Computer Organization & Architecture

In this Lecture…

- How do computers represent negative numbers?
- What about arithmetic operation on negative numbers
  - Add/Subtract
  - Multiply
  - Divide

Multiplying Negative Numbers

- This does not work!
- Solution 1
  - Convert to positive if required
  - Multiply as above
  - If signs were different, negate answer
- Solution 2
  - Booth's algorithm

Booth’s Algorithm

5 important term

1. Product Register
2. Multiplicand
3. Current Bit (least most)
4. Previous Bit(least most)
5. Operations

Bits	Description
00	Do Nothing, Just Shift
01	Add Multiplicand to Upper left half of Product Register and then Shift
10	Subtract Multiplicand to Upper left half of Product Register and then Shift
11	Do Nothing, Just Shift

Booth’s Algorithm Example

Right Shift Circulant RSC → Moving of Binary Numbers in a circle

ex: 11011 ———— > RSC ———> 11011

ex: 10100 ———— > RSC ———> 01010

Right Shift Arithmetic RSA → Make a copy of first bit and then copy next bits

ex: 11011 ———— > RSA ———> 111011

ex: 00100 ———— > RSA ———> 000100

Example

Multiply 14 times -5 using 5-bit numbers (10-bit result).

Multiplicand: 14 in binary: 01110

Multiplier: -5 in binary: 11011

For Simplicity: -14 in binary: 10010 (so we can add when we need to subtract the multiplicand)

Expected result: -70 in binary: 1110111010

Step	Multiplicand	Product Register	Current Bit	Previous Bit	Action
0	01110	00000 11011		0	Initialize

Step	Multiplicand	Product Register	Current Bit	Previous Bit	Action
0	01110	00000 11011		0	Initialize
1	01110	00000 11011	1	0	Subtract Multiplicand 00000 + 10010 = 10010

Step	Multiplicand	Product Register	Current Bit	Previous Bit	Action
0	01110	00000 11011		0	Initialize
1	01110	00000 11011	1	0	Subtract Multiplicand 00000 + 10010 =10010 10010 11011

Step	Multiplicand	Product Register	Current Bit	Previous Bit	Action
0	01110	00000 11011		0	Initialize
1	01110	00000 11011	1	0	Subtract Multiplicand 00000 + 10010 =10010 10010 11011 Right Shift Arithmetic RSA

Step	Multiplicand	Product Register	Current Bit	Previous Bit	Action
0	01110	00000 11011		0	Initialize
1	01110	00000 11011	1	0	Subtract Multiplicand 00000 + 10010 =10010 10010 11011 Right Shift Arithmetic RSA
1a		11001 01101		1	

Step	Multiplicand	Product Register	Current Bit	Previous Bit	Action
0	01110	00000 11011		0	Initialize
1	01110	00000 11011	1	0	Subtract Multiplicand 00000 + 10010 =10010 10010 11011 Right Shift Arithmetic RSA
1a		11001 01101		1	
2	01110	11001 01101	1	1	No-Operation just RSA

Step	Multiplicand	Product Register	Current Bit	Previous Bit	Action
0	01110	00000 11011		0	Initialize
1	01110	00000 11011	1	0	Subtract Multiplicand 00000 + 10010 =10010 10010 11011 Right Shift Arithmetic RSA
1a		11001 01101		1	
2	01110	11001 01101	1	1	No-Operation just RSA
3	01110	11100 10110	0	1	Add Multiplicand 11100 + 01110 = 01010 [Carry Ignored]

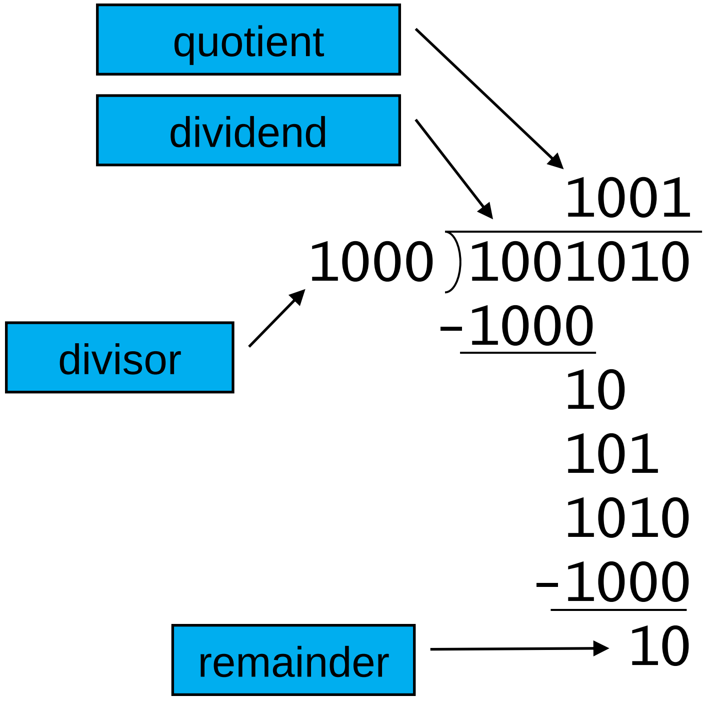
Step	Multiplicand	Product Register	Current Bit	Previous Bit	Action
0	01110	00000 11011		0	Initialize
1	01110	00000 11011	1	0	Subtract Multiplicand 00000 + 10010 =10010 10010 11011 Right Shift Arithmetic RSA
1a		11001 01101		1	
2	01110	11001 01101	1	1	No-Operation just RSA
3	01110	11100 10110	0	1	Add Multiplicand 11100 + 01110 = 01010 [Carry Ignored] 01010 10110 RSA
3a		00101 01011		0	

Step	Multiplicand	Product Register	Current Bit	Previous Bit	Action
0	01110	00000 11011		0	Initialize
1	01110	00000 11011	1	0	Subtract Multiplicand 00000 + 10010 =10010 10010 11011 Right Shift Arithmetic RSA
1a		11001 01101		1	
2	01110	11001 01101	1	1	No-Operation just RSA
3	01110	11100 10110	0	1	Add Multiplicand 11100 + 01110 = 01010 [Carry Ignored] 01010 10110 RSA
3a		00101 01011		0	
4	01110	00101 01011	1	0	Subtract Multiplicand

Step	Multiplicand	Product Register	Current Bit	Previous Bit	Action
0	01110	00000 11011		0	Initialize
1	01110	00000 11011	1	0	Subtract Multiplicand 00000 + 10010 =10010 10010 11011 Right Shift Arithmetic RSA
1a		11001 01101		1	
2	01110	11001 01101	1	1	No-Operation just RSA
3	01110	11100 10110	0	1	Add Multiplicand 11100 + 01110 = 01010 [Carry Ignored] 01010 10110 RSA
3a		00101 01011		0	
4	01110	00101 01011	1	0	Subtract Multiplicand 00101 + 10010 =10111 10111 01011 RSA
4a	01110	11011 10101		1	

Step	Multiplicand	Product Register	Current Bit	Previous Bit	Action
0	01110	00000 11011		0	Initialize
1	01110	00000 11011	1	0	Subtract Multiplicand 00000 + 10010 =10010 10010 11011 Right Shift Arithmetic RSA
1a		11001 01101		1	
2	01110	11001 01101	1	1	No-Operation just RSA
3	01110	11100 10110	0	1	Add Multiplicand 11100 + 01110 = 01010 [Carry Ignored] 01010 10110 RSA
3a		00101 01011		0	
4	01110	00101 01011	1	0	Subtract Multiplicand 00101 + 10010 =10111 10111 01011 RSA
4a	01110	11011 10101		1	
5	01110	11011 10101	1	1	No- Operation just RSA
		11101 11010			

Signed Division



n-bit operands yield *n*-bit quotient and remainder

- Check for 0 divisor
- Long division approach
  - If divisor ≤ dividend bits
    - 1 bit in quotient, subtract
  - Otherwise
    - 0 bit in quotient, bring down next dividend bit
- Restoring division
  - Do the subtract, and if remainder goes < 0, add divisor back
- Signed division
  - Divide using absolute values
  - Adjust sign of quotient and remainder as required