Converting an Unsigned Integer Product to a Signed Integer Product

Consider the problem of finding the product of two signed 2's complement <u>integers</u>. We'll do this using the same approach we used for fixed-point multiplication, i.e., compute the unsigned product and then modify it - <u>except that we need to keep the full double-length result rather than just the middle half</u>. We'll do it here using four-bit operands, but the principle is the same regardless of operand size.

First the math:

$$\begin{aligned} A_u B_u &= (2^3 A_3 + A_{2..0})(2^3 B_3 + B_{2..0}) = 2^6 A_3 B_3 + 2^3 (A_3 B_{2..0} + B_3 A_{2..0}) + A_{2..0} B_{2..0} \\ A_s B_s &= (-2^3 A_3 + A_{2..0})(-2^3 B_3 + B_{2..0}) = 2^6 A_3 B_3 - 2^3 (A_3 B_{2..0} + B_3 A_{2..0}) + A_{2..0} B_{2..0} \\ &= A_u B_u - 2^4 A_3 B_{2..0} - 2^4 B_3 A_{2..0} \end{aligned}$$

This equation tells us that for each negative operand, we should subtract the <u>three</u> least-significant bits of the other operand from the most-significant half of the unsigned product. When performing these subtractions, even though the math says to subtract only the three least-significant bits (i.e., $A_{2..0}$ or $B_{2..0}$ or both) from the unsigned product, <u>you can actually use the entire 4-bit operand (A_s or B_s)</u>:

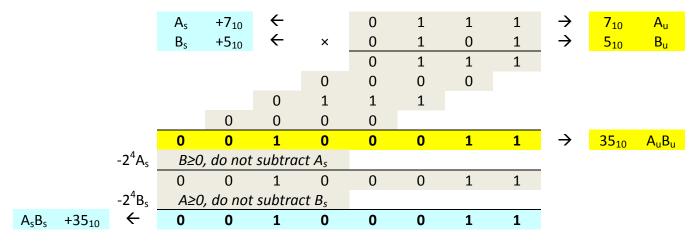
$$A_s B_s = A_u B_u - 2^4 A_3 B_s - 2^4 B_3 A_s$$

Case 1 - Both operands are positive: No subtractions are performed.

Case 2 - Operands of different signs: We subtract the positive operand from the most-significant half of the unsigned product. The fourth bit of the positive operand is zero, and thus has no effect on the result.

Case 3- Both operands are negative: The most-significant bits of both 4-bit operands are 1's. The first subtraction thus changes the sign of the result; the second subtraction changes it back – the same final result you would get if the subtraction used only the least-significant three bits (i.e., $A_{2..0}$ and $B_{2..0}$).

Example 1 - Positive × Positive:

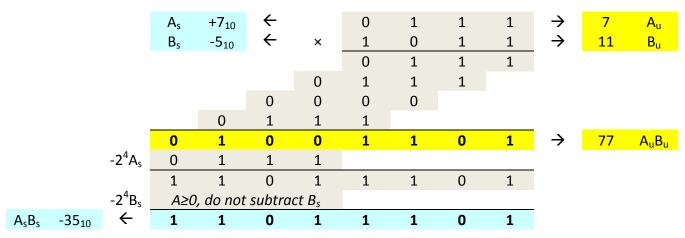


Converting an Unsigned Integer Product to a Signed Integer Product

Example 2a - Positive × Negative:

	A_s	-7 ₁₀	\leftarrow		1	0	0	1	\rightarrow	9	A _u
	B_s	+5 ₁₀	\leftarrow	×	0	1	0	1	\rightarrow	5	B _u
					1	0	0	1			
				0	0	0	0				
			1	0	0	1					
		0	0	0	0						
	0	0	1	0	1	1	0	1	\rightarrow	45	A_uB_u
-2^4A_s	<i>B</i> ≥0	B≥0, do not subtract A_s							_		
	0	0	1	0	1	1	0	1			
-2^4B_s	0	1	0	1					_		
A_sB_s -35 ₁₀ \leftarrow	1	1	0	1	1	1	0	1			

Example 2b - Negative × Positive:



Example 3 - Negative × Negative:

