

General Division Algorithm

Algorithm is described for N-bit unsigned integers. It assumes a double-length (2N-bit) dividend, an N-bit divisor and obtains N-bit quotient and remainder.

REM: Accumulator Register **QUO**: Quotient Register **DVR**: Divisor Register
 The algorithm treats **REM** as a high-end extension of **QUO**

```

DVR ← divisor
REM:QUO ← dividend    // REM is cleared if the dividend is only N-bits
for (int k = 1; k <= N; k++) {
  ShiftLeft( REM:QUO ) //Shift dividend 1 bit to the left
  if ( DVR <= REM ) { //Compare
    REM := REM - DVR    //Subtract
    SET low bit of QUO    //Set bit
  }
}
  
```

Quotient is in **QUO**. Remainder is in **REM**

Example Trace with BYTE (8-bit) integers

522 (0000 0010 0000 1010) / 17 (0001 0001) = 30 (0001 1110) Quotient
 12 (0000 1100) Remainder

<u>Iteration</u>	<u>REM</u>	<u>QUO</u>	<u>DVR</u>
0: Load	0000 0010	0000 1010	0001 0001
1: Shift	0000 0100	0001 0100	
2: Shift	0000 1000	0010 1000	
3: Shift	0001 0000	0101 0000	
4: Shift	0010 0000	1010 0000	
Subtract	0000 1111	1010 000 1 <-SET low bit	
5: Shift	0001 1111	0100 0010	
Subtract	0000 1110	0100 001 1 <-SET low bit	
6: Shift	0001 1100	1000 0110	
Subtract	0000 1011	1000 011 1 <-SET low bit	
7: Shift	0001 0111	0000 1110	
Subtract	0000 0110	0000 111 1 <-SET low bit	
8: Shift	0000 1100	0001 1110	