EECS 363: Digital Filtering
Lab 4 - 2/12/2017
Karan Shah

Code:

```
/****************************************************************************/
/* LNKX.CMD - COMMAND FILE FOR LINKING C PROGRAMS IN LARGE/HUGE MEMORY MODEL */
/*                                                                          */
/* Usage:                                                                   */
/*  cl55 <src files> -z -o<out file> -m<map file> lnkx.cmd -l<RTS library>  */
/*                                                                          */
/* Description: This file is a sample command file that can be used for     */
/*              linking programs built with the C Compiler.  Use it as a    */
/*              guideline; you  may want to change the allocation scheme     */
/*              according to the size of your program and the memory layout  */
/*              of your target system.                                       */
/*                                                                          */
/*   Notes: (1) You must specify the directory in which <RTS library> is    */
/*              located.  Either add a "-i<directory>" line to this file     */
/*              file, or use the system environment variable C55X_C_DIR to   */
/*              specify a search path for the libraries.                     */
/*                                                                          */
/****************************************************************************/

-stack    0x2000      /* Primary stack size   */
-sysstack 0x1000      /* Secondary stack size */
-heap     0x2000      /* Heap area size       */

-c                    /* Use C linking conventions: auto-init vars at runtime */
-u _Reset             /* Force load of reset interrupt handler                */

/* SPECIFY THE SYSTEM MEMORY MAP */

MEMORY
{
 PAGE 0:  /* ---- Unified Program/Data Address Space ---- */

  MMR    (RWIX): origin = 0x000000, length = 0x0000c0  /* MMRs */
  DARAM0 (RWIX): origin = 0x0000c0, length = 0x00ff40  /*  64KB - MMRs */
  SARAM0 (RWIX): origin = 0x010000, length = 0x010000  /*  64KB */
  SARAM1 (RWIX): origin = 0x020000, length = 0x020000  /* 128KB */
  SARAM2 (RWIX): origin = 0x040000, length = 0x00FE00  /*  64KB */
  VECS   (RWIX): origin = 0x04FE00, length = 0x000200  /*  512B */
  PDROM  (RIX): origin = 0xff8000, length = 0x008000  /*  32KB */

 PAGE 2:  /* -------- 64K-word I/O Address Space -------- */

  IOPORT (RWI) : origin = 0x000000, length = 0x020000
}

/* SPECIFY THE SECTIONS ALLOCATION INTO MEMORY */
```

```
SECTIONS
{
    .text      >> SARAM1|SARAM2|SARAM0   /* Code                      */

    /* Both stacks must be on same physical memory page          */
    .stack     >  DARAM0                  /* Primary system stack      */
    .sysstack  >  DARAM0                  /* Secondary system stack    */

    .data      >> DARAM0|SARAM0|SARAM1   /* Initialized vars          */
    .bss       >> DARAM0|SARAM0|SARAM1   /* Global & static vars      */
    .const     >> DARAM0|SARAM0|SARAM1   /* Constant data             */
    .sysmem    >  DARAM0|SARAM0|SARAM1   /* Dynamic memory (malloc)   */
    .switch    >  SARAM2                  /* Switch statement tables   */
    .cinit     >  SARAM2                  /* Auto-initialization tables */
    .pinit     >  SARAM2                  /* Initialization fn tables  */
    .cio       >  SARAM2                  /* C I/O buffers             */
    .args      >  SARAM2                  /* Arguments to main()       */
    .coeffs    >> DARAM0
    .dbuffer   >> DARAM0

    vectors    >  VECS                    /* Interrupt vectors         */

    .ioport    >  IOPORT PAGE 2           /* Global & static ioport vars */
}
```

<u>Output (In Q15 format):</u>

```
4096
8192
12288
16384
16384
16384
16384
16384
16384
16384
16384
16384
16384
16384
16384
16384
12288
8192
4096
0
```