$$= AB + C(A \oplus \mathrm{B})$$



Full adder using two half adders:

A full adder can indeed be constructed using two half adders and an additional OR gate. Here's how it's done:

- **Using Two Half Adders:**
  First, use one half adder to add the two input bits (A and B). This produces a partial sum (S1) and a carry (C1). Then, use another half adder to add the partial sum (S1) from the first half adder and the third input bit (Cin). This produces the final sum (S) and another carry (C2).
- **Using an OR Gate:**
  Finally, combine the carry outputs (C1 and C2) from the two half adders using an OR gate to obtain the overall carry-out (Cout) from the full adder. The resulting circuit effectively adds three input bits (A, B, and Cin), producing a sum output (S) and a carry-out (Cout). This arrangement allows for the construction of a full adder using only two half adders and an OR gate, making it a simple and efficient implementation of the full adder functionality.



7.3 Half-Subtractor:

A Half-Subtractor is a combinational circuit that can be used to subtract one binary digit from another to produce a DIFFERENCE output and a BORROW output. The BORROW output here specifies whether a '1' has been borrowed to perform the subtraction.

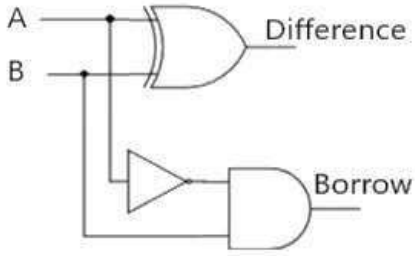
$$Difference = (A \oplus \mathrm{B})$$

$$Borrow = (A' \mathrm{B})$$

Truth Table:

Inputs		Outputs	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Logic Diagram:



7.4 Full-Subtractor:

A full subtractor performs subtraction operation on two bits, a minuend and a subtrahend, and also takes into consideration whether a '1' has already been borrowed by the previous adjacent lower minuend bit or not. This circuit has three inputs and two outputs. The three inputs, X, Y and Z denote the minuend, subtrahend, and previous borrow, respectively. The two outputs, D and B represent the difference and output borrow, respectively.

X	Y	Z	D	B
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

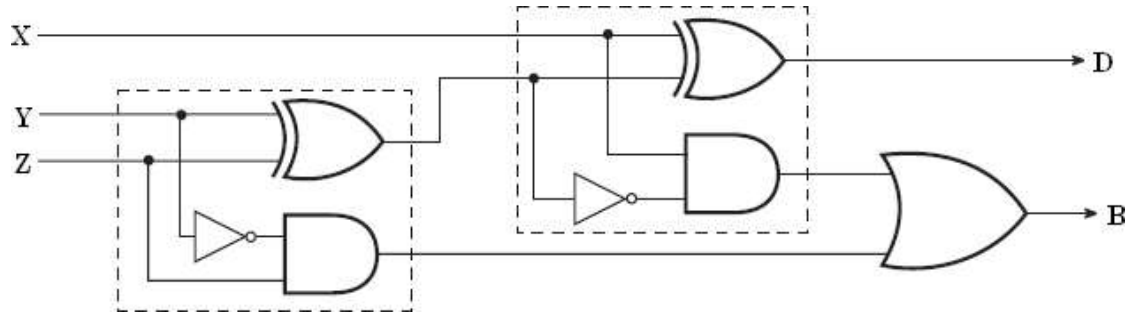
$$D = \sum m(1,2,4,7) = X'Y'Z + X'YZ' + XY'Z' + XYZ$$

$$B = \sum m\,(1,2,3,7) = X'Y + X'Z + YZ$$

From truth table,

$$D = X'Y'Z + X'YZ' + XY'Z' + XYZ = Z'(X'Y + XY') + Z(XY + X'Y') = X \oplus \mathrm{Y} \oplus \mathrm{Z}$$

$$B = X'Y'Z + X'YZ' + X'YZ + XYZ = X'Y + Z(X \oplus \mathrm{Y})'$$



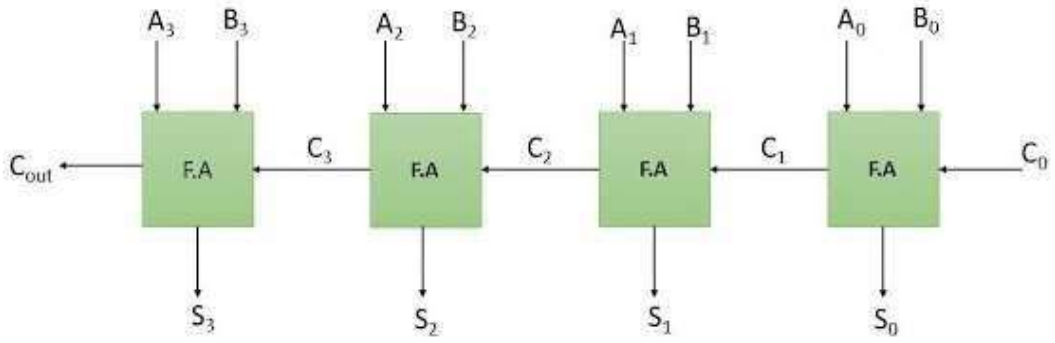
7.5 Binary Parallel Adder:

A 4-bit binary parallel adder adds two 4-bit binary numbers together, producing a sum and a possible carry-out. It can be constructed using a combination of full adders. Here's how it's typically organized:

Inputs: Two 4-bit binary numbers, A and B.

Outputs: A 4-bit sum (S) and a carry-out (Cout).

The circuit comprises four full adders, each handling one bit of the addition operation. The input bits from the two numbers A and B are fed into corresponding inputs of the full adders. Additionally, the carry-out (Cout) from each full adder is connected to the carry-in (Cin) of the next higher-order full adder. This carry propagation ensures that carry bits are correctly propagated across all bits during addition. The output sum (S) from each full adder forms the bits of the 4-bit sum, while the carry-out (Cout) from the last full adder represents the overall carry-out from the addition operation.



Advantages of parallel Adder/Subtractor –

- The parallel adder/subtractor performs the addition operation faster as compared to serial adder/subtractor.
- Time required for addition does not depend on the number of bits.
- The output is in parallel form i.e all the bits are added/subtracted at the same time.
- It is less costly.

8. SAQ's – Self Assessment Questions:

1. In which operation carry is obtained?
   a) Subtraction
   b) **Addition**
   c) Multiplication
   d) Both addition and subtraction

2. The difference between half adder and full adder is.
   a) Half adder has two inputs while full adder has four inputs

b) Half adder has one output while full adder has two outputs

c) Half adder has two inputs while full adder has three inputs

d) All of above

3. Half subtractor is used to perform subtraction of _____

a) 2 bits

b) 3 bits

c) 4 bits

d) 5 bits

4. How many outputs are required for the implementation of a subtractor?

a) 1

b) 3

c) 2

d) 4

9. Terminal Questions:

- Design a full adder using two half adders.
- Draw a half subtractor principle with input and output and truth table.
- Describe the full adder using a block diagram, list its truth table and output equations.
- Design a Full Adder circuit utilizing an appropriate decoder and OR gates.
- Realize the full adder using two half adders.
- Realize the output functional equations of Full adder using required logic gates.

10. References books:

- Computer System Architecture by M. Moris Mano
- Fundamentals of Digital Logic with Verilog HDL by Stephen Brown and ZvonkoVranesic

11. Sites and Web links:

- https://www.tutorialspoint.com/adders-and-subtractors-in-digital-electronics
- https://technobyte.org/half-adder-full-adder-half-subtractor-full-subtractor/