Note Title 3/22/2010

Let's look at the multiply process again, but in a different order.

$$
\begin{array}{lccccccccccl}
 & & & & & & \text{Bit} & 3 & 2 & 1 & 0 & \\
\text{Multiplicand} \rightarrow & & & & & & & 1 & 1 & 0 & 1 & \\
\text{Multiplier} \rightarrow & & & & & & \times & 1 & 0 & 1 & 1 & \\
\hline
 & & & & & & & 1 & 1 & 0 & 1 & i=0 \\
 & & & & & + & 1 & 1 & 0 & 1 & & i=1 \\
\hline
 & & & & 1 & 0 & 0 & 1 & 1 & 1 & & \\
 & & & + & 0 & 0 & 0 & 0 & & & & i=2 \\
\hline
 & & & & 1 & 0 & 0 & 1 & 1 & 1 & & \\
 & & + & 1 & 1 & 0 & 1 & & & & & i=3 \\
\hline
 & & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & &
\end{array}
$$

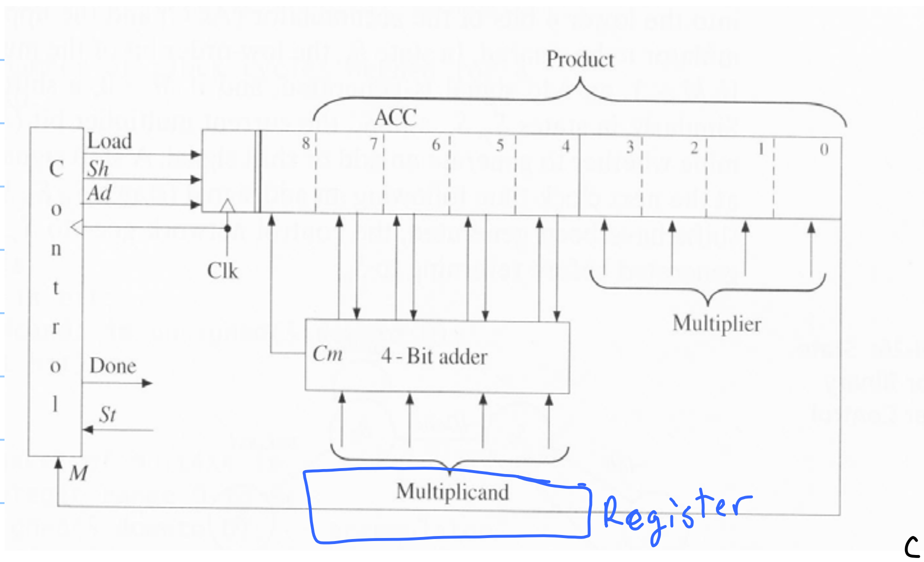
Partial Products

Note this time that the partial products can be viewed as either the multiplicand or Ø depending on whether Bit i of the multiplier is a 1 or a Ø respectively.

Note also that the i^{th} partial product is shifted left i bits.

So we could set this up by storing the multiplicand in a 4-bit register and then use a 4-bit adder to add the multiplicand to the current intermediate sum which is stored in a 9-bit shift register (the extra bit is to store the adder carry before we shift right).

(Figures are from Digital System Design Using VHDL, 2nd Edition
by Charles Roth)



We load the multiplier into the lower bits of the shift register (the accumulator) and use it to feed the control logic to determine if we need to perform an addition (along with a shift) or just a shift.

The shift signal moves all 9-bits to the right.

After we have done n-1 shifts (for n-bit numbers) we shift one more time to move all the bits of the answer into the low bits of ACC.

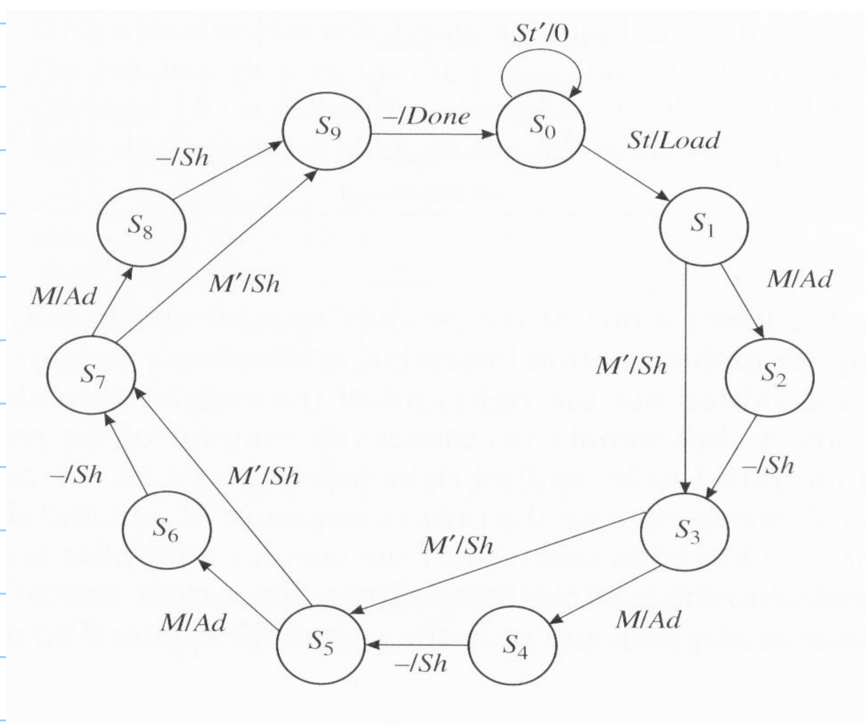
In the process this shifts the multiplier out of the register (ACC).

Let's look at the process and the contents of the register for our example of multiplying 13 x 11

Running total on left | Multiplier bits on right

	00000 1011
Add multiplicand as multiplier bit is 1	+1101
now shift this result →	01101 1011
After the shift	00110 1101
Add multiplicand →	+1101
then shift this →	10011 1101
After the shift	01001 1110
Multiplier bit is Ø so just shift →	00100 1111
Add multiplicand →	+1101
	10001 1111
One final shift	01000 1111

The controller for this register is:



Signed Multiplication

Now let's take a look at signed multiplication where we utilize both positive and negative numbers.

We will also take a look at fractions and not just integers.

Signed fractions are no different than signed integers. We still use 2's complement for these.

$$0.101 = \frac{1}{2} + \frac{1}{8} = \frac{5}{8}$$

$$\begin{array}{ccc} 2^{-1} & 2^{-2} & 2^{-3} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{8} \end{array}$$

What is the 2's complement of this?

$$\begin{array}{r} 1.010 \\ +\quad\ \ 1 \\ \hline 1.011 \end{array} = -5/8$$

The left most bit is still a sign bit to represent whether the number is positive or negative.

For signed multiplication we have four possibilities.

Multiplicand	Multiplier
+	+
−	+
+	−
−	−

Starting with two numbers that are both positive we see the multiplication process is straightforward.

$$\begin{array}{rl}
 & 0.111 \\
* & 0.101 \\
\hline
 & (0.00)0111 \\
+ & (0.)0111 \\
\hline
 & 0.100011
\end{array}$$

$+7/8$

$+5/8$

$1/8 * 7/8 = 7/64$

$1/2 * 7/8 = 7/16 = 28/64$

$35/64$

$\frac{1}{2}$ $\frac{1}{32}$ $\frac{1}{64}$

$32/64 + \frac{2}{64} + \frac{1}{64} = 35/64$

Now consider Multiplier (+) and Multiplicand (−)

$$\begin{array}{rl}
 & 1.101 \\
* & 0.101 \\
\hline
 & (1.11)1101 \\
+ & (1.)1101 \\
\hline
 & 1.110001
\end{array}$$

Sign Extend the negative number with leading 1's

$-3/8$

$+5/8$

$1/8 * (-\frac{3}{8}) = -3/64$

$1/2 * (-3/8) = -3/16 = -12/64$

$-15/64$

Take 2's complement of this negative number to double check its value.

$$\begin{array}{lrl}
 & & 1.110001 \\
\text{Invert} & & 0.001110 \\
\text{Add 1} & + & 1 \\
\hline
 & & 0.001111
\end{array}$$

$\frac{1}{8}$ $\frac{1}{16}$ $\frac{1}{32}$ $\frac{1}{64}$ $\Rightarrow \frac{8}{64} + \frac{4}{64} + \frac{2}{64} + \frac{1}{64} = 15/64$

Next, examine the case of

Multiplier (−) and Multiplicand (+)

Before we do we should look a little bit more closely at 2's complement numbers.

For pure fractional numbers (only the sign bit is to the left of the binary point) the bit pattern 1.g is the same as
$-1 + 0.g$

That is we can treat the fraction as positive (standard powers of two) and then add a value of −1

Consider the −3/8 from above ⇒ 1.101

We can verify this is −3/8 by taking its 2's complement and seeing the result is +3/8

$$
\begin{array}{lr}
 & 1.101 \\
\text{Invert} & 0.010 \\
\text{Add 1} & + \quad 1 \\
\hline
 & 0.011
\end{array}
= 0 + \frac{1}{4} + \frac{1}{8} = 3/8
$$

However we could also compute its value as:

$$
\begin{array}{c}
1.101 \\
-1 \quad +2^{-1} \quad +2^{-3}
\end{array}
$$

$$-1 + \frac{1}{2} + \frac{1}{8} = -\frac{8}{8} + \frac{4}{8} + \frac{1}{8} = \boxed{-3/8}$$

In general a 2's complement number can be written as

$$\left[(-1)^s * (\text{Bit MSB}) * 2^{MSB}\right] + \left[\sum_{i=LSB}^{MSB-1} (\text{Bit } i) * 2^i\right]$$

Example: 110.1011

Sign Bit is the MSB

This is a negative number because the sign bit is a 1. Take its 2's complement to determine its magnitude (or positive equivalent).

Invert 001.0100
Ad 1 $+ \quad\quad 1$

$001.0101 = 1 + \frac{1}{4} + \frac{1}{16} = 1\frac{5}{16} = 1.3125$

However we could also find its value as:

$$110.1011 = \left[(-1)^1 * 2^2\right] + (1*2^1) + (1*2^{-1}) + (1*2^{-3}) + (1*2^{-4})$$

$$= -4 + 2 + \frac{1}{2} + \frac{1}{8} + \frac{1}{16} = -4 + 2\frac{11}{16}$$

$$\boxed{= -1\frac{5}{16} = -1.3125}$$

So let's again multiply $-3/8$ and $+5/8$ but switching the operands

$$
\begin{array}{rrl}
 & 0.101 & +5/8 \\
* & 1.101 & -3/8 \\
\hline
 & 0.000101 & (+1/8) * (5/8) = 5/64 \\
+ & 0.0101 & (+1/2) * (5/8) = 5/16 = 20/64 \\
\hline
 & 0.011001 &
\end{array}
$$

Here the sign bit → $+1.011$ $\qquad (-1) * (5/8) = -5/8 = -\frac{40}{64}$

represents $(-1)^1 * 2^0 = -1$ $\qquad \boxed{1.110001} = -15/64$

So take 2's complement of multiplicand

$$-\frac{40}{64} + \frac{20}{64} + \frac{5}{64} = -15/64$$

Finally let us look at the case where both multiplier and multiplicand are negative.

$$
\begin{array}{rrl}
 & 1.101 & -3/8 \\
* & 1.101 & -3/8 \\
\hline
 & 1.111101 & (+\frac{1}{8}) * (-3/8) = -3/64 \\
+ & 1.1101 & (+1/2) * (-3/8) = -3/16 \\
\hline
 & 1.110001 &
\end{array}
$$

2's complement → $+0.011$ $\qquad (-1) * (-3/8) = +3/8$

because this is sign bit

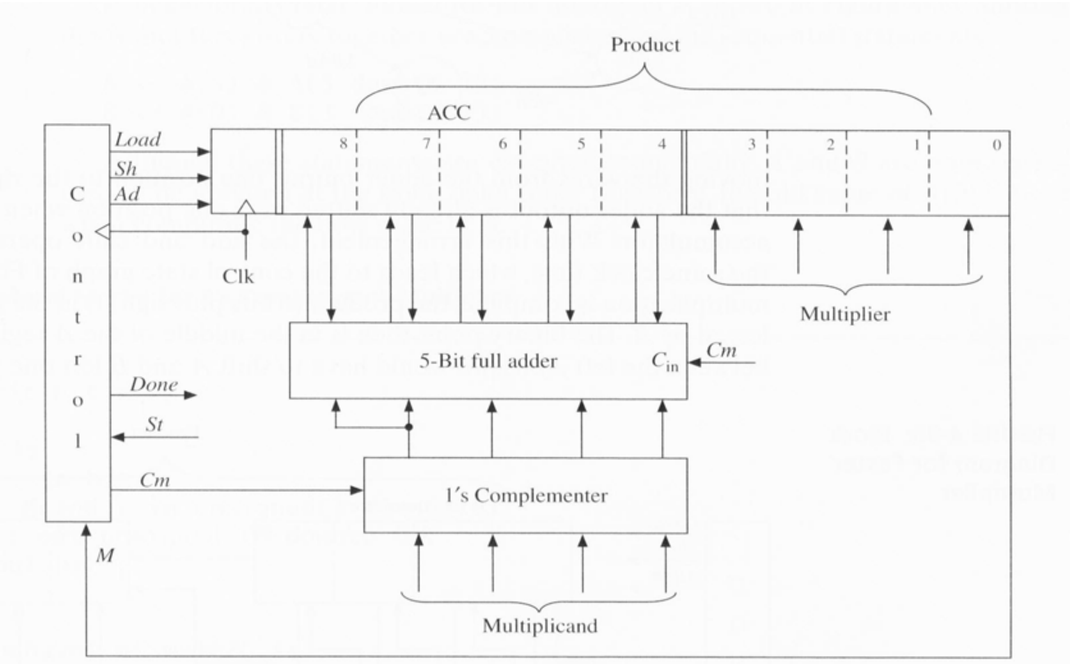
$$0.001001$$

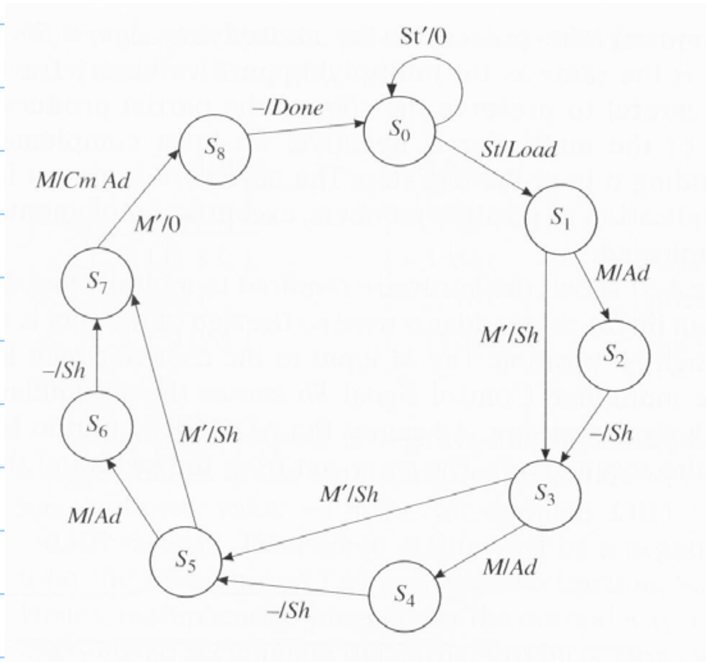
$$\frac{1}{8} + \frac{1}{64} = \frac{9}{64}$$

$$\frac{3}{8} - \frac{3}{16} - \frac{3}{64} =$$

$$\frac{24}{64} - \frac{12}{64} - \frac{3}{64} = \frac{9}{64}$$

To summarize: All multiplication steps proceed exactly as they do for unsigned multiplication. The only difference is for the sign bit of the multiplier (the last bit). If the sign bit is a 1 we add the 2's complement of the multiplicand, otherwise add ∅.





Note that in these previous circuits we are really wasting time and circuitry to load the result of the adder into register during one clock cycle and then use another clock cycle to shift it over.

It would be wiser to just hardwire the shift of the adder's result. This would eliminate the intermediate states needed for shifting.