Boolean Arithmetic III

CSCI 2050U - Computer Architecture

Randy J. Fortier
@randy_fortier

OntarioTech UNIVERSITY

Outline

- Shift and rotation
- Booth's algorithm

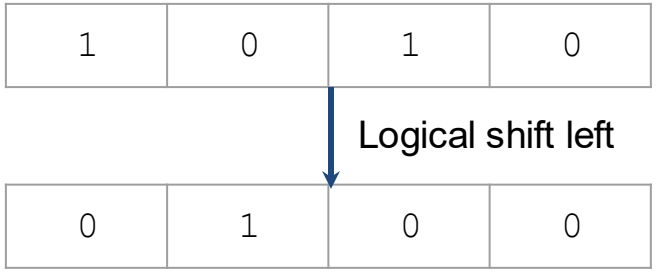
Shift and Rotation

CSCI 2050U - Computer Architecture

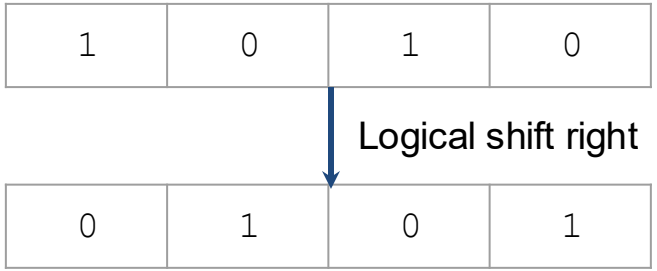
OntarioTech UNIVERSITY

Logical Shift

- Logical shift left
  - Shift all bits toward the left
  - Shift zero into the rightmost position
  - Leftmost bit gets shifted out



- Logical shift right
  - Shift all bits toward the right
  - Shift zero into the leftmost position
  - Rightmost bit gets shifted out



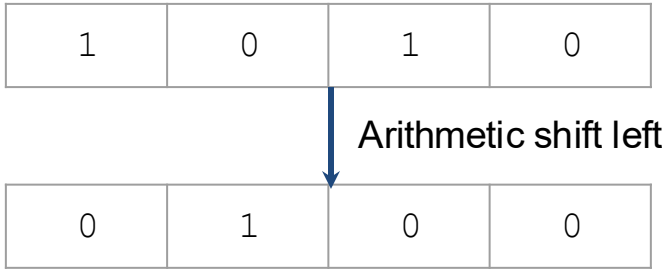
Logical Shift

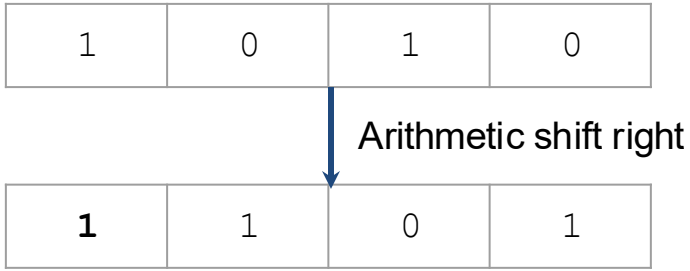
- Logical shift is used when you want to move bits into a more easily accessible position
  - e.g. If bits represent Boolean (flag) values
  - In some programming languages, the << and >> operators are used

Arithmetic Shift

- Arithmetic shift left
  - Shift all bits toward the left
  - Shift zero into the rightmost position
  - Leftmost bit gets shifted out
  - Identical to logical shift left

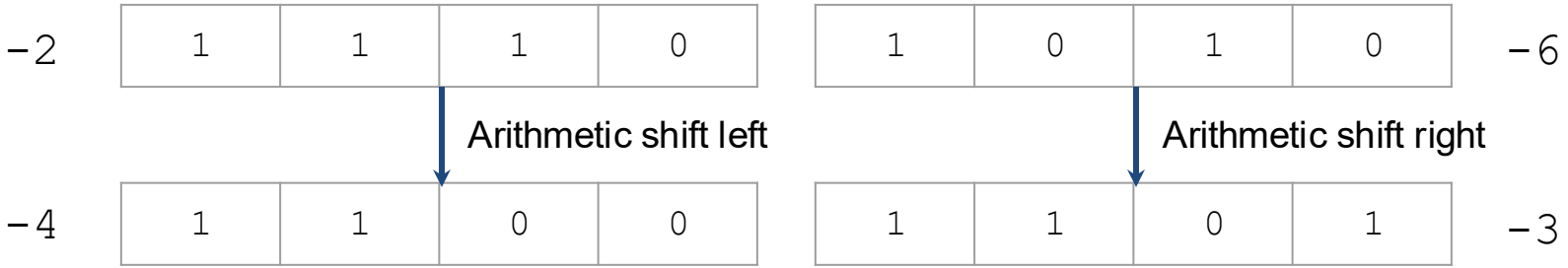
- Arithmetic shift right
  - Shift all bits toward the right
  - Duplicate the leftmost bit (the sign bit)
  - Rightmost bit gets shifted out





Arithmetic Shift

- Arithmetic shift is used when you want to multiply/divide by powers of 2
  - Arithmetic shift left by one bit → multiply by 2 (assuming no overflow)
    - Arithmetic shift left by `n` bits → multiply by 2^n
  - Arithmetic shift right by one bit → divide by 2 (integer division)
    - Arithmetic shift right by `n` bits → divide by 2^n



Arithmetic Shift

0	0	1	1	3
0	1	1	0	6
1	0	1	0	**-6**
1	1	0	1	**-3**

```
0101
   1
0110   (6)

0010
   1
0011   (3)
```

Multiplying Binary Numbers

- Arithmetic shift can be used to implement arbitrary multiplication
  - Multiply x by 13:
    - $13x = 8x + 4x + 1x = 2^3x + 2^2x + 2^0x$
    - Arithmetically shift x to the left by 3
    - Arithmetically shift x to the left by 2
    - Arithmetically shift x to the left by 0
    - Add up these results

Multiplying Binary Numbers

- Let's do an example:

```
x = -2

y = 11
```

Multiplying Binary Numbers

- Let's do an example:

1	1	1	1	1	1	1	0

x = -2

y = 11

Multiplying Binary Numbers

- Let's do an example:

1	1	1	1	1	1	1	0

$x = -2$

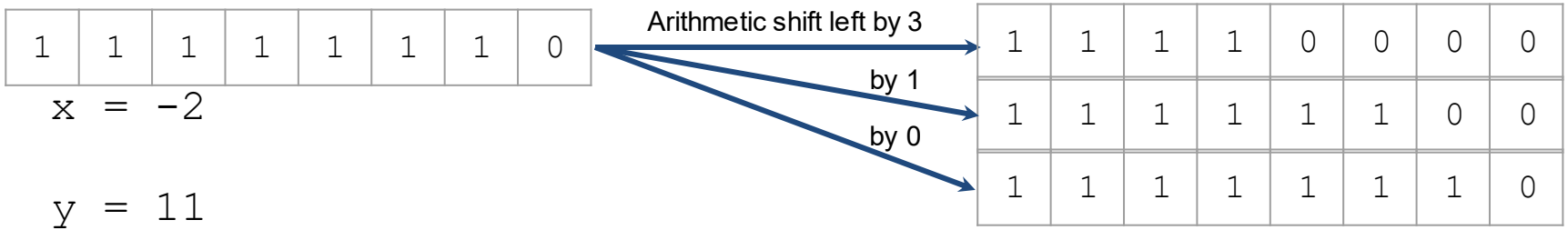
$y = 11$

$= 8 + 2 + 1$

$= 2^3 + 2^1 + 2^0$

Multiplying Binary Numbers

- Let's do an example:



$x = -2$

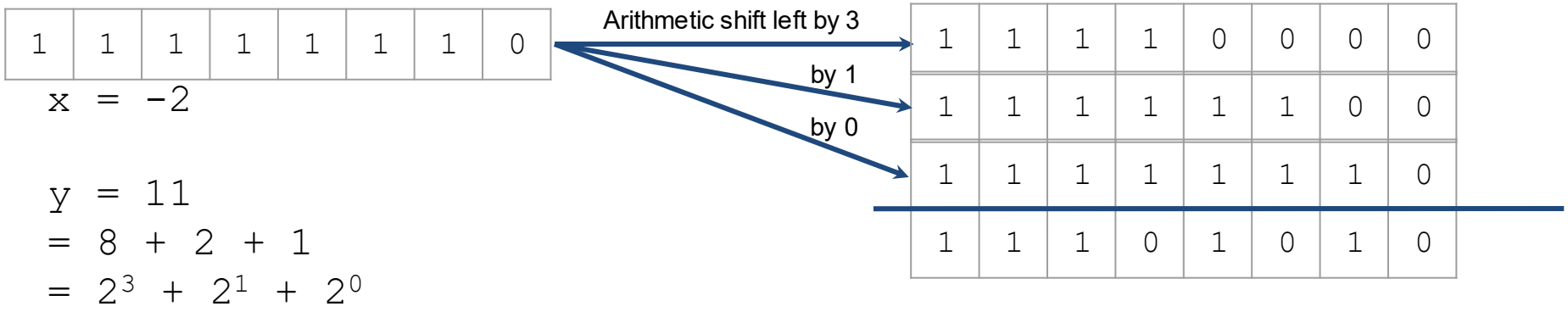
$y = 11$

$= 8 + 2 + 1$

$= 2^3 + 2^1 + 2^0$

Multiplying Binary Numbers

- Let's do an example:



$x = -2$

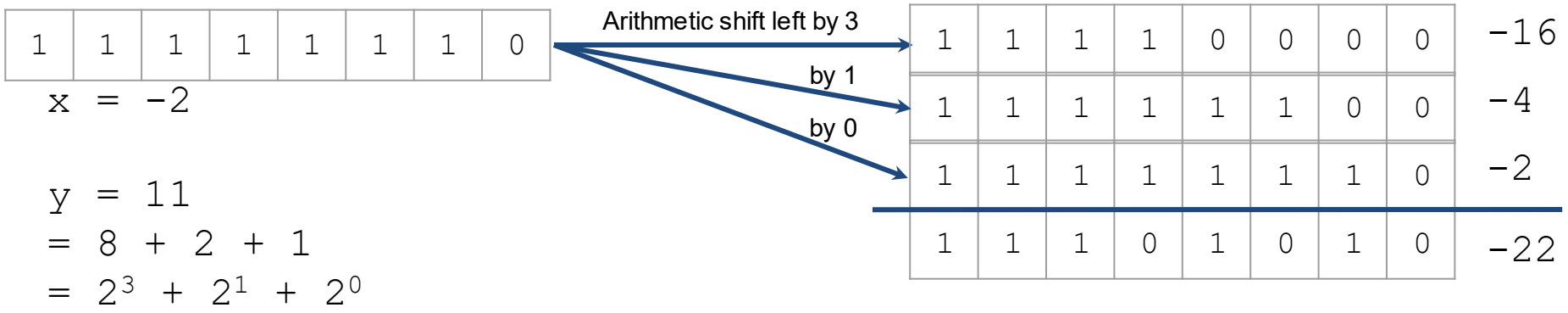
$y = 11$

$= 8 + 2 + 1$

$= 2^3 + 2^1 + 2^0$

Multiplying Binary Numbers

- Let's do an example:



$x = -2$

$y = 11$

$= 8 + 2 + 1$

$= 2^3 + 2^1 + 2^0$

Booth's Algorithm

CSCI 2050U - Computer Architecture

OntarioTech UNIVERSITY

Multiplying Binary Numbers - Booth's Algorithm

Booth's algorithm – a fast method of multiplying integers

- We'll use the following variables:
  - Q - multiplier
  - M – multiplicand
  - Q' – stores low order bit
  - A – stores high order portion of result (low order portion in Q)
- The number of steps is equal to the number of bits in M
- For each step, we compare Q's least significant bit and Q':
  - Arithmetic shift - shift all bits once to the right, keeping leftmost
  - Add – A = A + M
  - Subtract – A = A - M

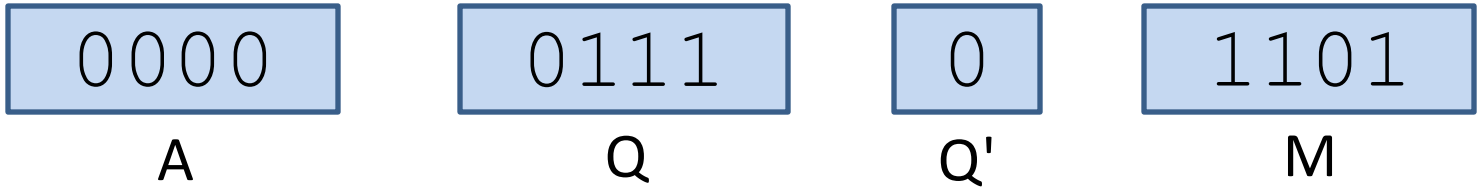
LSB of Q, Q'	Operation(s)
0,0	Arithmetic shift
0,1	Add, arithmetic shift
1,0	Subtract, arithmetic shift
1,1	Arithmetic shift

Multiplying Binary Numbers

Example:

- Let's multiply 7 * -3
  - Q = 7
  - M = -3
  - Q' = 0 (initially)
  - A = 0 (initially)

LSB of Q, Q'	Operation(s)
0,0	Arithmetic shift
0,1	Add, arithmetic shift
1,0	Subtract, arithmetic shift
1,1	Arithmetic shift



Multiplying Binary Numbers

Example:

- Step one
  - Operation(s) – Subtract, shift

LSB of Q, Q'	Operation(s)
0,0	Arithmetic shift
0,1	Add, arithmetic shift
1,0	**Subtract, arithmetic shift**
1,1	Arithmetic shift

0000	0111	0	1101
A	Q	Q'	M

Multiplying Binary Numbers

Example:

- Step one
  - Operation(s) – Subtract, shift
  - Subtract: A = A - M

LSB of Q, Q'	Operation(s)
0,0	Arithmetic shift
0,1	Add, arithmetic shift
1,0	**Subtract, arithmetic shift**
1,1	Arithmetic shift

0011	0111	0	1101
A	Q	Q'	M

```
 0000     0000
-1101    +0011
-----    -----
 0011     0011
```

Multiplying Binary Numbers

Example:

- Step one
  - Operation(s) – Subtract, shift
  - Arithmetic shift

LSB of Q, Q'	Operation(s)
0,0	Arithmetic shift
0,1	Add, arithmetic shift
1,0	**Subtract, arithmetic shift**
1,1	Arithmetic shift

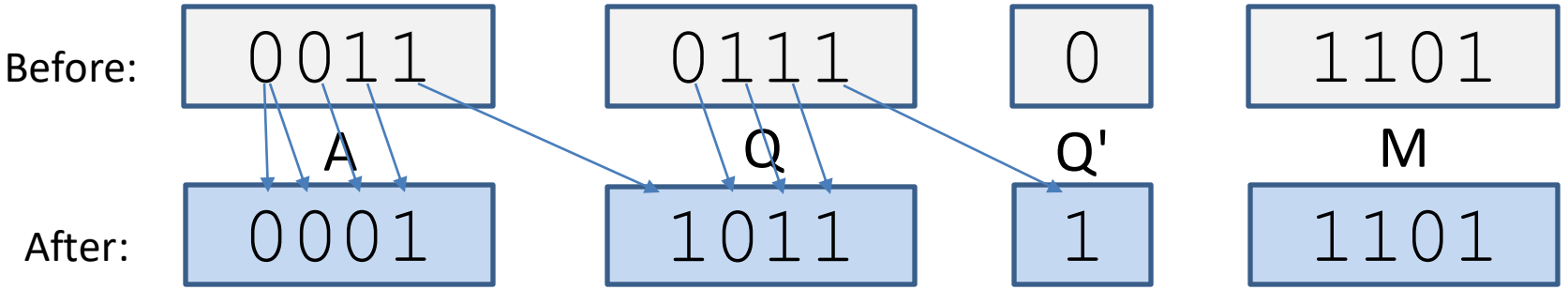
	A	Q	Q'	M
Before:	0011	0111	0	1101
After:	0001	1011	1	1101

Multiplying Binary Numbers

Example:

- Step one
  - Operation(s) – Subtract, shift
  - Arithmetic shift

LSB of Q, Q'	Operation(s)
0,0	Arithmetic shift
0,1	Add, arithmetic shift
1,0	**Subtract, arithmetic shift**
1,1	Arithmetic shift

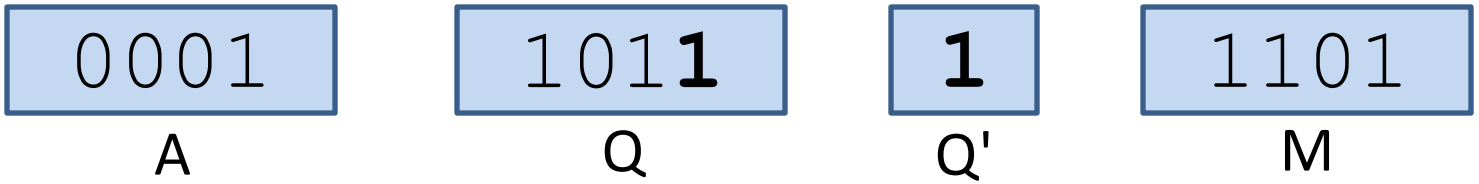


Multiplying Binary Numbers

Example:

- Step two
  - Operation(s) – shift
  - Arithmetic shift

LSB of Q, Q'	Operation(s)
0,0	Arithmetic shift
0,1	Add, arithmetic shift
1,0	Subtract, arithmetic shift
1,1	**Arithmetic shift**



Multiplying Binary Numbers

Example:

- Step two
  - Operation(s) – shift
  - Arithmetic shift

LSB of Q, Q'	Operation(s)
0,0	Arithmetic shift
0,1	Add, arithmetic shift
1,0	Subtract, arithmetic shift
1,1	**Arithmetic shift**

A	Q	Q'	M
0001	1011	1	1101
0000	1101	1	1101

Multiplying Binary Numbers

Example:

- Step three
  - Operation(s) – shift
  - Arithmetic shift

LSB of Q, Q'	Operation(s)
0,0	Arithmetic shift
0,1	Add, arithmetic shift
1,0	Subtract, arithmetic shift
1,1	**Arithmetic shift**

0000	1101**1**	**1**	1101
A	Q	Q'	M

Multiplying Binary Numbers

Example:

- Step three
  - Operation(s) – shift
  - Arithmetic shift

LSB of Q, Q'	Operation(s)
0,0	Arithmetic shift
0,1	Add, arithmetic shift
1,0	Subtract, arithmetic shift
1,1	**Arithmetic shift**

A	Q	Q'	M
0000	1101	1	1101
0000	0110	1	1101

Multiplying Binary Numbers

Example:

- Step four
  - Operation(s) – Add, shift

LSB of Q, Q'	Operation(s)
0,0	Arithmetic shift
0,1	**Add, arithmetic shift**
1,0	Subtract, arithmetic shift
1,1	Arithmetic shift

0000	0110	1	1101
A	Q	Q'	M

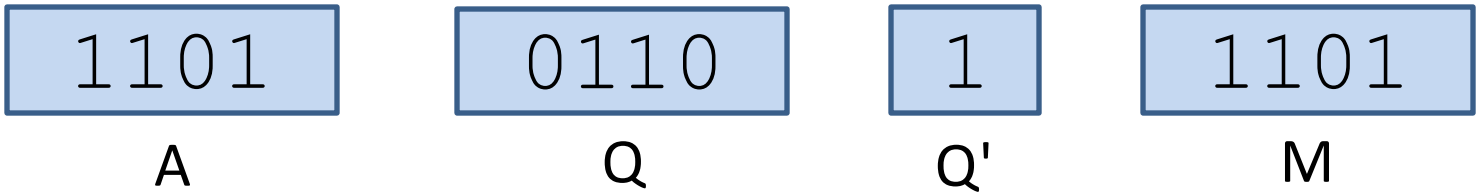
Multiplying Binary Numbers

Example:

- Step four
  - Operation(s) – Add, shift
  - Add: A = A + M

```
 0000
+1101
-----
 1101
```

LSB of Q, Q'	Operation(s)
0,0	Arithmetic shift
0,1	**Add, arithmetic shift**
1,0	Subtract, arithmetic shift
1,1	Arithmetic shift



Multiplying Binary Numbers

Example:

- Step four
  - Operation(s) – Add, shift
  - Arithmetic shift

LSB of Q, Q'	Operation(s)
0,0	Arithmetic shift
0,1	**Add, arithmetic shift**
1,0	Subtract, arithmetic shift
1,1	Arithmetic shift

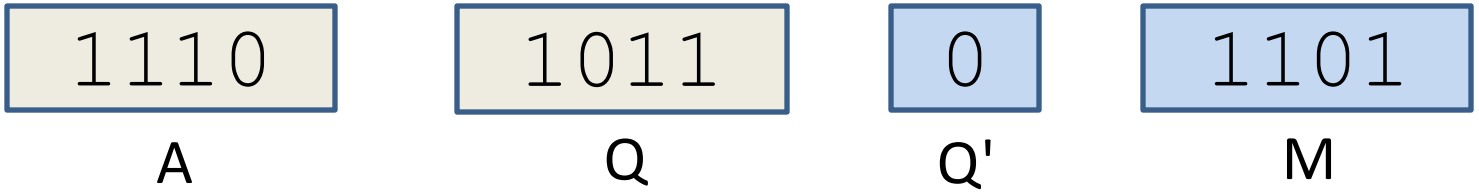
1101	0110	1	1101
A	Q	Q'	M
1110	1011	0	1101

Multiplying Binary Numbers

Example:

- Result: 1110 1011
  - This is a negative number
  - Complement it to find the magnitude
  - $0001\ 0101_2 = 21_{10}$
  - Thus, the result is 7 * -3 = -21

LSB of Q, Q'	Operation(s)
0,0	Arithmetic shift
0,1	Add, arithmetic shift
1,0	Subtract, arithmetic shift
1,1	Arithmetic shift



Wrap-Up

- Shift and rotation
  - Logical shift
  - Arithmetic shift
  - Rotation
- Booth's algorithm

What is next?

- Equivalent forms:
  - Boolean algebraic expressions
  - Truth tables
  - Digital circuit diagrams