Division Algorithms: Signed- Magnitude Data

Division Algorithms: Outline

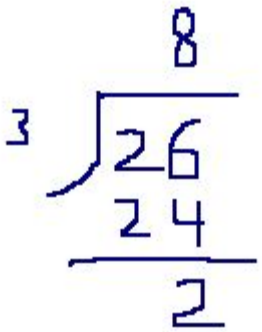
- Division: Shift – subtract Methodology
  - 3 ways to represent signed number
    - Signed Magnitude
    - 1’s Compliment
    - 2’s Compliment
  - Dividend = divisor * quotient + remainder
- Division By Hand: Long Division
- Hardware Algorithm: Restoring Method

Division algorithms--overview

Division with a paper and pencil

We repeatedly

- Comparison
- Shifting
- Subtraction



Division on computers: only deals with 0s and 1s

the same procedures, only easier than the decimals

How it is done in binary—an example

Dividend: 111011 (59)

Divisor: 11 (3)

Quotient: 10011 (19)

Remainder: 10 (2)

```
     10011 r 10
   _______
11)111011
  -11
   ______
     101
     -11
   ______
      101
       11
   ______
       10
```

Division Algorithm: Long Division

1. Determine Quotient Sign bit
2. Compare divisor with partial remainder
3. Subtract divisor from partial remainder if partial remainder > divisor
4. Shift divisor to the right
5. Repeat 2–4 until visit least significant bit of partial remainder
6. Combine the sign result from number 1 with the quotient and remainder

Signed-Magnitude

Dividend = -221 = 111011101

Divisor = 3 = 011

1. $Q_s = Divisor_s$ XOR $Dividend_s = 1$
   $Remainder_s = Dividend_s = 1$

2. – 4.

```
          1  0  0  1  0  0  1
      ┌──────────────────────
11    │ 1  1  0  1  1  1  0  1
        1  1
      ──────────────────────
           0  0  1  1
              1  1
      ──────────────────────
                 0  1  0  1
                    1  1
      ──────────────────────
                       1  0
```

Quotient = 11001001 = -73

Remainder = 110 = -2

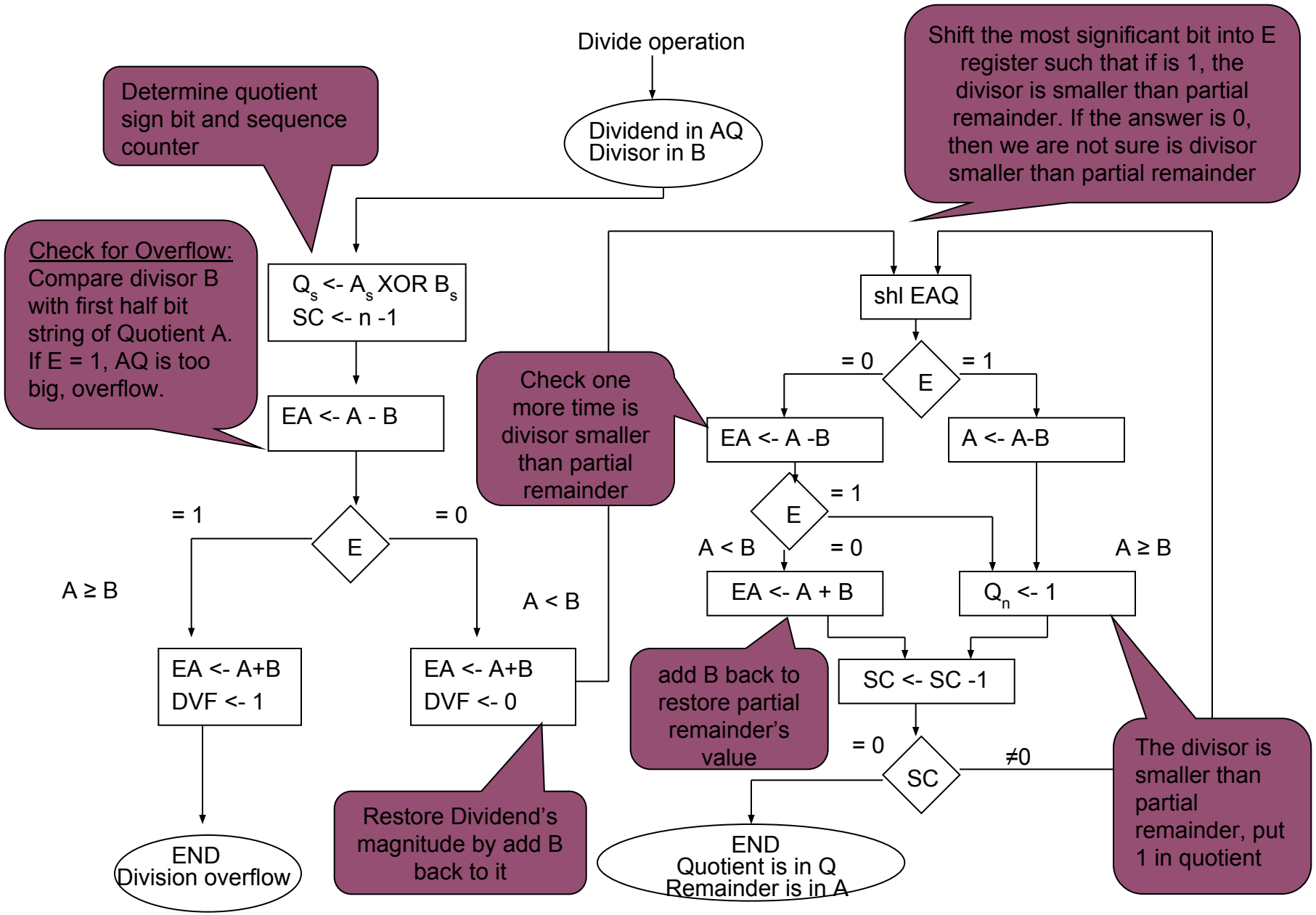
- 221 = -73 * 3 + -2

Division Algorithm: Restoring Method

- Division Overflow
  - Constraint by register size
  - Dividend = A Q; Divisor = B
    - A > B overflow
  - DVF: divide-overflow flip-flop
- Divide by Zero
  - Take care when check overflow

1. Determine Sign for quotient and remainder
2. Check for Overflow
3. Shift dividend to left
4. Check is partial remainder larger than divider
5. If larger, subtract divisor from partial remainder and put 1 for quotient
6. If smaller, do nothing
7. Repeat 3–6 for N times such that N = number of how many bit pattern a register can hold -1
8. Combine sign with quotient and remainder

Division Algorithm: Restoring Method



Example

- Ques: Perform Division algorithm on 448/17
- Sol: step 1: sign check, divisor n dividend both hv same sign.
- Step 2: 448- -> 0111000000
  - 17 10001
  - AQ dividend448 0111000000
  - A01110, Q-00000
  - B10001
  - B'+1 01111

  - Step 3: EAA-BA+B'+101110+01111=11101 CARRY BIT?
  - HERE E=0
  - STEP 4:

- Shl EAQ= 0 01110 00000
- 0 11100 00000

	E	A	Q	SC
INITIAL	0	01110	00000	101
Shl EAQ	0	11100	00000	
E=0,EA□ A-B		01111		
	1	01011	00001	100
Shl EAQ	0	10110	00010	
E=0, EA□A-B		01111		
	1	00101		
E=1, Qn□1			00011	011
Shl EAQ	0	01010	00110	
E=0, EA□A-B		01111		
	0	11001		
E=0, EA□A+B		10001		
	1	01010	00110	010
Shl EAQ	0	10100	01100	
E=0, EA□ A-B		01111		
	1	00011		
E=1,			01101	001
Shl EAQ	0	00110	11010	

ans

E - discard

A- remainder- 00110

Q – Quotient- 11010

+ve/-ve---- -ve result 2’s complement

-ve/ +ve---- -ve result 2’s complement

+ve/ +Ve - result

Division Algorithm: Restoring Method

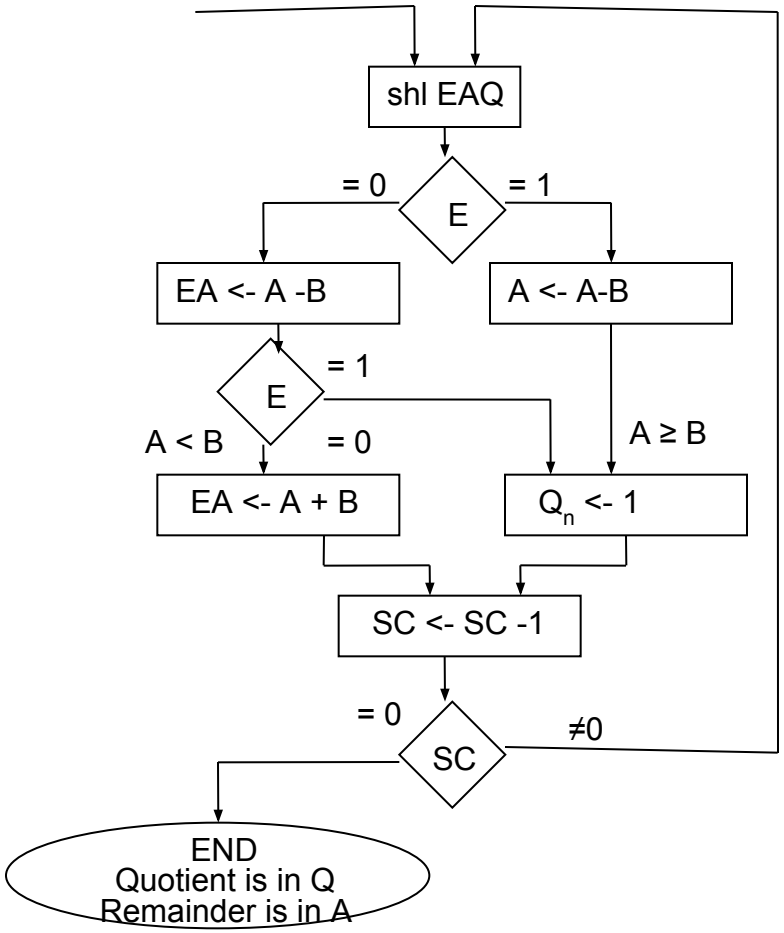
-15 / 3 = -5 …0

15 = 0000 1111

-15 = 1111 0001

3 = 0011

-3 = 1101



1. Qs = 1 XOR 0 = 1…negative

	E	A	Q	
	0	0 0 0 0	1 1 1 1	
		1 1 0 1		2. Check for Overflow
	0	1 1 0 1	1 1 1 1	E = 1 not overflow; restore value
		0 0 1 1		
	1	0 0 0 0	1 1 1 1	
	0	0 0 0 1	1 1 1 0	3&4. Shift to the Left
		1 1 0 1		Check is partial
	0	1 1 1 0	1 1 1 0	remainder big enough
		0 0 1 1		6. E =0. Restore original partial remainder
S3	1	0 0 0 1	1 1 1 0	
	0	0 0 1 1	1 1 0 0	3&4. Shift to the Left
		1 1 0 1		Check is partial
	1	0 0 0 0	1 1 0 1	remainder big enough
S2	0	0 0 0 1	1 0 1 0	E = 1; Qn = 1
		1 1 0 1		3&4. Shift to the Left
	0	1 1 1 0	1 0 1 0	Check is partial
		0 0 1 1		remainder big enough
	1	0 0 0 1	1 0 1 0	6. E =0. Restore original partial remainder
S1	0	0 0 1 1	0 1 0 0	
		1 1 0 1		3&4. Shift to the Left
	1	0 0 0 0	0 1 0 1	Check is partial
				remainder big enough

remainder

quotient