Intel
Hexadecimal Object File
Format Specification
Revision A, 1/6/88

1. Introduction

This document describes the hexadecimal object file format for the Intel 8-bit, 16-bit, and 32-bit microprocessors.   The hexadecimal format is suitable as input to PROM programmers or hardware emulators.

Hexadecimal object file format is a way of representing an absolute binary object file in ASCII. Because the file is in ASCII instead of binary, it is possible to store the file is non-binary medium such as paper-tape, punch cards, etc.; and the file can also be displayed on CRT terminals, line printers, etc.. The 8-bit hexadecimal object file format allows for the placement of code and data within the 16-bit linear address space of the Intel 8-bit processors. The 16-bit hexadecimal format allows for the 20-bit segmented address space of the Intel 16-bit processors. And the 32-bit format allows for the 32-bit linear address space of the Intel 32-bit processors.

The hexadecimal representation of binary is coded in ASCII alphanumeric characters.   For example, the 8-bit binary value 0011-1111 is 3F in hexadecimal.   To code this in ASCII, one 8-bit byte containing the ASCII code for the character '3' (0011-0011 or 033H) and one 8-bit byte containing the ASCII code for the character 'F' (0100-0110 or 046H) are required. For each byte value, the high-order hexadecimal digit is always the first digit of the pair of hexadecimal digits. This representation (ASCII hexadecimal) requires twice as many bytes as the binary representation.

A hexadecimal object file is blocked into records, each of which contains the record type, length, memory load address and checksum in addition to the data. There are currently six (6) different types of records that are defined, not all combinations of these records are meaningful, however. The record are:

·   Data Record (8-, 16-, or 32-bit formats)

·   End of File Record (8-, 16-, or 32-bit formats)

·   Extended Segment Address Record (16- or 32-bit formats)

·   Start Segment Address Record (16- or 32-bit formats)

·   Extended Linear Address Record (32-bit format only)

·   Start Linear Address Record (32-bit format only)

2.   General Record Format

```
+-----------------+--------------+------------+--------------+--------------+------------------+
|   RECORD  |  LOAD      |            |              |   INFO  |            |
|   MARK    |  RECLEN  | OFFSET  | RECTYP  |   or      | CHKSUM  |
|      ':'      |              |            |              |   DATA  |            |
+-----------------+--------------+------------+--------------+--------------+------------------+
      1-byte          1-byte        2-bytes       1-byte          n-bytes        1-byte
```

Each record begins with a RECORD MARK field containing 03AH, the ASCII code for the colon (':') character.

Each record has a RECLEN field which specifies the number of bytes of information or data which follows the RECTYP field of the record. Note that one data byte is represented by two ASCII characters. The maximum value of the RECLEN field is hexadecimal 'FF' or 255.

Each record has a LOAD OFFSET field which specifies the 16-bit starting load offset of the data bytes, therefore this field is only used for Data Records. In other records where this field is not used, it should be coded as four ASCII zero characters ('0000' or 030303030H).

Each record has a RECTYP field which specifies the record type of this record. The RECTYP field is used to interpret the remaining information within the record. The encoding for all the current record types are:

> '00'   Data Record
> '0l'   End of File Record
> '02'   Extended Segment Address Record
> '03'   Start Segment Address Record
> '04'   Extended Linear Address Record
> '05'   Start Linear Address Record

Each record has a variable length INFO/DATA field, it consists of zero or more bytes encoded as pairs of hexadecimal digits. The interpretation of this field depends on the RECTYP field.

Each record ends with a CHKSUM field that contains the ASCII hexadecimal representation of the two's complement of the 8-bit bytes that result from converting each pair of ASCII hexadecimal digits to one byte of binary, from and including the RECLEN field to and including the last byte of the INFO/DATA field. Therefore, the sum of all the ASCII pairs in a record after converting to binary, from the RECLEN field to and including the CHKSUM field, is zero.

3.   <u>Extended Linear Address Record</u>   (32-bit format only)

```
+----------------+--------------+------------+---------------+-------------+-----------------+
|   RECORD  |            |  LOAD  |            |          |            |
|   MARK    | RECLEN  | OFFSET | RECTYP  |  ULBA  | CHKSUM  |
|    ':'      |   '02'   | '0000' |   '04'   |          |            |
+----------------+--------------+------------+---------------+-------------+-----------------+
     1-byte        1-byte      2-bytes     1-byte        2-bytes      1-byte
```

The 32-bit Extended Linear Address Record is used to specify bits 16-31 of the Linear Base Address (LBA), where bits 0-15 of the LBA are zero. Bits 16-31 of the LBA are referred to as the Upper Linear Base Address (ULBA). The absolute memory address of a content byte in a subsequent Data Record is obtained by adding the LBA to an offset calculated by adding the LOAD OFFSET field of the containing Data Record to the index of the byte in the Data Record (0, 1, 2, ... n). This offset addition is done modulo 4G (i.e., 32-bits), ignoring any carry, so that offset wrap-around loading (from OFFFFFFFFH to OOOOOOOOOH) results in wrapping around from the end to the beginning of the 4G linear address defined by the LBA. The linear address at which a particular byte is loaded is calculated as:

          (LBA + DRLO + DRI) MOD 4G

where:

          DRLO is the LOAD OFFSET field of a Data Record.
          DRI is the data byte index within the Data Record.

When an Extended Linear Address Record defines the value of LBA, it may appear anywhere within a 32-bit hexadecimal object file. This value remains in effect until another Extended Linear Address Record is encountered. The LBA defaults to zero until an Extended Linear Address Record is encountered.

The contents of the individual fields within the record are:

RECORD MARK
          This field contains 03AH, the hexadecimal encoding of the ASCII colon (':') character.

RECLEN
          The field contains 03032H, the hexadecimal encoding of the ASCII characters '02', which is the length, in bytes, of the ULBA data information within this record.

LOAD OFFSET

       This field contains 030303030H, the hexadecimal encoding of the ASCII characters '0000', since this field is not used for this record.

RECTYP

       This field contains 03034H, the hexadecimal encoding of the ASCII character '04', which specifies the record type to be an Extended Linear Address Record.

ULBA

       This field contains four ASCII hexadecimal digits that specify the 16-bit Upper Linear Base Address value. The high-order byte is the 10th/llth character pair of the record. The low-order byte is the 12th/13th character pair of the record.

CHKSUM

       This field contains the check sum on the RECLEN, LOAD OFFSET, RECTYP, and ULBA fields.

4.   <u>Extended Segment Address Record</u> (16- or 32-bit formats)

```
+-----------------+--------------+------------+---------------+-------------+-----------------+
|    RECORD  |              |   LOAD  |               |             |             |
|    MARK    |  RECLEN  | OFFSET  |  RECTYP   |    USBA    |  CHKSUM   |
|     ':'    |   '02'   | '0000'  |   '02'    |            |           |
+-----------------+--------------+------------+---------------+-------------+-----------------+
     1-byte        1-byte      2-bytes    1-byte       2-bytes      1-byte
```

The 16-bit Extended Segment Address Record is used to specify bits 4-19 of the Segment Base Address (SBA), where bits 0-3 of the SBA are zero. Bits 4-19 of the SBA are referred to as the Upper Segment Base Address (USBA). The absolute memory address of a content byte in a subsequent Data Record is obtained by adding the SBA to an offset calculated by adding the LOAD OFFSET field of the containing Data Record to the index of the byte in the Data Record (0, 1, 2, ... n). This offset addition is done modulo 64K (i.e., 16-bits), ignoring any carry, so that offset wraparound loading (from OFFFFH to OOOOOH) results in wrapping around from the end to the beginning of the 64K segment defined by the SBA. The address at which a particular byte is loaded is calculated as:

       $SBA + ([DRLO + DRI] \ MOD \ 64K)$

where:

       DRLO is the LOAD OFFSET field of a Data Record.

DRI is the data byte index within the Data Record.

When an Extended Segment Address Record defines the value of SBA, it may appear anywhere within a 16-bit hexadecimal object file. This value remains in effect until another Extended Segment Address Record is encountered. The SBA defaults to zero until an Extended Segment Address Record is encountered.

The contents of the individual fields within the record are:

RECORD MARK

This field contains 03AH, the hexadecimal encoding of the ASCII colon (':') character.

RECLEN

The field contains 03032H, the hexadecimal encoding of the ASCII characters '02', which is the length, in bytes, of the USBA data information within this record.

LOAD OFFSET

This field contains 030303030H, the hexadecimal encoding of the ASCII characters '0000', since this field is not used for this record.

RECTYP

This field contains 03032H, the hexadecimal encoding of the ASCII character '02', which specifies the record type to be an Extended Segment Address Record.

USBA

This field contains four ASCII hexadecimal digits that specify the 16-bit Upper Segment Base Address value. The high-order byte is the 10th/11th character pair of the record. The low-order byte is the 12th/13th character pair of the record.

CHKSUM

This field contains the check sum on the RECLEN, LOAD OFFSET, RECTYP, and USBA fields.

5.   Data Record   (8-, 16-, or 32-bit formats)

```
+-----------------+--------------+------------+---------------+--------------+-----------------+
|  RECORD  |              | LOAD   |               |              |                 |
|  MARK    | RECLEN | OFFSET | RECTYP  |   DATA   | CHKSUM  |
|   ':'    |              |            |   '00'    |              |                 |
+-----------------+--------------+------------+---------------+--------------+-----------------+
    1-byte       1-byte     2-bytes    1-byte       n-bytes    1-byte
```

The Data Record provides a set of hexadecimal digits that represent the ASCII code for data bytes that make up a portion of a memory image. The method for calculating the absolute address (linear in the 8-bit and 32-bit case and segmented in the 16-bit case) for each byte of data is described in the discussions of the Extended Linear Address Record and the Extended Segment Address Record.

The contents of the individual fields within the record are:

RECORD MARK

This field contains 03AH, the hexadecimal encoding of the ASCII colon (':') character.

RECLEN

The field contains two ASCII hexadecimal digits that specify the number of data bytes in the record. The maximum value is 'FF' or 04646H (255 decimal).

LOAD OFFSET

This field contains four ASCII hexadecimal digits representing the offset from the LBA (see Extended Linear Address Record) or SBA (see Extended Segment Address Record) defining the address which the first byte of the data is to be placed.

RECTYP

This field contains 03030H, the hexadecimal encoding of the ASCII character '00', which specifies the record type to be a Data Record.

DATA

This field contains pairs of ASCII hexadecimal digits, one pair for each data byte.

CHKSUM

This field contains the check sum on the RECLEN, LOAD OFFSET, RECTYP, and DATA fields.

6.   Start Linear Address Record   (32-bit format only)

```
+------------------+---------------+-------------+----------------+--------------+-------------------+
|   RECORD   |               |  LOAD   |                |              |                   |
|   MARK    | RECLEN  | OFFSET | RECTYP   |   EIP    | CHKSUM    |
|    ':'       |   '04'     | '0000'  |   '05'      |              |                   |
+------------------+---------------+-------------+----------------+--------------+-------------------+
     1-byte          1-byte        2-bytes      1-byte         4-bytes       1-byte
```

The Start Linear Address Record is used to specify the execution start address for the object file. The value given is the 32-bit linear address for the EIP register. Note that this record only specifies the code address within the 32-bit linear address space of the 80386. If the code is to start execution in the real mode of the 80386, then the Start Segment Address Record should be used instead, since that record specifies both the CS and IP register contents necessary for real mode.

The Start Linear Address Record can appear anywhere in a 32-bit hexadecimal object file. If such a record is not present in a hexadecimal object file, a loader is free to assign a default start address.

The contents of the individual fields within the record are:

RECORD MARK

This field contains 03AH, the hexadecimal encoding of the ASCII colon (':') character.

RECLEN

The field contains 03034H, the hexadecimal encoding of the ASCII characters '04', which is the length, in bytes, of the EIP register content within this record.

LOAD OFFSET

This field contains 030303030H, the hexadecimal encoding of the ASCII characters '0000', since this field is not used for this record.

RECTYP

This field contains 03035H, the hexadecimal encoding of the ASCII character '05', which specifies the record type to be a Start Linear Address Record.

EIP

This field contains eight ASCII hexadecimal digits that specify the 32-bit EIP register contents. The high-order byte is the 10th/11th character pair.

CHKSUM

This field contains the check sum on the RECLEN, LOAD OFFSET, RECTYP, and EIP fields.

7.　　　　Start Segment Address Record　(16- or 32-bit formats)

```
+------------------+---------------+------------+---------------+--------------+------------------+
|    RECORD   |            |   LOAD   |          |          |              |
|    MARK     |  RECLEN  |  OFFSET  |  RECTYP  |  CS/IP   |  CHKSUM     |
|     ':'     |   '04'   |  '0000'  |   '03'   |          |              |
+------------------+---------------+------------+---------------+--------------+------------------+
      1-byte        1-byte      2-bytes     1-byte        4-bytes      1-byte
```

The Start Segment Address Record is used to specify the execution start address for the object file. The value given is the 20-bit segment address for the CS and IP registers. Note that this record only specifies the code address within the 20-bit segmented address space of the 8086/80186.

The Start Segment Address Record can appear anywhere in a 16-bit hexadecimal object file.   If such a record is not present in a hexadecimal object file, a loader is free to assign a default start address.

The contents of the individual fields within the record are:

RECORD MARK
> This field contains 03AH, the hexadecimal encoding of the ASCII colon (':') character.

RECLEN
> The field contains 03034H, the hexadecimal encoding of the ASCII characters '04', which is the length, in bytes, of the CS/IP register contents within this record.

LOAD OFFSET
> This field contains 030303030H, the hexadecimal encoding of the ASCII characters '0000', since this field is not used for this record.

RECTYP
> This field contains 03033H, the hexadecimal encoding of the ASCII character '03', which specifies the record type to be a Start Segment Address Record.

CS/IP
> This field contains eight ASCII hexadecimal digits that specify the 16-bit CS register and 16-bit IP register contents. The high-order byte of the CS register content is the 10th/11th character pair, the low-order byte is the 12th/13th character pair of the record. The high-order byte of the IP register content is the 14th/15th character pair, the low-order byte is the 16th/17th character pair of the record.

CHKSUM

> This field contains the check sum on the RECLEN, LOAD OFFSET, RECTYP, and CS/IP fields.

8.   <u>End of File Record</u>   (8-, 16-, or 32-bit formats)

```
+-----------------+--------------+------------+---------------+-----------------+
|   RECORD  |              |  LOAD  |           |                 |
|   MARK    | RECLEN  | OFFSET  | RECTYP  |   CHKSUM  |
|     ':'      |   '00'     |  '0000'  |    '01'    |                 |
+-----------------+--------------+------------+---------------+-----------------+
     1-byte         1-byte       2-bytes      1-byte         1-byte
```

The End of File Record specifies the end of the hexadecimal object file.

The contents of the individual fields within the record are:

RECORD MARK

> This field contains 03AH, the hexadecimal encoding of the ASCII colon (':') character.

RECLEN

> The field contains 03030H, the hexadecimal encoding of the ASCII characters '00'. Since this record does not contain any INFO/DATA bytes, the length is zero.

LOAD OFFSET

> This field contains 030303030H, the hexadecimal encoding of the ASCII characters '0000', since this field is not used for this record.

RECTYP

> This field contains 03031H, the hexadecimal encoding of the ASCII character '01', which specifies the record type to be an End of File Record.

CHKSUM

> This field contains the check sum an the RECLEN, LOAD OFFSET, and RECTYP fields. Since all the fields are static, the check sum can also be calculated statically, and the value is 04646H, the hexadecimal encoding of the ASCII characters 'FF'.