Number System

M. Sachdev,  
Dept. of Electrical & Computer Engineering  
University of Waterloo

General Radix Representation

- A decimal number such as 7392 represents  
  7392 = 7 \times 10^3 + 3 \times 10^2 + 9 \times 10^1 + 2 \times 10^0

- It is practical to write only coefficients and deduce power of 10s from position  
  - In general, any radix (base) can be used

- Define coefficients \( a_i \) in radix \( r \)
  
  \[ 0 \leq a_i < r \]
  
  \[ a_n r^n + a_{n-1} r^{n-1} + \ldots + a_0 r^0 + \ldots + a_m r^m \]

  Common radics – \( r = 2, 4, 8, 10, 16 \)
### General Radix Representation

<table>
<thead>
<tr>
<th>$r = 10$ (Dec.)</th>
<th>$r = 2$ (Binary)</th>
<th>$r = 8$ (Octal)</th>
<th>$r = 16$ (Hex)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0000</td>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>0001</td>
<td>01</td>
<td>1</td>
</tr>
<tr>
<td>02</td>
<td>0010</td>
<td>02</td>
<td>2</td>
</tr>
<tr>
<td>03</td>
<td>0011</td>
<td>03</td>
<td>3</td>
</tr>
<tr>
<td>04</td>
<td>0100</td>
<td>04</td>
<td>4</td>
</tr>
<tr>
<td>05</td>
<td>0101</td>
<td>05</td>
<td>5</td>
</tr>
<tr>
<td>06</td>
<td>0110</td>
<td>06</td>
<td>6</td>
</tr>
<tr>
<td>07</td>
<td>0111</td>
<td>07</td>
<td>L7</td>
</tr>
<tr>
<td>08</td>
<td>1000</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>09</td>
<td>1001</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
<td>12</td>
<td>A</td>
</tr>
<tr>
<td>11</td>
<td>1011</td>
<td>13</td>
<td>B</td>
</tr>
<tr>
<td>12</td>
<td>1100</td>
<td>14</td>
<td>C</td>
</tr>
</tbody>
</table>

### General Radix Representation

<table>
<thead>
<tr>
<th>$r = 10$ (Dec.)</th>
<th>$r = 2$ (Binary)</th>
<th>$r = 8$ (Octal)</th>
<th>$r = 16$ (Hex)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>1101</td>
<td>15</td>
<td>D</td>
</tr>
<tr>
<td>14</td>
<td>1110</td>
<td>16</td>
<td>E</td>
</tr>
<tr>
<td>15</td>
<td>1111</td>
<td>17</td>
<td>F</td>
</tr>
</tbody>
</table>

Usually, radix is shown as subscript

$(1234.4)_5 = 1 \times 5^3 + 2 \times 5^2 + 3 \times 5^1 + 4 \times 5^0 + 4 \times 5^{-1} = (513.4)_{10}$

$(\text{F75C.B})_{16} = 15 \times 16^3 + 7 \times 16^2 + 5 \times 16^1 + 12 \times 16^0 + 11 \times 16^{-1} = (63,324.6875)_{10}$
Radix Conversion

The integral part of a decimal number to radix r, repeatedly divide by r with reminders becoming a_i

- Convert $(77)_{10}$ to binary

<table>
<thead>
<tr>
<th>Integer</th>
<th>Remainder</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>1</td>
<td>a0</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>a1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>a2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>a3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>a4</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>a5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>a6</td>
</tr>
</tbody>
</table>

$(77)_{10} = (1001101)_{2}$

Examples

- Convert $(173)_{10}$ to r = 7

<table>
<thead>
<tr>
<th>Integer</th>
<th>Remainder</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>173</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>5</td>
<td>a0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>a1</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>a2</td>
</tr>
</tbody>
</table>

$(173)_{10} = (335)_{2}$

- Converting from Binary to decimal

$(101101)_{2} = 1 \times 2^{5} + 0 \times 2^{4} + 1 \times 2^{3} + 1 \times 2^{2} + 0 \times 2^{1} + 1 \times 2^{0} = (45)_{10}$
Fraction Conversion

- To convert the fractional part of a number to radix r, repeatedly multiply by r; integral parts of products becoming $a_i$
- Convert $0.7215_{10}$ to binary

<table>
<thead>
<tr>
<th>$0.7215 \times 2$</th>
<th>= 1.443</th>
<th>$a_1 = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.443 \times 2$</td>
<td>= 0.866</td>
<td>$a_2 = 0$</td>
</tr>
<tr>
<td>$0.866 \times 2$</td>
<td>= 1.772</td>
<td>$a_3 = 1$</td>
</tr>
<tr>
<td>$0.772 \times 2$</td>
<td>= 1.544</td>
<td>$a_4 = 1$</td>
</tr>
<tr>
<td>0.544</td>
<td>= 1.088</td>
<td>$a_5 = 1$</td>
</tr>
</tbody>
</table>

$(0.7215)_{10} = (0.10111...)_{2}$

Conversion between Binary/Octal/Hex

- Nice simple ways to convert between these three number systems, since all are a power of 2
- Binary to Octal simply requires grouping bits into groups of 3-bits and converting
- Binary to Hex simply requires grouping bits into groups of 4-bits and converting
- Going the other direction (Octal to Binary or Hex to Binary) should follow...
- Example $(100111100101)_2 = (4745)_8 = (9E5)_{16}$
Number Systems- Book Sections

- Number representations and conversions are covered in Chapter 1, Sections 1.1 through 1.4

Complements

- In computers, the representation and manipulation of –ve numbers is often performed using complements
- Complements for a radix come in two forms
  - R’s complement (Radix complement)
  - (r-1)’s complement (Diminished radix complement)
- For binary system -- 2’s complement, & 1’s complement
r’s Complement

- Given a +ve number \( N \) with \( n \) digits
  \[ N = (a_{n-1}a_{n-2} \ldots a_0) \]
- \( r \)'s complement is defined as
  \[ r^n - N \text{ for } N \neq 0; \text{ zero otherwise} \]
- Examples – 10’s complement
  \[(37218)_{10} = 10^5 - 37218 = 62782\]
  \[(0.12345)_{10} = 10^0 - 0.12345 = 0.87655\]
- Examples – 2’s complement
  \[(101110)_2 = 2^6 - 101110 = 010010\]
  \[(0.0110)_2 = 2^0 - 0.0110 = 0.1010\]

(r-1)’s Complement

- Given a +ve number \( N \) with \( n \) digit integer part & \( m \) digit fractional part
  \[ N = (a_{n-1}a_{n-2} \ldots a_0.a_1a_2 \ldots a_m) \]
- \( (r-1) \)'s complement is defined as
  \[ r^n - r^m - N \]
- Examples – 9’s complement
  \[(37218)_{10} = 10^5 - 1 - 37218 = 62781\]
  \[(0.12345)_{10} = 10^0 - 10^{-5} - 0.12345 = 0.87654\]
- Examples – 1’s complement
  \[(101110)_2 = 2^6 - 2^0 - 101110 = 111111 - 101100 = 010011\]
  \[(0.0110)_2 = 2^0 - 2^{-4} - 0.0110 = 0.1111 - 0.0110 = 0.1001\]
Interesting Facts

- The complement of a complement returns the original number.
- Since we work with binary numbers a lot in digital systems, it is really worth nothing that:
  - The 1’s complement of a number is obtained by flipping bits.
  - The 2’s complement of a number is obtained by flipping bits and adding 1.

Arithmetic Operations

- Arithmetic operations - addition, subtraction, multiplication can be performed in any radix.
- Example, $r = 2$.
**Addition (Unsigned Numbers)**

- Binary addition of unsigned numbers is done just like in decimal. Add digits and generate carries.

\[
\begin{array}{ccc}
1010101 & + & 0011001 \\
+ & 1101110 & 110 \\
\end{array}
\quad
\begin{array}{ccc}
1010101 & + & 1011001 \\
+ & 10101110 & 174 \\
\end{array}
\]

**Important:** We can get a non-zero *carry out*; if we are limited to n-bits, this becomes an *overflow* condition.

**Subtraction (Unsigned Numbers)**

- Binary subtraction is also similar to decimal subtraction. Use borrow when needed.

\[
\begin{array}{ccc}
1010101 & \ - & 0011001 \\
\ - & 0111100 & 60 \\
\end{array}
\quad
\begin{array}{ccc}
1010101 & \ - & 1011001 \\
\ - & 01011100 & \ ???? \\
\end{array}
\]

**Important:** Doesn’t seems to work if second term is smaller!! A negative result is often considered an underflow.
**Subtraction (Unsigned Numbers) Using r’s Complement**

- Direct method (using borrows) is fine if done by hand, but a hassle in a digital system
  - Usage of complement makes subtraction easier to implement in hardware
- The subtraction of two numbers \( (A-B)_r \) can be done as:
  - Add r’s complement of B to A
    \[ Y = A + (r^n - B) = (A - B) + r^n \]

---

**Subtraction …**

- If \( A \geq B \) then carry out \( (r^n) \) occurs; ignore it, the rest is the subtracted result
- If \( A < B \)
  \[ Y = A + (r^n - B) = r^n - (A - B) \]
- Note: \( (A - B) \) is a -ve number! Hence, take r’s complement and treat it as -ve number
  \[ = r^n - Y = r^n - \{ r^n - (A - B) \} = (A - B) \]
Subtraction …Examples

- 10’s complement subtraction
  A = 72532, A’ = 27468
  B = 03250, B’ = 96750
  A – B = 72532  B – A = 03250
      + 96750     + 27468
  1 69282    → 30718  →  -69282

Subtraction …Examples

- 2’s complement subtraction
  A = 1010100, A’ = 0101100
  B = 1000100, B’ = 0111100
  A – B = 1010100  B – A = 1000100
      + 0111100     + 0101100
  1 0010000  → 1110000
        → -0010000
Subtraction (Unsigned Numbers) Using (r-1)’s Complement

- Similar to r’s complement subtraction
- If A > B
  \[ Y = A + \{(r^n - r^m) - B\} = (A - B) + r^n - r^m \]
  If end carry \((r^n)\) is discarded & least significant carry \((r^m)\) is added, result is the subtracted value
- If A = B
  \[ Y = A + \{(r^n - r^m) - B\} = (A - B) + r^n - r^m = 0 + r^n - r^m \]

Subtraction (Unsigned Numbers) Using (r-1)’s Complement

- If A < B
  \[ Y = A + \{(r^n - r^m) - B\} = -(B - A) + r^n - r^m \]
  \[ = r^n - r^m - (B - A) \]
- We take \((r-1)\)’s complement & treat it as a –ve number
  \[ r^n - r^m -Y = r^n - r^m - \{(r^n - r^m - (B - A)\} = (A - B) \]
**Subtraction …Examples**

- **9’s complement subtraction**
  
  \[
  A = 72532, \ A' = 27467 \\
  B = 03250, \ B' = 96749 \\
  A – B = 72532 \quad B – A = 03250 \\
  + 96749 \quad + 27467 \\
  1 \ 69281 \quad 30717 \Rightarrow -69282 \\
  + \ 00001 \\
  69282
  \]

- **1’s complement subtraction**
  
  \[
  A = 1010100, \ A' = 0101011 \\
  B = 1000100, \ B' = 0111011 \\
  A – B = 1010100 \quad B – A = 1000100 \\
  + 0111011 \quad + 0101011 \\
  1 0010000 \quad 1101111 \\
  0000001 \Rightarrow -0010000 \\
  0010000
  \]
Signed Numbers

- Computers handle signed numbers as well
  - Numbers are represented in a fixed # of bits
- Often required to represent both +ve & -ve numbers in the same n-bit format
  - Left most bit (Most Significant Bit) represents the sign
    0 → +ve; 1 → -ve
- Three common representations
  - Sign & magnitude
  - Signed 1’s complement
  - Signed 2’s complement
- For +ve numbers, all three have same representation

---

Signed Numbers

- Sign & magnitude
  - MSB is the sign bit; rest is the magnitude
  - For 8 bit word
    + 9 → 0000 1001; -9 → 1000 1001
- Signed 1’s complement
  - MSB is the sign bit
    - For +ve number → actual value
    - For –ve number → 1’s complement
  + 9 → 0000 1001; -9 → 1111 0110
  - Has 2 representations for 0; not widely used
Signed Numbers

- Signed 2’s complement
  - Common method of representing signed #s
  - Restrict the number range to (n-1) bits; Use 2’s complement for –ve number representation
  - + 9 → 0000 1001; -9 → 1111 0111

- With 5 bit word size, in signed 2’s complement we can represent number from -16 to +15
  
  \[ Y = (-a_{n-1})2^{n-1} + a_{n-2}2^{n-2} + \ldots + a_02^0 \]

- Exercise – write all number from -16 to +15

Addition of Signed Binary Numbers

- 2’s complement
  - Perform the addition, ignore the carry

\[
\begin{array}{c|c}
00000110 & + 6 \\
+ 0001101 & + 13 \\
\hline
00010011 & + 19 \\
\end{array}
\]

\[
\begin{array}{c|c}
11111010 & - 6 \\
+ 00001101 & + 13 \\
\hline
100000111 & + 7 \\
\end{array}
\]

\[
\begin{array}{c|c}
00000110 & + 6 \\
+ 11110011 & - 13 \\
\hline
01111001 & - 7 \\
\end{array}
\]

\[
\begin{array}{c|c}
11111010 & - 6 \\
+ 11110011 & - 13 \\
\hline
11101101 & - 19 \\
\end{array}
\]
Subtraction of Signed Binary Numbers

- 2’s complement
  - Take 2’s complement of the # to be subtracted
    \[ (+/-A) - (+/-B) = (+/-A) + (-/+B) \]
  - A, B are in signed 2’s complement
- Example, (-6) – (-13) = +7
  \[
  \begin{align*}
  (1111 1010) - (1111 0011) &= 1111 1010 + 0000 1101 \\
  &= 1 0000 0111
  \end{align*}
  \]
  - Ignore the carry out (9th bit)

Codes

- Decimal numbers are coded using binary bit patterns

<table>
<thead>
<tr>
<th>Decimal</th>
<th>BCD (8421)</th>
<th>2421</th>
<th>Excess-3</th>
<th>Gray</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
<td>0000</td>
<td>0011</td>
<td>0000</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
<td>0001</td>
<td>0100</td>
<td>0001</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
<td>0010</td>
<td>1001</td>
<td>0011</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
<td>0011</td>
<td>0110</td>
<td>0010</td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
<td>0100</td>
<td>0111</td>
<td>0110</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
<td>0101</td>
<td>1000</td>
<td>0111</td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
<td>0110</td>
<td>1001</td>
<td>0101</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
<td>0111</td>
<td>1010</td>
<td>0100</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>1000</td>
<td>1011</td>
<td>1100</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>1111</td>
<td>1100</td>
<td>1101</td>
</tr>
</tbody>
</table>
BCD Addition

- BCD addition can be carried out as follows:
  
  4  0100  8  1000
  +8  +1000  +9  +1001
  12  1100  17  10001

  0110  0110

  1 0010  1 0111

Complements, Signed Arithmetic, Codes - Book Sections

- Complements and Signed Arithmetic are covered in Chapter 1, Sections 1.5 through 1.7