```
 11
x 23
 253
```

The result is 000011111101
which is 253 in decimal

```
  001011
x 010111
 000011111101
```

	M	A	Q	β	Count	Math/ALU
Initialization	001011	000000	010111	0	6	
Subtraction	001011	110101	010111	0	6	
Shift	001011	111010	101011	1	5	
Shift	001011	111101	010101	1	4	
Shift	001011	111110	101010	1	3	
Addition	001011	001001	101010	1	3	
Shift	001011	000100	110101	0	2	
Subtraction	001011	111001	110101	0	2	
Shift	001011	111100	111010	1	1	
Addition	001011	000111	111010	1	1	
Shift	001011	000011	111101	0	0	

Pseudo Code | Info

Booth's Multiplication Algorithm

```
01 boothMultiply(multiplicand,
multiplier){
02   Register M=multiplicand
03   Register A=0
04   Register Q=multiplier
05   Bit β=0
06   Integer count=REGISTER_SIZE
07
08   while (count > 0) {
09
10     switch ([leastSignificantBit
(Q),β]) {
11       case [1,0]: A=A-M
12         break
13
14       case[0,1]: A=A+M
15         break
16     }
17
18     //Shift A, Q, and β 1 bit
19     signPreservingRightShift(1,
A, Q, β)
20
21     count--
22   }
23 }
```

34
35