UNIVERSITY OF MICROCHIP
UOFM
MASTERS 2007

11072 DEE

Data EEPROM Emulation for PIC18, PIC24 and dsPIC33F Flash Devices

Class Objective

- **When you finish this class you will:**
  - Know how AN1095 Data EE emulation works
  - Understand configuration options
  - Be able to implement AN1095

Agenda

- Nonvolatile memory options
- AN1095 DEE emulation algorithm
- Customization options
- Hands-on: Use AN1095 DEE emulation

Nonvolatile Memory Options

Nonvolatile Memory Options

- **What is Data EEPROM (DEE)?**
  - Nonvolatile
  - Word addressable programming and reading
  - Higher endurance
  - Lower data rates
  - Can be within MCU or external

Nonvolatile Memory Options

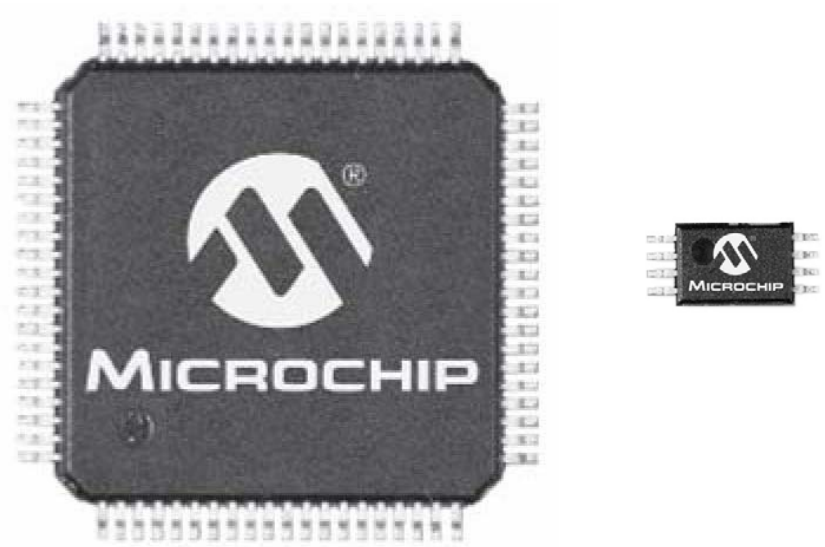
- **What kind of data uses DEE?**
  - Infrequently updated data
    - **Identification information**
    - **Operating parameters**
    - **Observed ranges (min and max)**
  - Smaller amounts of data
  - Lower speed access
  - Retained when power is removed

Nonvolatile Memory Options

- **PIC18J/PIC24/dsPIC33 devices have:**
  - Low cost
  - High performance
  - Wide range of peripherals
  - Self-writable program memory

Nonvolatile Memory Options

- **External - Serial EEPROM**
  - High endurance
  - Consumes I/O pins
  - Requires I²C™, SPI or External Memory Interface peripheral or equivalent
  - Requires additional component, board space



Nonvolatile Memory Options

- **Internal – Flash Program Memory**
  - Simplest to implement
  - Low endurance
  - Requires low-level algorithm implementation

Nonvolatile Memory Options

- **Data EE Emulation**
  - Internal to the microcontroller
    - **No extra components required**
    - **No I/O pins or special peripherals required**
  - Extended memory endurance
  - Implementation details are transparent to the application

Nonvolatile Memory Options

- **AN1095 DEE Emulation Goals**
  - Support a limited amount of data
  - Extend memory endurance
  - Require minimal overhead
  - Provide a simple application interface
  - Allow flexible application configuration

Nonvolatile Memory Options

- **Issues to address**
  - Flash architecture differences
  - Uneven address write distribution
  - Endurance maximization
  - Reset tolerance
  - CPU stall on program memory write

UNIVERSITY OF MICROCHIP
UofM
MASTERs 2007

How AN1095 DEE Emulation Works

How AN1095 DEE Emulation Works

- **Program Memory architecture**
  - Erase block size (page size): 512 instructions
  - Many “J” devices support single-word programming
  - Each location can be programmed only once per erase

How AN1095 DEE Emulation Works

- **Program Memory Architecture**
  - PIC18 data format

MSB <15:8>	LSB <7:0>
DEE Data (8 bits)	DEE Address

  - PIC24/dsPIC33 data format

MSW <31:16>		LSW <15:0>
Phantom Byte 0x00	DEE Address	DEE Data (16 bits)

How AN1095 DEE Emulation Works

- **A minimum of two Flash pages are required**
  - Flash pages are used sequentially
- **Page status information and erase/write count are held in the first locations of every page**

How AN1095 DEE Emulation Works

- **Address/data pairs are written to the Flash page**
  - Flash pages are filled sequentially
  - No write occurs if current value matches
- **When a page is filled**
  - Most recent data copied to the next page (“packed”)

How AN1095 DEE Emulation Works - Initialization

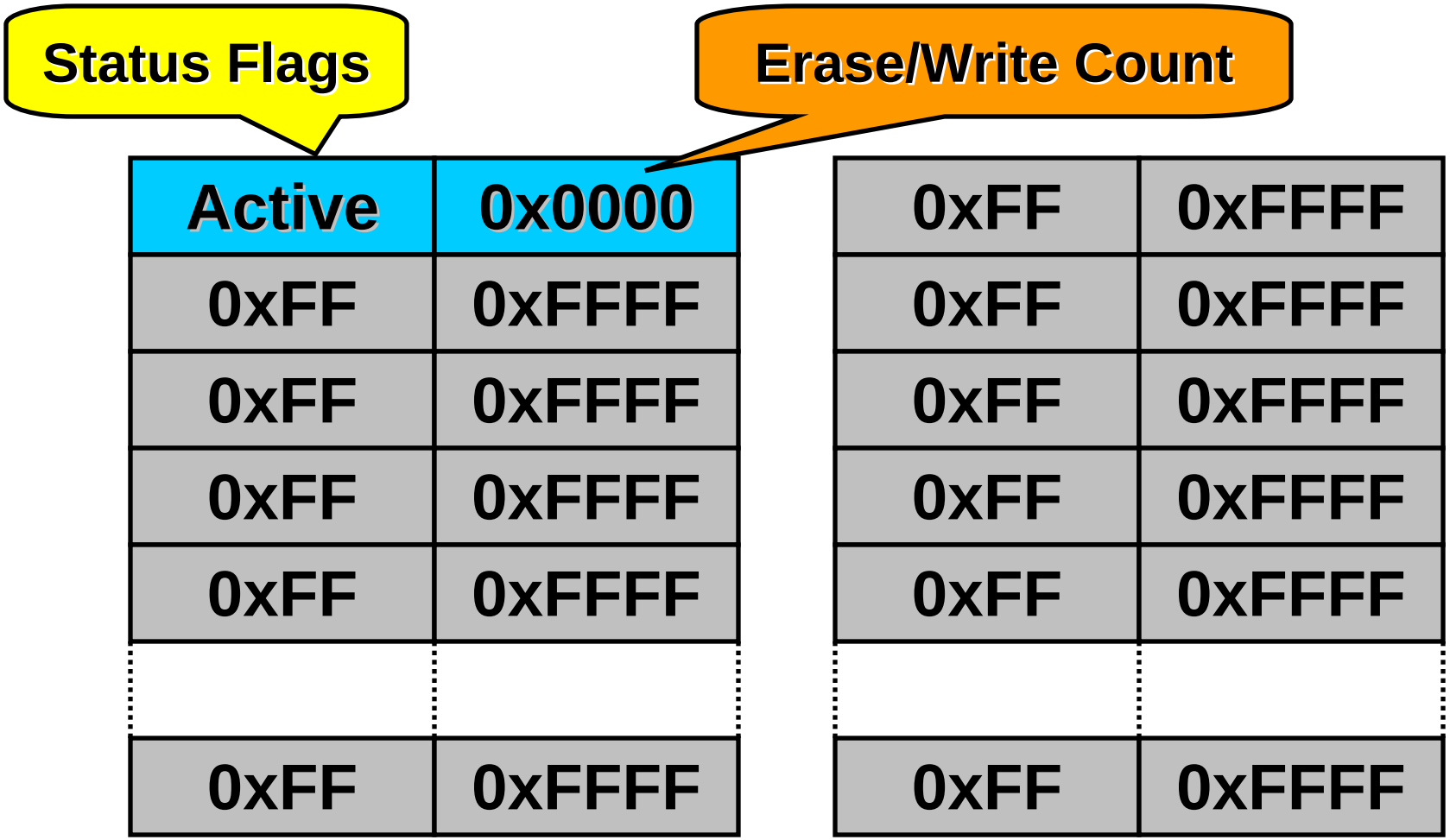
- Blank pages of PIC24/dsPIC33 memory

0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF

0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF

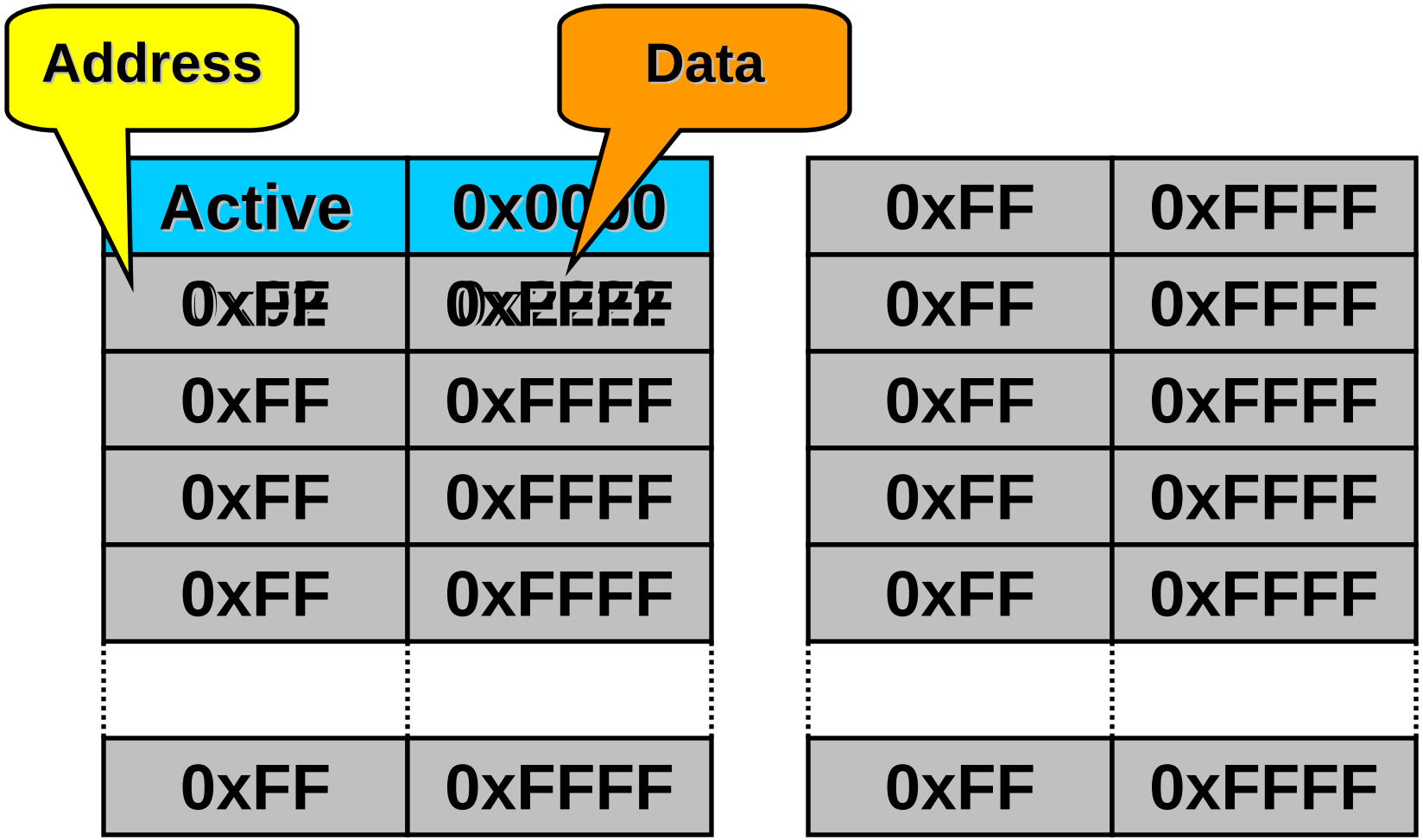
How AN1095 DEE Emulation Works - Initialization

- Initialized pages of PIC24/dsPIC33 memory



How AN1095 DEE Emulation Works - Write

- Write the value 0x2222 to location 2



How AN1095 DEE Emulation Works - Write

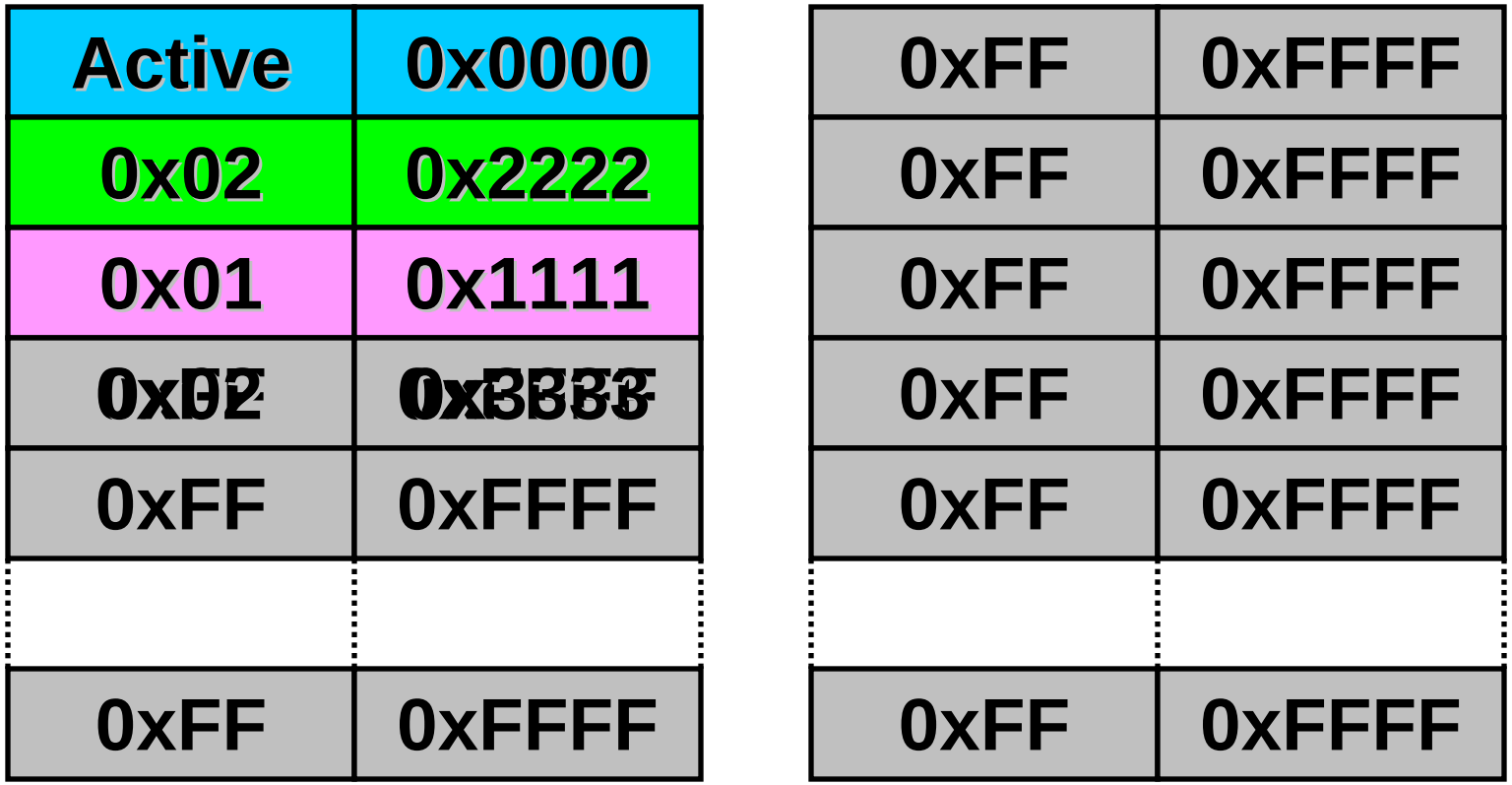
- Write the value 0x1111 to location 1

Active	0x0000
0x02	0x2222
0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF

0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF

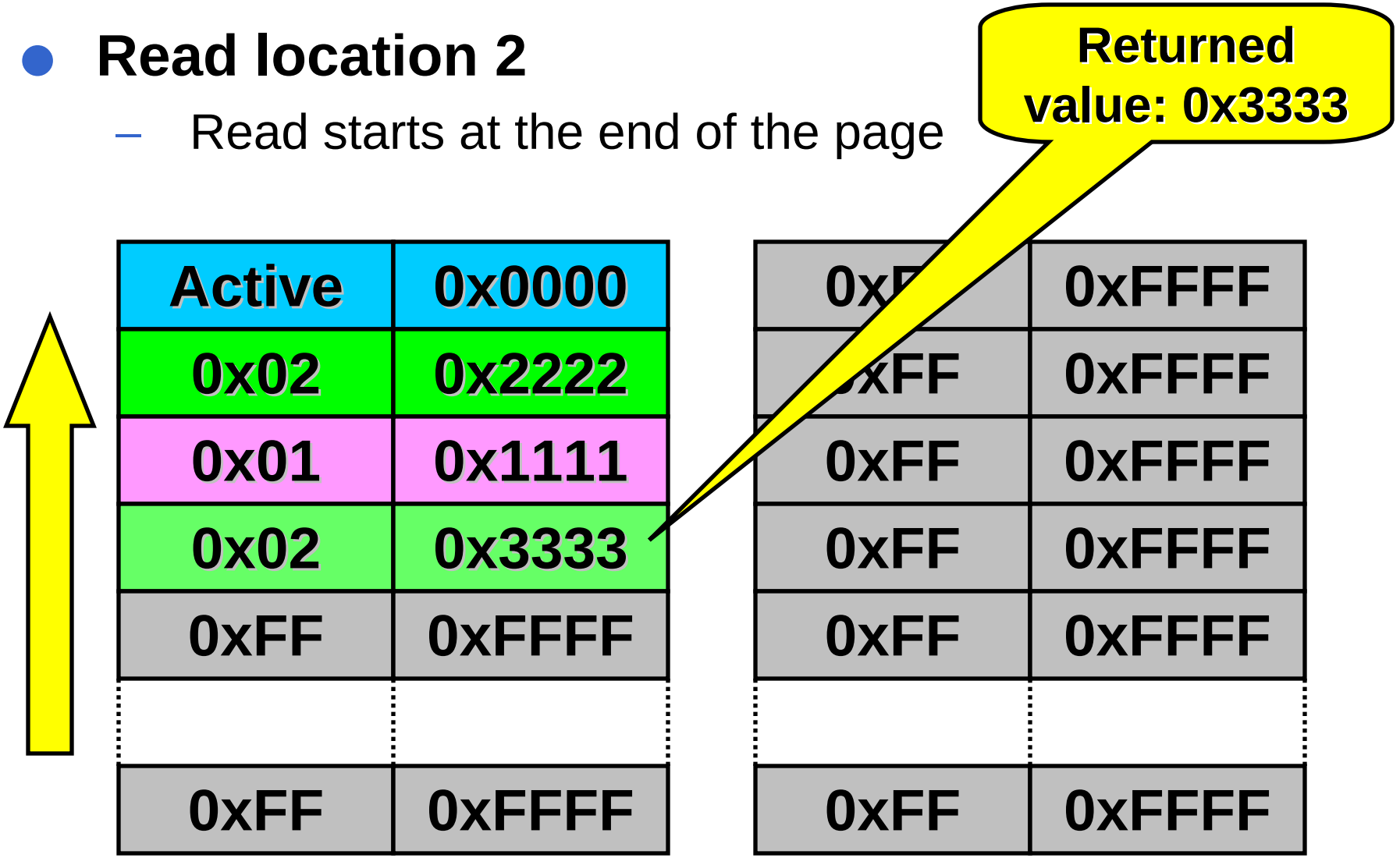
How AN1095 DEE Emulation Works - Write

- Write the value 0x3333 to location 2



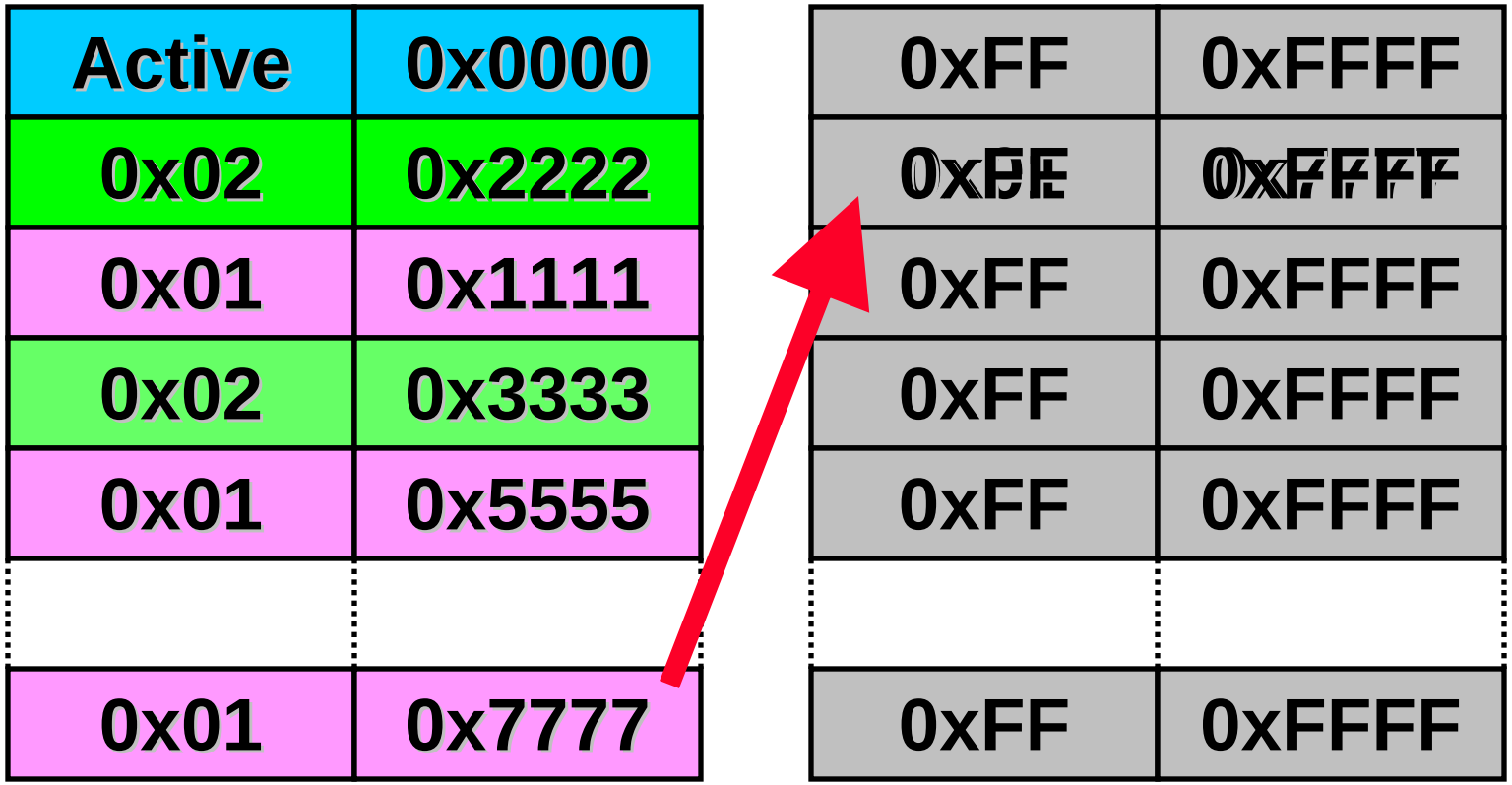
How AN1095 DEE Emulation Works - Read

- **Read location 2**
  - Read starts at the end of the page



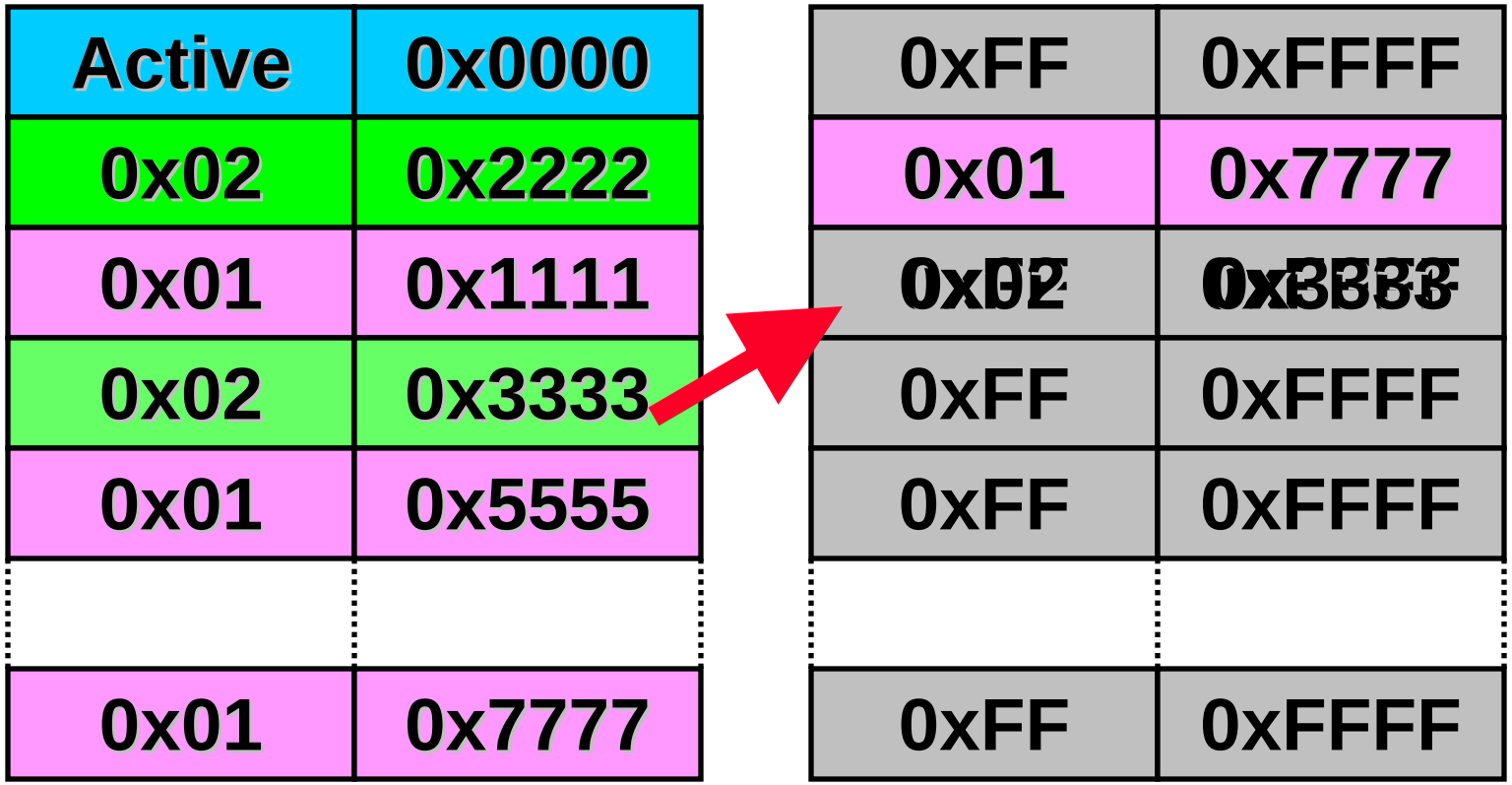
How AN1095 DEE Emulation Works - Pack

- **First pack into second page**
  - Sorts and writes data into new page



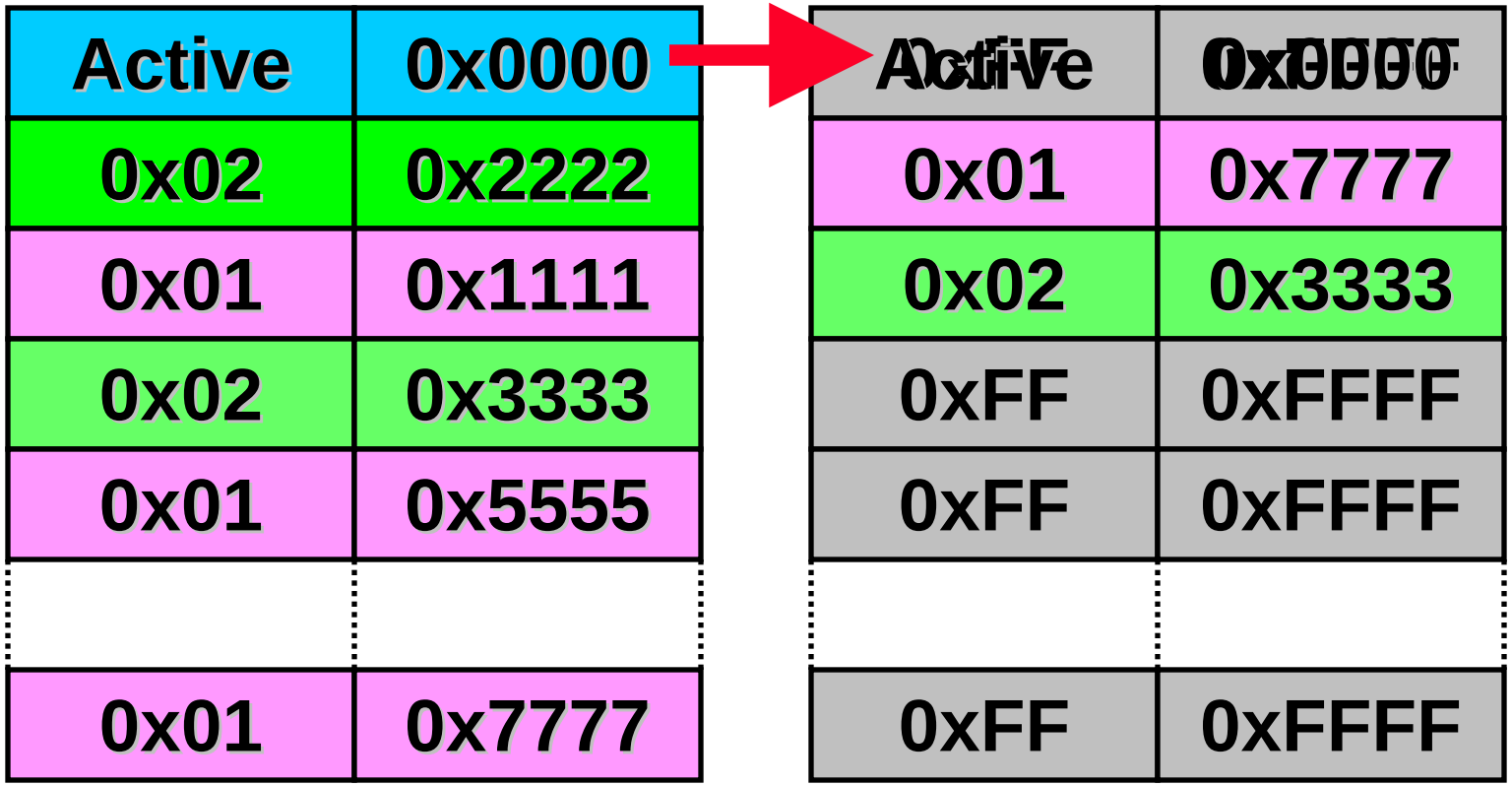
How AN1095 DEE Emulation Works - Pack

- **First pack into second page**
  - Sorts and writes data into new page



How AN1095 DEE Emulation Works - Pack

- **First pack into second page**
  - Copies Status and Erase/Write count



How AN1095 DEE Emulation Works - Pack

- **First pack into second page**
  - Erases old page

Data is verified before the erase

Active	0x0000
0x01	0x7777
0x02	0x3333
0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF

How AN1095 DEE Emulation Works - Pack

- **Pack into first page**
  - Increments Erase/Write count

Active	0x0001
0x01	0x8888
0x02	0x4444
0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF

Active	0x0000
0x01	0x7777
0x02	0x3333
0x02	0x4444
0x01	0x9999
0x01	0x8888

How AN1095 DEE Emulation Works - Pack

- **Pack into first page**
  - Increments Erase/Write count

Active	0x0001
0x01	0x8888
0x02	0x4444
0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF

0xFE	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF
0xFF	0xFFFF

How AN1095 DEE Emulation Works

- **Blank page of PIC24/dsPIC33 memory**

Page + 0	0xFF (Page Status)	0xFFFF (Erase/Write Count)
Page + 2	0xFF DEE Address	0xFFFF DEE Data
Page + 4	0xFF	0xFFFF
Page + 6	0xFF	0xFFFF
Page + 8	0xFF	0xFFFF
Page + 1022	0xFF	0xFFFF

How AN1095 DEE Emulation Works

- **Initialized Active Page for PIC24/dsPIC33**

Page + 0	0xF3	0
Page + 2	0xFF	0xFFFF
Page + 4	0xFF	0xFFFF
Page + 6	0xFF	0xFFFF
Page + 8	0xFF	0xFFFF

Page + 1022	0xFF	0xFFFF

How AN1095 DEE Emulation Works

- **Active Page after WriteDEE (0x0202,2)**

Page + 0	0xF3	0
Page + 2	2 (DEE Address)	0x0202 (DEE Data)
Page + 4	0xFF	0xFFFF
Page + 6	0xFF	0xFFFF
Page + 8	0xFF	0xFFFF
Page + 1022	0xFF	0xFFFF

How AN1095 DEE Emulation Works

- **Active Page after WriteDEE (0x0707,7)**

Page + 0	0xF3	0
Page + 2	2	0x0202
Page + 4	7	0x0707
Page + 6	0xFF	0xFFFF
Page + 8	0xFF	0xFFFF

Page + 1022	0xFF	0xFFFF

How AN1095 DEE Emulation Works

- **Active Page after WriteDEE (0x2222,2)**

Page + 0	0xF3	0
Page + 2	2	0x0202
Page + 4	7	0x0707
Page + 6	2	0x2222
Page + 8	0xFF	0xFFFF

Page + 1022	0xFF	0xFFFF

How AN1095 DEE Emulation Works

- **Active Page after WriteDEE (0x0A0A,0xA)**

Page + 0	0xF3	0
Page + 2	2	0x0202
Page + 4	7	0x0707
Page + 6	2	0x2222
Page + 8	0xA	0x0A0A

Page + 1022	0xFF	0xFFFF

How AN1095 DEE Emulation Works

- **Active Page after WriteDEE (0x7777,0x7)**

Page + 0	0xF3	0
Page + 2	2	0x0202
Page + 4	7	0x0707
Page + 6	2	0x2222
Page + 8	0xA	0x0A0A

Page + 1022	7	0x7777

How AN1095 DEE Emulation Works

- **New Active Page after Pack operation**

Page + 1024	0xF3	0
Page + 1026	2	0x2222
Page + 1028	7	0x7777
Page + 102A	0xA	0xAAAA
Page + 102C	0xFF	0xFFFF

Page + 2046	0xFF	0xFFFF

AN1095 DEE Emulation API

- **DataEEInit()**
  - Must be called before any other DEE emulation function
  - Initializes program memory
    - **Resets erase/write cycle counter**
    - **Initializes Page Status**
    - **Designates first Active page**
  - Is reset tolerant

AN1095 DEE Emulation API

- **WriteDEE(data, address)**
  - Uses word-write capability
  - Stores information in the next available location in Active page
  - Address 255 (0xFF) is reserved
  - Data size is architecture dependent
    - **PIC18 – 8 bits**
    - **PIC24/dsPIC – 16 bits**

AN1095 DEE Emulation API

- **ReadDEE(address)**
  - Performs reverse search of Active page for DEE address
  - Returns corresponding DEE data
  - If address is not found
    - **Sets a status flag**
    - **Returns 0xFFFF (0xFF)**

AN1095 DEE Emulation API

- **PackEE()**
  - Called automatically by WriteDEE() when Active page is full
  - Increments Erase/Write counter
  - Can also be called by the application

AN1095 DEE Emulation API PackEE()

- Pack moves data to next page of program memory
- After last page has been filled, data is moved back into first page
- Erase/write counter is incremented after all pages have been used

AN1095 DEE Emulation API PackEE()

- Pack operations take time
- Application may want to ensure that the pack does not interfere with other functionality
- GetNextAvailCount()
  - Returns the number of available locations in Active page
- Can be used to schedule pack at a convenient time

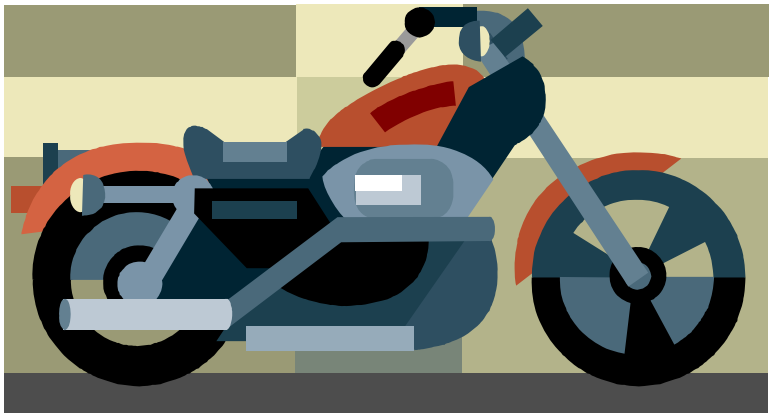
How AN1095 DEE Emulation Works

- **Reset tolerant operation**
  - Two pages of Flash are required
  - DEE information is always held in Flash
    - **Temporary copies only in RAM**
  - Initialization routine counts number of Active pages to determine context

AN1095 DEE Emulation API

- **Status flags set by the algorithm**
  - Address not found
  - Expired page
  - Pack before page is full
  - Pack before initialization
  - Pack skipped
  - Illegal address
  - Page corrupt
  - Write error
- **Macros available to clear flags**

Customization Options



Customization Options

- **Compile-Time Options**
  - Application dependent
    - **Amount of emulated DEE**
    - **Number of program memory pages**
    - **Maximum erase/write count**
  - MCU dependent
    - **Program memory opcodes**
    - **Page erase size**
    - **Row programming size**

Customization Options

- **Effective Endurance**
  - Number of program memory pages
  - Amount of emulated DEE
  - Maximum erase/write count
  - Distribution of writes across DEE address range

Endurance increase can be >500X

UNIVERSITY OF MICROCHIP
UOFM
MASTERS 2007

Hands-on: Implement AN1095 DEE Emulation

Hands-on: Implement AN1095 DEE Emulation

- **Extract AN1095 files from 11072DEE folder to C:**
  - **Launch MPLAB® IDE and open project**

    C:\Microchip Solutions\DEE Emulation Demo 16-bit.mcp

- **Configure algorithm for a DEE size of 50 and an erase/write limit of 100**
  - Refer to AN1095 page 13, section "PIC24/dsPIC33F Emulation Checklist"

How AN1095 DEE Emulation Works

- **Highlights**
  - Increased effective endurance
  - Simple DEE interface
  - Highly configurable
  - Low requirements: program/data memory, peripherals, pins
  - No reserved data values
  - Nonvolatile
  - PIC18, PIC24 and dsPIC33 support

Summary

- Compared the advantages of various nonvolatile memory options
- Discussed how data EE emulation works
- Reviewed configuration options
- Implemented AN1095 DEE emulation algorithm

Dev Tools Used in this Class

- **Explorer 16 Development Board (DM240001)**
  - PIC24FJ128GA010 PIM (Included)
- **MPLAB® REAL ICE™ In-Circuit Emulator (DV244005)**
- **MPLAB IDE (SW007002)**

References

- **AN1095 Emulating Data EEPROM for PIC18 & PIC24 Microcontrollers and dsPIC® Digital Signal Controllers**
  - **Available on Microchip Technology's website www.microchip.com**

University of Microchip UofM MASTERs 2007

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KeeLoq, KeeLoq logo, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.