ICS 233
COMPUTER ARCHITECTURE

Integer Arithmetic
Multiplication

Lecture 13

Lecture Outline

- Integer Representation
- Integer Arithmetic
- Real Number Representation
- Floating-point Arithmetic

Integer Arithmetic

❑ Multiplication

➢ **Pencil and Paper method** (Multiplication of two unsigned numbers)

➢ **Booth's Algorithm** (Multiplication of Two's Complement numbers)

Integer Arithmetic

❑ **Pencil and Paper method (Multiplication of two unsigned numbers)**

❖ **The multiplication is carried out as follows :**

➢ **Multiplication involves the generation of partial products, one for each digit in the multiplier.**

➢ **These partial products are then summed to produce the final product**

- **The partial products are easily defined.**
- **When the multiplier bit is 0, the partial product is 0.**
- **When the multiplier is 1, the partial product is the multiplicand.**

➢ **The total product is produced by summing the partial products.**

- **For this operation, each successive partial product is shifted one position to the left relative to the preceding partial product.**

• **The multiplication of two n-bit binary integers results in a product of up to 2n bits in length.**

Integer Arithmetic

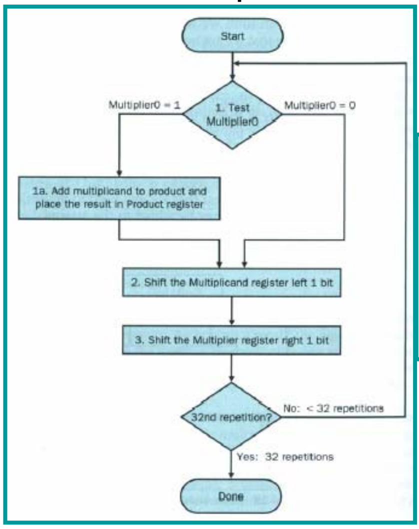
❑ **Multiplication - Shift & Add Method**

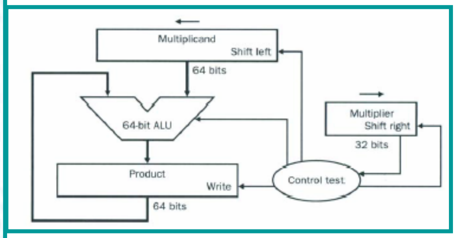
➢ Example : Compute 11 x 13

```
Multiplicand        Multiplier
    1011     x       1101
    -------------------------
     1011
    0000
   1011
  1011
 -------------
10001111    (143)
 -------------
```


First version of Multiplication Algorithm & Hardware

32-bit multiplication





First version of Multiplication Algorithm

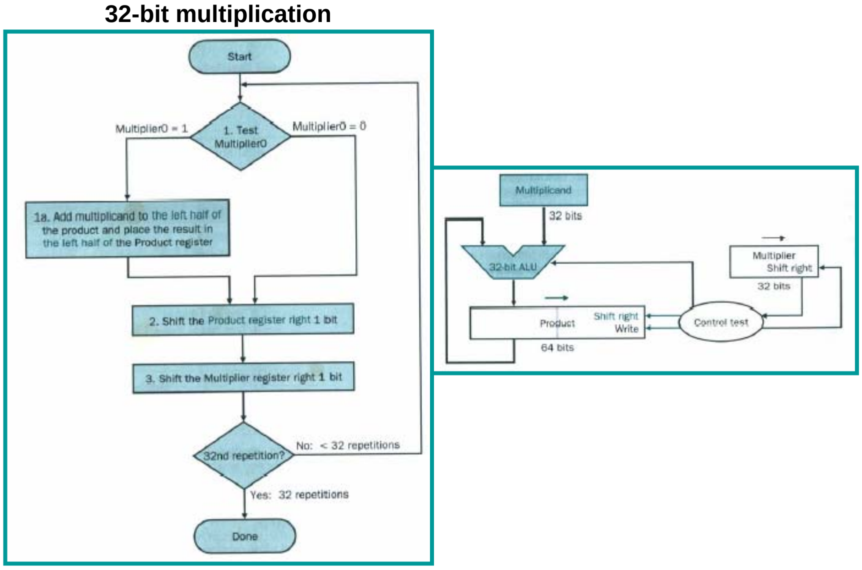
Example : Multiply 7 by 5 = 0111 x 0101 Q= 0101, M=0111

Iteration Counter	Action	Multiplier Q	Multiplicand M	Product P
4	Initialization	0101	0000 0111	0000 0000
3	Q0=1, P← P + M Shift Left M Shift Right Q	0101 0010	0000 0111 0000 1110	0000 0111
2	Q0=0, No operation Shift Left M Shift Right Q	0010 0001	0000 1110 0001 1100	0000 0111
1	Q0=1, P← P + M Shift Left M Shift Right Q	0001 0000	0001 1100 0011 1000	0010 0011
0	Q0=0, No operation Shift Left M Shift Right Q	0000 0000	0011 1000 0111 0000	**0010 0011**

Result : 0111 x 0101 = 0010 0011 = 35

Second version of Multiplication Algorithm & Hardware

32-bit multiplication



Second version of Multiplication Algorithm

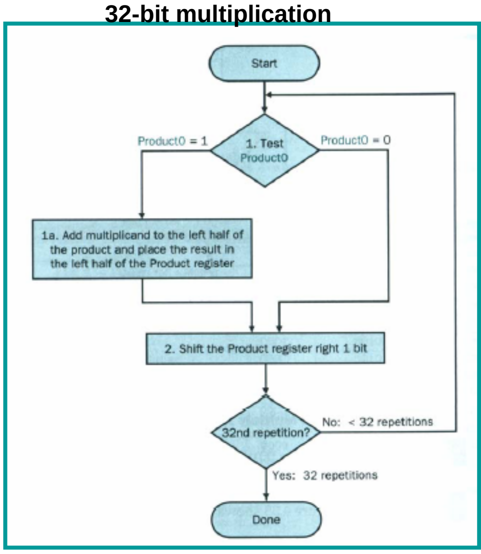
Example : Multiply 7 by 5 = 0111 x 0101 Q= 0101, M=0111

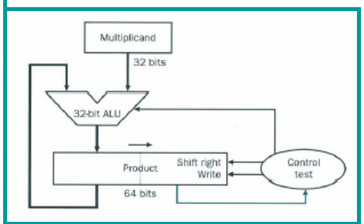
Iteration Counter	Action	Multiplier Q	Multiplicand M	Product P
4	Initialization	0101	0111	0000 0000
  3	Q0=1, P← lefthalf(P)+ M Shift Right P Shift Right Q	0101 0010	0111	0111 0000 0011 1000
  2	Q0=0, No operation Shift Right P Shift Right Q	0010 0001	0111	0011 1000 0001 1100
  1	Q0=1, P← lefthalf(P)+ M Shift Right P Shift Right Q	0001 0000	0111	1000 1100 0100 0110
  0	Q0=0, No operation Shift Right P Shift Right Q	0000 0000	0111	0010 0011 **0010 0011**

Result : 0111 x 0101 = 0010 0011 = 35

Final version of Multiplication Algorithm & Hardware

32-bit multiplication





Final version of Multiplication Algorithm

Example : Multiply 7 by 5 = 0111 x 0101 P= 0101, M=0111

Iteration Counter	Action	Multiplicand M	Product P
4	Initialization	0111	0000 **0101**
3	P_0=1, P← lefthalf(P)+ M Shift Right P	0111	0111 0101 0011 1010
2	P_0=0, No operation Shift Right P	0111	0011 1010 0001 1101
1	P_0=1, P← lefthalf(P)+ M Shift Right P	0111	1000 1101 0100 0110
0	P_0=0, No operation Shift Right P	0111	0100 0110 **0010 0011**

Result : 0111 x 0101 = 0010 0011 = 35

Integer Arithmetic

Booth's Algorithm

Two's Complement Multiplication (Multiplication of Signed numbers)

Booth's Multiplication Algorithm

Run of 1's

end of run

0	1	1	1	1	1	1	0

beginning of run

middle of run

Perform b x a : b – Multiplicand, a - Multiplier

Current bit a_i	Bit to the right a_{i-1}	Meaning	Operation
0	0	Middle of run of 0's	No operation
0	1	End of run of 1's	Add b
1	0	Beginning of run of 1's	Subtract b
1	1	Middle of run of 1's	No operation

Examining 2 bits at a time & Computing the product 1 bit at a time

Integer Arithmetic

- ❑ **Booth's Algorithm - Two's Complement Multiplication (Multiplication of Signed numbers)**

- ➢ **One Adder/Subtractor**
- ➢ **Registers Required :**

Three n-bit registers - A, Q, M and one 1-bit register, Q_{-1}

- The multiplier is loaded into register Q.
- The multiplicand is loaded into register M.
- Register A is used as a working register and is initialized to 0.
- There is also a 1-bit register placed logically to the right of the least significant bit (Q0) of the Q register and designated as Q_{-1} and initialized to 0

- **The results of the multiplication will appear in the A and Q registers**

Integer Arithmetic

❑ Booth's Algorithm

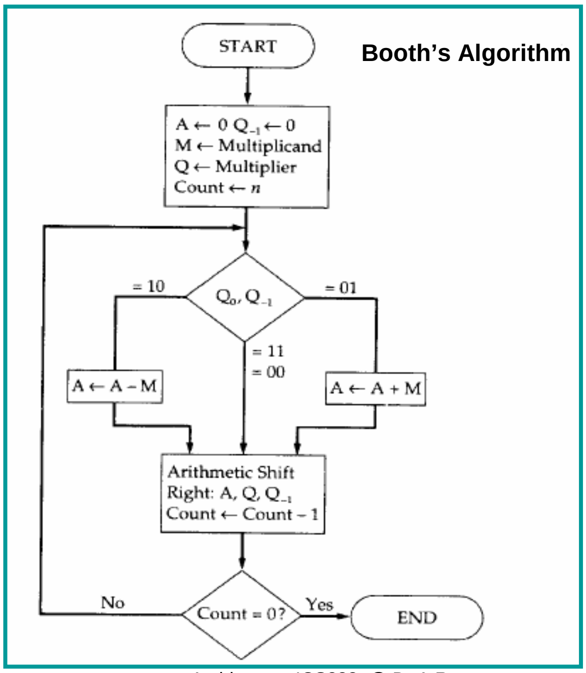
➢ **Step 1**
- Examine bits of the multiplier one at a time.
- As each bit is examined, the bit to its right is also examined.
  i.e., bits Q_0 Q_{-1} are examined

➢ **Step 2**
- If the two bits Q_0 Q_{-1} are the same 11 or 00, then
- shift all of the bits of the A, Q and Q_{-1} registers to the right 1 bit

- **The right shift operation is done such that the leftmost bit of A, namely An-1, not only is shifted into An-2 , but also remains in An-1.**
- **This is required to preserve the sign of the number in A and Q. It is also known as an arithmetic shift, since it preserves the sign bit.**

➢ **Step 3**
- If the two bits Q_0 Q_{-1} differ, then
  Add the multiplicand to the A register if two bits Q_0 Q_{-1} = 01
  OR Subtract the multiplicand from A register if two bits Q_0 Q_{-1} = 10
- Shift all of the bits of the A, Q, and Q_{-1} registers to the right 1 bit

- **Repeat the above steps for each bit of the original multiplier.**
- **The resulting 2n-bit product is contained in the A and Q registers**.



Booth's Multiplication Algorithm

Example : Multiply - 3 by 7 = 1101 x 0111 Q= 0111, M=1101

Iteration Counter	Action	A	Multiplier Q	Q_{-1}
4	Initialization	0000	0111	0
3	Q_0,Q_{-1}=10, A ← A – M Arithmetic Shift right A,Q,Q_{-1}	0011 0001	1011	1
2	Q_0,Q_{-1}=11, No operation Arithmetic Shift right A,Q,Q_{-1}	0000	1101	1
1	Q_0,Q_{-1}=11, No operation Arithmetic Shift right A,Q,Q_{-1}	0000	0110	1
0	Q_0,Q_{-1}=01, A ← A + M Arithmetic Shift right A,Q,Q_{-1}	1101 1110	0110 1011	1 0

Result : 1101 x 0111 = 11101011 = - 21

Booth Multiplier - HARDWARE IMPLEMENTATION