ソースです。

・見たままです。
・8bit (64kbyte 以内) 専用。
・コピペミスがあるかも。エラーがでたら直してください。
・ライセンスは NYSL で。

```
#include <stdio.h>
#include <stdlib.h>

#define LINEMAX 0x100

/* pseudo memory space */
unsigned char memory[0x10000] ; /* 64Kbyte Space */

/* hex to decimal converter */
static int hextoint(char a)
{
  if ((a >= '0') && (a <='9')) return a - '0' ;
  if ((a >= 'A') && (a <='F')) return a - 'A' + 0x0A ;
  if ((a >= 'a') && (a <='f')) return a - 'a' + 0x0A ;
  return 0 ;
}

static int hex2toint(char *a)
{
  return (hextoint(a[0]) * 0x10 +  hextoint(a[1])) ;
}

static int hex4toint(char *a)
{
  return (hextoint(a[0]) * 0x1000 +   hextoint(a[1]) * 0x100 + hextoint(a[2])*0x10 +  hextoint(a[3]))
;
}

void main(int argc,char *argv[])
{
  char line[LINEMAX] ;
  unsigned int memtop,memend,i ;
  FILE *fpi,*fpo ;

  memend = 0x0000 ; /* end address of output coe file */

  printf("Intel HEX to Xilinx memory initfile converter (8bit)\n",argv[0]) ;


  /* help message */
  if (argc < 3) {
    fprintf(stderr,"%s [infile.hex] [outfile.coe]\n",argv[0]) ;
    exit(-1) ;
  }

  /* open input file */
  if ((fpi = fopen(argv[1],"r")) == NULL) {
    fprintf(stderr,"Can't open input file [%s]\n",argv[1]) ;
    exit(-1) ;
  }

  /* read hex file and distribute bits */
  while(fgets(line,LINEMAX,fpi) != NULL) {
    unsigned int reclen,recofs,rectyp ;

    /* +0123456789A           */
    /*  :LL0000TTDD...DDCC[CR] */
    /*  LL   - Data Count */
    /*  0000 - Offset Address */
    /*  TT   - Record Type, 00 : DATA , 01 : END , ignore other types */

    /* [0] is always ':' */
    if (line[0] != ':') continue ;

    /* [1,2] is record length */
    reclen = hex2toint(&line[1]) ;
    if (reclen == 0) continue ;

    /* [3,4,5,6] is record offset */
    recofs = hex4toint(&line[3]) ;

    /* [7,8] is recore type */
    rectyp = hex2toint(&line[7]) ;
    if (rectyp != 0) continue ; /* 01 is END but ignore here */

    /* write one record to pseudo memory (no error check :-) */
    for (i = 0;i < reclen;i++) {
      int data ;
      data = hex2toint(&line[9+i*2]) ;
      memory[recofs] = (unsigned char)data & 0xFF;
      if (recofs > memend) memend = recofs ;
      recofs ++ ;
    }
  }
  fclose(fpi) ;

#define DEBUG
#ifdef DEBUG
  /* dump memory map (for DEBUG) */
  printf("Memory Content is ....\n") ;
  for (i = 0;i <= memend;i += 16) {
    unsigned int j ;
    printf("%04X :",i) ;
    for (j = i;(j < i + 16) && (j <= memend);j++) {
      printf(" %02X",memory[j]) ;
    }
    printf("\n") ;
  }
#endif

  /* open output file */
  if ((fpo = fopen(argv[2],"w"))== NULL) {
    fprintf(stderr,"Can't open output file [%s]\n",argv[2]) ;
    exit(-1) ;
  }

  /* Output .coe format */
  fprintf(fpo,"MEMORY_INITIALIZATION_RADIX= 16 ;\n") ;
  fprintf(fpo,"MEMORY_INITIALIZATION_VECTOR= ") ;
  for (i = 0;i <= memend;i += 16) {
    unsigned int j ;
    fprintf(fpo,"\n") ;
    for (j = i;j < i + 16;j++) {
       fprintf(fpo,"%02X",memory[j]) ;
      if (j == memend) break ;
      fprintf(fpo,",") ;
    }
  }
  fprintf(fpo,";") ;
  fclose(fpo) ;
}
```