Architecture Overview

CSE3442 EMBEDDED SYSTEMS I

J Losh

Architecture Topics

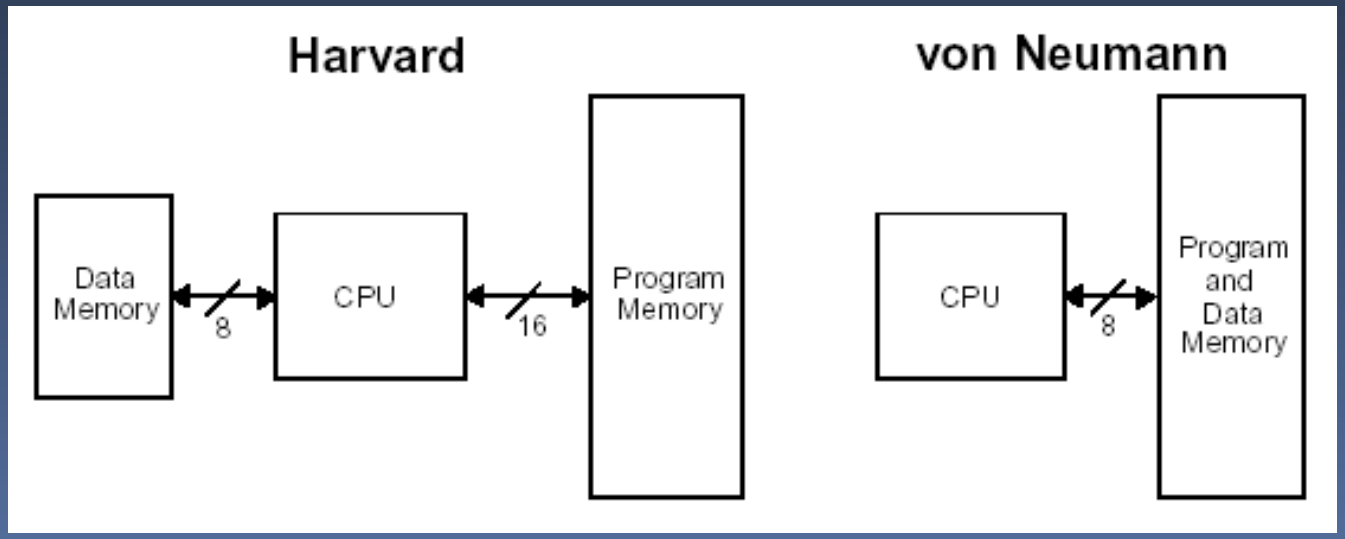
- Harvard v. von Neumann
- RISC v. CISC
- Microcontrollers v. Microprocessors
- M4F Microcontroller Architecture
  - Pipelines
  - ALU and CPU
  - Data memory
  - Program memory
  - Reset and interrupt operation
  - Peripherals

Resources

- Resources on the Class Server
  - TM4C123GH6PM datasheet
  - DDI0439D M4 Processor Technical Reference Manual
- All figures are from the TM4C123GH6PM datasheet unless specified and are copyright the owner

Harvard v. von Neumann

- von Neumann Architecture shares a memory space for programs and data
- Harvard Architecture has different memory spaces for program and data



From Microchip 18C Family Reference

Harvard v. von Neumann

- von Neumann Architecture
  - Shares a memory space for programs and data
  - Programs can be altered or loaded easily
  - Code storage may not be optimal and may require multiple fetches to form an instruction
  - Contention exists between program fetches and data flow, causing a performance hit
- Harvard Architecture
  - Has different memory spaces for program and data
  - Allows different size buses for program and data memories
  - Allows optimal sizing of program data bus width to facilitate fast execution of instructions

RISC v. CISC

- CISC (Complex instruction set computer)
  - Commonly many hundreds of instructions
  - Specialized instructions perform more complex functions
  - Complicated instructions can take 50+ cycles (*, /)
- RISC (Reduced instruction set computer)
  - Generally less than 100 instructions
  - Instructions are less powerful on average, so many instructions may be needed to perform a CISC equivalent instruction
  - Instructions normally execute in 1 instruction cycle leading to higher MIPS

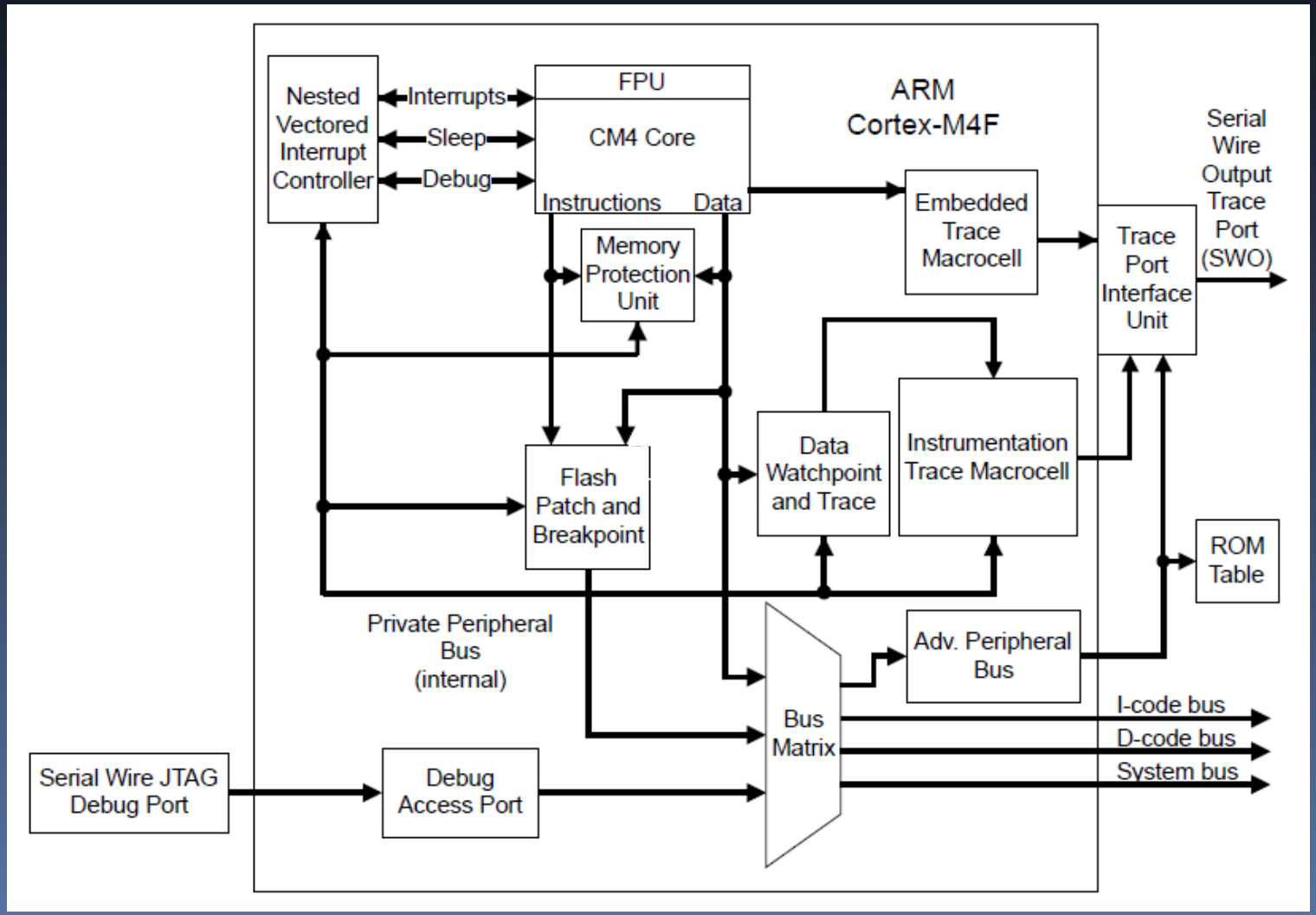
Microcontrollers v. uP's

- Microprocessors
  - Provide external data and address buses
  - Generally require external memory
  - Require many external peripherals for typical applications
- Microcontrollers
  - Generally do not expose data and address buses
  - Generally do not require external memories
  - Have many peripherals built-in

TM4C123GH6M Controller

- 32-bit ARM® M4F core @ 80MHz
- Harvard architecture, 3-stage pipeline
- Thumb-2 16-/32-bit instruction set
- IEEE754 compliant floating point unit
- Single-instruction, multiple-data (SIMD) multiply and multiple and accumulate (MAC) instructions
- M4 peripherals
  - Memory protection unit (MPU)
  - Nested interrupt controller (NVIC)
  - System tick timer (SysTick)
  - System control block (SCB)

TM4C123GH6M Controller

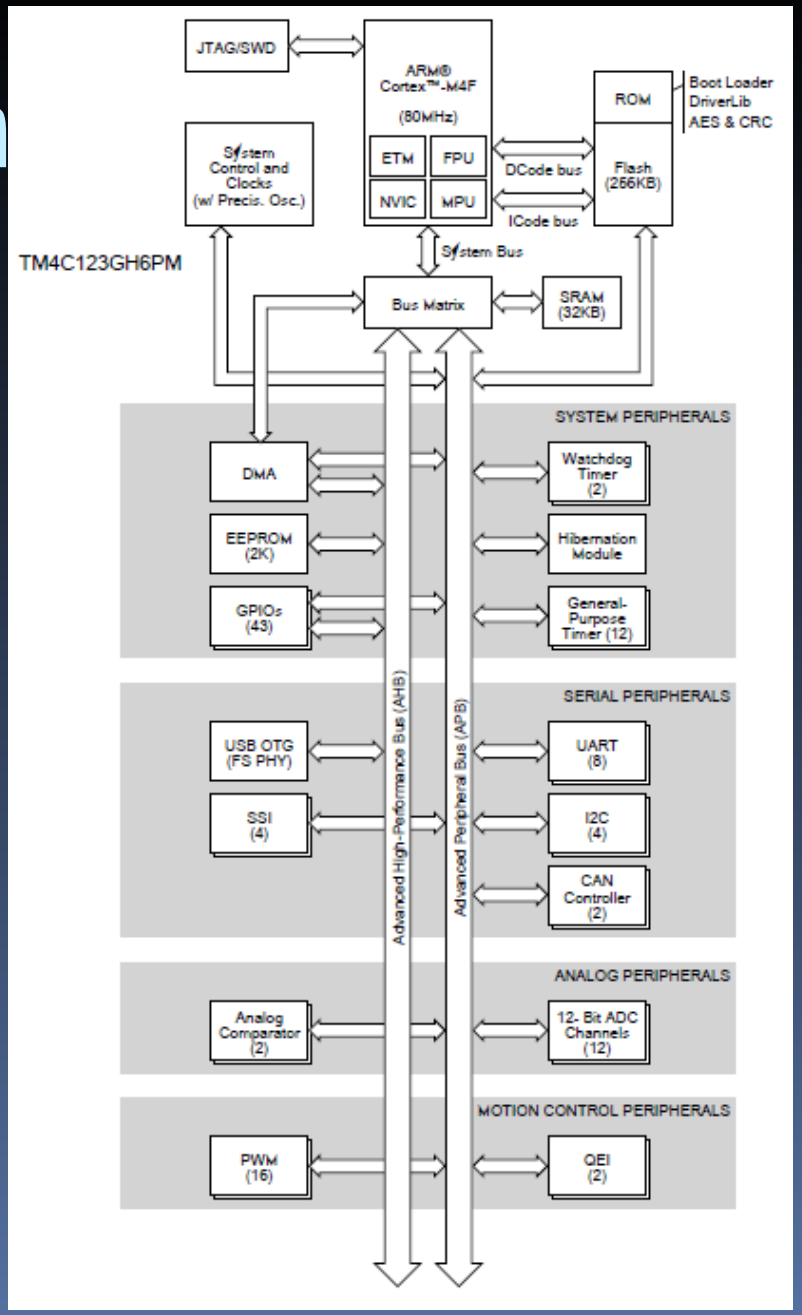


AMBA Types

- Two different Advanced Microprocessor Bus Architecture (AMBA) types in use
  - Advanced high-performance bus (AHB) offers high bandwidth, full-duplex, pipelined access mainly to memories
  - Advanced peripheral bus (APB) offers simple interface to low bandwidth peripherals

Block Diagram

- Bus Usage
  - Most peripherals are on APB
  - EEPROM and USB-OTG are on AHB
  - GPIO can be routed to AHB or APB
  - DMA can access AHB and APB (memory-to-memory, peripheral-to-memory, memory-to-peripheral)



Memory Model

- 0x00000000-0x0003FFFF
  - 256 KiB on-chip Flash (I-code AHB-lite bus interface)
- 0x20000000-0x20007FFF
  - 32KiB on-chip SRAM (D-code AHB-lite bus interface)
- 0x22000000-0x220FFFFF (system bus AHB-lite interface)
  - Bitband alias of SRAM (atomic bit ops – no read-modify-write)
- 0x40000000-0x400FFFFF (system bus AHB-lite interface)
  - Peripherals (UART, SPI, I2C, GPIO, USB, …)
- 0x42000000-0x43FFFFFF (system bus AHB-lite interface)
  - Bitband alias of peripheral area
- 0xE0000000-0xFFFFFFFF (private peripheral bus APB

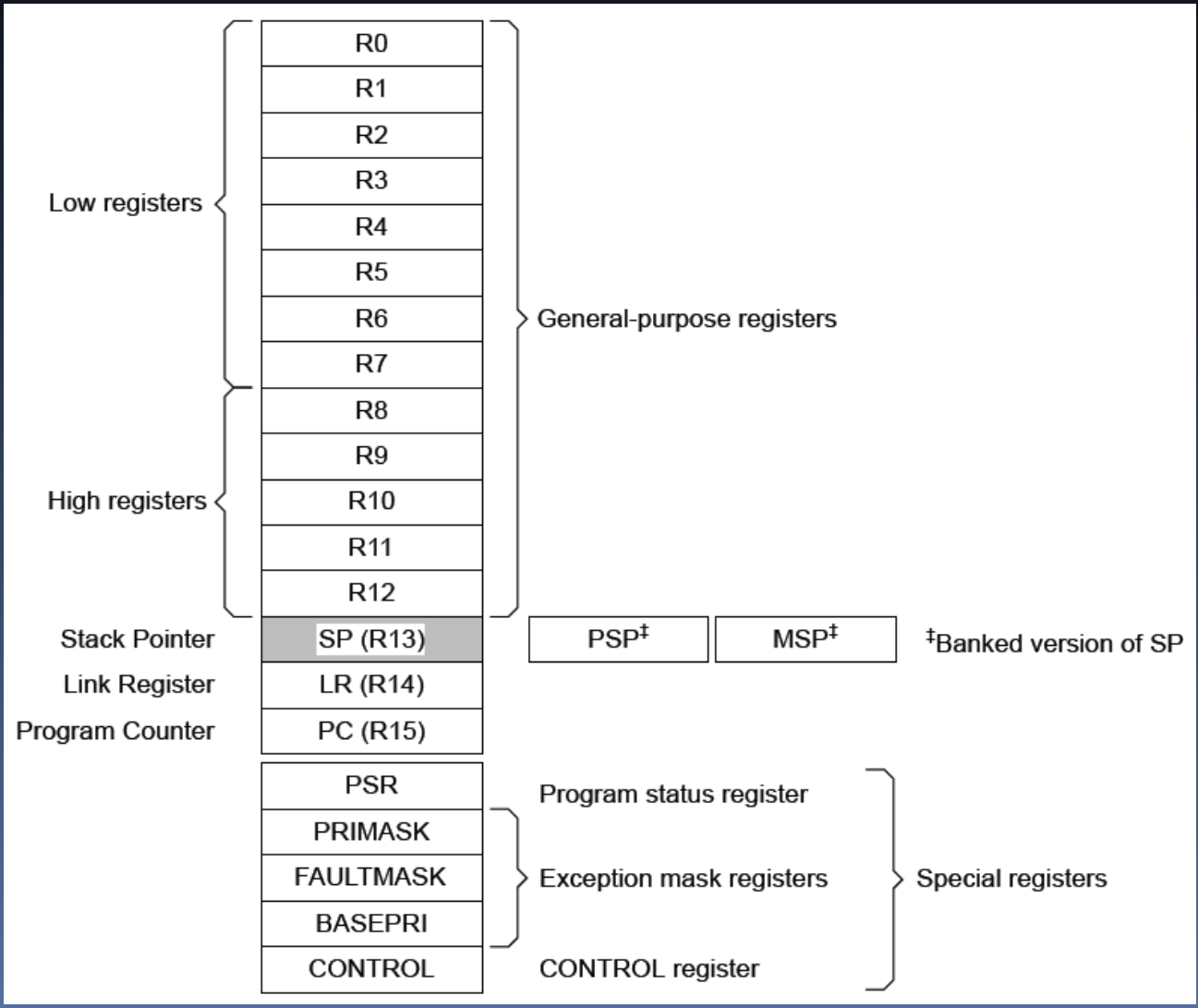
Bit-banding

- Bit-banding allows a single bit in the SRAM or peripheral space to be set or cleared without reading the entire 32-bit value, modifying the bit, and then re-writing the 32-bit value
- There is a byte in alias memory that maps to a bit in the SRAM or peripheral space allows a single write to be set or cleared

Programming Model

- Two modes of operation
  - Thread mode for normal tasks and handler mode for interrupt/exception processing
- Two privilege levels are possible
  - Privileged (only for trusted code)
    - Access to all memory and instructions
    - Used by operating system, including interrupt/exception handler
  - Unprivileged
    - Limited access to memory, no access to private peripheral bus, some instructions cannot be accessed
    - Used by user tasks (threads)

Registers


R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
Low registers
High registers
General-purpose registers
Stack Pointer
SP (R13)
PSP‡
MSP‡
‡Banked version of SP
Link Register
LR (R14)
Program Counter
PC (R15)
PSR
PRIMASK
FAULTMASK
BASEPRI
CONTROL
Program status register
Exception mask registers
CONTROL register
Special registers

Registers

- R0-12 are general purpose registers
- SP/PSP/MSP/LR are related to stack operations
- PC is the program counter
- PSR is the program status register
- PRIMASK, FAULTMASK, and BASEPRI are for interrupt control
- CONTROL is for FPU and activating privileged mode
- FPSC is for floating-point system status and control

Stacks

- PSP is the process stack pointer (for threads)
- MSP is the priviledged stack pointer (for exceptions/OS)
- SP is mapped to MSP when the active stack pointer (ASP bit in CONTROL register) = 0
  (this is always the case in handler mode)
- SP is mapped to PSP when the active stack pointer (ASP bit in CONTROL register) = 1
- LR is the link register used to allow local variable allocation on the stack in subroutines
- SP is pre-decremented with PUSH (RTL: SP<-SP-4, [SP] <- reg)
- Data is stored in little-endian format

On-chip Peripherals

- 8 UARTs
- 4 Synchronous Serial Interface (SSI) for SPI
- 4 Inter-IC (I^2C)
- 2 CAN2.0A/B controllers
- 1 USB Host/Device/OTG controller
- 6 General purpose timers (GPTM)
- 2 Watchdog timers (WDT)
- 16 PWM outputs
- 2 Quadrature encoders (QEI)
- 2 12-bit, 1 Msps analog-to-digital converters (ADC) with 8 digital comparators multiplexed to 12 inputs
- 2 analog comparators
- Up to 43 GPIO pins
- 32-channel DMA controller (µDMA)
- 2kB EEPROM