M03: Memory& uDMA

3.1. Internal Memory

Prof. Rosa Zheng

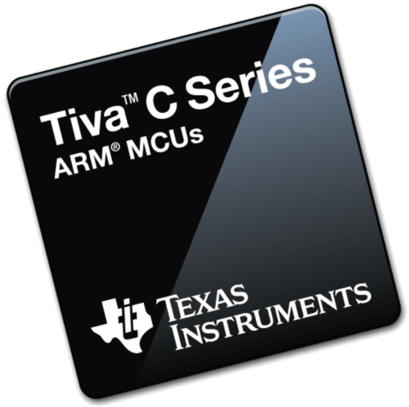
References:

1.TM4C123GH6PM data sheet (**spms376e.pdf**) Chapter 8.

2. TM4C123 Workshop Lab 8 slides

3. Bit-banding Blog at https://spin.atomicobject.com/2013/02/08/bit-banding/

TM4C123GH6PM Memory



256KB Flash memory

- Single-cycle up to 40MHz
- Pre-fetch buffer and speculative branch improves performance above 40 MHz

32KB single-cycle SRAM w/ bit-banding& uDMA

Internal ROM loaded with TivaWare software

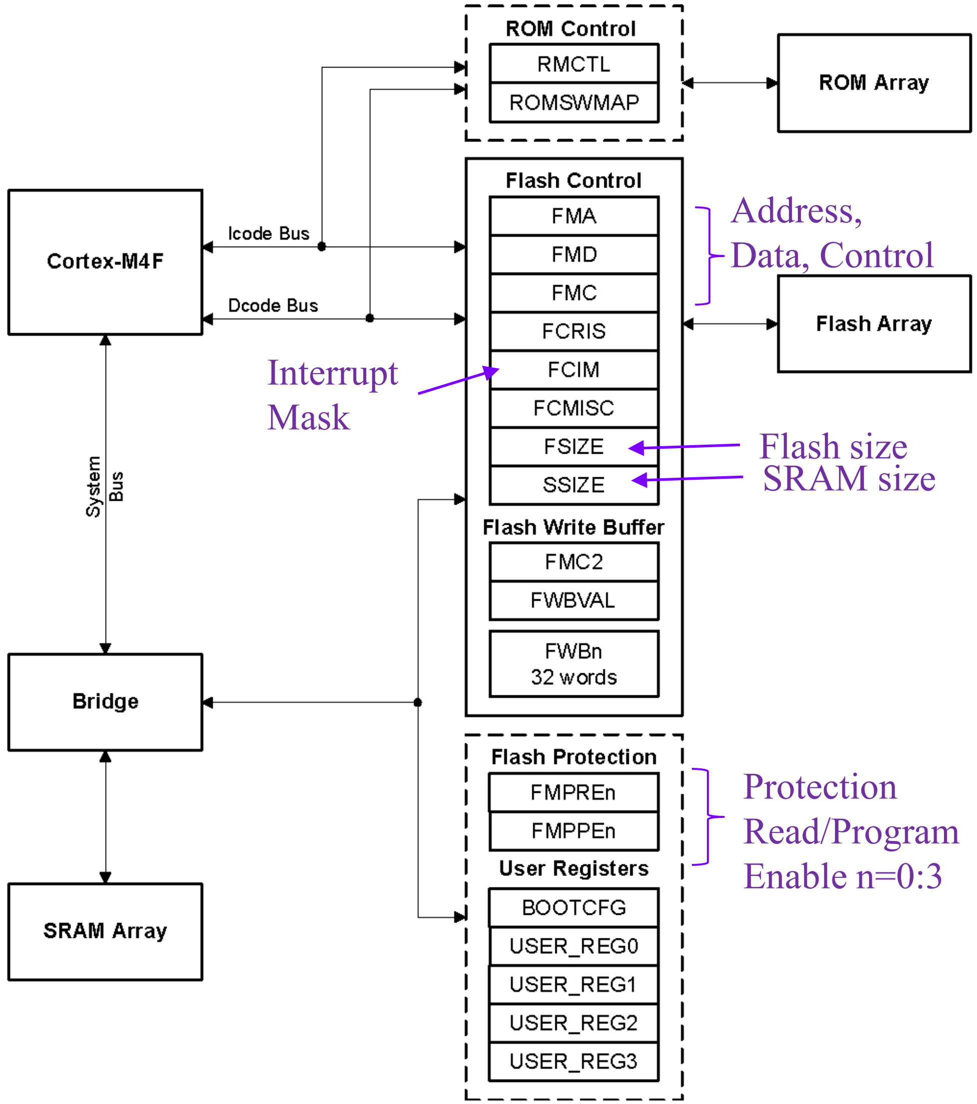
- Peripheral Driver Library (ROM_xxx() in diverlib)
- Boot Loader (start program at Flash w/o debugger)
- Advanced Encryption Standard (AES) cryptography tables
- Cyclic Redundancy Check (CRC) error detection functionality

2KB EEPROM (fast, saves board space)

- Wear-leveled 500K program/erase cycles
- Thirty-two 16-word blocks
- Can be bulk or block erased
- 10 year data retention
- 4 clock cycle read time

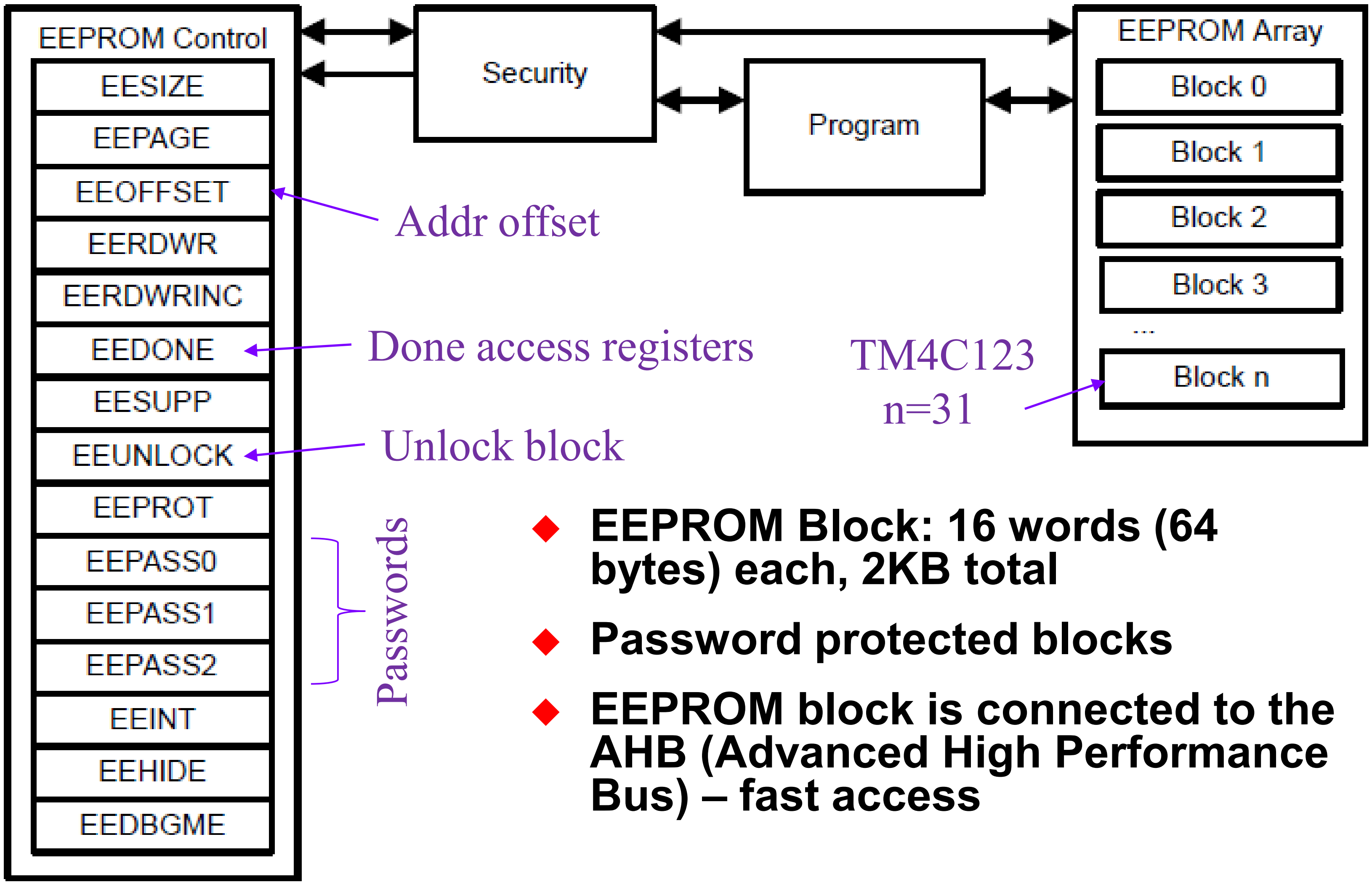
0x00000000 Flash
0x01000000 ROM
0x20000000 SRAM
0x22000000 Bit-banded SRAM
0x40000000 Peripherals & EEPROM
0x42000000 Bit-banded Peripherals
0xE0000000 Instrumentation, ETM, etc.

Flash, SRAM and ROM Control



- **ROM Control:**
  - ROMCTL – 32 bits reserved, 1 bit set to enable boot loader
  - ROMSWMAP – table of address (driverlib/rom_map.h) for driverlib functions (driverlib/rom.h)
- **Flash Control**
- **Flash Write Buffer:** to speed up the flash programming
- **Flash Protection:** four modes
- **User Registers:** user defined power-on reset behavior.
- **SRAM:**
  - Separate bus to work w/ uDMA
  - Bit-banding

EEPROM Control



- EEPROM Block: 16 words (64 bytes) each, 2KB total
- Password protected blocks
- EEPROM block is connected to the AHB (Advanced High Performance Bus) – fast access

EEPROM

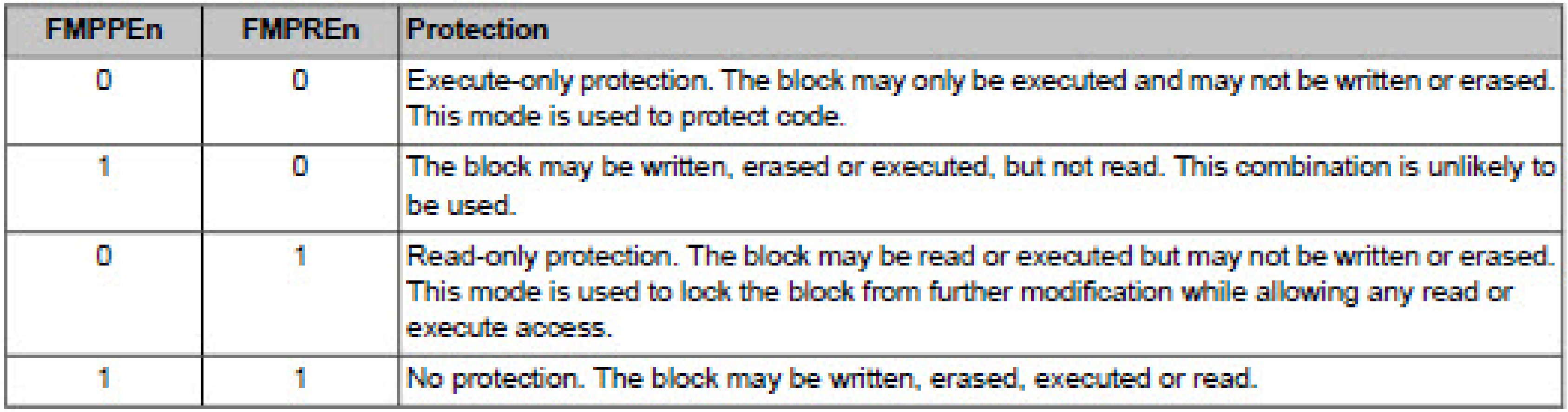
- **2KB of memory starting at 0x400AF000 in Peripheral space**
- **Built-in wear leveling with endurance of 500K writes**
- **Lock protection option for the whole peripheral as well as per block using 32-bit to 96-bit password codes**
- **Interrupt support for write completion to avoid polling**
- **Random and sequential read/write access (4 cycles max/word)**
- **driverlib/eeprom.h APIs:**
  - Reading: EEPROMRead() EEPROMSizeGet()
  - Programming: EEPROMMassErase(), EEPROMProgram() and EEPROMProgramNonBlocking().
  - Protection: EEPROMBlockProtectGet(), EEPROMBlockProtectSet(), EEPROMBlockPasswordSet(), EEPROMBlockLock(), EEPROMBlockUnlock() and EEPROMBlockHide().
  - Interrupt: EEPROMIntEnable(), EEPROMIntDisable(), EEPROMIntStatus(), EEPROMIntClear().

Flash

- Holds user program and data. Contents remain when power off.
- 256KB / 40MHz starting at 0x00000000
- Organized in 1KB independently erasable blocks
- Code fetches and data access occur over separate buses
- Below 40MHz, Flash access is single cycle
- Above 40MHz, the prefetch buffer fetches two 32-bit words/cycle. No wait states for sequential code.
- Branch speculation avoids wait state on some branches
- Programmable write and execution protection available
- Simple programming interface: driverlib/flash.h
  - FlashErase(), FlashProgram(), FlashUserGet(), and FlashUserSet();
  - FlashProtectGet(), FlashProtectSet(), and FlashProtectSave();
  - FlashIntRegister(), FlashIntUnregister(), FlashIntEnable(), FlashIntDisable(), FlashIntGetStatus(), and FlashIntClear().

Securing Your IP

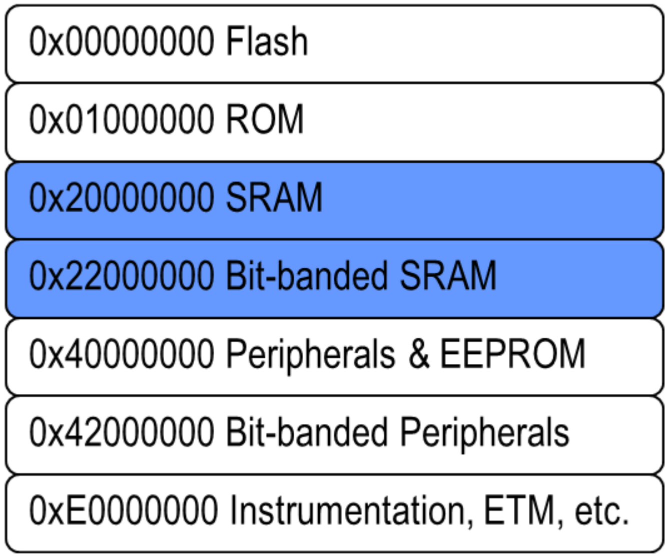
- Flash memory can be protected (per 2KB memory block). Prohibited access attempts will generate a bus fault.



FMPPEn	FMPREn	Protection
0	0	Execute-only protection. The block may only be executed and may not be written or erased. This mode is used to protect code.
1	0	The block may be written, erased or executed, but not read. This combination is unlikely to be used.
0	1	Read-only protection. The block may be read or executed but may not be written or erased. This mode is used to lock the block from further modification while allowing any read or execute access.
1	1	No protection. The block may be written, erased, executed or read.

- The JTAG and SWD ports can be disabled. DBG0 = 0 and DBG1 = 1 (in BOOTCFG register) for debug to be available. The user should be careful to provide a mechanism, for instance via the bootloader of enabling the ports since this is permanent.
- There is a set of steps in the UG for recovering a "locked" microcontroller, but this will result in the mass erase of flash memory.

SRAM & Bit-Banding

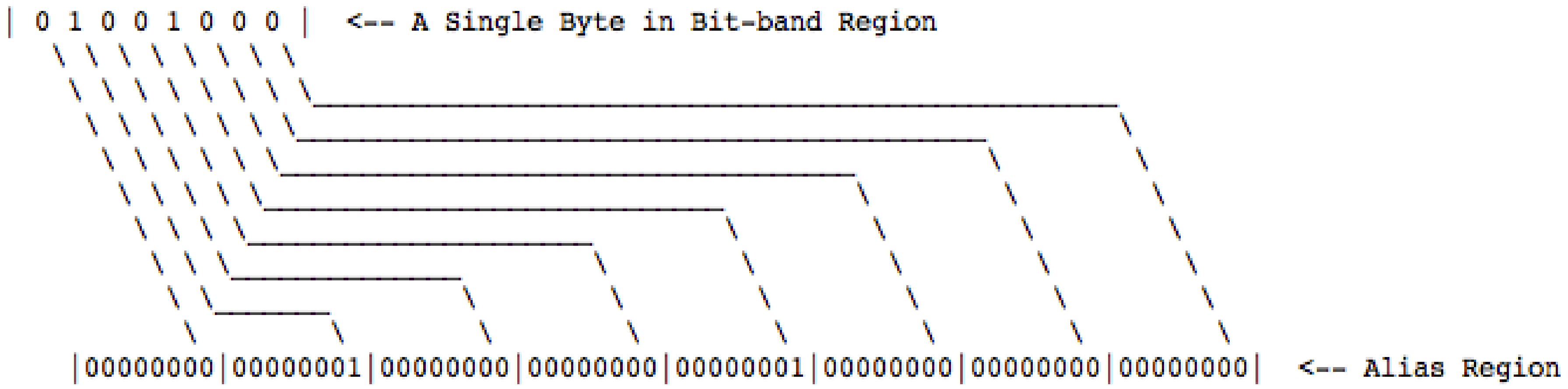

0x00000000 Flash
0x01000000 ROM
0x20000000 SRAM
0x22000000 Bit-banded SRAM
0x40000000 Peripherals & EEPROM
0x42000000 Bit-banded Peripherals
0xE0000000 Instrumentation, ETM, etc.

- **32KB / 80MHz**
- **Can hold code or data**
- **Contents will be lost when power off**
- **Bit banding: fast access**
- **uDMA (Micro Direct Memory Access)**

- **Bit-band reduces the number of read-modify-write operations**
- **Base addresses:**
  - **SRAM starts at 0x20000000**
    Bit-banded SRAM starts at 0x2200000
  - **Peripheral space starts at 0x40000000**
    Bit-banded peripheral space starts at 0x42000000

Bit-Banding

Blog at https://spin.atomicobject.com/2013/02/08/bit-banding/



The bit-band alias is calculated by using the formula:

```
bit-band alias = bit-band base + (byte offset*0x20) + (bit number*4)
```

For example, if bit-7 at address 0x20002000 is to be modified, then bit-band alias is:

```
0x20002000 + (0x2000 * 0x20) + (7 * 4) = 0x2204001C
```

Memory Protection Unit (MPU)

- Defines 8 separate memory regions plus a background region accessible only from privileged mode
- Regions of 256 bytes or more are divided into 8 equal-sized sub-regions
- MPU definitions for all regions include: Location, Size, Access permissions, Memory attributes
- Accessing a prohibited region causes a memory management fault
- Driverlib/mpu.h:
  - MPURegionGet(), MPURegionSet(), MPURegionCountGet();
  - MPURegionEnable(), MPURegionDisable().
  - MPUIntRegister(), MPUIntUnregister().

Cortex M4 Privilege Levels

- Privilege levels offer additional protection for software, particularly operating systems (see datasheet chapter 2)
- Unprivileged (User mode): software has …
  - Limited access to the Priority Mask register
  - No access to the system timer, NVIC, or system control block
  - Possibly restricted access to memory or peripherals (FPU, MPU, etc)
- Privileged (other modes): software has …
  - use of all the instructions and has access to all resources
- ISRs operate in privileged mode
- Thread code operates in unprivileged mode unless the level is changed via the Thread Mode Privilege Level (TMPL) bit in the CONTROL register