1a

```
shl R2 R1 3 ; R2 = 8R1
sub R2 R2 R1 ; R2 = R2-R1 = 8R1-R1 = 7R1
```

~~3cycles~~ Mult takes 4, so this implementation will be faster ~~if cycles cost~~ (4/3 times faster)

~~1b~~

~~shl R2 R1 2 ; R2 = 4R1~~

~~shl R1 R1 3 ; R1 = 8R1~~

~~add R2 R2 R1 ; R2 = R2+R1~~

~~sll R1 R1 1~~

1b

```
shl R2 R1 3 ; ~~R2 = n×8b~~ R2 = 8n
shl R1 R1 2 ; R1 = 4n
add R2 R2 R1 ; R2 = 12n
shl R1 R2 2 ; R1 = 48n
add R2 R2 R1 ; R2 = 60n
shl R1 R2 4 ; R1 = 960n
add R2 R2 R1 ; R2 = 1020n
```

~~1,4~~ 10 cycles. Mult takes 4cycles, so this implementation is 2.5 times slower

1c

Booth's algorithm would go through C and examine pairs of adjacent bits.

$1020 = \underbrace{01}_{\text{add } 10000000000_2 \times n}1111111\underbrace{100}_{\text{subtruct } 100_2 \times n}{}_2$

```
shl R2 R1 10 ; R2 = 1024n
shl R1 R1 2 ; R1 = 4n
sub R2 R2 R1 ; R2 = 1020n
```

This implementation takes 4 cycles, which is exactly as fast as simply using mult instruction.