

This chip is a 16bits chip which contains all functions that mentioned in the question sheet, and the 7th pin is multiply.

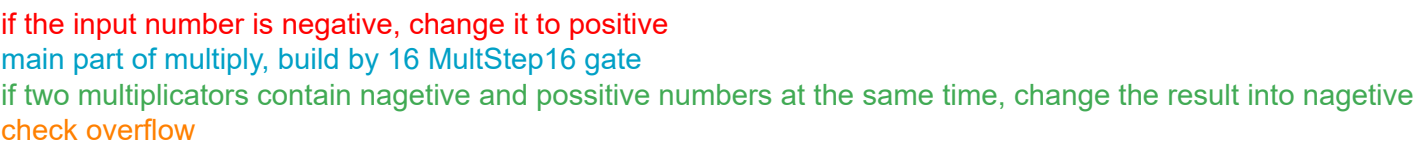
My method is :

- In order to get the negative number, my method is:

- * Put result into 'Not16' gate.
- * Use 'Add16' chip plus the result with A 16-bits bus full of 1 (false, 111.....1, which same as plus -1 or minus 1).

* Build two 16-bits bus, one is 'hi' and the other is 'lo', treat them as a 32-bits bus. For two 16-bits multipliers, the result is up to 32-bits, therefore register 'hi' will never overflow.

The following diagram is the main part of Mult16.HDL, please scale up to 200 %.



The diagram illustrates the proposed architecture's data flow. It starts with two 16-bit inputs, x_{lo} and x_{hi} , and a 'false' signal. These inputs feed into two multiplexers, Mux16 and Mux10. Mux16 selects between x_{lo} and the 'false' signal to produce xx_{lo} . Mux10 selects between x_{hi} and the output of Mux16 to produce xx_{hi} . These intermediate results, along with $precut_{lo}$ and $precut_{hi}$, are then processed by two 16-bit adders (Add16) to produce out_{lo} and out_{hi} . Finally, the outputs of the adders and the $precut$ signals are fed into a 16-bit integer shifter (IShift16) to produce the final outputs $xout_{lo}$ and $xout_{hi}$.