EE 354
Notes on Hex format

The hex format used for programming flash memory dates back to the early days of microprocessors. The format now in use comes originally from Intel and is referred to as the Intel Hex-record format. This format takes binary data, such as from an object file, and encodes each byte as two ascii hexadecimal numbers. Thirty-two such bytes are encoded into a single record. The record contains address information and other information related to the type of data being encoded. It ends with a check sum so that the receiver can verify the data.

The contents of a single record are summarized in Table 1.

Field		**Size**	**Comment**
1	Start byte	1	An ASCII colon character **:**
2	Byte count	2	The number of bytes of data being sent
3	Address	4	A 16 bit address of where the data is to be loaded.
4	Record Type[1]	2	00, 01, or 02
5	Data	0 to 2n	Indicates 0 to n bytes of data or code. A record is 20h bytes long or less.
6	Checksum	2	The mod 256 twos complement sum of the values represented by all the pairs of characters in the record except the start code and checksum.

Table 1
The Intel Hex record format.

Note 1: There are three record types: 00 indicates a record containing data; 01 indicates a termination record; 10 indicates a segment base address record and is generally ignored.

Example
Here is an 8051 assembly program, a segment of the list file showing the op codes, and a copy of the hex file to be loaded into memory.

```
;Port 1
;This program increment port 1 forever
MAIN SEGMENT CODE
CSEG at 0000h
LJMP Start
RSEG MAIN
Start:mov P1, #0
Loop1:inc P1
      sjmp Loop1
END
```

Figure 1
A 8051 assembly program to increment port 1 forever.

```
LOC  OBJ            LINE     SOURCE

                       1      ;Port 1
                       2      ;This program increment port 1 forever
                       3      MAIN SEGMENT CODE
----                   4      CSEG at 0000h
0000 020000   F        5      LJMP Start
----                   6      RSEG MAIN
0000 759000            7      Start:mov P1, #0
0003 0590              8      Loop1:inc P1
0005 80FC              9      sjmp Loop1
                      10      END
```

Figure 2

A list file segment that corresponds the assembly program in Figure 1

```
:03000000020003F8
:0700030075900059080FCE0
:00000001FF
```

Figure 3

The Intel Hex format for the program in Figure 1