Sept 10^{th}

Note Title 10-09-2011

Non- Restoring Algorithm

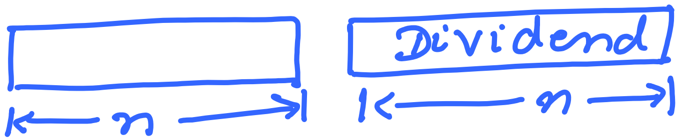
(2x faster than restoring algorithm)

1) Left shift

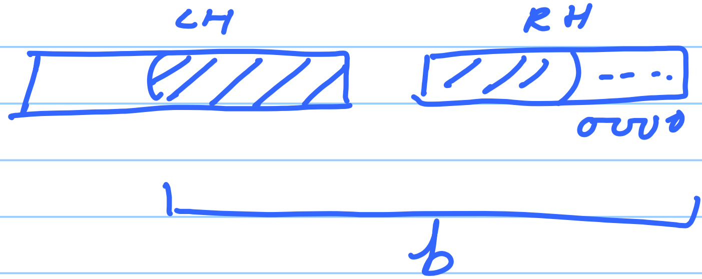
2) (Something) - - - .

3) Write quotient bit

Idea



Any point of time: assume that quotient bits are zero.



If we are subtracting the divisor (d)

The number that we get is:

$$b - d \times 2^n$$

Left shift.

$2b - d \times 2^{n+1}$ (Successful $LH > d$)

$(LH < d)$

Not successful:

Restoring Algorithm

1) Doesn't subtract (b)
2) Left shift $(2b)$
3) $(LH > d)$: Subtract.

$2b - d \times 2^n$

Non-restoring Algo.

1) Subtract: $b - d \times 2^n$
2) Left shift:

$2b - 2 \times d \times 2^n$

3) Add: $d \times 2^n$

$2b - d \times 2^n$

Example

Restoring

$10 \div 3$

$(1010) \quad (011)$

	1010
1) 0001	0100
2) 0010	1000
3) 0101	000
-011	
00010	0001

=

Non-restoring

$-3 = 1101$

1) 0001	1010
1101	
$-$ 1110	0100
2) 1100	100
$+$ 11	
$-$ 1111	1000
3) 1111	000
$+$ 11	
$+$ 0010	0001

4) $\boxed{0100}$ $\boxed{001}$

$-\ 11$

R $\boxed{0001}$ $\boxed{0011}$
Q

=

4) $\boxed{0100}$ $\boxed{001}$

$-\ 0\ 11$

$+$ $\boxed{0\ 0\ 0\ 1}$ $\boxed{0\ 0\ 1\ 1}$
R Q

Non-Restoring Algorithm

For every stage

1) Left shift $O(1)$

$O(\log(n))$ 2) If LH is positive
subtract divisor from LH

If LH is negative
add divisor to LH

3) Quotient Bit:

0 $\longrightarrow$ LH is neg. O(1)

1 $\longrightarrow$ LH is pos.

At the end: (R) can be negative

in this case: add the divisor to R.

Floating Point

Next Class

- IEEE 754
- Denormal Numbers
- Operations.