CpE 213
Digital Systems Design

Lecture 13

Thursday 10/9/2003

Announcements

- Homework 4 has been posted on Blackboard. It is due on Thursday 10/16.
- Check your grades on Blackboard and collect your assignments from my office.

Intel HEX Format

- The "Intel-Standard" HEX file is one of the most popular and commonly used formats in the 8051 world.
- The standard is used to burn the 8051 program into an EPROM, PROM, etc. For example, an 8052 assembler will usually generate an Intel Standard HEX file which can then be loaded into an EPROM programmer and burned into the chip.

Intel HEX File

- An Intel Standard HEX file is an ASCII file with one "record" per line. Each line has the following format:

Position	Description
1	**Record Marker**: The first character of the line is always a colon (ASCII 0x3A) to identify the line as an Intel HEX file
2 – 3	**Record Length**: This field contains the number of data bytes in the register represented as a 2-digit hex number. This is the total number of *data* bytes, not including the checksum byte nor the first 9 characters of the line.
4 - 7	**Address**: This field contains the address (in hex) where the data should be loaded into the chip
8 - 9	**Record Type**: This field indicates the type of record for this line. The possible values are: **00**=Register contains normal data. **01**=End of File. **02**=Extended address.

(continued)

Intel HEX File (continued)

Position	Description
10 - ?	**Data Bytes**: The following bytes are the actual data that will be burned into the EPROM. The data is represented as 2-digit hex values.
Last 2 characters	**Checksum**: The last two characters of the line are a checksum for the line. The checksum value is calculated by taking the two's complement of the sum of all the preceding data bytes, excluding the checksum byte itself and the colon at the beginning of the line.

Intel Hex format

```
:0B|00|0300|E4FF0FBF14FC7F14DFFE22 9F
:03|00|0000|02000E ED
:0C|00|0E00|787FE4F6D8FD758107020003 3E
```

- Breaking the second line into its components we have:
- **Record Length**: 03 (3 bytes of data)
  Address: 0000 (the 3 bytes will be stored at 0000, 0001, and 0002)
  Record Type: 00 (normal data)
  Data: 02, 00, 0E
  Checksum: ED
- Checksum: -(03+00+00+00+02+00+0E) = ED

Disassembly example

```
0000 00 02000E ED
000E 00 787F E4 F6 D8FD 758107 020003

  0000 02 000E LJMP 000E
  000E 78 7F   MOV R0,#7f
  0010 E4      CLR A
  0011 F6      MOV @R0,A
  0012 D8 FD   DJNZ R0,-3 ;jmp 0011
  0014 75 8107 MOV SP, #7 ;sp=sfr 81
  0017 02 0003 LJMP 3
```