```
/************************************************************************
 *
 *  File:   flp_func
 *
 *  Purpose:      For Erase suspend and resume
 *
 *  Author: K.Narayanan
 *
 *  Date:
 *  Assumptions  :USE NO CNTRS IN THIS MODULE FOR PM OVERLAY SWAPPING
 *************************************************************************/
.module/seg = inthigh_pm  flp_func;

/* --------------------------------------------------------------------- */
/* Include Files  */
/* ------------   */
#include "../stdinc/def2181.h"
#include "../stdinc/globid.h"

.entry Delay20usec;
.entry WriteSusResBytesToFlash;
/*=======================================================================
 * Delay20usec:
 *    Args: none
 *    Ret:  Nothing
 *    Bugs:
 *-----------------------------------------------------------------------*/
Delay20usec:
     ar = 20;
     af = pass ar;
MulOf1usec:
     ar=26;
OneusL:
     ar = ar - 1;
     if gt jump OneusL;

     af = af -1;
     if gt jump MulOf1usec;
rts;
/*=======================================================================
 * WriteSusResBytesToFlash - Write bytes to flash prom thru BDMA
 *    Args: si: data ptr, my1: cnt, mr0: Addr Lo, mr1: Addr Hi.
 *    Ret:  Nothing
 *    Bugs:
 *-----------------------------------------------------------------------*/
WriteSusResBytesToFlash:

     dm (BDMA_Internal_Address) = si; /* store int mem data addr */

     dm (BDMA_External_Address) = mr0; /* only lower 14 bits valid */

     sr  = lshift mr0 by -6 (lo); /* b15..14 as b9..8 of page num */
     ay1 = 0x0300;
     af  = sr0 and ay1;
                         /* mr1 = Ext Addr Hi */
     sr  = lshift mr1 by 10 (lo); /* b15..10 of page num */
     ar  = sr0 or af;        /* combine b15..10 & b9..8 */
     ay1 = 0x0007;                 /* to store LSB into boot mem */
     ar  = ar or ay1;
     dm (BDMA_Control) = ar;
```

```
        dm (BDMA_Word_Count) = my1;  /* write xfer cnt and init BDMA xfer */


WriteSusResBytesWaitL:
!     idle;                      /* no forground activity */
      ar = dm (BDMA_Word_Count);
      ar = pass ar;
      if ne jump WriteSusResBytesWaitL;
rts;                     /* byte xfer over */


/* ---------------------------------------------------------------------- */
.endmod;          /* flp_func */
/* ---------------------------------------------------------------------- */
```