Design and implement a MAC (multiply and accumulate) unit in VHDL. The design is composed of three main components: and adder, a multiplier, and a register. Figure 1 shows how these components are connected.

*At each rising edge of the CLK signal, the register is updated with a new value. Assume that the clock signal is long enough to allow two new *a* and *b* values go through the multiplier and adder within a single clock.

* The bit width of the system is 16: inputs (a and b), output, accumulator, adder, and multiplier are all 16 bits. There are different ways to define and use 16 bit data. The simplest way is using bit_vector.

* If the multiplier or adder results are more than 16 bits, then an overflow signal should be asserted.

* For this design, first implement a multiplier, an adder and a register in data flow. For data flow sum and multiply, simply use the + and * operators. This way, the architecture of the adder, for example, is simply written as
O<=a * b;
Look in your VHDL books how to use + and * with bit_vector. Then, connect them by a structural VHDL code.
* Write a testbench to pass at least 10 pairs of 16-bit input numbers to the design.

* Should you need more assumptions, signals, parameters, and any other stuff not mentioned in the quiz description, make it yourself.

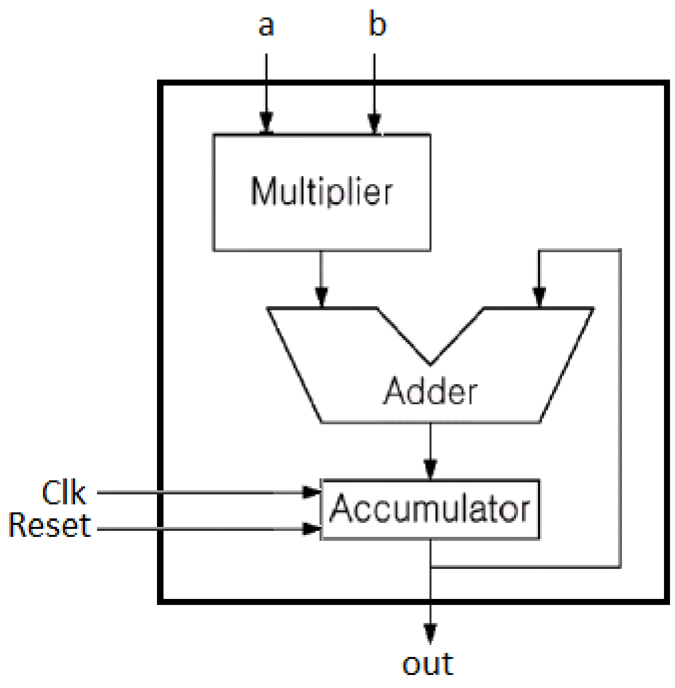

Figure 1. The architecture of a MAC unit