

IN3160 Oblig 7

Chris René Lerner

Task 1

Adding two binary numbers end up with a maximum of 1 overflow bit. Take the worst case as an example where every bit gets shifted one place to the right:

$$\begin{array}{r} 1111\ 1111\ 1111\ 1111 \\ +\ 1111\ 1111\ 1111\ 1111 \\ \hline =\ 1\ 1111\ 1111\ 1111\ 1110 \end{array} \quad (1)$$

Therefore, the answer is 17 bits.

Task 2

Multiplying the two numbers ends up with a less obvious amount of bits. Lets take the worst case, do a multiplication in base 10, and converting to binary.

$$(1111\ 1111\ 1111\ 1111)_2 = (65\ 535)_{10} \quad (2)$$

$$(65\ 535)_{10} \cdot (65\ 535)_{10} = (4\ 294\ 836\ 225)_{10} \quad (3)$$

$$(4\ 294\ 836\ 225)_{10} = (FFFE0001)_{16} = (1111\ 1111\ 1111\ 1110\ 0000\ 0000\ 0000\ 0001)_2 \quad (4)$$

The end result of the worst case multiplication is 32 bits.

Task 3

Say all four numbers are FFFF. $a + b$ results in 17 bits. $c + d$ results in 17 bits.

$$\begin{array}{r}
 1\ 1111\ 1111\ 1111\ 1110 \\
 \hline
 +\quad 1\ 1111\ 1111\ 1111\ 1110 \\
 \hline
 =\ 11\ 1111\ 1111\ 1111\ 1100
 \end{array} \tag{5}$$

The addition of these two 17 bit sums will then result in 3FFFC which is 18 bits.

Task 4

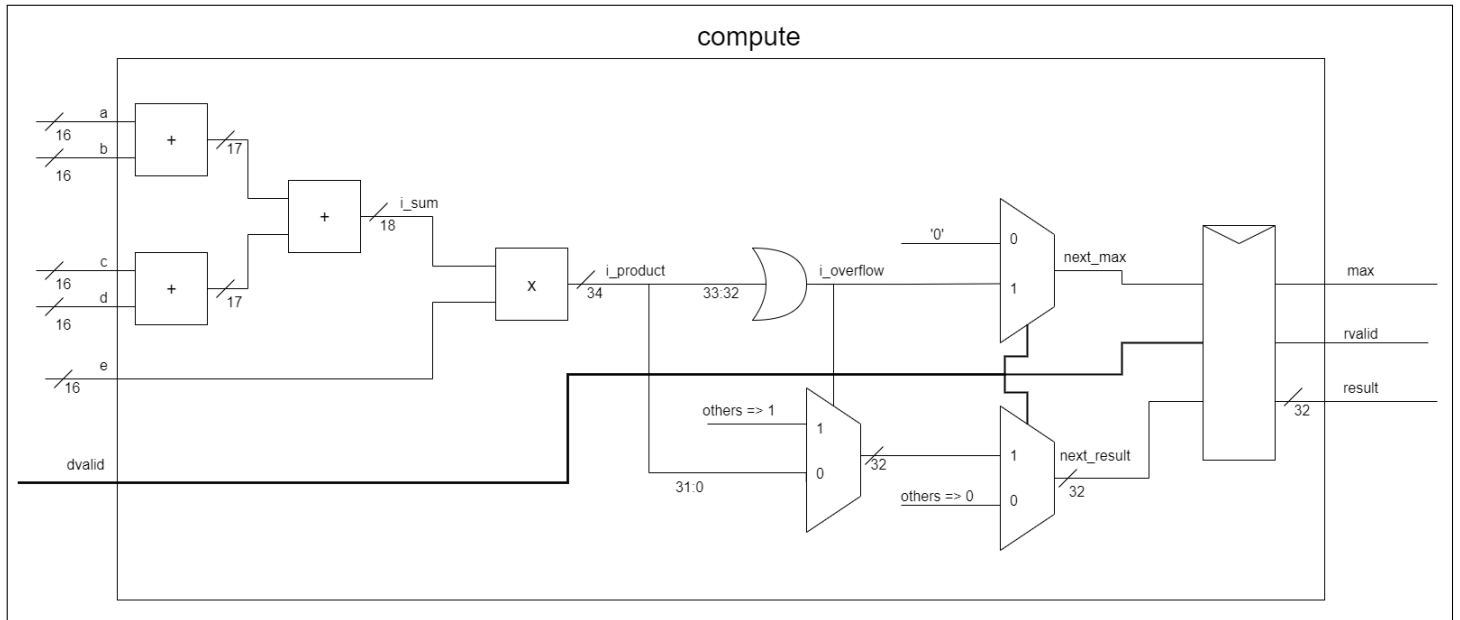
We also assume that all the numbers are FFFF, including e. We then end up with multiplying 18 bits with 16 bits. The multiplication goes like this:

$$\begin{aligned}
 (3\ FFFC)_{16} \cdot (FFFF)_{16} &= (262\ 140)_{10} \cdot (65\ 535)_{10} \\
 &= (17\ 179\ 344\ 900)_{10} = (3\ FFF8\ 0004)_{16} \\
 &= (11\ 1111\ 1111\ 1111\ 1000\ 0000\ 0000\ 0000\ 0100)_2
 \end{aligned} \tag{6}$$

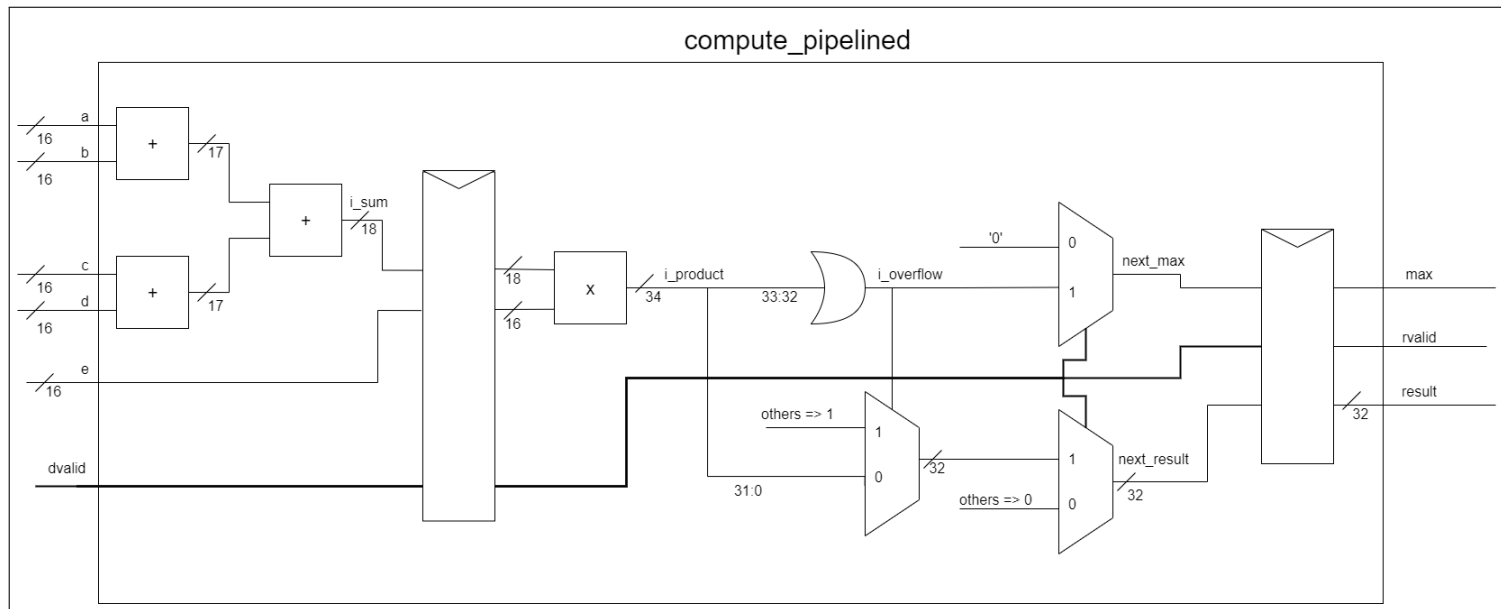
The resulting binary number is 34 bits long.

Task 5

a)



b)



Task 6