# CS2208a

Lab #1 September 23, 2013

#### **General Information**

- Graduate Teaching Assistants (TA's)
  - Abdulwahab Kabini, akabani 5 @uwo.ca
  - Mike Molnar, mmolnar2@uwo.ca
  - Sakif Pritom, spritom@uwo.ca
- Email any questions to all three TA's.
- Check the book and lecture notes before emailing us.
- Email assignment marking issues to the TA that marked it.



### Lab #1 Outline

- Number Systems
- Conversion Methods
- Binary Addition
- Binary Subtraction
- Binary Multiplication
- Fraction Conversions



## **Number Systems**

- We will only consider four numbering systems:
  - Binary
- Octal
- Decimal
- Hexadecimal

Base 2	Base 8	Base 10	Base 16
0	0	0	0
1	1	1	1
	2	2	2
	3	3	3
	4	4	4
	5	5	5
	6	6	6
	7	7	7
		8	8
		9	9
			A
			В
			C
			D
			E
			F



#### **Conversion Methods**

- Division Method
  - Divide the number by the new base.
  - Keep the whole number of the result and the remainder.
  - Use the whole number for the next division.
  - Repeat until the whole number is 0.
  - Converts from decimal to binary, octal, or hexadecimal.

#### **Division Method**

- Convert 14<sub>10</sub> to binary:
  - 14/2 = 7 Remainder: 0
  - 7/2 = 3 Remainder: 1
  - 3/2 = 1 Remainder: 1
  - 1/2 = 0 Remainder: 1
- The remainders from the bottom to the top is the result.
  - $14_{10} = 1110_2$



#### **Division Method**

• Convert 2477<sub>10</sub> to base 16:

$$2477/16 = 154$$
 Remainder:  $13 = D$   
 $154/16 = 9$  Remainder:  $10 = A$   
 $9/16 = 0$  Remainder:  $9 = 9$ 

10 = A 11 = B 12 = C 13 = D 14 = E 15 = F

• The remainders from the bottom to the top is the result.  $2477_{10} = 9 AD_{16}$ 

#### **Conversion Methods**

- Positional Representation
  - Find the value that each digit represents.
  - The position of the digit is important.
  - Add the values to find the results.
  - Converts to decimal from binary, octal, or hexadecimal.



## Positional Representation

• Convert 2E8<sub>16</sub> to decimal:

$$2E8_{16} = 2 \times 16^{2} + E \times 16^{1} + 8 \times 16^{0}$$
$$= 2 \times 256 + 14 \times 16 + 8 \times 1$$
$$= 512 + 224 + 8$$
$$= 744_{10}$$

• Convert 361<sub>8</sub> to decimal:

$$361_8 = 3 \times 8^2 + 6 \times 8^1 + 1 \times 8^0$$
  
=  $3 \times 64 + 6 \times 8 + 1 \times 1$   
=  $192 + 48 + 1$   
=  $241_{10}$ 



## **Grouping Method**

• Convert 1100 1101, to octal:

Base 2	011	001	101
Base 8	3	1	5

• Convert 743<sub>8</sub> to binary:

Base 8	7	4	3
Base 2	111	100	011

0 = 000	
1 = 001	
2 = 010	
3 = 011	
4 = 100	
5 = 101	
6 = 110	
7 = 111	

#### **Conversion Methods**

- Grouping Method
- Convert between binary and octal or hexadecimal.
- Base  $8 = 2^3$  for binary, so group bits in three's.
- Base  $16 = 2^4$  for binary, so group bits in four's.
- Pad the left most group with 0's if needed.
- Converts between binary and octal or hexadecimal.



## **Grouping Method**

 $\bullet$  Convert 0111 1101 0101  $_2$  to hexadecimal:

Base 2	0111	1101	0101
Base 16	7	D	5

• Convert FA9<sub>16</sub> to binary:

Base 16	F	A	9
Base 2	1111	1010	1001

0 = 00001 = 00012 = 00103 = 00114 = 01005 = 01016 = 01107 = 01118 = 10009 = 1001A = 1010B = 1011C = 1100D = 1101E = 1110F = 1111



## **Grouping Method**

- To convert from hexadecimal to octal, you do it through binary conversion:
- Convert FA9<sub>16</sub> to octal
- FA9<sub>16</sub> **→** 1111 1010 1001
  - **→** 111 110 101 001
  - **→** 7 6 5 1
- $FA9_{16} \rightarrow 7651_8$

0 = 0000 1 = 0001 2 = 0010

3 = 00114 = 0100

5 = 01016 = 0110

7 = 01118 = 10009 = 1001

A = 1010B = 1011

C = 1100 D = 1101E = 1110

F = 1110

## **Grouping Method**

- To convert from octal to hexadecimal you do it through binary conversion:
- Convert 315<sub>8</sub> to hexadecimal
- 315<sub>8</sub> → 011 001 101
  - **→**11001101
  - → 1100 1101 → C D
- $315_8 \rightarrow CD_{16}$

0 = 0000 1 = 0001 2 = 0010

3 = 00114 = 0100

5 = 01016 = 0110

7 = 01118 = 1000

9 = 1001 A = 1010

B = 1011C = 1100

D = 1101E = 1110

F = 1111

## **Binary Addition**

- $\bullet 0 + 0 = 0$
- $\bullet$  0 + 1 = 1
- 1 + 1 = 0 Carry 1
- $\bullet 0 + 0 + 0 = 0$
- $\bullet$  0 + 0 + 1 = 1
- 0 + 1 + 1 = 0 Carry 1
- 1 + 1 + 1 = 1 Carry 1



Binary Addition – Two's Complement

1 1 0 0 1 0 0 1 - 1 1 1 1 1 1 1 1



Binary Addition – Two's Complement

Binary Addition – Two's Complement

1 1 0 0 1 0 0 1

1 1 1

17

Binary Addition – Two's Complement

0 0 0

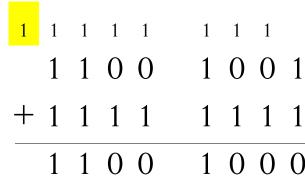
19

Binary Addition – Two's Complement

Binary Addition - Unsigned

25

Binary Addition - Unsigned

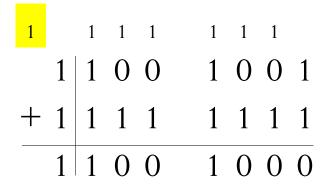


•Overflow occurred because there was a carry out.

Binary Addition - Sign and Magnitude

27

Binary Addition - Sign and Magnitude



•Overflow since the magnitude of the result was too large to fit in 7 bits.

•The sign is arbitrated separately.

Binary Addition – Two's Complement

29

Binary Addition – Two's Complement

Binary Addition – Two's Complement



Binary Addition – Two's Complement

Binary Addition – Two's Complement

Binary Addition - Unsigned



Binary Addition - Unsigned

•No overflow occurred because there was no carry out.

Binary Addition - Sign and Magnitude

37

Binary Addition - Sign and Magnitude

•Overflow since the magnitude of the result was too large to fit in 7 bits.

•The sign is arbitrated separately.

Binary Addition – Two's Complement

1 1 0 1 1 1 0 0 + 0 1 0 0 0 0 1 0



41

Binary Addition – Two's Complement

1

$$\frac{+\ 0\ 1\ 0\ 0}{0\ 0\ 1} \frac{0\ 0\ 1\ 0}{1\ 1\ 1\ 0}$$

42

Binary Addition – Two's Complement

1 1

•Overflow is **not possible** since the signs of the operands differ.

•The carry out is ignored in two's complement.

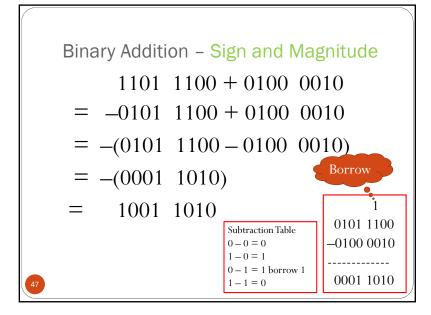
Binary Addition - Unsigned



Binary Addition – Sign and Magnitude

1 1 0 1 1 1 0 0

+ 0 1 0 0 0 1 0



#### Binary Subtraction –Two's Complement

- Two's complement does not use subtraction.
- Find two's complement of subtrahend and add.
- Overflow if both numbers have the same sign but the result is the opposite sign.

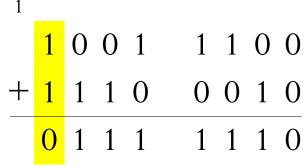
#### Binary Subtraction – Two's Complement

- Let's try -100 30
- The first step is to find the two's complement of 30 and 100.
- $30_{10} = 0001\ 1110_2$   $100_{10} = 0110\ 0100_2$
- Flip (complement) the bits and add 1 to get the two's complement.

• Now we can add -100 and -30.



## Binary Subtraction –Two's Complement



•Overflow because the sign of the sum differs from the sign of operands.

# Binary Multiplication

- $0 \times (0 \text{ or } 1) = 0$
- $1 \times 1 = 1$
- Algorithm for unsigned integers:
  - $\circ$  Start with the least significant bit of the multiplier.
  - ${\color{blue}\circ}$  If it is a 1 then copy the multiplicand to the partial product.
  - o Check the next bit and repeat.
  - $\circ$  Shift the partial product left each time.



## **Binary Multiplication**

- $5 \times 6 = 30_{10}$
- $5_{10} = 0101$ , = Multiplicand
- $6_{10} = 0110_2 = Multiplier$
- $30_{10} = 11110_2$

- 0 1 0 1
- $\times$  0 1 1 0
- 0 1 0 1



## **Binary Multiplication**

- Binary multiplication by powers of 2 uses shifting.
- Shift left arithmetic is used for multiplication.
  - $\times$  2 = shift left once
  - $\times$  4 = shift left twice
  - $\bullet \times 8 = \text{shift left three times}$
- Division by powers of 2 works in a similar fashion.
  - Shifts are to the right instead of left.
  - Shift logical instead of arithmetic.
  - This maintains the sign of the value.



#### **Fraction Conversions**

- Convert from decimal to binary.
- Convert the whole number with division method.
- Convert the fractional part with a multiplication method.
- Convert 12.6875<sub>10</sub> to binary.
- Convert 12<sub>10</sub> to binary first:
  - 12/2 = 6 R=0
  - 6/2 = 3 R=0
  - 3/2 = 1 R=1
  - 1/2 = 0 R=1
- Reading upwards gives 1100<sub>2</sub> as the whole part in binary.



## **Binary Multiplication**

- Multiply 3 by 2, 8, and 16:
  - $\bullet$  3<sub>10</sub> = 0000 0011,
  - $\times$  2 = one shift left = 0000 0110<sub>2</sub> = 6<sub>10</sub>
  - $\times$  8 = three shifts left = 0001 1000<sub>2</sub> = 24<sub>10</sub>
  - $\times$  16 = four shifts left = 0011 0000, = 48<sub>10</sub>
- Divide –32 by 4 and 8:
  - $-32_{10} = 11100000_2$
  - $\div 4$  = two shifts right = 1111 1000<sub>2</sub> =  $-8_{10}$
  - $\div 8$  = three shifts right = 1111 1100<sub>2</sub> =  $-4_{10}$
  - When shifting right we **replicate** the sign of the number.



#### **Fraction Conversions**

- Next we have to convert the fractional part.
- Similar to division method but uses multiplication.
- The result is read from the top to bottom.
- Keep whole part and use remaining fraction for next digit.
- Convert .6875<sub>10</sub> to binary:
  - $.6875 \times 2 = 1.375 \text{ WP} = 1$
- $.375 \times 2 = 0.75$  WP=0
- $.75 \times 2 = 1.5$  WP=1
- $.5 \times 2 = 1.0$  WP=1
- $\bullet$  Reading from top to bottom gives  $.1011_2$  as the result.



## **Fraction Conversions**

- Combining the two results gives the binary representation.
- $12.6875_{10} = 1100.1011_2$
- Find the 32-bit representation of the result:
  - $\bullet$  The normalized result is 1.1001011<sub>2</sub>
  - The exponent in binary is  $127+3 = 130_{10} = 10000010_2$
  - The sign bit is 0 because the number is positive.



## **Fraction Conversions**

- We can now convert the result to octal or hexadecimal.
- Use the grouping method to find the answer.

Base 2	001	000	001	010	010	110	000	000	000	000	000
Base 8	1	0	1	2	2	6	0	0	0	0	0

Base 2	0100	0001	0100	1011	0000	0000	0000	0000
Base 16	4	1	4	В	0	0	0	0

- The answer in octal is  $10122600000_8$ .
- The answer in hexadecimal is 414B0000<sub>16</sub>.

