

Booth's Multiplication Algorithm

```
01 boothMultiply(multiplicand, multiplier){
02   Register M=multiplicand
03   Register A=0
04   Register Q=multiplier
05   Bit  $\beta$ =0
06   Integer count=REGISTER_SIZE
07
08   while (count > 0) {
09
10     switch ([leastSignificantBit(Q), $\beta$ ]) {
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  $\beta$  1 bit
19     signPreservingRightShift(1, A, Q,  $\beta$ )
20
21     count--
22   }
23 }
```

Decrement Count

11
x 23

001011
x 010111

	M	A	Q	β	Count
Initialization	0 0 1 0 1 1	0 0 0 0 0 0	0 1 0 1 1 1	0	6
Subtraction	0 0 1 0 1 1	1 1 0 1 0 1	0 1 0 1 1 1	0	6
Shift	0 0 1 0 1 1	1 1 1 0 1 0	1 0 1 0 1 1	1	5

Math/ALU