RESTORING DIVISION

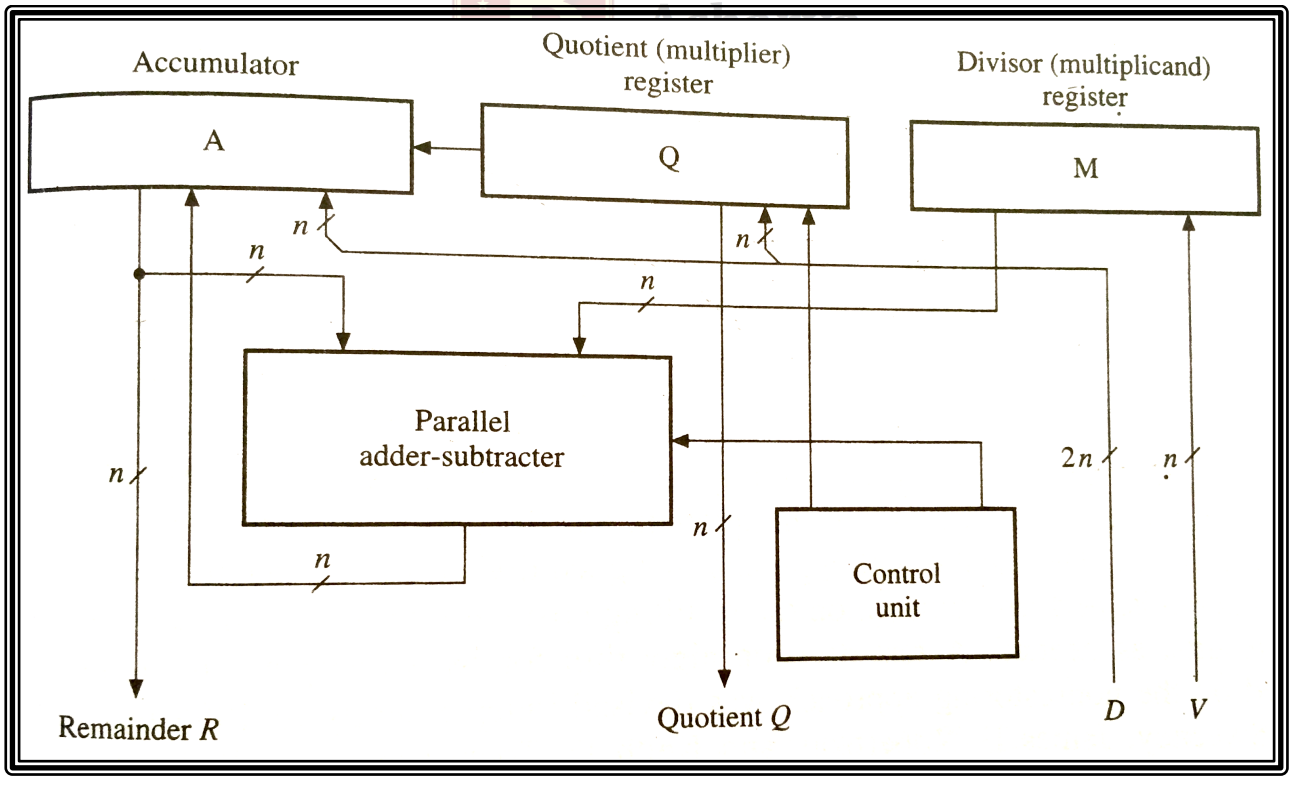
1) Let Q register hold the divided, M register holds the divisor and A register is 0.
2) On completion of the algorithm, Q will get the quotient and A will get the remainder.

Algorithm:

The **number of steps** required is equal to the **number of bits in the Dividend**.

1) At each step, **left shift the dividend by 1 position**.
2) **Subtract the divisor from A** (perform A - M).
3) If the **result is positive** then the step is said to be **"Successful"**.
   In this case **quotient bit will be "1"** and **Restoration is NOT Required**.
4) If the **result is negative** then the step is said to be **"Unsuccessful"**.
   In this case **quotient bit will be "0".**
   Here Restoration is performed by adding back the divisor.
   Hence the method is called Restoring Division.

Repeat steps 1 to 4 for **all bits** of the Dividend.



Example: (6) / (4)

Dividend (Q) = 6
Divisor (M) = 4
Accumulator (A) = 0

6 = 0110 4 = 0100
-6 = 1010 -4 = 1100

In case of doubts, Call #BharatSir @ 98204 08217	ACCUMULATOR A (0)	DIVIDEND Q (6)	DIVISOR M (4)
Initial Values	0000	0110	0100
Step 1:			
Left-Shift	0000	110_	
A – M	+1100		
Unsuccessful (-ve)	1100		
Restoration	0000	1100	
Step 2:			
Left-Shift	0001	100_	
A – M	+1100		
Unsuccessful (-ve)	1101		
Restoration	0001	1000	
Step 3:			
Left-Shift	0011	000_	
A – M	+1100		
Unsuccessful (-ve)	1111		
Restoration	0011	0000	
Step 4:			
Left-Shift	0110	000_	
A – M	+1100		
Successful (+ve)	0010		
No Restoration	0000	0001	
	Remainder(0)	**Quotient(1)**	

Example: (19) / (7)

Dividend (Q) = 19 Divisor (M) = 7 Accumulator (A) = 0

19 = 010011 7 = 000111
-19 = 101101 -7 = 111001

In case of doubts, Call #BharatSir @ 98204 08217	ACCUMULATOR A (0)	DIVIDEND Q (19)	DIVISOR M (7)
Initial Values	000000	010011	000111
Step 1:			
Left-Shift	000000	10011_	
A – M	+111001		
Unsuccessful (-ve)	111001		
Restoration	000000	100110	
Step 2:			
Left-Shift	000001	00110_	
A – M	+111001		
Unsuccessful (-ve)	111010		
Restoration	000001	001100	
Step 3:			
Left-Shift	000010	01100_	
A – M	+111001		
Unsuccessful (-ve)	111011		
Restoration	000010	011000	
Step 4:			
Left-Shift	000100	11000_	
A – M	+111001		
Unsuccessful (-ve)	111101		
Restoration	000100	110000	
Step 5:			
Left-Shift	001001	10000_	
A – M	+111001		
Successful (+ve)	000010		
No Restoration	000010	100001	
Step 6:			
Left-Shift	000101	00001_	
A – M	+111001		
Unsuccessful (-ve)	111110		
Restoration	000101	000010	
	Remainder(5)	Quotient(2)	

Example: (23) / (3)

Dividend (Q) = 23 Divisor (M) = 3 Accumulator (A) = 0

23 = 010111 3 = 000011
-23 = 101001 -3 = 111101

In case of doubts, Call #BharatSir @ 98204 08217	ACCUMULATOR A (0)	DIVIDEND Q (23)	DIVISOR M (3)
Initial Values	000000	010111	000011
Step 1:			
Step 2:			
Step 3:			
Step 4:			
Step 5:			
Step 6:			
	Remainder(2)	Quotient(7)	

NON-RESTORING DIVISION

1) Let Q register hold the divided, M register holds the divisor and A register is 0.
2) On completion of the algorithm, Q will get the quotient and A will get the remainder.

Algorithm:

The **number of steps** required is equal to the **number of bits in the Dividend**.

1) At each step, **left shift the dividend by 1 position**.
2) **Subtract the divisor from A** (perform A - M).
3) If the **result is positive** then the step is said to be **"Successful"**.
   In this case **quotient bit will be "1"** and **Restoration is NOT Required**.
   The **Next Step** will also be **Subtraction**.
4) If the **result is negative** then the step is said to be **"Unsuccessful"**.
   In this case **quotient bit will be "0".**
   Here Restoration is NOT Performed.
   Instead the next step will be ADDITION in place of subtraction.
   As restoration is not performed, the method is called Non-Restoring Division.

Repeat steps 1 to 4 for **all bits** of the Dividend.

Note: Restoration may only be performed in the FINAL step, if it is unsuccessful, as there is no next step thereafter. In such a case, restoration is performed by adding back the divisor.

Example: (7) / (5)

Dividend (Q) = 7
Divisor (M) = 5
Accumulator (A) = 0

7 = 0111 5 = 0101
-7 = 1001 -5 = 1011

In case of doubts, Call #BharatSir @ 98204 08217	ACCUMULATOR A (0)	DIVIDEND Q (7)	DIVISOR M (5)
Initial Values	0000	0111	0101
Step 1:			
Left-Shift	0000	111_	
A – M	+1011		
Unsuccessful (-ve)	1011	1110	
Next Step: "Add"			
Step 2:			
Left-Shift	0111	110_	
A + M	+0101		
Unsuccessful (-ve)	1100	1100	
Next Step: "Add"			
Step 3:			
Left-Shift	1001	100_	
A + M	+0101		
Unsuccessful (-ve)	1110	1000	
Next Step: "Add"			
Step 4:			
Left-Shift	1101	000_	
A + M	+0101		
Successful (+ve)	0010	0001	
	Remainder(2)	Quotient(1)	

Example: (7) / (3)

Dividend (Q) = 7
Divisor (M) = 3
Accumulator (A) = 0

7 = 0111 3 = 0011
-7 = 1001 -3 = 1101

In case of doubts, Call #BharatSir @ 98204 08217	ACCUMULATOR A (0)	DIVIDEND Q (7)	DIVISOR M (3)
Initial Values	0000	0111	0011
Step 1:			
Left-Shift	0000	111_	
A – M	+1101		
Unsuccessful (-ve)	1101	1110	
Next Step: "Add"			
Step 2:			
Left-Shift	1011	110_	
A + M	+0011		
Unsuccessful (-ve)	1110	1100	
Next Step: "Add"			
Step 3:			
Left-Shift	1101	100_	
A + M	+0011		
Successful (+ve)	0000	1001	
Next Step: "Sub"			
Step 4:			
Left-Shift	0001	001_	
A - M	+1101		
Unsuccessful (-ve)	1110		
Restore: (Add back divisor)	0001	0010	
	Remainder(1)	**Quotient(2)**	

Example: (19) / (4)

Dividend (Q) = 19 Divisor (M) = 4 Accumulator (A) = 0

19 = 010011 4 = 000100
-19 = 101101 -4 = 111100

In case of doubts, Call #BharatSir @ 98204 08217	ACCUMULATOR	DIVIDEND	DIVISOR
	A (0)	Q (19)	M (4)
Initial Values	000000	010011	000100
Step 1:			
Left-Shift	000000	10011_	
A – M	+111100		
Unsuccessful (-ve)	111100	100110	
Next Step: "Add"			
Step 2:			
Left-Shift	111001	00110_	
A + M	+000100		
Unsuccessful (-ve)	111101	001100	
Next Step: "Add"			
Step 3:			
Left-Shift	111010	01100_	
A + M	+000100		
Unsuccessful (-ve)	111110	011000	
Next Step: "Add"			
Step 4:			
Left-Shift	111100	11000_	
A + M	+000100		
Successful (+ve)	000000	110001	
Next Step: "Sub"			
Step 5:			
Left-Shift	000001	10001_	
A – M	+111100		
Unsuccessful (-ve)	111101	100010	
Next Step: "Add"			
Step 6:			
Left-Shift	111011	00010_	
A + M	+000100		
Unsuccessful (-ve)	111111		
Restore: (Add back divisor)	000011	000100	
	Remainder(3)	Quotient(4)	

Example: (23) / (3)

Dividend (Q) = 23 Divisor (M) = 3 Accumulator (A) = 0

23 = 010111 3 = 000011
-23 = 101001 -3 = 111101

In case of doubts, Call #BharatSir @ 98204 08217	ACCUMULATOR A (0)	DIVIDEND Q (23)	DIVISOR M (3)
Initial Values	000000	010111	000011
Step 1:			
Step 2:			
Step 3:			
Step 4:			
Step 5:			
Step 6:			
	Remainder(2)	Quotient(7)	

RESTORING DIVISION FOR SIGNED NUMBERS

1) Let M register hold the divisor, Q register hold the divided.
2) A register should be the signed extension of Q.
3) On completion of the algorithm, Q will get the quotient and A will get the remainder.

Algorithm:

The **number of steps** required is equal to the **number of bits in the Dividend**.

1) At each step, **left shift the dividend by 1 position**.
2) If Sign of A and M is the same then **Subtract the divisor from A** (perform A - M),
   Else **Add M to A**
3) After the operation,
   If **Sign of A remains the same** or the **dividend** (in A and Q) **becomes zero**,
   then the step is said to be **"Successful"**.
   In this case **quotient bit will be "1"** and **Restoration is NOT Required**.
4) If **Sign of A changes**, then the step is said to be **"Unsuccessful"**.
   In this case **quotient bit will be "0".**
   Here Restoration is Performed.
   Hence, the method is called Restoring Division.

Repeat steps 1 to 4 for **all bits** of the Dividend.

Note: The result of this algorithm is such that, the quotient will always be positive and the remainder will get the same sign as the dividend.

Example: (5) / (3)

5 = 0101 3 = 0011
-5 = 1011 -3 = 1101

In case of doubts, Call #BharatSir @ 98204 08217	ACCUMULATOR A (Sign Extension)	DIVIDEND Q (5)	DIVISOR M (3)
Initial Values	0000	0101	0011
Step 1:			
Left-Shift	0000	101_	
Sign (A, M) Same: A - M	+1101		
Sign Changes: Unsuccessful	1101		
Restore	0000	1010	
Step 2:			
Left-Shift	0001	010_	
Sign (A, M) Same: A - M	+1101		
Sign Changes: Unsuccessful	1110		
Restore	0001	0100	
Step 3:			
Left-Shift	0010	100_	
Sign (A, M) Same: A – M	+1101		
Sign Changes: Unsuccessful	1111		
Restore	0010	1000	
Step 4:			
Left-Shift	0101	000_	
Sign (A, M) Same: A - M	+1101		
Sign still Same: Successful	0010		
Restore not required	0010	0001	
	Remainder(2)	Quotient(1)	

Example: (-19) / (7)

19 = 010011 7 = 000111
-19 = 101101 -7 = 111001

In case of doubts, Call #BharatSir @ 98204 08217	ACCUMULATOR	DIVIDEND	DIVISOR
	A (Sign Extension)	Q (-19)	M (7)
Initial Values	111111	101101	000111
Step 1:			
Left-Shift	111111	01101_	
Sign (A, M) Different: A + M	+000111		
Sign Changes: Unsuccessful	000110		
Restore	111111	011010	
Step 2:			
Left-Shift	111110	11010_	
Sign (A, M) Different: A + M	+000111		
Sign Changes: Unsuccessful	000101		
Restore	111110	110100	
Step 3:			
Left-Shift	111101	10100_	
Sign (A, M) Different: A + M	+000111		
Sign Changes: Unsuccessful	000100		
Restore	111101	101000	
Step 4:			
Left-Shift	111011	01000_	
Sign (A, M) Different: A + M	+000111		
Sign Changes: Unsuccessful	000010		
Restore	111011	010000	
Step 5:			
Left-Shift	110110	10000_	
Sign (A, M) Different: A + M	+000111		
Sign still Same: Successful	111101		
Restore not required	111101	100001	
Step 6:			
Left-Shift	111011	00001_	
Sign (A, M) Different: A + M	+000111		
Sign Changes: Unsuccessful	000010		
Restore	111011	000010	
	Remainder(-5)	Quotient(2)	

Example: (-8) / (-4)

8 = 01000 4 = 00100
-8 = 11000 -4 = 11100

In case of doubts, Call #BharatSir @ 98204 08217	ACCUMULATOR	DIVIDEND	DIVISOR
	A (Sign Extension)	Q (-8)	M (-4)
Initial Values	11111	11000	11100
Step 1:			
Left-Shift	11111	1000_	
Sign (A, M) Same: A - M	+00100		
Sign Changes: Unsuccessful	00011		
Restore	11111	10000	
Step 2:			
Left-Shift	11111	0000_	
Sign (A, M) Same: A - M	+00100		
Sign Changes: Unsuccessful	00011		
Restore	11111	00000	
Step 3:			
Left-Shift	11110	0000_	
Sign (A, M) Same: A - M	+00100		
Sign Changes: Unsuccessful	00010		
Restore	11110	00000	
Step 4:			
Left-Shift	11100	0000_	
Sign (A, M) Same: A - M	+00100		
Dividend(A,Q)=0: Successful	00000		
Restore not required	00000	00001	
Step 5:			
Left-Shift	00000	0001_	
Sign (A, M) Different: A + M	+11100		
Sign Changes: Unsuccessful	11100		
Restore	00000	00010	
	Remainder(0)	**Quotient(2)**	

Example: (-14) / (2)

14 = 01110 2 = 00010
-14 = 10010 -2 = 11110

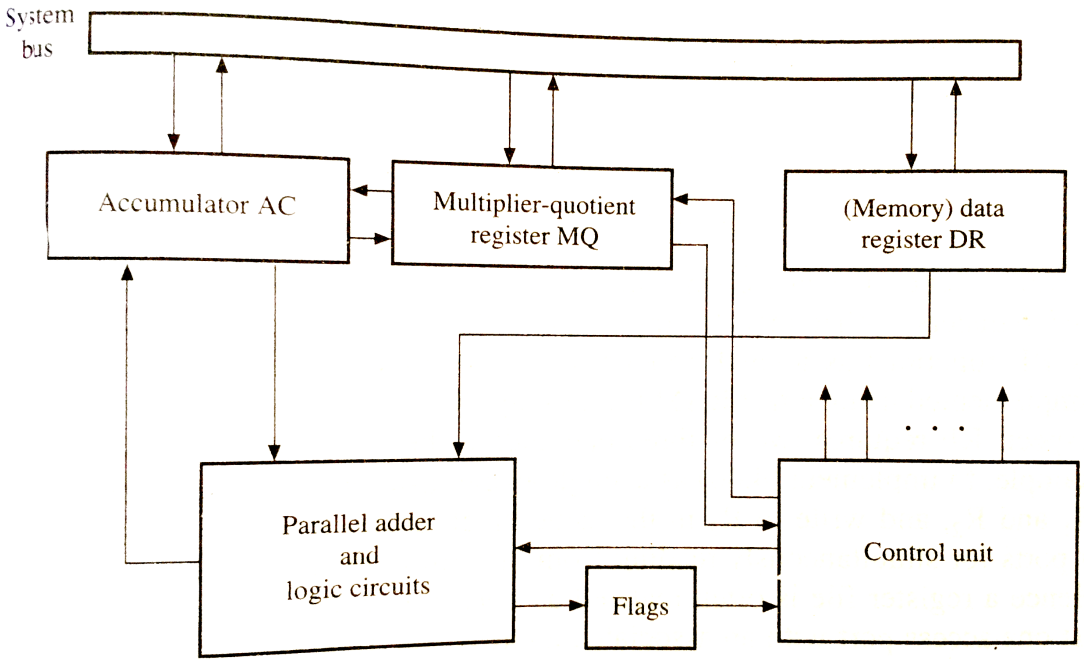
In case of doubts, Call #BharatSir @ 98204 08217	ACCUMULATOR	DIVIDEND	DIVISOR
	A (Sign Extension)	Q (-14)	M (2)
Initial Values	11111	10010	00010
Step 1:			
Left-Shift	11111	0010_	
Sign (A, M) Different: A + M	+00010		
Sign Changes: Unsuccessful	00001		
Restore	11111	00100	
Step 2:			
Left-Shift	11110	0100_	
Sign (A, M) Different: A + M	+00010		
Sign Changes: Unsuccessful	00000	*Note that dividend part in (A, Q) is not zero*	
Restore	11110	01000	
Step 3:	*In case of doubts, call #BharatSir @ 98204 08217*		
Left-Shift	11100	1000_	
Sign (A, M) Different: A + M	+00010		
Sign still Same: Successful	11110		
Restore not required	11110	10001	
Step 4:			
Left-Shift	11101	0001_	
Sign (A, M) Different: A + M	+00010		
Sign still Same: Successful	11111		
Restore not required	11111	00011	
Step 5:			
Left-Shift	11110	0011_	
Sign (A, M) Different: A + M	+00010		
Dividend(A,Q)=0: Successful	00000	*Note that dividend part in (A, Q) is Zero!*	
Restore not required	00000	00111	
	Remainder(0)	**Quotient(7)**	

Example: (14) / (-2)

14 = 01110 2 = 00010
-14 = 10010 -2 = 11110

In case of doubts, Call #BharatSir @ 98204 08217	ACCUMULATOR A (Sign Extension)	DIVIDEND Q (-14)	DIVISOR M (-2)
Initial Values	00000	01110	11110
Step 1:			
Step 2:			
Step 3:	*In case of doubts, call #BharatSir @ 98204 08217*		
Step 4:			
Step 5:			
	Remainder(0)	Quotient(7)	

TYPICAL ALU DESIGN



Addition	AC := AC + DR
Subtraction	AC := AC – DR
Multiplication	AC.MQ := DR × MQ
Division	AC.MQ := MQ/DR
AND	AC := AC ***and*** DR
OR	AC := AC ***or*** DR
EXCLUSIVE-OR	AC := AC ***xor*** DR
NOT	AC := ***not***(AC)