Binary Multiplication
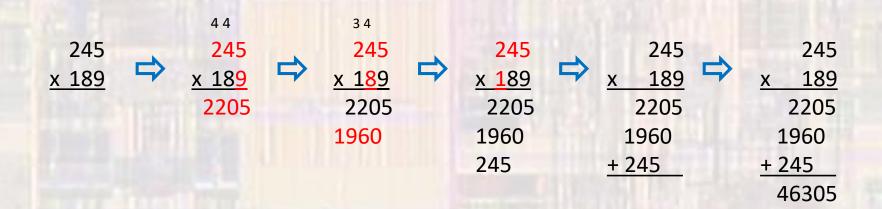
Common – last updated 11/7/18

Multiplication

- Elementary school concepts

  - Multiply digits, shift, repeat … add

```
                    4 4              3 4
      245           245              245           245          245            245
    x 189         x 189            x 189         x 189        x   189        x   189
                   2205            2205          2205          2205           2205
                                  1960           1960          1960           1960
                                                 245          + 245         + 245
                                                                            46305
```

Multiplication

- Sizing
  - In elementary school we usually don't worry about how many digits the result requires
    - If we think about it – the maximum number of digits is the sum of the number of digits of the multiplicand and the multiplier

1x1→2	3x1→4	3x5→8

```
        9              999                999
      x 9            x    9          x     99999
       81             8991              99899001
```

Binary Multiplication

- Sizing
  - In binary addition – we are generally representing something that ultimately is to be executed in hardware
    - Our hardware cannot change the number of bits (wires) it can hold
    - We must establish a maximum number size
    - For multiplication the size of the result must be the sum of the sizes of the multiplier and multiplicand

Unsigned

$$
\begin{array}{r}
\text{3 bit multiplication} \quad 111 \\
x \quad 111 \\
\hline
111 \\
111 \\
111 \\
\hline
110001
\end{array}
\qquad
\begin{array}{r}
7 \\
x \quad 7 \\
\hline
49
\end{array}
$$

Binary Multiplication - unsigned

- ## Unsigned multiplication
  - ### No additional concerns

Unsigned

```
3 bit multiplication     111              7
                  x    111            x  7
                       111              49
                      111
                     111
                    110001
```

Binary Multiplication - signed

- ## Signed multiplication
  - Requires special consideration for negative (2's complement) numbers

<div align="center">2's complement</div>

	3 bits	4 bits	8 bits	16 buts
-5	not possible	1011	1111 1011	1111 1111 1111 1011

  - The additional values out to the MSB position is called sign extension
    - This is true for both positive and negative numbers
    - We just don't usually write out the 0's

<div align="center">2's complement</div>

	3 bits	4 bits	8 bits	16 buts
5	not possible	0101	0000 0101	0000 0000 0000 0101
-5	not possible	1011	1111 1011	1111 1111 1111 1011

Binary Multiplication - signed

- ## Signed multiplication
  - ## Requires special consideration for negative (2's complement) numbers

<p align="center">signed</p>

	3 bit multiplication		
	101	-3
	x 011	x 3
	101	- 9
	101	⇕ ERROR
	000	
	001111	15

Binary Multiplication - signed

- Signed Multiplication

  - In 2's complement you must sign extend to the product bit width

4b x 4b

```
          0  1  1  1              7
      x   0  1  1  0          x   6
```

```
      0  0  0  0  0  1  1  1              7
  x   0  0  0  0  0  1  1  0          x   6
```

Must carry out all the bits

```
      0  0  0  0  0  0  0  0
      0  0  0  0  0  1  1  1
      0  0  0  0  1  1  1
  +   0  0  0  0  0  0  0
  x  x  x  0  0  1  0  1  0  1  0          42
```

Stop after 8 bits

Binary Multiplication - signed

- Signed Multiplication

  - In 2's complement you must sign extend to the product bit width

4b x 4b

				1	0	0	1		-7
			x	1	0	1	0	x	-6

	1	1	1	1	1	0	0	1		-7
x	1	1	1	1	1	0	1	0	x	-6

		0	0	0	0	0	0	0	0
	1	1	1	1	**1**	**0**	**0**	**1**
0	0	0	0	0	0	0	0	
1	1	1	1	**1**	**0**	**0**	**1**	
1	1	1	1	**1**	**0**	**0**	**1**	

Must carry out all the bits

	1	1	1	1	**1**	**0**	**0**	**1**
	1	1	1	1	**1**	**0**	**0**	**1**
	1	1	1	1	**1**	**0**	**0**	**1**
+	1	1	1	1	**1**	**0**	**0**	**1**

x	0	0	1	0	1	0	1	0		42

Stop after 8 bits

Binary Multiplication - signed

- Signed Multiplication
  - In 2's complement you must sign extend to the product bit width
  - When doing it by hand - where possible multiply by the positive value

4b x 4b

```
        0  1  1  1              7
    x   1  0  1  0      x      -6


        0  0  0  0  0  1  1  1              7
    x   1  1  1  1  1  0  1  0      x      -6
```

Must carry out all the bits

```
        0  0  0  0  0  0  0  0
     0  0  0  0  0  1  1  1
  0  0  0  0  0  0  0  0
     0  0  0  0  0  1  1  1
  0  0  0  0  0  1  1  1
  0  0  0  0  0  1  1  1
0  0  0  0  0  1  1  1
+ 0  0  0  1  0  1  1  1
  x  1  1  0  1  0  1  1  0
```

Stop after 8 bits   **-42**

4b x 4b

```
        1  0  1  0             -6
    x   0  1  1  1      x       7


  1  1  1  1  1  0  1  0             -6
x 0  0  0  0  0  1  1  1      x       7
  1  1  1  1  1  0  1  0
  1  1  1  1  1  0  1  0
  1  1  1  1  1  0  1  0
+ 0  0  0  0  0  0  0  0
```

Must carry out all the bits

Others are 0   x x x | 1 1 0 1 0 1 1 0   **-42**

Stop after 8 bits

Common   10   © tj