Binary division - Kenyan (?) algorithm

Agustín T. Ferrari Nicolay - Buenos Aires - April 2007

1 - Given a DIVIDEND and a DIVISOR, initialize QUOTIENT =0 and PNTR_DSOR =0
2 - Duplicate DIVISOR repeatedly. Increase PNTR_DSOR every time you double DIVISOR.
3 - Repeat the doubling until getting a value **equal to**, or the **closest below** DIVIDEND.
4 - Substract the highest double value from the DIVIDEND and set b0 of QUOTIENT. Shift QUOTIENT to the left. Decrement PNTR_DSOR
Succesively try to substract every doubled DIVISOR value down in the list, from the remainder above.
5 - For every possible substraction, keep setting b0 of QUOTIENT and shifting it to the left. If not possible, just shift QUOTIENT to the left. In any case, decrement PNTR_DSOR.
6 - Once finished (PNTR_DSOR again =0), the remainder is the result of the last substraction. Quotient in the corresponding register.

1st example - Decimal values.

DIVIDEND	79807	PNTR_DSOR
DIVISOR	34	0
double above value	68	1
double above value	136	2
double above value	272	3
double above value	544	4
double above value	1088	5
double above value	2176	6
double above value	4352	7
double above value	8704	8
double above value	17408	9
double above value	34816	10
double above value	69632	11

Found the closest value **below** DIVIDEND

QUOTIENT = 2347

b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	1	1

If possible, substract:	Remainder	Substraction is:	PNTR_DSOR	
79807 - 69632 =	10175	**possible**	11	set b11
10175 - 34816 =	10175	not possible	10	
10175 - 17408 =	10175	not possible	9	
10175 - 8704 =	1471	**possible**	8	set b8
1471 - 4352 =	1471	not possible	7	
1471 - 2176 =	1471	not possible	6	
1471 - 1088 =	383	**possible**	5	set b5
383 - 544 =	383	not possible	4	
383 - 272 =	111	**possible**	3	set b3
111 - 136 =	111	not possible	2	
111 - 68 =	43	**possible**	1	set b1
43 - 34 =	9	**possible**	0	set b0

2nd example - Decimal values.

DIVIDEND	1349827	PNTR_DSOR
DIVISOR	793	0
double above value	1586	1
double above value	3172	2
double above value	6344	3
double above value	12688	4
double above value	25376	5
double above value	50752	6
double above value	101504	7
double above value	203008	8
double above value	406016	9
double above value	812032	10

Found the closest value **below** DIVIDEND

QUOTIENT = 1702

b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	1	1	0	1	0	1	0	0	1	1	0

If possible, substract:	Remainder	Substraction is:	PNTR_DSOR	
1349827 - 812032 =	537795	**possible**	10	set b10
537795 - 406016 =	131779	**possible**	9	set b9
131779 - 203008 =	131779	not possible	8	
131779 - 101504 =	30275	**possible**	7	set b7
30275 - 50752 =	30275	not possible	6	
30275 - 25376 =	4899	**possible**	5	set b5
4898 - 12688 =	4899	not possible	4	
4899 - 6344 =	4899	not possible	3	
4899 - 3172 =	1727	**possible**	2	set b2
1727 - 1586 =	141	**possible**	1	set b1
141 - 793 =	141	not possible	0	

```
      ;DIV 216U KENYAN.ASM

DIV_3216U_KEN               ;unsigned 32/16-bit values division (KENYAN)

;Agustin T. Ferrari Nicolay - Buenos Aires - April 2007

;Algorithm implemented:

;Given DIVIDEND and DIVISOR, initialize QUOTIENT =0 and INDEX_DSOR =0

;Duplicate DIVISOR repeatedly. Increase PNTR_DSOR every time DIVISOR is doubled

;Repeat doubling to get a DSOR equal to, or the closest below DIVIDEND.

;Substract the highest DSOR from DIVIDEND setting b0 of QUOTIENT.

;Shift QUOTIENT to left and decrement PNTR_DSOR.

;Succesively try to substract from the resulting remainder above, every DIVISOR

;value, down in the list.

;For every possible substraction, keep setting b0 of QUOTIENT and shifting it

;to left. If not possible, just shift QUOTIENT to the left.

;In any case, always decrement PNTR_DSOR.

;Once finished (PNTR_DSOR again =0), the remainder is the result of the last

;substraction. QUOTIENT in the corresponding register.


;To call the routine:
;dividend in DEND_3:0 (max val H'FFFE 0001' = H'FFFF' * H'FFFF' =4.294.836.225)
;divisor in DSOR_1:0 (max value H'FFFF' =65.535) - DSOR_3:2 used internally in
;the routine

;User to ensure being within range or if division by zero is attempted.

;The routine gives:
;Result of DEND_3:DEND_0 / DSOR_1:DSOR_0 => QUOT_H:QUOT_L.
;Remainder in DEND_3:0

    CLRF DSOR_3
    CLRF DSOR_2
    CLRF QUOT_H
    CLRF QUOT_L
    CLRF PNTR_DSOR

DIV_3216U_KEN_INC_DSOR_LOOP
    BCF STATUS,C            ;ensure b0 of DSOR_0 is clear after the shifting
    RLCF DSOR_0,F           ;DSOR =DSOR*2
    RLCF DSOR_1,F
    RLCF DSOR_2,F
    RLCF DSOR_3,F
    INCF PNTR_DSOR,F

    MOVF DSOR_3,W
    SUBWF DEND_3,W
    BNZ CHKIF_DSOR3_GT_DEND3

    MOVF DSOR_2,W
    SUBWF DEND_2,W
    BNZ CHKIF_DSOR2_GT_DEND2

    MOVF DSOR_1,W
    SUBWF DEND_1,W
    BNZ CHKIF_DSOR1_GT_DEND1

    MOVF DSOR_0,W
    SUBWF DEND_0,W
    BC DIV_3216U_KEN_INC_DSOR_LOOP

DIV_3216U_KEN_SUBST_DSOR_LOOP
    TSTFSZ PNTR_DSOR
    BRA DIV_3216U_KEN_DECR_PNTR
    RETURN

DIV_3216U_KEN_DECR_PNTR
    DECF PNTR_DSOR          ;we look down in the list of double DSOR values

    BCF STATUS,C            ;ensure b0 of QUOT_L is clear after shifting
    RLCF QUOT_L,F           ;shift QUOT to the left to have it ready
    RLCF QUOT_H,F           ;for next substraction

    BCF STATUS,C            ;ensure b7 of DSOR_3 is clear after shifting
    RRCF DSOR_3,F           ;shift DSOR
    RRCF DSOR_2,F           ;to the right
    RRCF DSOR_1,F           ;to get
    RRCF DSOR_0,F           ;DSOR =DSOR/2

    MOVF DSOR_3,W
    SUBWF DEND_3,W
    BNZ CHKIF_DSOR3_LT_DEND3

    MOVF DSOR_2,W
    SUBWF DEND_2,W
    BNZ CHKIF_DSOR2_LT_DEND2

    MOVF DSOR_1,W
    SUBWF DEND_1,W
    BNZ CHKIF_DSOR3_LT_DEND3

    MOVF DSOR_0,W
    SUBWF DEND_0,W
    BNC DIV_3216U_KEN_SUBST_DSOR_LOOP

DIV_3216U_KEN_SUBST_DSOR    ;substract DSOR_3:0 from DEND_3:0
    MOVF DSOR_0,W           ;LSB, borrow
    SUBWF DEND_0,F          ;is NOT used
    MOVF DSOR_1,W           ;borrow
    SUBWFB DEND_1,F         ;IS used

    MOVF DSOR_2,W           ;borrow
    SUBWFB DEND_2,F         ;IS used

    MOVF DSOR_3,W           ;borrow
    SUBWFB DEND_3,F         ;IS used

    BSF QUOT_L,0            ;flag "a valid substraction from dividend occurred"
    BRA DIV_3216U_KEN_SUBST_DSOR_LOOP

CHKIF_DSOR3_GT_DEND3
    BNC DIV_3216U_KEN_SUBST_DSOR_LOOP
    BRA DIV_3216U_KEN_INC_DSOR_LOOP

CHKIF_DSOR2_GT_DEND2
    BNC DIV_3216U_KEN_SUBST_DSOR_LOOP
    BRA DIV_3216U_KEN_INC_DSOR_LOOP

CHKIF_DSOR1_GT_DEND1
    BNC DIV_3216U_KEN_SUBST_DSOR_LOOP
    BRA DIV_3216U_KEN_INC_DSOR_LOOP

CHKIF_DSOR3_LT_DEND3
    BC DIV_3216U_KEN_SUBST_DSOR
    BRA DIV_3216U_KEN_SUBST_DSOR_LOOP

CHKIF_DSOR2_LT_DEND2
    BC DIV_3216U_KEN_SUBST_DSOR
    BRA DIV_3216U_KEN_SUBST_DSOR_LOOP

CHKIF_DSOR1_LT_DEND1
    BC DIV_3216U_KEN_SUBST_DSOR
    BRA DIV_3216U_KEN_SUBST_DSOR_LOOP
```