TOPIC 5 – THE ARITHMETIC LOGIC UNIT (ALU)

LEARNING OUTCOMES

By the end of this topics, you will be able to:

1. describe the construction of the ALU.
2. explain the binary half-adder, full adder and binary-coded-decimal (BCD) adder.
3. explain positive and negative BCD numbers, shift operation and high speed arithmetic.

INTRODUCTION

This topic explains about the Arithmetic Logic Unit (ALU) construction, binary half-adder, binary full adder, binary coded decimal (BCD), positive and negative BCD, shift operation and high speed arithmetic.

5.1 CONSTRUCTION OF THE ALU

The arithmetic logic unit (ALU) is a part of computer that performs arithmetic and bitwise operations on integer binary numbers which contrast to a floating-point unit (FPU) that operates on floating point numbers. It consists of combinational digital electronic circuit that becomes a fundamental building block of many types of computing circuits, including the central processing unit (CPU) of computers.

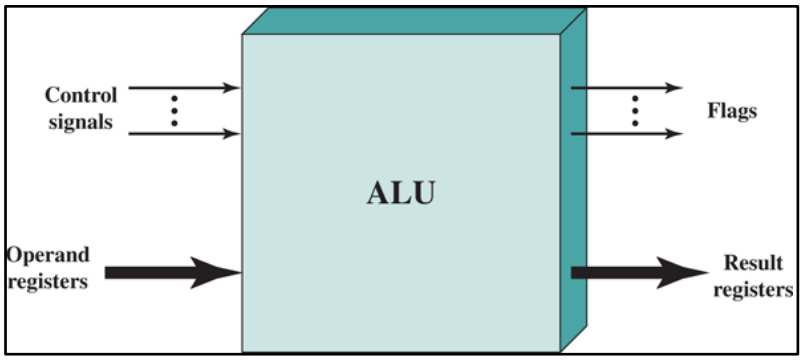

Figure 5.28

Figure 5.28 is showing the direction of control signals and operand register to ALU for processing and the flags and result register produced after the processing.

SELF CHECK 5.1

1. Describe the ALU.

5.2 BINARY HALF-ADDER

The binary half adder is a logic circuit (Figure 5.29) that has 2 inputs; A and B that added together. It has two outputs; sum (S) and carry (C). The value of the sum in decimal system is $2C + S$. The carry signal represents an overflow into the next digit of a multi-digit addition.

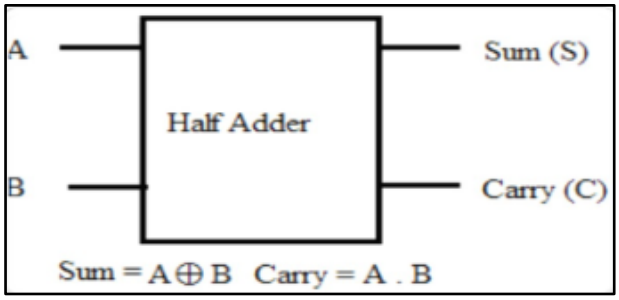

Figure 5.29

The simplest half-adder design shown in Figure 5.30. It incorporates an XOR gate for S and an AND gate for C. With the addition of an OR gate to combine their carry outputs, two half adders can be combined to make a full adder.

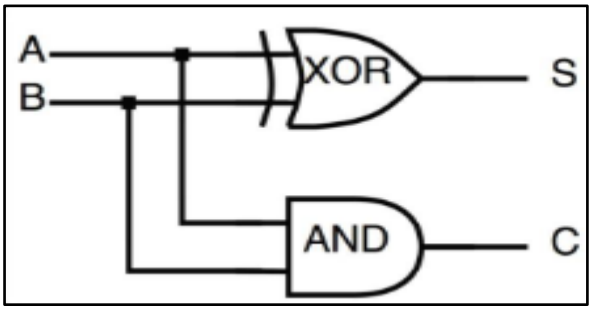

Figure 5.30

The half-adder truth table is as shown in Table 5.9 where S= A ⊕ B (Exclusive OR) and C = A.B (AND)

Table 5.9

A	B	Sum	Carry-Out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

SELF CHECK 5.2

1. Explain the binary half-adder.
2. Draw a truth table of half-adder.

5.3 FULL ADDER

The full-adder has 3 inputs; A, B and C_{in} that adds three one-bit binary numbers. The outputs are 2 one-bit binary numbers, a sum (S) and a carry (C_{out}). The full-adder is usually a component in a cascade of adders that add 8, 16, 32 binary numbers. The logic circuit of full adder is as shown in Figure 5.31.

Figure 5.31

The simplest way to construct a full adder logic diagram is to connect two half- adder and an OR gate as shown in Figure 5.32. The full-adder is then the fundamental logic circuit incorporated in digital computers to perform arithmetic functions.

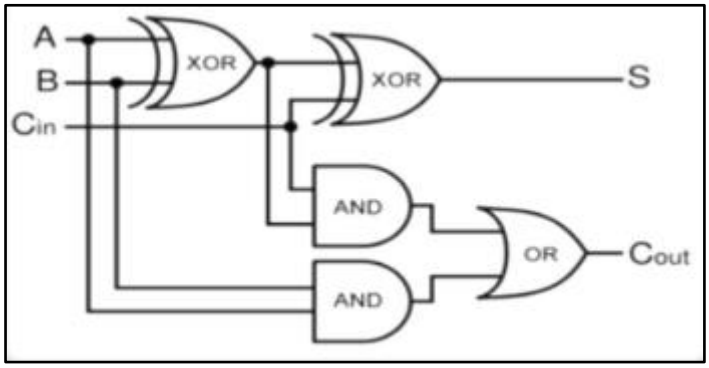

Figure 5.32

The full adder truth table is represented in Table 5.10 where $S= A \oplus B \oplus C_{in}$ and $C=AB + C_{in} (A \oplus B)$.

Table 5.10

A	B	Carry-In	Sum	Carry-Out
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

SELF CHECK 5.3

1. Explain the binary full adder.
2. Draw a truth table of full adder.

5.4 BINARY CODED DECIMAL (BCD) ADDER

Binary Coded Decimal (BCD) adder has capability of adding two 4-bit words format. The input value should only apply from 0..9 to the inputs of the adder. This is because the remaining values A..F are undefined for BCD arithmetic. Click the hex-switches or use the 'a' and 'b' bindkeys to select the input values for the adder.

For example, the inputs are A = 0111 (7) and B = 1000 (8). The output is Y= 1 0101 the value of binary of A and B is sum add and will give the result 1111 (15) however with BCD, the sum will be 10101 where 1 is 000 in binary and 5 is 0101 in binary.

SELF CHECK 5.4

1. Explain the BCD.

5.5 POSITIVE AND NEGATIVE BCD NUMBERS

In computing and electronic systems, binary-coded decimal (BCD) is a class of binary encodings of decimal numbers where each decimal digit is represented by a fixed number of bits, usually four or eight. Special bit patterns are sometimes used for a sign or for other indications

In byte-oriented systems (i.e. most modern computers), the term unpacked BCD usually implies a full byte for each digit (often including a sign), whereas packed BCD typically encodes two decimal digits within a single byte by taking advantage of the fact that four bits are enough to represent the range 0 to 9. The precise 4-bit encoding may vary however, for technical reasons,

SELF CHECK 5.5

1. Describe the positive and negative BCD numbers.

5.6 SHIFT OPERATION

Shift operation has been vacated by zero-filled in the bit positions. For signed numbers, the sign bit is used to fill the vacated bit positions. In other words, if the number is positive, 0 is used, and if the number is negative, 1 is used.

The left-shift operator causes the bits in *shift-expression* to be shifted to the left by the number of positions specified by *additive-expression*. The bit positions that have been vacated by the shift operation are zero-filled. A left shift is a logical shift (the bits that are shifted off the end are discarded, including the sign bit).

SELF CHECK 5.6

1. Explain the shift operation.

5.7 HIGH SPEED ARITHMETIC

High speed arithmetic is a method of obtaining high speed in addition, multiplication, and division in parallel binary computers. It is then compared with each other as to efficiency of operation and cost.

The transit time of a logical unit is used as a time base in comparing the operating speeds of different methods, and the number of individual logical units required is used in the comparison of costs.

SELF CHECK 5.7

1. What is high speed arithmetic?

SUMMARY

In this topic you have learnt that:

The Arithmetic Logic Unit (ALU) construction, binary half-adder, binary full adder, binary coded decimal (BCD), positive and negative BCD, shift operation and high-speed arithmetic.

KEY TERMS

Arithmetic Operations The arithmetic operations consist of addition, subtraction, multiplication, and division operations.

REFERENCES

Stallings, W., (2019). *Computer Organization and Architecture Designing for Performance.* 11th ed. New York: Pearson.