Objectives

1. Arithmetic operations in binary number system \((addition, subtraction, multiplication)\)
2. Arithmetic operations on other number systems
3. Converting from Decimal to other Bases
4. Converting from Binary to Octal and Hexadecimal Bases
5. Other base conversions
Arithmetic Operation in base-$r$

- Arithmetic operations with numbers in base-$r$ follow the same rules as for decimal numbers
- Be careful!
  - Only $r$ allowed digits
Binary Addition

One bit addition:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ 0</td>
<td></td>
<td>+ 1</td>
<td>+ 0</td>
<td>+ 1</td>
</tr>
<tr>
<td>-----</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Augend (/aw-jend/)
Addend
Sum
Carry

2 doesn’t exist in binary!

Ahmad Almulhem, KFUPM 2009
Binary Addition (cont.)

Example:

\[
\begin{array}{cccccccc}
1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
+ & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
\hline
& & & & & & & & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1
\end{array}
\]

Q: How to verify?
A: Convert to decimal

\[
\begin{array}{cccc}
783 \\
+ & 490 \\
\hline
1273
\end{array}
\]
**Binary Subtraction**

In binary addition, there is a **sum** and a **carry**.

In binary subtraction, there is a **difference** and a **borrow**.

Note: 0 – 1 = 1 borrow 1

---

### One bit subtraction:

<table>
<thead>
<tr>
<th>Minuend</th>
<th>Subtrahend</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>-0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
</tbody>
</table>

*Minuend* /men-u-end/

*Subtrahend* /sub-tra-hend/

*Difference*
Binary Subtraction (cont.)

Subtract 101 - 011

1
0 \ 0 1
- 0 1 1

--------------------------

0 1 0

Larger binary numbers

1 1 1 1
\ \ \ \ 0 1 1 1 1
- 0 1 1 1 0 1 0 1 0

--------------------------

0 1 0 0 1 0 0 1 0 1

Verify In decimal,

783
- 490

--------

293

- In Decimal subtraction, the borrow is equal to 10.
- In Binary, the borrow is equal to 2. Therefore, a ‘1’ borrowed in binary will generate a \((10)_2\), which equals to \((2)_{10}\) in decimal
Binary Subtraction (cont.)

• Subtract \((11110)_2\) from \((10011)_2\)

\[
\begin{array}{c}
10011 \\
-11110
\end{array} \quad \begin{array}{c}
11110 \\
-10011
\end{array}
\]

\[
\begin{array}{c}
-01011
\end{array} \quad \begin{array}{c}
01011
\end{array}
\]

\text{borrow}

• Note that \((10011)_2\) is smaller than \((11110)_2\) \(\Rightarrow\) result is negative
Binary Multiplication

Multiply 1011 with 101:

\[
\begin{array}{cccc}
1 & 0 & 1 & 1 \\
\times & 1 & 0 & 1 \\
\hline
\end{array}
\]

----------

\[
\begin{array}{cccc}
1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 \\
\hline
1 & 1 & 0 & 1 & 1 \\
\end{array}
\]

Rules (short cut):

1. A ‘1’ digit in the multiplier implies a simple copy of the multiplicand
2. A ‘0’ digit in the multiplier implies a shift left operation with all 0’s
Hexadecimal addition

Add \((59F)_{16}\) and \((E46)_{16}\)

\[
\begin{array}{cccc}
5 & 9 & F & \text{F} + 6 = (21)_{10} = (16 \times 1) + 5 = (15)_{16} \\
+ & E & 4 & 6 \\
\hline
1 & 3 & E & 5 \\
\end{array}
\]

Rules:

1. For adding individual digits of a Hexadecimal number, a mental addition of the decimal equivalent digits makes the process easier.

2. After adding up the decimal digits, you must convert the result back to Hexadecimal, as shown in the above example.
### Octal Multiplication

Multiply \((762)_8\) with \((45)_8\)

<table>
<thead>
<tr>
<th>Octal</th>
<th>Octal</th>
<th>Decimal</th>
<th>Octal</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 6 2</td>
<td>5 x 2</td>
<td>((10)_{10}) = ((8 \times 1) + 2) = 12</td>
<td></td>
</tr>
<tr>
<td>x 4 5</td>
<td>5 x 6 + 1 = ((31)_{10}) = ((8 \times 3) + 7) = 37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>---------------</td>
<td>-------</td>
</tr>
<tr>
<td>4 6 7 2</td>
<td>5 x 7 + 3 = ((38)_{10}) = ((8 \times 4) + 6) = 46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 7 1 0</td>
<td>4 x 2</td>
<td>((8)_{10}) = ((8 \times 1) + 0) = 10</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>---------------</td>
<td>-------</td>
</tr>
<tr>
<td>4 3 7 7 2</td>
<td>4 x 6 + 1 = ((25)_{10}) = ((8 \times 3) + 1) = 31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>---------------</td>
<td>-------</td>
</tr>
<tr>
<td>4 3 7 7 2</td>
<td>4 x 7 + 3 = ((31)_{10}) = ((8 \times 3) + 7) = 37</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We use decimal representation for ease of calculation.
Converting Decimal Integers to Binary

- Divide the decimal number by ‘2’
- Repeat division until a quotient of ‘0’ is received
- The sequence of remainders in reverse order constitute the binary conversion

Example:

\[(41)_{10} = (101001)_{2}\]

Verify: \[1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (41)_{10}\]
Converting Decimal Integer to Octal

1. Divide the decimal number by 8.
2. Repeat division until a quotient of 0 is received.
3. The sequence of remainders in reverse order constitute the binary conversion.

Example:

\[(153)_{10} = (231)_{8}\]

Verify: \[2 \times 8^2 + 3 \times 8^1 + 1 \times 8^0 = (153)_{10}\]
Converting Decimal Fraction to Binary

- Multiply the decimal number by ‘2’
- Repeat multiplication until a fraction value of ‘0.0’ is reached or until the desired level of accuracy is reached
- The sequence of integers before the decimal point constitute the binary number

Example:

\[(0.6875)_{10} = (0.1011)_{2}\]

Verify: \[1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} = (0.6875)_{10}\]
Converting Decimal Fraction to Octal

• Multiply the decimal number by ‘8’
• Repeat multiplication until a fraction value of ‘0.0’ is reached or until the desired level of accuracy is reached
• The sequence of integers before the decimal point constitute the octal number

Example:
\((0.513)_{10} = (0.4065\ldots)_{8}\)

Verify: \(4\times 8^{-1} + 0 \times 8^{-2} + 6 \times 8^{-3} + 5 \times 8^{-4} = (0.513)_{10}\)
Q. How to convert a number that has both integral and fractional parts?
A. Convert each part separately, combine the two results with a point in between.

Example: Consider the “decimal -> octal” examples in previous slides

\[(153.513)_{10} = (231.407)_{8}\]
Converting Binary to Octal

- Group 3 bits at a time
- Pad with 0s if needed
- Example: \((11001.11)_2 = (011 \ 001.110)_2 = (31.6)_8\)
Converting Binary to Hexadecimal

- Group 4 bits at a time
- Pad with 0s if needed
- Example: \((11001.11)_2 = (0001\ 1001.1100)_2 = (19.C)_{16}\)

<table>
<thead>
<tr>
<th>Group of 4 Binary Bits</th>
<th>Hexadecimal Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>1</td>
</tr>
<tr>
<td>0 0 1 0</td>
<td>2</td>
</tr>
<tr>
<td>0 0 1 1</td>
<td>3</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>4</td>
</tr>
<tr>
<td>0 1 0 1</td>
<td>5</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>6</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>7</td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>8</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>9</td>
</tr>
<tr>
<td>1 0 1 0</td>
<td>A</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>B</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>C</td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>D</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>E</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>F</td>
</tr>
</tbody>
</table>

Ahmad Almulhem, KFUPM 2009
Converting between other bases

Q. How to convert between bases other than decimal; e.g from base-4 to base-6?

A. Two steps:
   1. convert source base to decimal
   2. convert decimal to destination base.

Exercise: $(123)_4 = (\ ? )_6$ ?
Example

Convert \((211.6250)_{10}\) to binary?

Steps:

- Split the number into integer and fraction
- Perform the conversions for the integer and fraction part separately
- Rejoin the results after the individual conversions
Example (cont.)

<table>
<thead>
<tr>
<th>Division</th>
<th>Quotient</th>
<th>Remainder</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\frac{211}{2})</td>
<td>105</td>
<td>1</td>
</tr>
<tr>
<td>(\frac{105}{2})</td>
<td>52</td>
<td>1</td>
</tr>
<tr>
<td>(\frac{52}{2})</td>
<td>26</td>
<td>0</td>
</tr>
<tr>
<td>(\frac{26}{2})</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>(\frac{13}{2})</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>(\frac{6}{2})</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>(\frac{3}{2})</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>(\frac{1}{2})</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Combining the results gives us:

\[(211.6250)_{10} = (11011011.101)_{2}\]
## Decimal to binary conversion chart

<table>
<thead>
<tr>
<th>Decimal</th>
<th>binary equivalent (3-bits)</th>
<th>binary equivalent (4-bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000</td>
<td>0000</td>
</tr>
<tr>
<td>1</td>
<td>001</td>
<td>0001</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
<td>0010</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
<td>0011</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>0100</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>0101</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
<td>0110</td>
</tr>
<tr>
<td>7</td>
<td>111</td>
<td>0111</td>
</tr>
<tr>
<td>8</td>
<td>Not convertible (need more bits)</td>
<td>1000</td>
</tr>
<tr>
<td>9</td>
<td>Not convertible (need more bits)</td>
<td>1001</td>
</tr>
<tr>
<td>10</td>
<td>Not convertible (need more bits)</td>
<td>1010</td>
</tr>
<tr>
<td>11</td>
<td>Not convertible (need more bits)</td>
<td>1011</td>
</tr>
<tr>
<td>12</td>
<td>Not convertible (need more bits)</td>
<td>1100</td>
</tr>
<tr>
<td>13</td>
<td>Not convertible (need more bits)</td>
<td>1101</td>
</tr>
<tr>
<td>14</td>
<td>Not convertible (need more bits)</td>
<td>1110</td>
</tr>
<tr>
<td>15</td>
<td>Not convertible (need more bits)</td>
<td>1111</td>
</tr>
<tr>
<td>16</td>
<td>Not convertible (need more bits)</td>
<td>Not convertible (need more bits)</td>
</tr>
</tbody>
</table>
Conclusions

- When performing arithmetic operations in base-\(r\), remember allowed digits \(\{0, \ldots r-1\}\).
- To convert from decimal to base-\(r\), divide by \(r\) for the integral part, multiply by \(r\) for the fractional part, then combine.
- To convert from binary to octal (hexadecimal) group bits into 3 (4).
- To convert between bases other than decimal, first convert source base to decimal, then convert decimal to the destination base.