Sample Driver 5

Introduction

This chapter contains a sample TargetFFS-NAND driver for an 8MB device with an 8-bit interface. This driver can be used with most NAND devices simply by modifying the BLOCK_SIZE and NUM_BLOCKS definitions and by supplying equivalent code to read the RDY input and control the ALE, CE, CLE, and WP outputs. ECC encoding and decoding is done by calling the TargetFFS-NAND EncodeEcc() and DecodeEcc() routines, but these calls could be substituted with accesses to special registers if hardware ECC support is provided.

8MB Volume with 8-bit Interface

```
/*****************************************************************************/
/*                                                                           */
/*   Module:  nanddrvr.c                                                     */
/*   Version: 2005.0                                                         */
/*   Purpose: NAND Flash File System Driver                                  */
/*                                                                           */
/*****************************************************************************/
#include "../posix.h"
#if INC_NAND_FS
#include "../include/libc/string.h"
#include "../include/targetos.h"
#include "../include/sys.h"
#include "../include/fsprivate.h"

#include "xp860t.h"
#include "mpc860.h"

/*****************************************************************************/
/* Symbol Definitions                                                        */
/*****************************************************************************/
#define BLOCK_SIZE      (8 * 1024)
#define NUM_BLOCKS      1024
#define PAGE_SIZE       512

/*****************************************************************************/
/* Global Variable Definitions                                               */
/*****************************************************************************/
static int ExtraBytes;  /* TRUE iff chip is set to access extra bytes */

/*****************************************************************************/
/* Function Prototypes                                                       */
/*****************************************************************************/
void perror(const char *s);
```

```
void SysWait50ms(void);

/*****************************************************************************/
/* Local Function Definitions                                                */
/*****************************************************************************/

/*****************************************************************************/
/* page_erased: Check if page's data bytes and extra bytes are erased  */
/*                                                                           */
/*      Inputs: addr = the page address                                      */
/*              vol = driver's volume pointer                                */
/*                                                                           */
/*     Returns: TRUE if erased, FALSE if any bit is not 1                   */
/*                                                                           */
/*****************************************************************************/
static int page_erased(ui32 addr, void *vol)
{
  int n = PAGE_SIZE + 16;

  /*-------------------------------------------------------------------------*/
  /* Clear any pre-existing interrupt and begin access.                      */
  /*-------------------------------------------------------------------------*/
  MPC860_SIPEND = IRQ3;
  lowerCE();

  /*-------------------------------------------------------------------------*/
  /* Send command to start reading at first half of NAND data page.    */
  /*-------------------------------------------------------------------------*/
  raiseCLE();
  NandPort = 0x00;  /* read command */
  ExtraBytes = FALSE;
  lowerCLE();

  /*-------------------------------------------------------------------------*/
  /* Send address as three separate bytes.                                   */
  /*-------------------------------------------------------------------------*/
  raiseALE();
  NandPort = 0;
  NandPort = (ui8)(addr >> 9);
  NandPort = (ui8)(addr >> 17);
  lowerALE();

  /*-------------------------------------------------------------------------*/
  /* Wait until device is ready.                                             */
  /*-------------------------------------------------------------------------*/
  while ((MPC860_SIPEND & IRQ3) == FALSE) ;

  /*-------------------------------------------------------------------------*/
  /* Check one byte at a time.                                               */
  /*-------------------------------------------------------------------------*/
  do
    if (NandPort != 0xFF)
    {
      raiseCE();
      return FALSE;
    }
  while (--n);
```

```
  /*-------------------------------------------------------------------*/
  /* End device access and return success.                             */
  /*-------------------------------------------------------------------*/
  raiseCE();
  return TRUE;
}

/*********************************************************************/
/*  write_page: ECC encode and write one NAND flash page             */
/*                                                                   */
/*      Inputs: buffer = pointer to buffer containing data to write  */
/*              type = page type flag                                */
/*              addr = the page address                              */
/*              wear = wear count of this page's block               */
/*              vol = driver's volume pointer                        */
/*                                                                   */
/*     Returns: 0 for success, -1 for chip error, 1 for verify error */
/*                                                                   */
/*********************************************************************/
static int write_page(void *buffer, ui32 addr, ui32 type, ui32 wear,
                      void *vol)
{
  int n;
  ui8 *dst, ecc[10], status;

  PfAssert(page_erased(addr, vol));

  /*-------------------------------------------------------------------*/
  /* Ensure the NAND address pointer is back to the data page.         */
  /*-------------------------------------------------------------------*/
  if (ExtraBytes)
  {
    /*-----------------------------------------------------------------*/
    /* Clear any pre-existing interrupt and begin access.              */
    /*-----------------------------------------------------------------*/
    MPC860_SIPEND = IRQ3;
    lowerCE();

    /*-----------------------------------------------------------------*/
    /* Send command to start reading at first half of NAND data page.  */
    /*-----------------------------------------------------------------*/
    raiseCLE();
    NandPort = 0x00;  /* read command */
    lowerCLE();

    /*-----------------------------------------------------------------*/
    /* Send address 0 as three separate bytes.                         */
    /*-----------------------------------------------------------------*/
    raiseALE();
    NandPort = 0;
    NandPort = 0;
    NandPort = 0;
    lowerALE();

    /*-----------------------------------------------------------------*/
    /* Wait until device is ready.                                     */
```

```
  /*-------------------------------------------------------------------*/
  while ((MPC860_SIPEND & IRQ3) == FALSE) ;

  /*-------------------------------------------------------------------*/
  /* Read one byte and then end the device access.                     */
  /*-------------------------------------------------------------------*/
  NandPort;
  raiseCE();

  /*-------------------------------------------------------------------*/
  /* Clear flag since NAND pointer is reset to start of data page.     */
  /*-------------------------------------------------------------------*/
  ExtraBytes = FALSE;
}

/*---------------------------------------------------------------------*/
/* Compute page's ECC encoding.                                        */
/*---------------------------------------------------------------------*/
EncodeEcc(buffer, ecc);

/*---------------------------------------------------------------------*/
/* Clear any pre-existing interrupt and begin write access.            */
/*---------------------------------------------------------------------*/
MPC860_SIPEND = IRQ3;
lowerCE();

/*---------------------------------------------------------------------*/
/* Send Serial Data Input command.                                     */
/*---------------------------------------------------------------------*/
raiseCLE();
NandPort = 0x80;  /* serial data input command */
lowerCLE();

/*---------------------------------------------------------------------*/
/* Send address as three separate bytes.                               */
/*---------------------------------------------------------------------*/
raiseALE();
NandPort = 0;
NandPort = (ui8)(addr >> 9);
NandPort = (ui8)(addr >> 17);
lowerALE();

/*---------------------------------------------------------------------*/
/* Write the data portion.                                             */
/*---------------------------------------------------------------------*/
for (dst = buffer, n = 0; n < PAGE_SIZE; ++n)
  NandPort = *dst++;

/*---------------------------------------------------------------------*/
/* Write the 10-byte ECC portion.                                      */
/*---------------------------------------------------------------------*/
for (n = 0; n < 10; ++n)
  NandPort = ecc[n];

/*---------------------------------------------------------------------*/
/* Write the four type bytes.                                          */
/*---------------------------------------------------------------------*/
```

```
NandPort = (ui8)(type >> 24);
NandPort = (ui8)(type >> 16);
NandPort = (ui8)(type >> 8);
NandPort = (ui8)type;

/*-------------------------------------------------------------------*/
/* Send Auto Program command.                                        */
/*-------------------------------------------------------------------*/
raiseCLE();
NandPort = 0x10;  /* auto program command */
lowerCLE();

/*-------------------------------------------------------------------*/
/* Wait until device is ready.                                       */
/*-------------------------------------------------------------------*/
while ((MPC860_SIPEND & IRQ3) == FALSE) ;

/*-------------------------------------------------------------------*/
/* Send Status Read command.                                         */
/*-------------------------------------------------------------------*/
raiseCLE();
NandPort = 0x70;  /* status read command */
lowerCLE();

/*-------------------------------------------------------------------*/
/* Read device status and end write access.                          */
/*-------------------------------------------------------------------*/
status = NandPort;
raiseCE();

/*-------------------------------------------------------------------*/
/* Return -1 if chip reports program error.                          */
/*-------------------------------------------------------------------*/
if (status != 0xC0)
  return -1;

/*-------------------------------------------------------------------*/
/* Clear any previous interrupt and begin read-verify access.        */
/*-------------------------------------------------------------------*/
MPC860_SIPEND = IRQ3;
lowerCE();

/*-------------------------------------------------------------------*/
/* Send command to start reading at first half of NAND data page.    */
/*-------------------------------------------------------------------*/
raiseCLE();
NandPort = 0x00;  /* read command */
lowerCLE();

/*-------------------------------------------------------------------*/
/* Send address as three separate bytes.                             */
/*-------------------------------------------------------------------*/
raiseALE();
NandPort = 0;
NandPort = (ui8)(addr >> 9);
NandPort = (ui8)(addr >> 17);
lowerALE();
```

```
  /*-------------------------------------------------------------------*/
  /* Wait until device is ready.                                       */
  /*-------------------------------------------------------------------*/
  while ((MPC860_SIPEND & IRQ3) == FALSE) ;

  /*-------------------------------------------------------------------*/
  /* Read-verify the data portion.                                     */
  /*-------------------------------------------------------------------*/
  for (dst = buffer, n = 0; n < PAGE_SIZE; ++n)
    if (*dst++ != NandPort)
    {
      raiseCE();
      return 1;
    }

  /*-------------------------------------------------------------------*/
  /* Read-verify the 10-byte ECC portion.                              */
  /*-------------------------------------------------------------------*/
  for (n = 0; n < 10; ++n)
    if (NandPort != ecc[n])
    {
      raiseCE();
      return 1;
    }

  /*-------------------------------------------------------------------*/
  /* End device access and return success.                             */
  /*-------------------------------------------------------------------*/
  raiseCE();
  return 0;
}

/*********************************************************************/
/*  write_type: Write page's type value                              */
/*                                                                   */
/*      Inputs: addr = the page address                              */
/*              type = page type (0xFFFF0000)                        */
/*              vol = driver's volume pointer                        */
/*                                                                   */
/*        Note: Updates four-byte type flag written by wr_page()     */
/*                                                                   */
/*********************************************************************/
static int write_type(ui32 addr, ui32 type, void *vol)
{
  ui8 status;

  /*-------------------------------------------------------------------*/
  /* Ensure the NAND address pointer is set to the extra bytes.        */
  /*-------------------------------------------------------------------*/
  if (ExtraBytes == FALSE)
  {
    /*-----------------------------------------------------------------*/
    /* Clear any pre-existing interrupt and begin access.              */
    /*-----------------------------------------------------------------*/
    MPC860_SIPEND = IRQ3;
    lowerCE();
```

```
  /*-------------------------------------------------------------------*/
  /* Send command to start reading at the NAND extra bytes.            */
  /*-------------------------------------------------------------------*/
  raiseCLE();
  NandPort = 0x50;  /* read command */
  lowerCLE();

  /*-------------------------------------------------------------------*/
  /* Send address 0 as three separate bytes.                           */
  /*-------------------------------------------------------------------*/
  raiseALE();
  NandPort = 0;
  NandPort = 0;
  NandPort = 0;
  lowerALE();

  /*-------------------------------------------------------------------*/
  /* Wait until device is ready.                                       */
  /*-------------------------------------------------------------------*/
  while ((MPC860_SIPEND & IRQ3) == FALSE) ;

  /*-------------------------------------------------------------------*/
  /* Read one byte and then end the device access.                     */
  /*-------------------------------------------------------------------*/
  NandPort;
  raiseCE();

  /*-------------------------------------------------------------------*/
  /* Set flag since NAND pointer is set to the extra bytes.            */
  /*-------------------------------------------------------------------*/
  ExtraBytes = TRUE;
}

/*-------------------------------------------------------------------*/
/* Clear any pre-existing interrupt and begin write access.          */
/*-------------------------------------------------------------------*/
MPC860_SIPEND = IRQ3;
lowerCE();

/*-------------------------------------------------------------------*/
/* Send Serial Data Input command.                                   */
/*-------------------------------------------------------------------*/
raiseCLE();
NandPort = 0x80;  /* serial data input command */
lowerCLE();

/*-------------------------------------------------------------------*/
/* Send address as three separate bytes.                             */
/*-------------------------------------------------------------------*/
raiseALE();
NandPort = 10;    /* start just past ECC bytes */
NandPort = (ui8)(addr >> 9);
NandPort = (ui8)(addr >> 17);
lowerALE();

/*-------------------------------------------------------------------*/
```

```
  /* Write the four type bytes.                                          */
  /*---------------------------------------------------------------------*/
  NandPort = (ui8)(type >> 24);
  NandPort = (ui8)(type >> 16);
  NandPort = (ui8)(type >> 8);
  NandPort = (ui8)type;

  /*---------------------------------------------------------------------*/
  /* Send Auto Program command.                                          */
  /*---------------------------------------------------------------------*/
  raiseCLE();
  NandPort = 0x10;  /* auto program command */
  lowerCLE();

  /*---------------------------------------------------------------------*/
  /* Wait until device is ready.                                         */
  /*---------------------------------------------------------------------*/
  while ((MPC860_SIPEND & IRQ3) == FALSE) ;

  /*---------------------------------------------------------------------*/
  /* Send Status Read command.                                           */
  /*---------------------------------------------------------------------*/
  raiseCLE();
  NandPort = 0x70;  /* status read command */
  lowerCLE();

  /*---------------------------------------------------------------------*/
  /* Read device status and end write access.                            */
  /*---------------------------------------------------------------------*/
  status = NandPort;
  raiseCE();

  /*---------------------------------------------------------------------*/
  /* Return -1 if chip reports program error. Else return success.       */
  /*---------------------------------------------------------------------*/
  if (status != 0xC0)
    return -1;
  else
    return 0;
}

/***********************************************************************/
/*   read_type: Read page's write_page() type value                    */
/*                                                                     */
/*      Inputs: addr = the page address                                */
/*              vol = driver's volume pointer                          */
/*                                                                     */
/*     Returns: The four byte type value written during write_page()   */
/*              and possibly updated by write_type().                  */
/*                                                                     */
/***********************************************************************/
static ui32 read_type(ui32 addr, void *vol)
{
  ui32 type;

  /*---------------------------------------------------------------------*/
  /* Clear any pre-existing interrupt and begin access.                  */
```

```
  /*-------------------------------------------------------------------*/
  MPC860_SIPEND = IRQ3;
  lowerCE();

  /*-------------------------------------------------------------------*/
  /* Send read extra bytes command and set flag.                       */
  /*-------------------------------------------------------------------*/
  raiseCLE();
  NandPort = 0x50;  /* read extra bytes command */
  ExtraBytes = TRUE;
  lowerCLE();

  /*-------------------------------------------------------------------*/
  /* Send address as three separate bytes.                             */
  /*-------------------------------------------------------------------*/
  raiseALE();
  NandPort = 10;    /* start just past ECC bytes */
  NandPort = (ui8)(addr >> 9);
  NandPort = (ui8)(addr >> 17);
  lowerALE();

  /*-------------------------------------------------------------------*/
  /* Wait until device is ready.                                       */
  /*-------------------------------------------------------------------*/
  while ((MPC860_SIPEND & IRQ3) == FALSE) ;

  /*-------------------------------------------------------------------*/
  /* Read the four type bytes.                                         */
  /*-------------------------------------------------------------------*/
  type = NandPort;
  type = (type << 8) | NandPort;
  type = (type << 8) | NandPort;
  type = (type << 8) | NandPort;

  /*-------------------------------------------------------------------*/
  /* End the device access and return the coded type value.            */
  /*-------------------------------------------------------------------*/
  raiseCE();
  return type;
}

/*********************************************************************/
/*   read_page: Read and ECC correct one NAND flash page             */
/*                                                                   */
/*      Inputs: buffer = pointer to buffer to copy data to           */
/*              addr = the page address                              */
/*              wear = wear count of this page's block               */
/*              vol = driver's volume pointer                        */
/*                                                                   */
/*     Returns: 0 for success, -1 for uncorrectable error            */
/*                                                                   */
/*********************************************************************/
static int read_page(void *buffer, ui32 addr, ui32 wear, void *vol)
{
  int n;
  ui8 *dst = buffer;
  ui8 ecc[10];
```

```
  /*-------------------------------------------------------------------*/
  /* Clear any pre-existing interrupt and begin access.                */
  /*-------------------------------------------------------------------*/
  MPC860_SIPEND = IRQ3;
  lowerCE();

  /*-------------------------------------------------------------------*/
  /* Send command to start reading at first half of NAND data page.    */
  /*-------------------------------------------------------------------*/
  raiseCLE();
  NandPort = 0x00;  /* read command */
  ExtraBytes = FALSE;
  lowerCLE();

  /*-------------------------------------------------------------------*/
  /* Send address as three separate bytes.                             */
  /*-------------------------------------------------------------------*/
  raiseALE();
  NandPort = 0;
  NandPort = (ui8)(addr >> 9);
  NandPort = (ui8)(addr >> 17);
  lowerALE();

  /*-------------------------------------------------------------------*/
  /* Wait until device is ready.                                       */
  /*-------------------------------------------------------------------*/
  while ((MPC860_SIPEND & IRQ3) == FALSE) ;

  /*-------------------------------------------------------------------*/
  /* Read the data portion.                                            */
  /*-------------------------------------------------------------------*/
  for (n = 0; n < PAGE_SIZE; ++n)
    *dst++ = NandPort;

  /*-------------------------------------------------------------------*/
  /* Read the 10-byte ECC portion.                                     */
  /*-------------------------------------------------------------------*/
  for (n = 0; n < 10; ++n)
    ecc[n] = NandPort;

  /*-------------------------------------------------------------------*/
  /* End the device access.                                            */
  /*-------------------------------------------------------------------*/
  raiseCE();

  /*-------------------------------------------------------------------*/
  /* Perform ECC decoding, return error if uncorrectable error.       */
  /*-------------------------------------------------------------------*/
  if (DecodeEcc(buffer, ecc))
    return -1;
  else
    return 0;
}

/*********************************************************************/
/* erase_block: Erase entire flash block                             */
```

```
/*                                                                      */
/*      Inputs: addr = base address of block to be erased               */
/*              vol = driver's volume pointer                           */
/*                                                                      */
/*     Returns: 0 on success, -1 on failure                             */
/*                                                                      */
/************************************************************************/
static int erase_block(ui32 addr, void *vol)
{
  ui8 status;

  /*--------------------------------------------------------------------*/
  /* Clear any pre-existing interrupt and begin access.                 */
  /*--------------------------------------------------------------------*/
  MPC860_SIPEND = IRQ3;
  lowerCE();

  /*--------------------------------------------------------------------*/
  /* Send Auto Block Erase Setup command.                               */
  /*--------------------------------------------------------------------*/
  raiseCLE();
  NandPort = 0x60;  /* auto block erase setup command */
  lowerCLE();

  /*--------------------------------------------------------------------*/
  /* Send address as two separate bytes.                                */
  /*--------------------------------------------------------------------*/
  raiseALE();
  NandPort = (ui8)(addr >> 9);
  NandPort = (ui8)(addr >> 17);
  lowerALE();

  /*--------------------------------------------------------------------*/
  /* Send Erase Start command.                                          */
  /*--------------------------------------------------------------------*/
  raiseCLE();
  NandPort = 0xD0;  /* erase start command */
  lowerCLE();

  /*--------------------------------------------------------------------*/
  /* Wait until device is ready.                                        */
  /*--------------------------------------------------------------------*/
  while ((MPC860_SIPEND & IRQ3) == FALSE) ;

  /*--------------------------------------------------------------------*/
  /* Send Status Read command.                                          */
  /*--------------------------------------------------------------------*/
  raiseCLE();
  NandPort = 0x70;  /* status read command */
  lowerCLE();

  /*--------------------------------------------------------------------*/
  /* Read device status.                                                */
  /*--------------------------------------------------------------------*/
  status = NandPort;

  /*--------------------------------------------------------------------*/
```

```
  /* End device access and return success.                              */
  /*--------------------------------------------------------------------*/
  raiseCE();
  return status == 0xC0 ? 0 : -1;
}

/**********************************************************************/
/* Global Function Definitions                                        */
/**********************************************************************/

/**********************************************************************/
/* NandDriverModule: Flash driver's module list entry                 */
/*                                                                    */
/*      Inputs: req = module request code                             */
/*              ... = additional parameters specific to request       */
/*                                                                    */
/**********************************************************************/
void *NandDriverModule(int req, ...)
{
  FfsVol vol;

  /*--------------------------------------------------------------------*/
  /* Implement only the initialization request.                         */
  /*--------------------------------------------------------------------*/
  if (req == kInitMod)
  {
    /*------------------------------------------------------------------*/
    /* Reset the NAND device.                                           */
    /*------------------------------------------------------------------*/
    lowerCE();
    raiseCLE();
    NandPort = 0xFF;    /* reset command */
    lowerCLE();
    raiseCE();

    /*------------------------------------------------------------------*/
    /* Wait until the device is ready.                                  */
    /*------------------------------------------------------------------*/
    MPC860_SIEL &= ~ED3; /* -> level mode */
    SysWait50ms();
    while ((MPC860_SIPEND & IRQ3) == FALSE) ;
    MPC860_SIEL |= ED3;  /* -> edge mode */

    /*------------------------------------------------------------------*/
    /* Remove write protection, allowing the device to be programmed.  */
    /*------------------------------------------------------------------*/
    raiseWP();

    /*------------------------------------------------------------------*/
    /* Register flash NAND volume with TargetFFS.                       */
    /*------------------------------------------------------------------*/
    vol.name = "nand";
    vol.type = FFS_NAND;
    vol.flags = 0;
    vol.page_size = PAGE_SIZE;
    vol.block_size = BLOCK_SIZE;
    vol.num_blocks = NUM_BLOCKS;
```

```
    vol.mem_base = 0;
    vol.vol = NULL;
    vol.driver.nand.max_bad_blocks = 10;   /* from data sheet */
    vol.driver.nand.write_page = write_page;
    vol.driver.nand.write_type = write_type;
    vol.driver.nand.read_page = read_page;
    vol.driver.nand.read_type = read_type;
    vol.driver.nand.page_erased = page_erased;
    vol.driver.nand.erase_block = erase_block;
    if (FfsAddVol(&vol))
      perror("FfsAddVol() failed");
  }

  return NULL;
}
#endif /* INC_NAND_FS */
```

Intentionally left blank.