Computer Organization & Architecture

2-6 Multiplication of Signed Numbers

Wang Guohua

School of Software Engineering

Contents of this lecture

- Booth algorithm
  - Directly handle the multiplication with signed numbers

Booth Algorithm (1)

- Straightforward multiplication will not work in the case of a negative operand or two negative operands
- Example: M = 1001, Q = 0011, calculate $P = M \times Q$
  - Straightforward multiplication, P=00011011
  - If interpret M, Q, and P as unsigned non-negative numbers, then M = 9, Q = 3, P = 27
  - If interpret M, Q, and P as signed 2's complement numbers, then M = -7, Q = 3, P = 27

Booth Algorithm (2)

- Why straightforward multiplication not work?
  - Example: M = 1001, Q = 0011
  - Sign Extension

$$
\begin{array}{cccccccccc}
 & & & & 1 & 0 & 0 & 1 & & \\
 & & & \times & 0 & 0 & 1 & 1 & & \\
\hline
1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & & \\
1 & 1 & 1 & 1 & 0 & 0 & 1 & & & \\
0 & 0 & 0 & 0 & 0 & 0 & & & & \\
0 & 0 & 0 & 0 & 0 & & & & & \\
\hline
1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & & (-21)
\end{array}
$$

Booth Algorithm (3)

- Convert to unsigned, multiply, then convert back to signed
  - Works but is slow and a bit cumbersome
- Booth Algorithm can directly handle the multiplication with signed numbers
  - Recodes the bits of the multiplier
  - Works with 2’s complement numbers directly

Booth Algorithm (2)

- Reducing number of partial products
  - Fewer partial products generated for groups of consecutive 0's and 1's
    - Group of consecutive 0's in multiplier – no new partial product – only shift partial product right one bit position for every 0
    - Group of *m* consecutive 1's in multiplier – less than *m* partial products generated
    - Example: 0011110 = 0100000 – 0000010

      (decimal notation: 30 = 32 – 2)

Booth Algorithm (3)

- Recoding of a Multiplier

Multiplier		Version of Multiplicand Selected by bit i
Bit i	Bit i-1	
0	0	$0 \times M$
0	1	$+1 \times M$
1	0	$-1 \times M$
1	1	$0 \times M$

- Example
  - 0 0 1 0 1 1 0 0 1 1 1 0 1 0 1 1 0 0
  - 0+1 -1+1 0 -1 0 +1 0 0 -1+1 -1+1 0 -1 0 0

Booth Algorithm (4)

- Example

$$\begin{array}{rl} 0\,1\,1\,0\,1 & (+13) \\ \times\ 1\,1\,0\,1\,0 & (-6) \\ \hline \end{array}$$

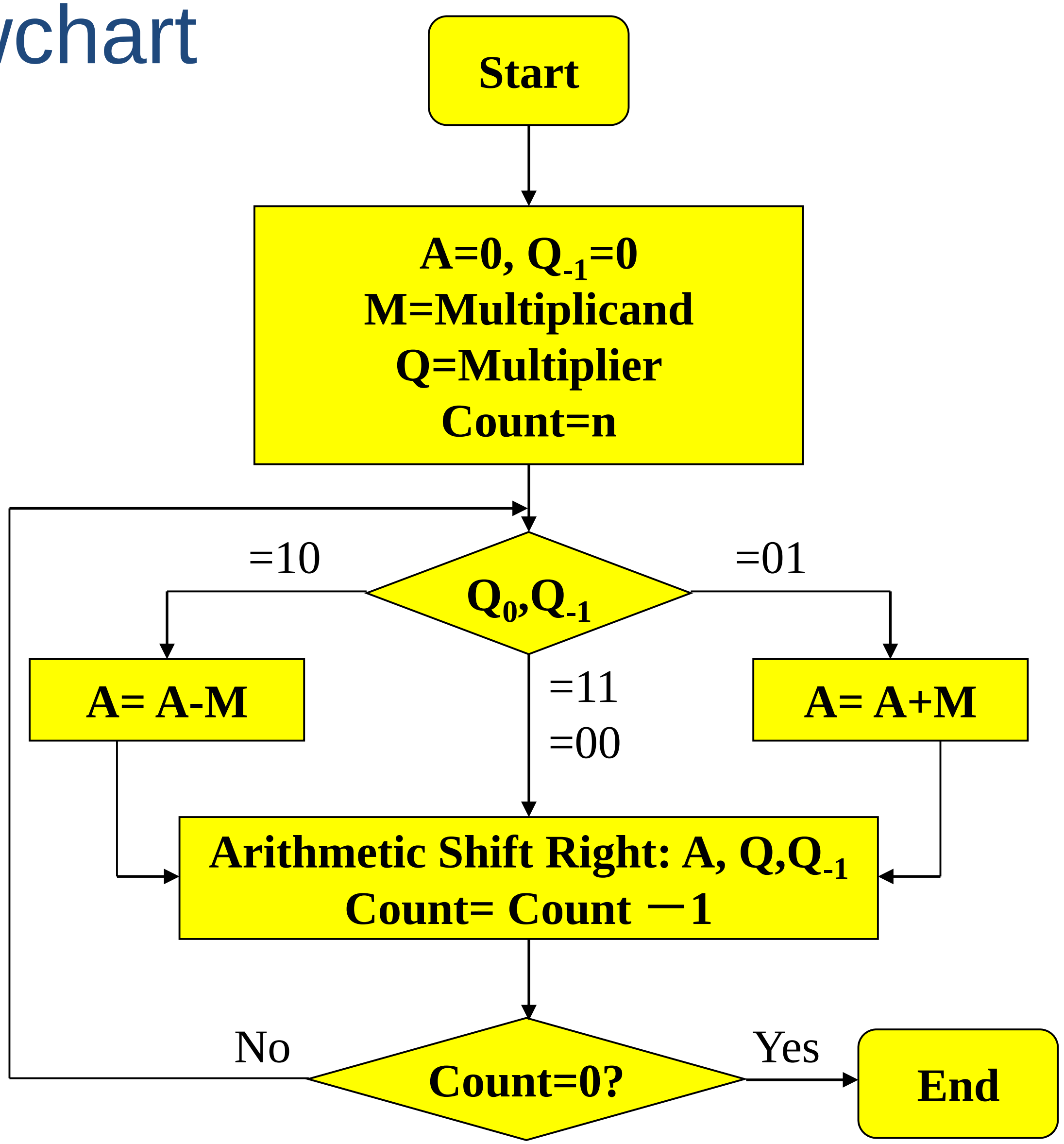
$$\begin{array}{cccccccccc l}
 & & & & & 0 & 1 & 1 & 0 & 1 & \\
 & & & & & \mathbf{0} & \mathbf{-1} & \mathbf{+1} & \mathbf{-1} & \mathbf{0} & \\
\hline
\color{blue}0 & \color{blue}0 & \color{blue}0 & \color{blue}0 & \color{blue}0 & 0 & 0 & 0 & 0 & 0 & \\
\color{blue}1 & \color{blue}1 & \color{blue}1 & \color{blue}1 & 1 & 0 & 0 & 1 & 1 & & \\
\color{blue}0 & \color{blue}0 & \color{blue}0 & 0 & 1 & 1 & 0 & 1 & & & \\
\color{blue}1 & \color{blue}1 & 1 & 0 & 0 & 1 & 1 & & & & \\
\color{blue}0 & 0 & 0 & 0 & 0 & 0 & & & & & \\
\hline
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & (-78)
\end{array}$$

Booth Algorithm (5)

- Hardware Organization
  - As before, the multiplicand and multiplier are placed in the M and Q registers, respectively.
  - There is a 1-bit register placed logically to the right of the least significant bit (Q_0) of the Q register and designated Q_{-1}.
  - Control logic scans the bits of the multiplier one at a time. Now, as each bit is examined, the bit to its right is also examined.
  - The results of the multiplication will appear in the A and Q registers.

Booth Algorithm (6)

- Flowchart



Arithmetic Shift Right:

The left-most of A, namely A_{n-1}, not only is shifted into A_{n-2}, but also remains in A_{n-1}.

Booth Algorithm (7)

- Example

	A	Q	Q_{-1}	
	M: 0 1 1 1			Initial Configuration
	0 0 0 0	0 0 1 1	0	
A=A-M	1 0 0 1	0 0 1 1	0	First Cycle
Shift	1 1 0 0	1 0 0 1	1	
Shift	1 1 1 0	0 1 0 0	1	Second Cycle
A=A+M	0 1 0 1	0 1 0 0	1	Third Cycle
Shift	0 0 1 0	1 0 1 0	0	
Shift	0 0 0 1	0 1 0 1	0	Fourth Cycle

Booth Algorithm (8)

- Advantages
  - Handle the multiplication of both positive and negative numbers uniformly.
  - Achieve some efficiency in the number of additions required when the multiplier has a few large blocks of 1s.
    - On average, the speed of doing multiplication with the Booth algorithm is the same as with the normal algorithm.

Booth Algorithm (9)

- Examples

Good multiplier	0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1
	0 0 0 +1 0 0 0 0 –1 0 0 0 +1 0 0 –1
Ordinary multiplier	1 1 0 0 0 1 0 1 1 0 1 1 1 1 0 0
	0 –1 0 0 +1 –1 +1 0 – 1 +1 0 0 0 –1 0 0
Worst-case multiplier	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
	+1 –1 +1 –1 +1 –1 +1 –1 +1 –1 +1 –1 +1 –1 +1 –1

Quiz (1)

1. In the hardware circuit of implementing Booth algorithm, the multiplier is placed in the Q register. And a 1-bit register Q_{-1} is placed logically to the right of the least significant bit (Q_0) of the Q register. The results of the multiplication will appear in the A and Q registers. At the end of each cycle of computing the product, A, Q, Q_{-1} are _______ one bit position.

   A. shifted left
   B. shifted right
   C. arithmetically shifted left
   D. arithmetically shifted right

Quiz (2)

2. About Booth algorithm, which of the following is not true?
   A. It can handle both positive and negative multipliers uniformly
   B. It achieves some efficiency in the number of additions required when the multiplier has a few large blocks of 1s
   C. Its speed of doing multiplication is always faster than the normal algorithm
   D. The multiplier 111100110 will generate three partial products

3. What are the main advantages of the Booth algorithm?
   – Directly handle the multiplication with the signed numbers
   – Fewer partial products generated for groups of consecutive 0’s and 1’s

Quiz (3)

4. Multiply A and B (2's-complement numbers) using the Booth algorithm. Assume that A is the multiplicand and B is the multiplier.

A=110011 and B=101100

						1	1	0	0	1	1
						-1	+1	0	-1	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	1	1	0	1		
0	0	0	0	0	0	0	0	0			
1	1	1	1	0	0	1	1				
0	0	0	1	1	0	1					
0	0	0	1	0	0	0	0	0	1	0	0

M

1 1 0 0 1 1

	A	Q	Q_{-1}	
	0 0 0 0 0 0	1 0 1 1 0 0	0	Initial Configuration
Shift	0 0 0 0 0 0	0 1 0 1 1 0	0	First Cycle
Shift	0 0 0 0 0 0	0 0 1 0 1 1	0	Second Cycle
A=A-M	0 0 1 1 0 1	0 0 1 0 1 1	0	Third Cycle
Shift	0 0 0 1 1 0	1 0 0 1 0 1	1	
Shift	0 0 0 0 1 1	0 1 0 0 1 0	1	Fourth Cycle
A=A+M	1 1 0 1 1 0	0 1 0 0 1 0	1	Fifth Cycle
Shift	1 1 1 0 1 1	0 0 1 0 0 1	0	
A=A-M	0 0 1 0 0 0	0 0 1 0 0 1	0	Sixth Cycle
Shift	0 0 0 1 0 0	0 0 0 1 0 0	1	