**TASK – 1**

The Arithmetic Logic Unit (ALU) is the heart of any CPU. An ALU performs three kinds of operations, i.e.

* Arithmetic operations such as Addition/Subtraction,
* Logical operations such as AND, OR, etc. and
* Data movement operations such as Load and Store

ALU derives its name because it performs arithmetic and logical operations. A simple ALU design is constructed with Combinational circuits. ALUs that perform multiplication and division are designed around the circuits developed for these operations while implementing the desired algorithm. More complex ALUs are designed for executing Floating point, Decimal operations and other complex numerical operations. These are called Coprocessors and work in tandem with the main processor.

The design specifications of ALU are derived from the Instruction Set Architecture. The ALU must have the capability to execute the instructions of ISA. An instruction execution in a CPU is achieved by the movement of data/datum associated with the instruction. This movement of data is facilitated by the Datapath. For example, a LOAD instruction brings data from memory location and writes onto a GPR. The navigation of data over datapath enables the execution of LOAD instruction. We discuss Datapath more in details in the next chapter on Control Unit Design. The trade-off in ALU design is necessitated by the factors like Speed of execution, hardware cost, the width of the ALU.

**Combinational ALU**

A primitive ALU supporting three functions AND, OR and ADD is explained in figure 11.1. The ALU has two inputs A and B. These inputs are fed to AND gate, OR Gate and Full ADDER. The Full Adder also has CARRY IN as an input. The combinational logic output of A and B is statically available at the output of AND, OR and Full Adder. The desired output is chosen by the Select function, which in turn is decoded from the instruction under execution. Multiplexer passes one of the inputs as output based on this select function. Select Function essentially reflects the operation to be carried out on the operands A and B. Thus A and B, A or B and A+B functions are supported by this ALU. When ALU is to be extended for more bits the logic is duplicated for as many bits and necessary cascading is done. The AND and OR logic are part of the logical unit while the adder is part of the arithmetic unit.

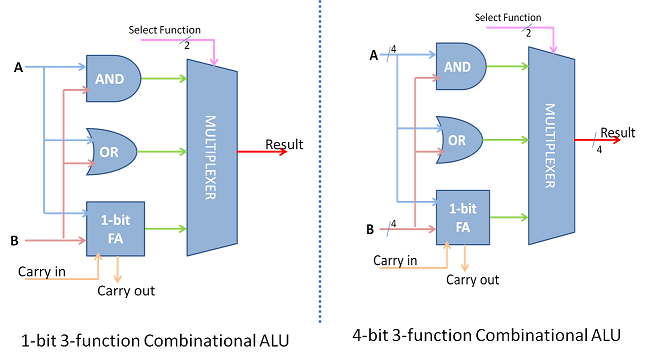


Fig 1: 1-bit 3 and 4-bit 3 function combinational ALU

The simplest ALU has more functions that are essential to support the ISA of the CPU. Therefore the ALU combines the functions of 2's complement, Adder, Subtractor, as part of the arithmetic unit. The logical unit would generate logical functions of the form f(x,y) like AND, OR, NOT, XOR etc. Such a combination supplements most of a CPU's fixed point data processing instructions.

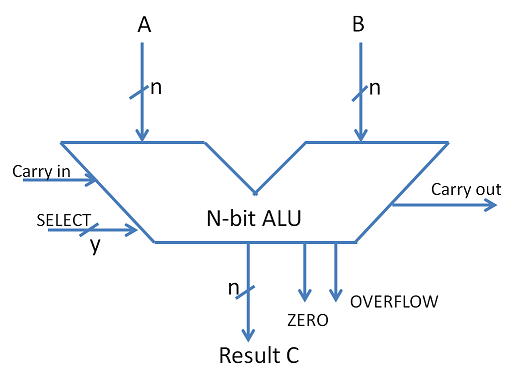
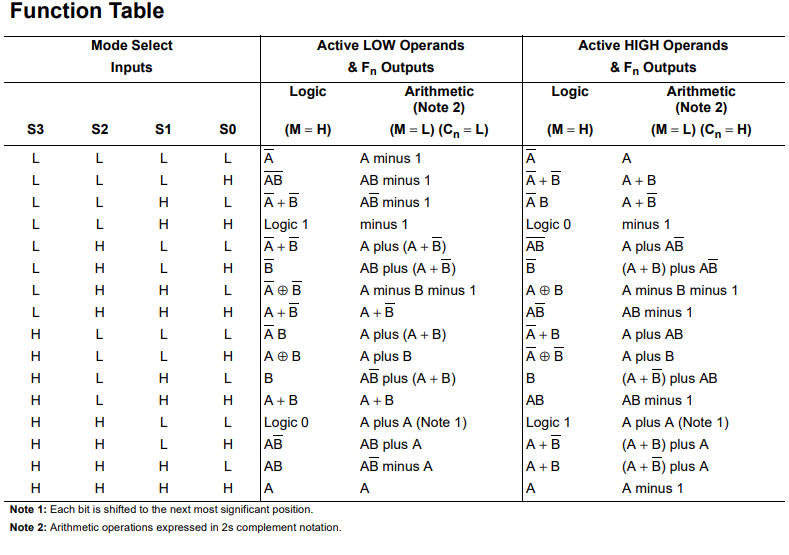


Fig 2: ALU symbol

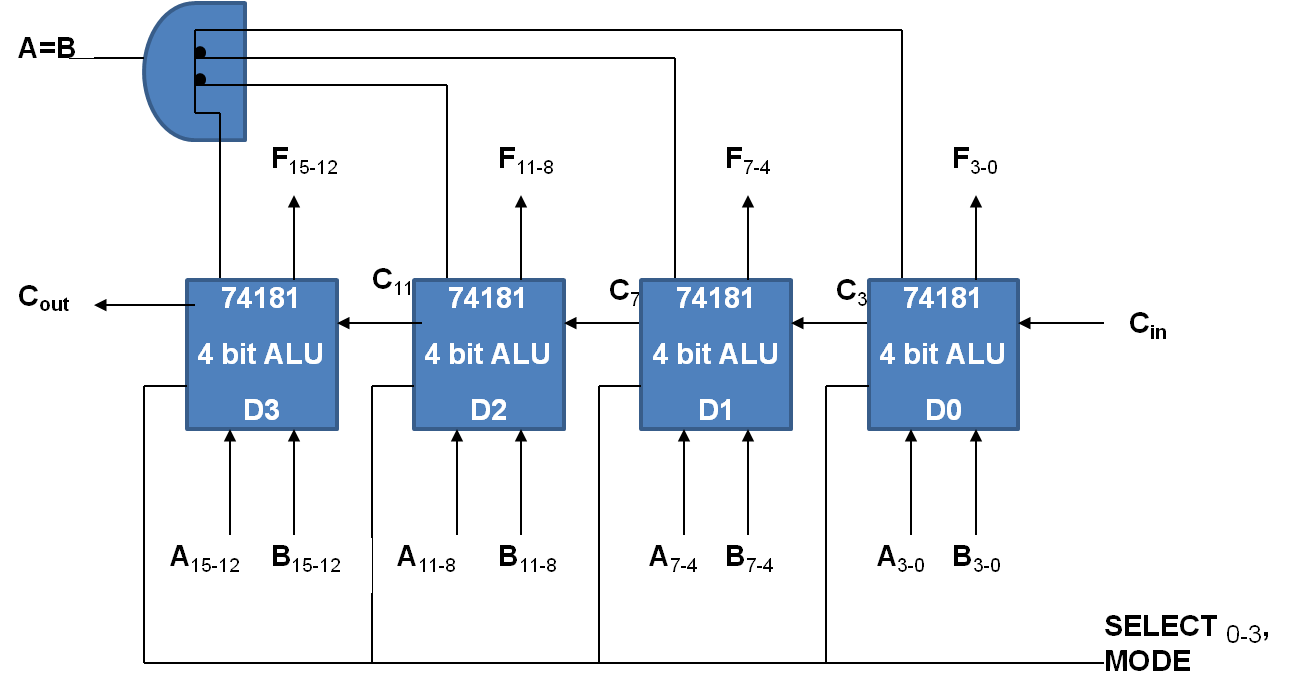
## ****74181 Arithmetic Logic Unit Integrated Chip****

74181 is cascadable ALU of the 1960s and first of the kind. ALU operation and complexity is better understood by the features of 74181, although much more is offered by modern ALUs. Even today 74181 is of academic interest in teaching Computer architecture.

* 4-bit Arithmetic and Logical Unit for fixed-point operations
* Inputs: Two operands A and B of 4-bit width
* Output: F 0-3 ( 4 bit width)
* Mode selection with M - defines Arithmetic or logical mode
* Function select with 4 lines (S0-3)
* 16 sets of Arithmetic and 16 Logical Operations possible ( as detailed in Function table)
* Carry in used as a special input; Cin disabled for logical operations
* Look Ahead Carry Adder Principle employed for faster output propagation with P, G outputs
* Carry Output
* A=B comparator output



As we see from the table, the logical operations are AND, OR, NOT, NAND, NOR, XOR, A Not, B Not etc. The arithmetic operations are ADD, Subtract, Shift, 2's complement, compare, Double, etc. There are few bizarre functions too which are rarely used. The functions and logic in the table is an example of what an ALU is. In the microprocessor era, 74181 is not in use. ALU is in-built in the microprocessor. The p, g and Cout outputs are intended to allow k-copies of the 74181 to be combined either using ripple-carry propagation or carry look ahead to form 4k bit ALU. This ALU is expandable to more word width by cascading and is shown in figure 11.3. AMD 2901 is a microprocessor-based 4-bit bit-sliced cascadable ALU. This supports 3 arithmetic and 5 logical functions.



## ****ALU Expansion****

ALU of size **m** can easily be expanded to handle operands of size **n = km** or any word size **n > m** and is done in two ways.

**Spatial expansion (bit sliced ALU)**: Connect **k**-copies of **m**-bit ALU in the manner of a ripple carry adder to form a single ALU capable of processing **km** bit words directly. The resulting circuit is said to be bit sliced because each block of ALU concurrently processes a separate "slice" of **m** bits from each of the **km** bit operands.

**Temporal Expansion (Multi-cycle ALU)**: Use one copy of **m**-bit ALU chip to perform operations on **km**-bit words in **k** consecutive steps i.e. **k** clock cycles. In each step, the ALU processes a separate **m** bit slice of each operand. Hence this processing is called multi-cycle.

The hardware cost of bit sliced ALU is directly proportional to the number of slices **k**, while the CPI per instruction remains the same. In multi-cycle ALU the performance is inversely proportional to **k** (here **k** refers to cycles) but the amount of hardware required remains constant. Multi-cycle ALU must be controlled by a micro-programmed control unit.