Fixed Point Binary

In denary we represent number, or parts of numbers, that are less than one like this:

100	10	1	.	1/10	1/100	1/1000
3	6	9	.	7	5	2

In binary, we use the same tactic:

8	4	2	1	.	½	¼	$^1/_8$	$^1/_{16}$
1	0	1	0	.	1	1	0	0
			10	.	$75_{(10)}$			

Converting fixed point binary to denary

Convert the integer part as normal, then add the fractions together:

$0.1_{(2)}$	=	½	=	$0.5_{(10)}$
$0.01_{(2)}$	=	¼	=	$0.25_{(10)}$
$0.001_{(2)}$	=	$^1/_8$	=	$0.125_{(10)}$
$0.0001_{(2)}$	=	$^1/_{16}$	=	$0.0625_{(10)}$

Converting denary to fixed point binary

Convert the integer part as normal, then remove the fractions:

E.G. 11.6875 [= 1010.1011]

Integer part = $1010_{(2)}$	
Remove 0.5 from the fractional part = $1010.1_{(2)}$	Remaining: $0.1875_{(10)}$
Can't remove 0.25 from the fractional part = $1010.10_{(2)}$	Remaining: $0.1875_{(10)}$
Remove 0.125 from the fractional part = $1010.101_{(2)}$	Remaining: $0.0625_{(10)}$
Remove 0.0625 from the fractional part = $1010.1011_{(2)}$	Remaining: $0_{(10)}$

Binary Multiplication

Binary multiplication uses the following rules (obvious when you think about it):

0 x 0 = 0
0 x 1 = 0
1 x 0 = 0
1 x 1 = 1

E.G. 1 – 1100 x 0010 [= 11000]

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

Multiply by the right hand number:        0 0 0 0
Multiply by the next number:            1 1 0 0
Multiply by the next number:          0 0 0 0
Multiply by the next number:        0 0 0 0
                                   ____________
                                        1 1 0 0 0
```

We can effectively ignore any 0s in the bottom number.

E.G 2 – 0010 1010 x 0001 0010 [= 1 1010 0100]

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

    0 0 1 0 1 0 1 0
0 0 1 0 1 0 1 0        +
___________________
  1 1 0 1 0 0 1 0 0
```