CS1101: Lecture 12

Radix Numbers: Conversions, Negatives & Arithmetic

Dr. Barry O'Sullivan
`b.osullivan@cs.ucc.ie`

Course Homepage
`http://www.cs.ucc.ie/~osullb/cs1101`

Lecture Outline

- **Reading**: Tanenbaum, Appendix A.
- Conversions
  - Binary-Decimal Conversion
  - Binary-Decimal Conversion by Doubling
  - Decimal-Octal & Decimal-Hexadecimal
- Binary Arithmetic
- Negatives
  - Negative Binary Numbers
  - Negative Numbers – Comparison
  - Arithmetic in One's and Two's Complement
  - Errors

Binary-Decimal Conversion

- Binary integers can also be converted to decimal in two ways. One method consists of summing up the powers of 2 corresponding to the 1 bits in the number.
- Example:

$$10110 = 2^4 + 2^2 + 2^1 = 16 + 4 + 2 = 22$$

CS1020: Computer Systems I *Radix Numbers: Conversions, Negatives & Arithmetic*

Binary-Decimal Conversion by Doubling

- The binary number is written vertically, one bit per line, with the leftmost bit on the bottom.
- The bottom line is called line 1, the one above it line 2, and so on.
- The decimal number will be built up in a parallel column next to the binary number.
- Begin by writing a 1 on line 1.
- The entry on line n consists of two times the entry on line $n - 1$ plus the bit on line n (either 0 or 1).
- The entry on the top line is the answer.

Binary-Decimal Conversion by Doubling

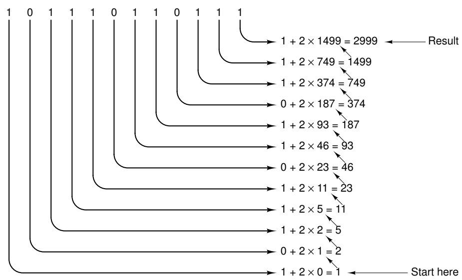

Figure A.6 Conversion of the binary number 101110110111 to decimal by successive doubling.

Decimal-Octal & Decimal-Hexadecimal

- Decimal-to-octal and decimal-to-hexadecimal conversion can be accomplished either by first converting to binary
- Then convert to the desired system or convert by subtracting powers of 8 or 16.

Binary Arithmetic

Addend	0	0	1	1
Augend	+0	+1	+0	+1
Sum	0	1	1	0
Carry	0	0	0	1

Figure A-8: The addition table in binary.

Binary Arithmetic

- Two binary numbers can be added, starting at the rightmost bit and adding the corresponding bits in the addend and the augend.
- If a carry is generated, it is carried one position to the left, just as in decimal arithmetic.
- In one's complement arithmetic, a carry generated by the addition of the leftmost bits is added to the rightmost bit. This process is called an end-around carry.
- In two's complement arithmetic, a carry generated by the addition of the leftmost bits is merely thrown away.

Negative Binary Numbers

- Four different systems for representing negative numbers have been used in digital computers at one time or another.
- The first one is called **signed magnitude**. In this system the leftmost bit is the sign bit (0 is + and I is -) and the remaining bits hold the absolute magnitude of the number.
- The second system, called **one's complement**, also has a sign bit with 0 used for plus and 1 for minus.
- To negate a number, replace each 1 by a 0 and each 0 by a 1.
- This holds for the sign bit as well.
- One's complement is obsolete!

Negative Binary Numbers

- The third system, called **two's complement**, also has a sign bit that is 0 for plus and 1 for minus.
- Negating a number is a two-step process.
  - First, each 1 is replaced by a 0 and each 0 by a 1, just as in one's complement.
  - Second, 1 added to the result.
- Binary addition is the same as decimal addition except that a carry is generated if the sum is greater than 1 rather than greater than 9.
- For example, converting 6 to two's complement is done in two steps:

```
00000110 (+6)
11111001 (-6 in one's complement)
11111010 (-6 in two's complement)
```

- If a carry occurs from the leftmost bit, it is thrown away.

Negative Binary Numbers

- The fourth system, which for m-bit numbers is called **excess** $2^{(m-1)}$ represents a number by storing it as the sum of itself and $2^{(m-1)}$
- For example, for 8-bit numbers, m = 8, the system is called excess 128 and a number is stored as its true value plus 128.
- Therefore, -3 becomes -3 + 128 = 125, and -3 is represented by the 8-bit binary number for 125 (01111101).
- The numbers from -128 to +127 map onto 0 to 255, all of which are expressible as an 8-bit positive integer.
- Interestingly enough, this system is identical to two's complement with the sign bit reversed.

Negative Numbers – Comparison

N decimal	N binary	–N signed mag.	–N 1's compl.	–N 2's compl.	–N excess 128
1	00000001	10000001	11111110	11111111	01111111
2	00000010	10000010	11111101	11111110	01111110
3	00000011	10000011	11111100	11111101	01111101
4	00000100	10000100	11111011	11111100	01111100
5	00000101	10000101	11111010	11111011	01111011
6	00000110	10000110	11111001	11111010	01111010
7	00000111	10000111	11111000	11111001	01111001
8	00001000	10001000	11110111	11111000	01111000
9	00001001	10001001	11110110	11110111	01110111
10	00001010	10001010	11110101	11110110	01110110
20	00010100	10010100	11101011	11101100	01101100
30	00011110	10011110	11100001	11100010	01100010
40	00101000	10101000	11010111	11011000	01011000
50	00110010	10110010	11001101	11001110	01001110
60	00111100	10111100	11000011	11000100	01000100
70	01000110	11000110	10111001	10111010	00111010
80	01010000	11010000	10101111	10110000	00110000
90	01011010	11011010	10100101	10100110	00100110
100	01100100	11011010	10011011	10011100	00011100
127	01111111	11111111	10000000	10000001	00000001
128	Nonexistent	Nonexistent	Nonexistent	10000000	00000000

Figure A-7: Negative 8-bit numbers in four systems.

Binary Arithmetic

Addend	0	0	1	1
Augend	+0	+1	+0	+1
Sum	0	1	1	0
Carry	0	0	0	1

Figure A-8: The addition table in binary.

Binary Arithmetic

- Two binary numbers can be added, starting at the rightmost bit and adding the corresponding bits in the addend and the augend.
- If a carry is generated, it is carried one position to the left, just as in decimal arithmetic.
- In one's complement arithmetic, a carry generated by the addition of the leftmost bits is added to the rightmost bit. This process is called an end-around carry.
- In two's complement arithmetic, a carry generated by the addition of the leftmost bits is merely thrown away.

Binary Arithmetic

Decimal	1's complement	2's complement
10	00001010	00001010
+ (–3)	11111100	11111101
+7	1 00000110	1 00000111
	carry 1	discarded
	00000111	

Figure A-9: Addition on one's complement and two's complement.

Binary Arithmetic – Errors

- If the addend and the augend are of opposite signs, overflow error cannot occur.
- If they are of the same sign and the result is of the opposite sign, overflow error has occurred and the answer is wrong.
- In both one's and two's complement arithmetic, overflow occurs if and only if the carry into the sign bit differs from the carry out of the sign bit.
- Most computers preserve the carry out of the sign bit, but the carry into the sign bit is not visible from the answer.
- For this reason, a special overflow bit is usually provided.