Radix Conversions, Characters Codes, Parity

Tom Kelliher, CS 220

Sept. 10, 2003

1 Administrivia

Announcements

Study binary, hex addition/subtraction on your own. Responsible for assigned readings.

Assignment

Read 2.1–2.

From Last Time

Outline

1. Radix conversions.
2. Character encodings.
3. Parity.

Coming Up

Binary logic, gates, Boolean algebra.

2 From Last Time

Binary	Hexadecimal	Decimal
11100111	E7	231
00111010	3A	58
11000111	C7	199
00011111	1F	31
11101110	EE	238

3 Radix Conversion

Binary or hexadecimal to decimal is simple enough.

Decimal to binary algorithm:

```
/* d: decimal number
 * b: binary number
 * b[i]: bit i of b
 */

b = 0;
i = 0;

while (d != 0)
{
   b[i] = d % 2;   /* d modulo 2 (radix) */
   d = d / 2;      /* Integer division */
   ++i;
}
```

Example: convert $(77)_{10}$ to binary.

How do we modify this for hexadecimal? Repeat the example.

4 Character Representation

1. So far, all we can represent is unsigned numbers. How can we represent characters?

2. ASCII character code. A few examples using hex encodings:

   (a) A: 41, a: 61.

   (b) Z: 5A, z: 7A.

   Collating sequence.

   (c) 0: 30, 9: 39.

   (d) !: 21, =: 3D, ’ ’: 20.

   (e) nl: 0A, cr: 0D.

   C code to convert an integer numeric string to integer value:

   ```
   char s[] = "123";
   int val;
   int i;

   val = 0;
   i = 0;

   while (isdigit(s[i])
   {
      val = val * 10;
      val = s[i] - ’0’;
      ++i;
   }
   ```

3. ASCII is a seven-bit code; characters stored in bytes.

   What about characters for non-English languages, math characters, etc? Unicode: 16-bit character code.

5 Parity

1. Used to *detect* data errors in memory or during simple data communications (serial lines).

Detects single bit errors. Misses double bit errors.

2. Other mechanisms: ECC, CRC.

3. Idea: Maintain one extra bit which keeps total number of one bits even or odd.

   Odd parity examples: 0011010 becomes 00011010; 0110011 becomes 10110011.