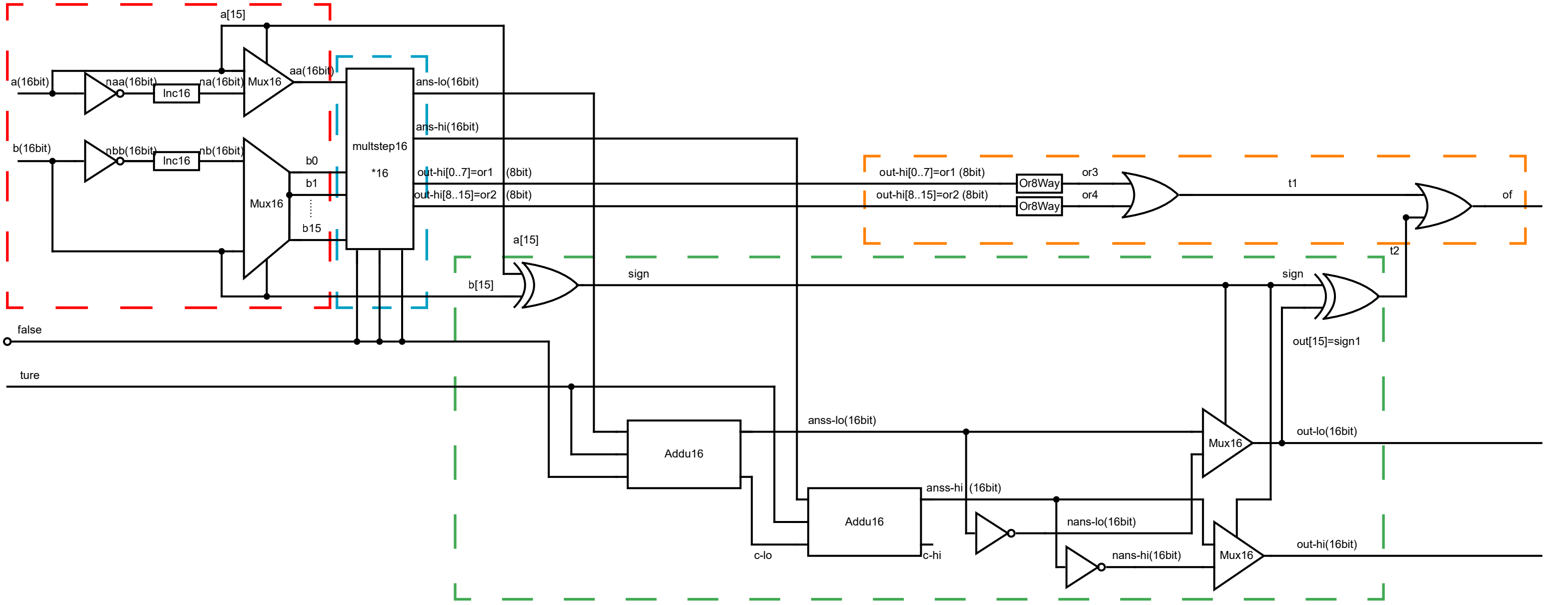
cALU16:
This chip is a 16bits chip which contains all functions that mentioned in the question sheet, and the 7th pin is multiply.
To activate 7th pin, the control bit should like this: nx=0, zy=0, ny=0, no=0, cx=0, re=0, op=1.

The Hardwaresimulator use two's complement, therefore it is complex for multiply between nagetive numbers. My method is :
* Change negative number into possitive number and do the multiply.
* Check the signal of the result and decide whether to change the result into its opposite (for example, if two multiplicators contain nagetive and possitive numbers at the same time, the result should be nagetive).

In order to get the opposite number(negative), my method is:
* Put result into ‘Not16’ gate.
* Use ‘Addu16’ chip (custom chip) plus the result with A 16-bits bus full of 1 (false, 111.....1, which same as plus -1 or minus 1).

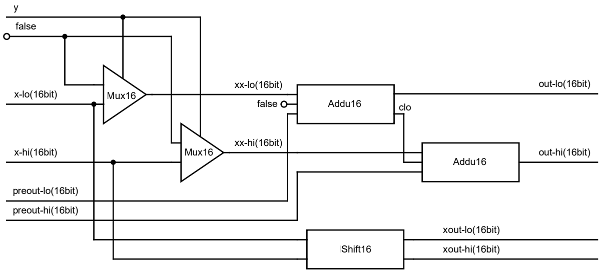
I choose to check the overflow when the result is still a posstive number. The way I deal with overflow-checking is:
* Build two 16-bits bus, one is ‘hi’ and the other is ‘lo’, treat them as a 32-bits bus. For two 16-bits multiplicators , the result is up to 32-bits, therefore register ‘hi’ will never overflow.
* If it is not ‘000...0’ or ‘111...1’ in bus ‘hi’, or the sign bit of ‘lo’ is wrong (for example, the result of two possitive multiplicators is negative), then an overflow occurs.

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

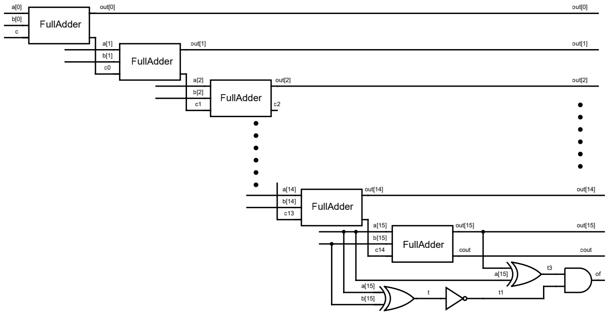


if the input number is negative, change it to positive
main part of multiply, build by 16 MultStep16 gate
if two multiplicators contain nagetive and possitive numbers at the same time, change the result into nagetive
check overflow

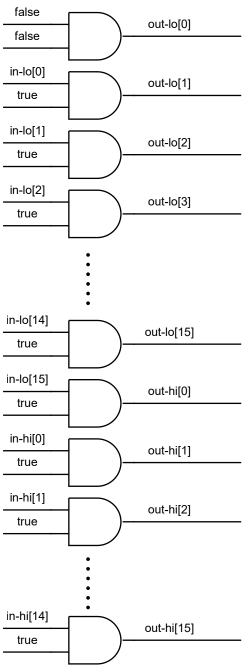
MultStep16.HDL



Addu16.HDL



lShift16.HDL



Powered by: Circuit Diagram Adobe Illustrator