Microprocessor and Interfacings

Lec_1 Review of CPU and Intro to ARM Cortex-M3

This lecture note was adapted from the following materials:

1) Lecture notes: Embedded Computer Systems by Dr. Gul N. Khan
2) The Definitive Guide to the ARM Cortex-M3 by Joseph Yiu
3) Fundamental of Embedded Software with the ARM Cortex-M3 by Daniel W. Lewis

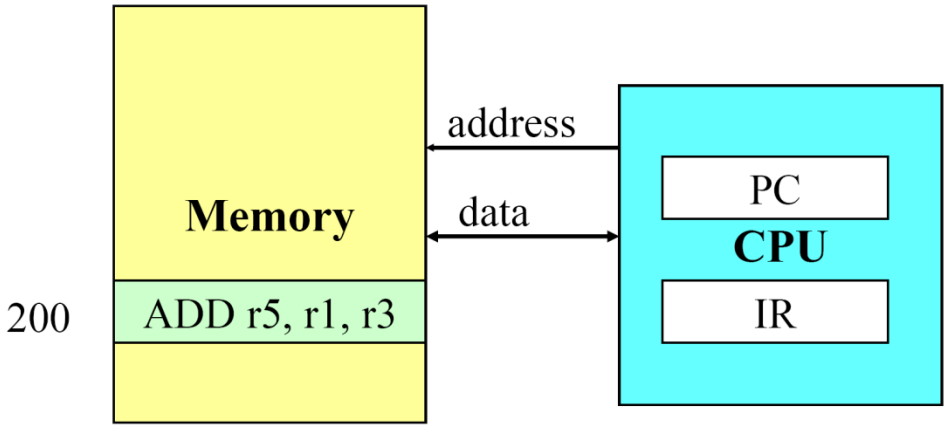
Outline

Topic:

- Review of Microprocessor
  - Architecture
  - Memory
  - Input Output
- Introduction to ARM Cortex-M3
  - Profile of ARM processor
  - Register
  - Internal organization

Processor Architecture

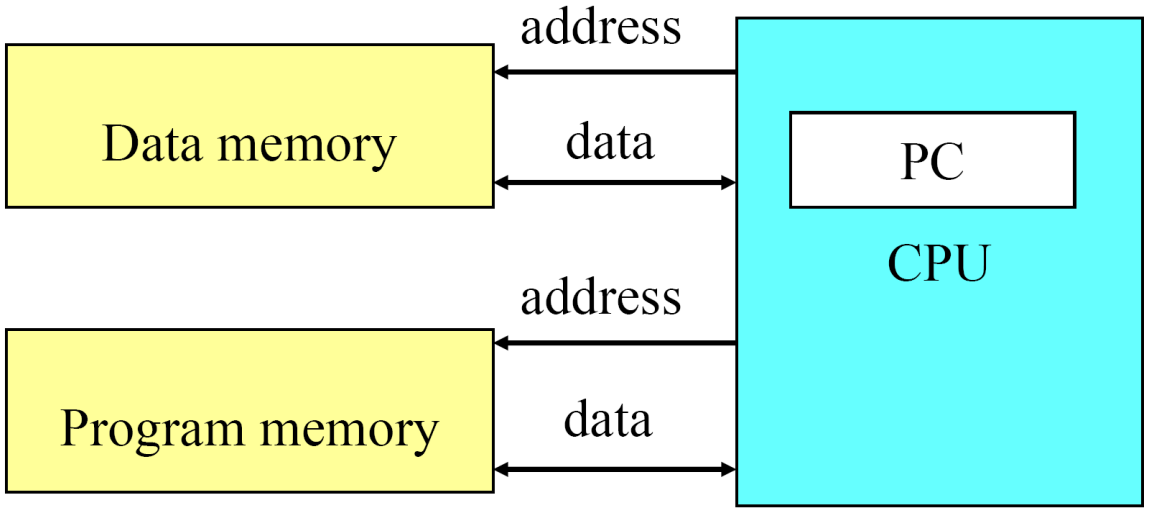
von Neumann Architecture (single memory)



- Memory holds both data and instructions.
- Central processing unit (CPU) fetches instructions from memory.

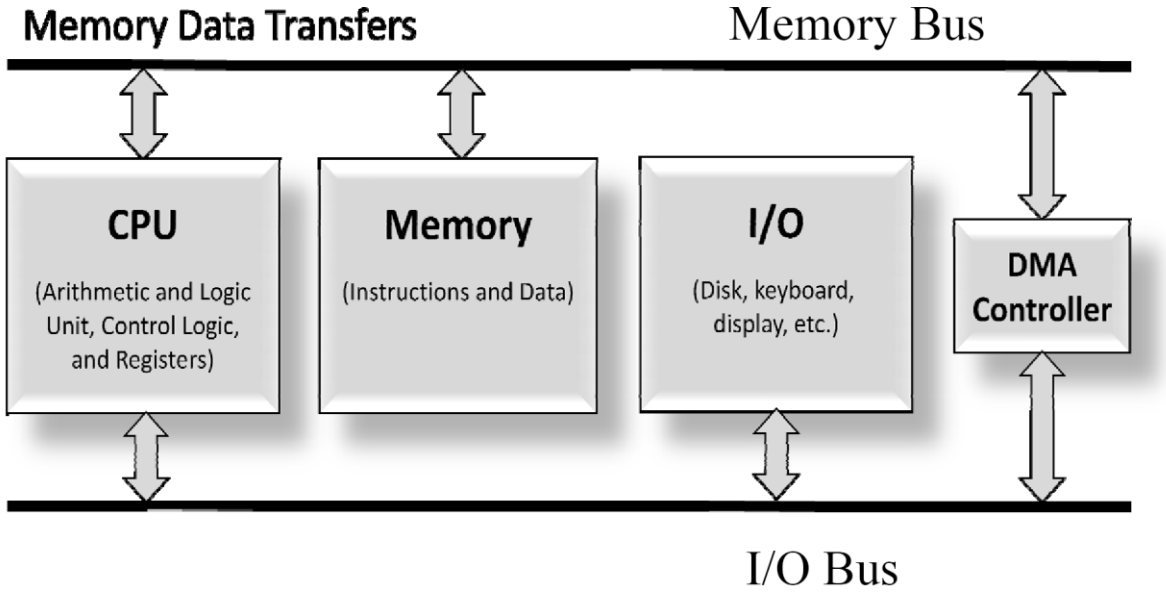
Processor Architecture

Harvard Architecture (dual memory)



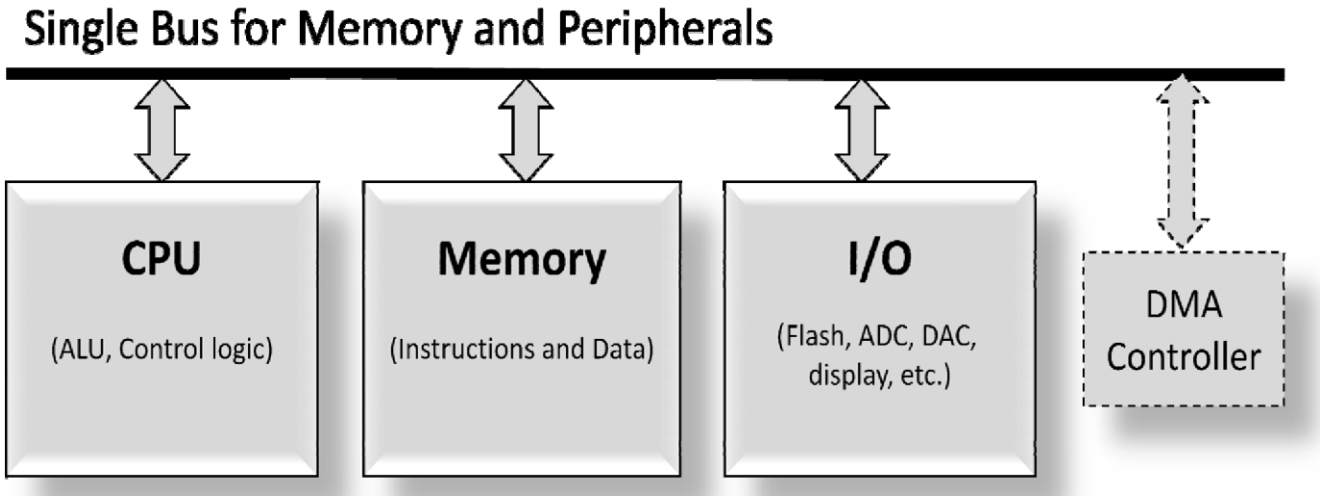
- Separate program memory and data memory
- Allows two simultaneous memory fetches.
- ARM Cortex-M3 is considered as Harvard Architecture

Typical Desktop Processor



- One set of buses to connect the CPU and memory
- Another set of buses to connect the CPU and I/O
- Data transferred between memory and I/O go through and is coordinated by CPU
- Greater data transferred are achieved by DMA

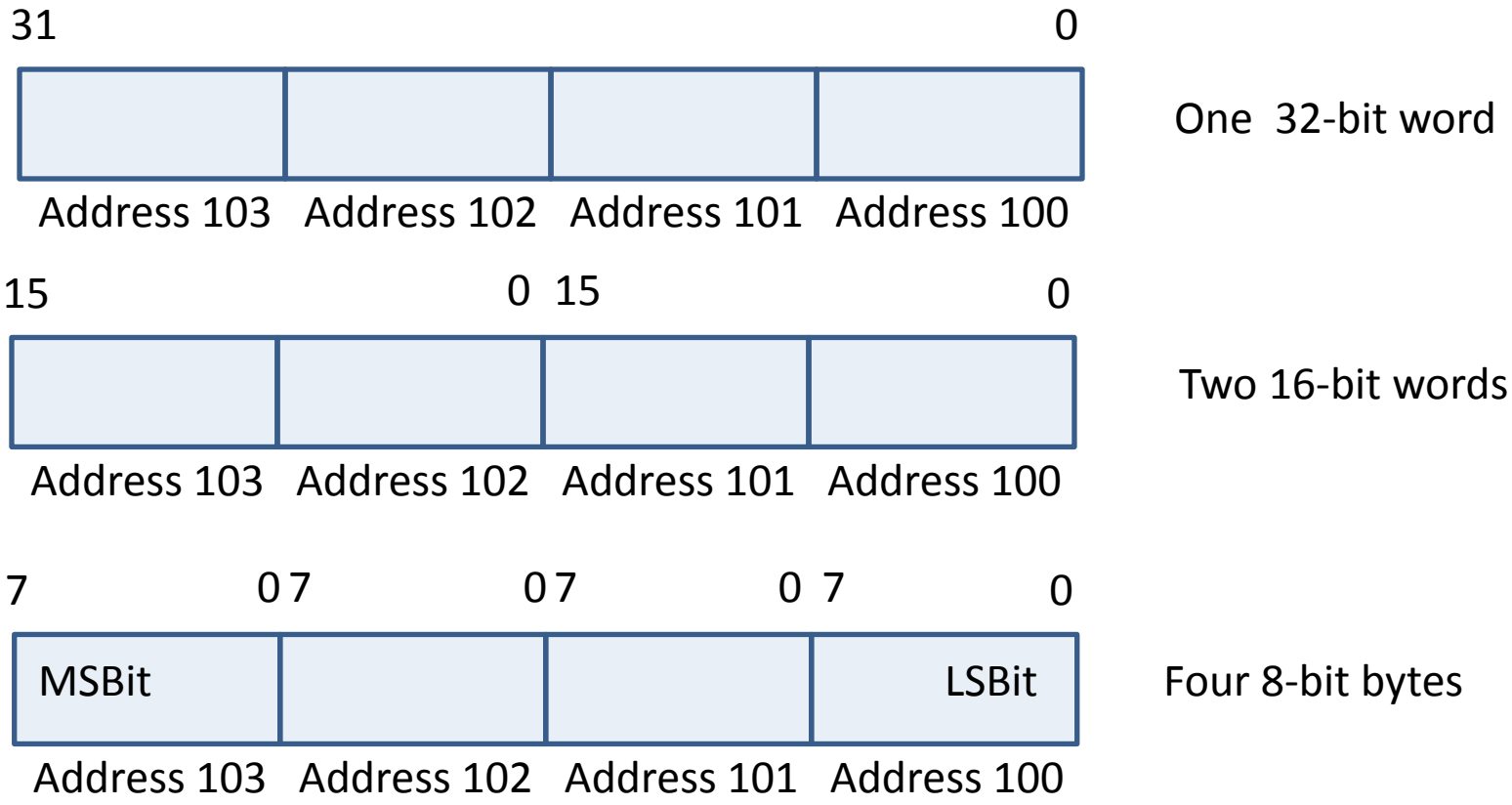
Typical Embedded Processor



- Use single bus configuration: " memory-map I/O"
- Portion of address space is reserved for and decoded by the I/O devices
  - Eliminates the need for a separate "I/O instruction" zall of the regular instruction for access memory can be used to access I/O devices
  - Most embedded applications do not require the higher data transfer rate and omit the DMA controller entirely

Memory: Byte Ordering

- Byte addressable (each byte has unique address)



- Cortex-M3 is little endian

Memory : Data Alignment

Address 15	Address 14	Address 13	Address 12
Address 11	Address 10	Address 9	Address 8
Address 7 (MSbyte)	Address 6	Address 5	Address 4 (LSbyte)
Address 3	Address 2	Address 1	Address 0

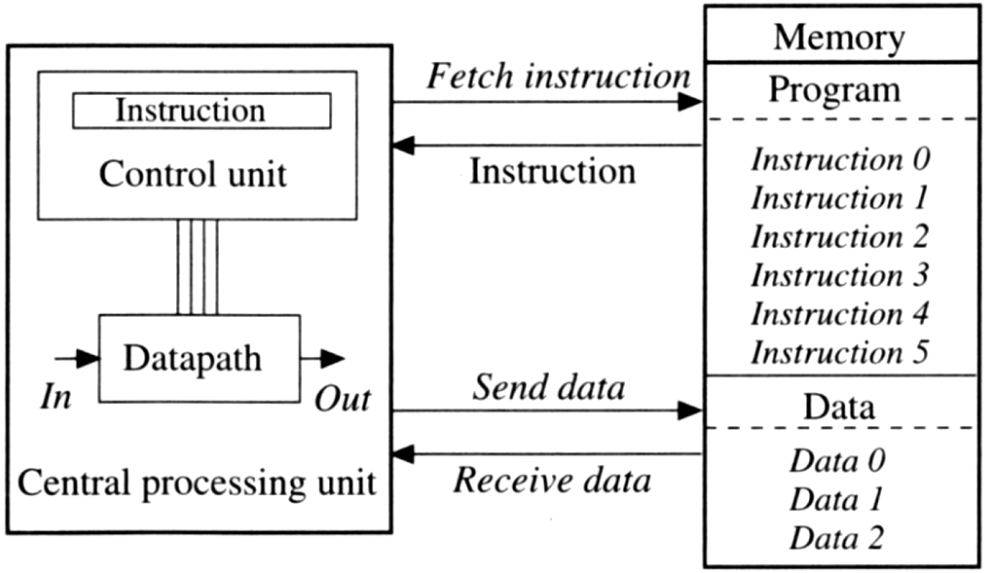
Address 15	Address 14	Address 13	Address 12
Address 11	Address 10	Address 9 (Msbyte)	Address 8
Address 7	Address 6 (LSbyte)	Address 5	Address 4
Address 3	Address 2	Address 1	Address 0

		Address 7	Address 6

Address 9	Address 8		

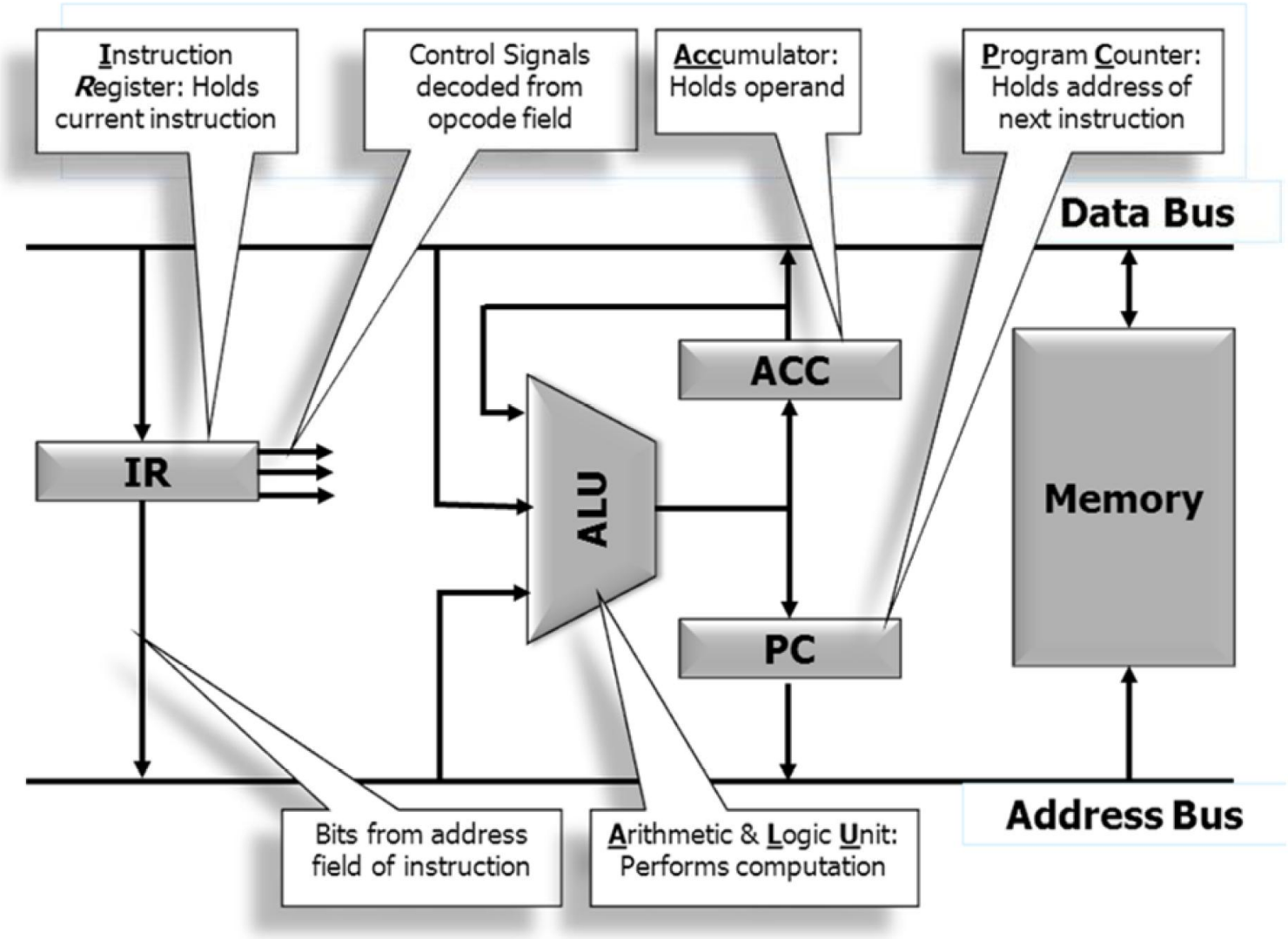
Address 9	Address 8	Address 7	Address 6

Instruction Execution Process



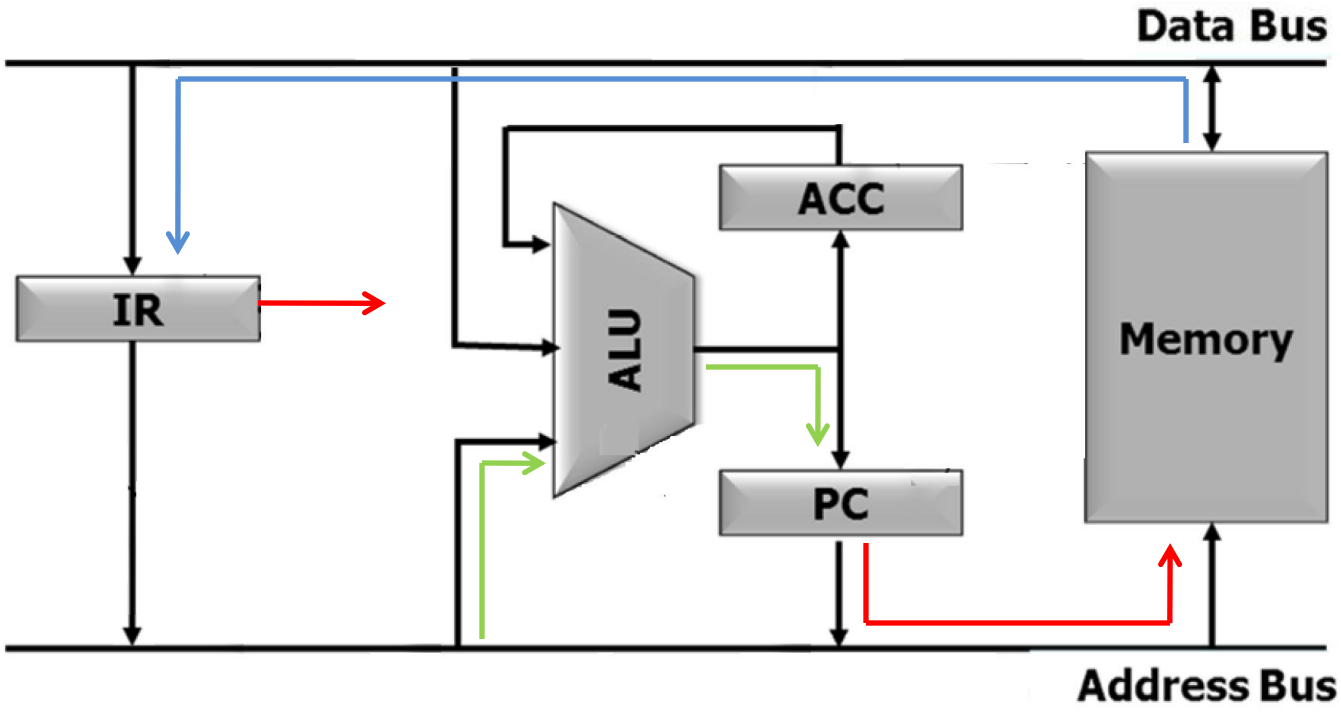
- **Instruction Fetch**
  - Reads next instruction into instruction register (IR). Address is in PC
- **Instruction Interpretation (Decode)**
  - Decodes the op-code. Gets the required operands and routes them to ALU.
  - Determines the address of next instruction and loads it into the PC
- **Execution**
  - Generates control signals of ALU for execution

EX: Single Accumulator CPU



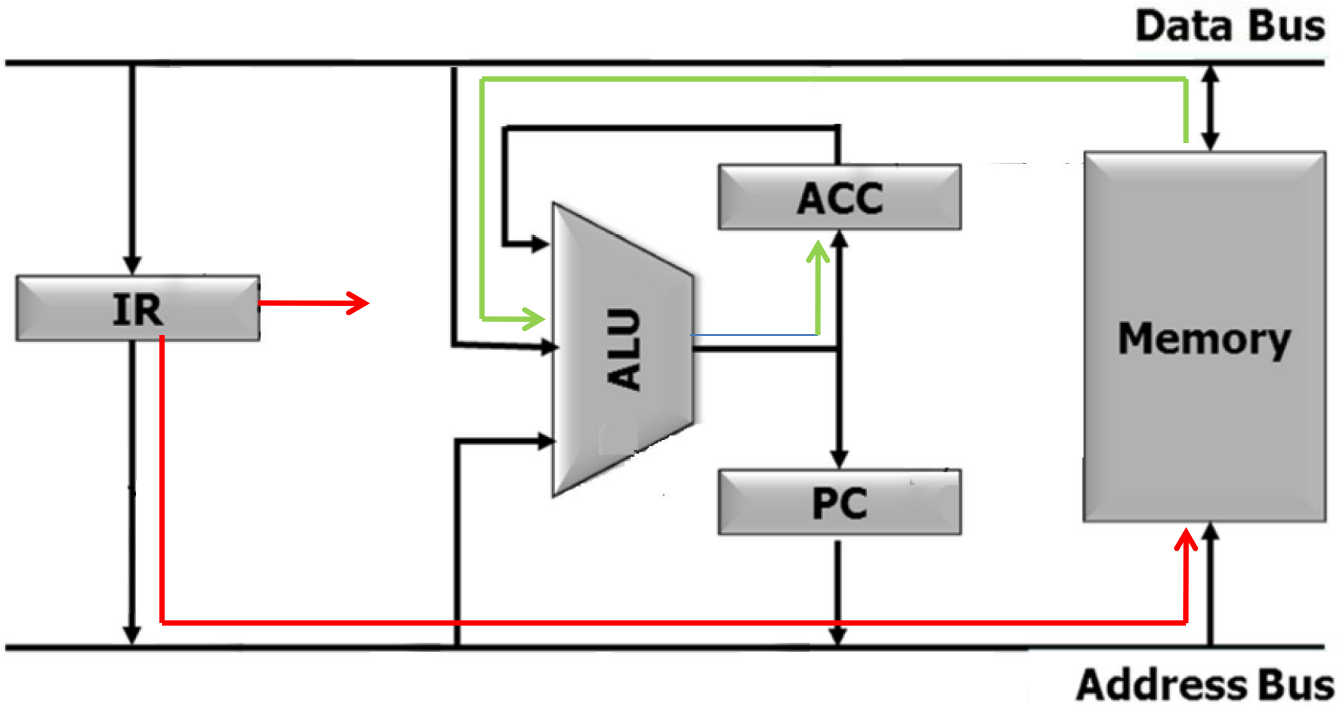
EX: Single Accumulator CPU

- CPU components used during the fetch phase



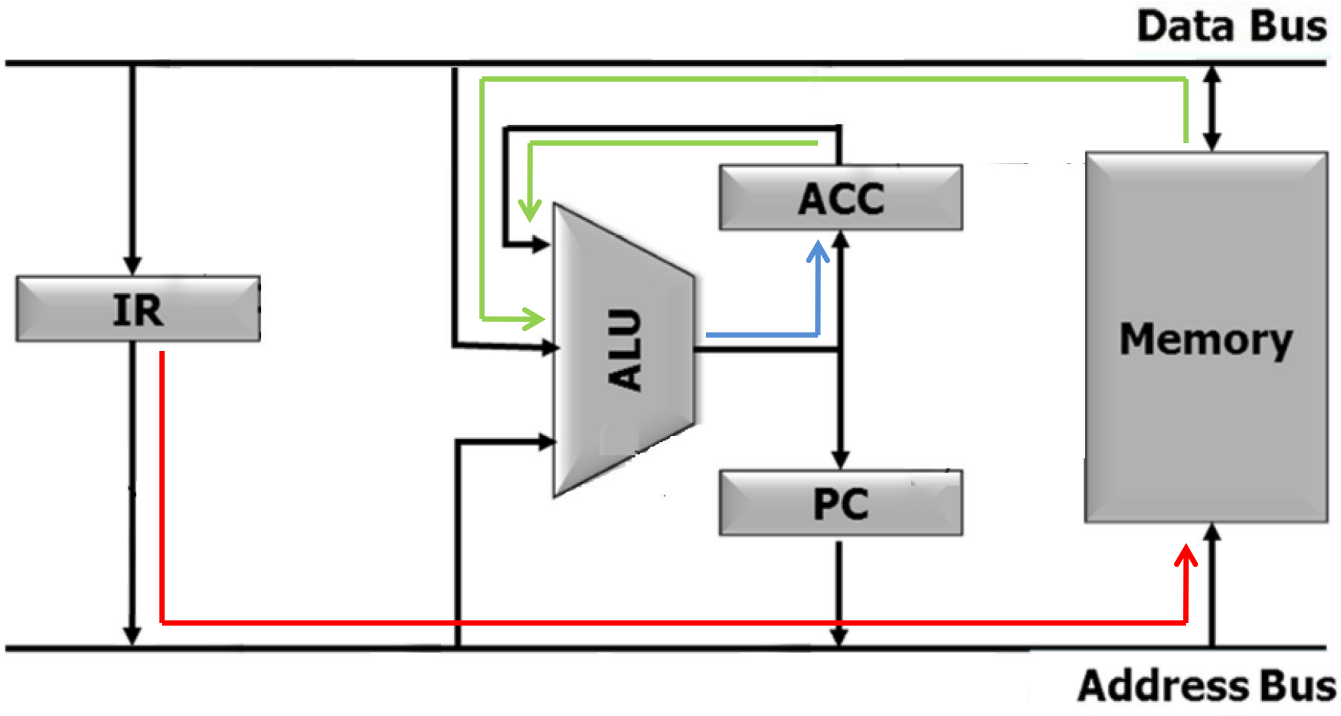
EX: Single Accumulator CPU

- CPU components used during execution phase to load register from memory



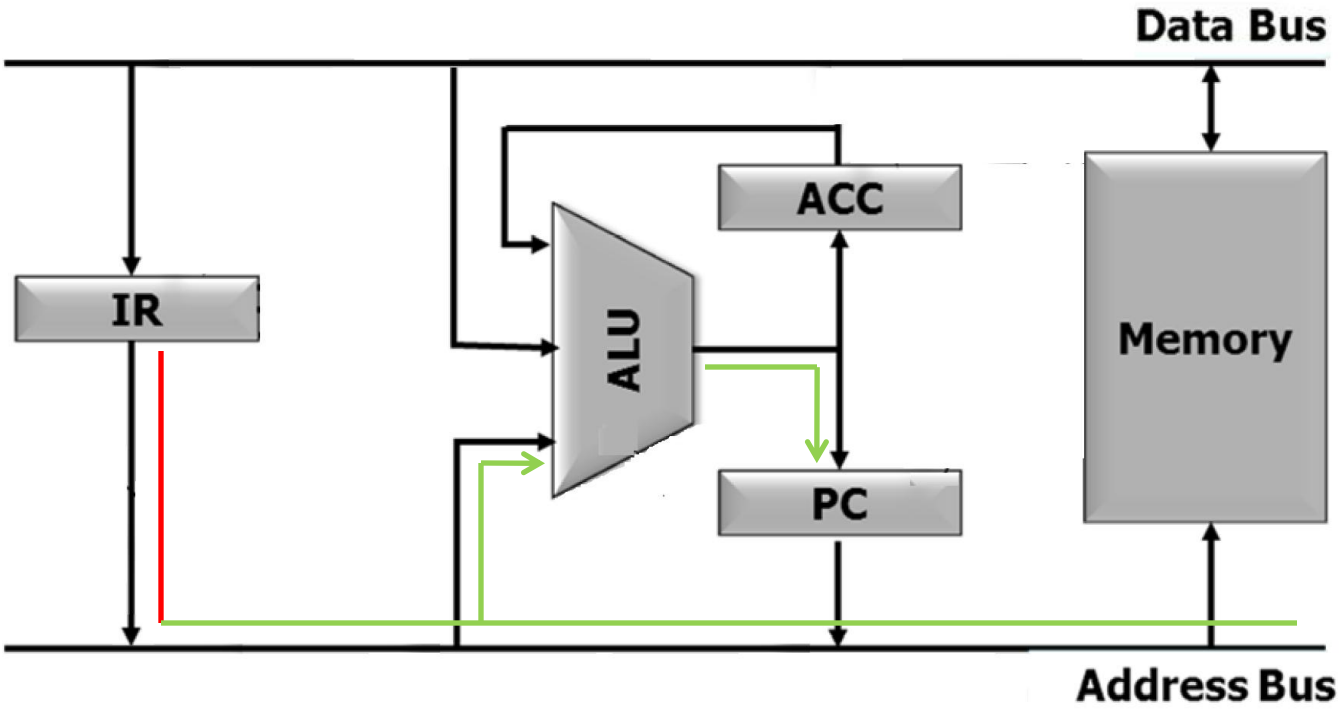
EX: Single Accumulator CPU

- CPU components used during execution phase of an ADD instruction



EX: Single Accumulator CPU

- CPU components used during execute phase of a branch instruction



Input/Output

- I/O system includes everything other than CPU and memory
- Usually operates at speed that are **unrelated** to and **much slower** than the processor
- Some form of handshaking to coordinate transfer between I/O and processor is always necessary.
  - Not output data faster than output devices can accept it.
  - Not try to process input data that has not yet become available

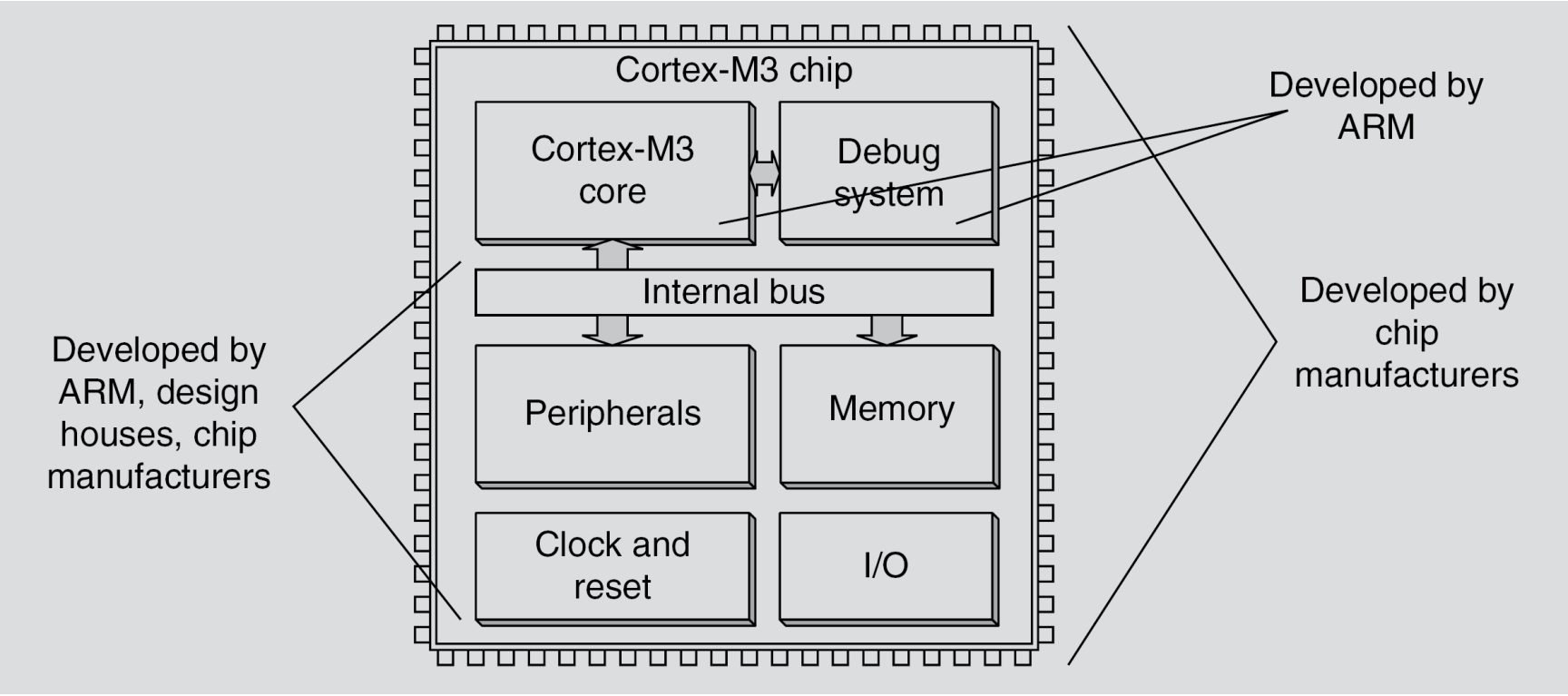
ARM Cortex™-M3 processor

Highlight Features

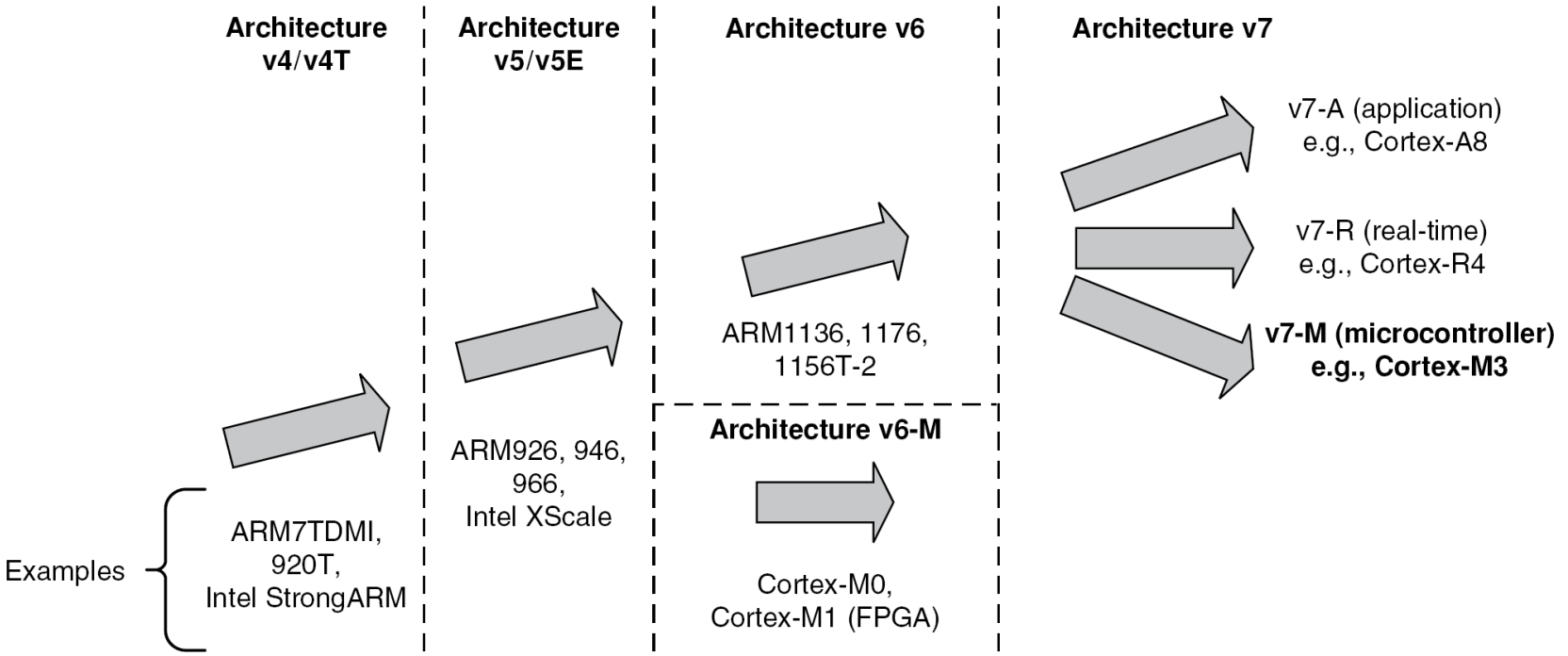
- 32-bit microcontroller
- *Greater performance efficiency*:
  - Allowing more work to be done without increasing the frequency or power requirements
- *Low power consumption*:
  - Enabling longer battery life, especially critical in portable products including wireless networking applications
- *Reduce System cost*:
  - A single, more powerful device can potentially replace three or four traditional 8-bit devices
  - Improving the amount of **code reuse** across all systems

ARM Cortex™-M3 processor

Intellectual Property (IP) Licensing



ARM Architecture Versions



Profile of ARMv7

- The *A profile* is designed for high-performance open application platforms.
  - high-end embedded operating systems (OSs) such as high-end mobile phones. Ex Cortex-A8
- The *R profile* is designed for high-end embedded systems in which real-time performance is needed. Ex. Cortex-R4
  - high-end breaking systems
  - hard drive controllers
- The *M profile* is designed for deeply embedded microcontroller-type systems. Ex. Cortex-M0, Cortex-M3, now Cortex-M4

ARM Processors

Table 1.1 ARM Processor Names

Processor Name	Architecture Version	Memory Management Features	Other Features
ARM7TDMI	ARMv4T		
ARM7TDMI-S	ARMv4T		
ARM7EJ-S	ARMv5E		DSP, Jazelle
ARM920T	ARMv4T	MMU	
ARM922T	ARMv4T	MMU	
ARM926EJ-S	ARMv5E	MMU	DSP, Jazelle
ARM946E-S	ARMv5E	MPU	DSP
ARM966E-S	ARMv5E	DSP	
ARM968E-S	ARMv5E		DMA, DSP
ARM966HS	ARMv5E	MPU (optional)	DSP
ARM1020E	ARMv5E	MMU	DSP
ARM1022E	ARMv5E	MMU	DSP
ARM1026EJ-S	ARMv5E	MMU or MPU	DSP, Jazelle
ARM1136J(F)-S	ARMv6	MMU	DSP, Jazelle
ARM1176JZ(F)-S	ARMv6	MMU + TrustZone	DSP, Jazelle
ARM11 MPCore	ARMv6	MMU + multiprocessor cache support	DSP, Jazelle
ARM1156T2(F)-S	ARMv6	MPU	DSP
Cortex-M0	ARMv6-M		NVIC
Cortex-M1	ARMv6-M	FPGA TCM interface	NVIC
Cortex-M3	ARMv7-M	MPU (optional)	NVIC

ARM Processors

Table 1.1 ARM Processor Names *Continued*

Processor Name	Architecture Version	Memory Management Features	Other Features
Cortex-R4	ARMv7-R	MPU	DSP
Cortex-R4F	ARMv7-R	MPU	DSP + Floating point
Cortex-A8	ARMv7-A	MMU + TrustZone	DSP, Jazelle, NEON + floating point
Cortex-A9	ARMv7-A	MMU + TrustZone + multiprocessor	DSP, Jazelle, NEON + floating point

Instruction Set

Instruction Set Development

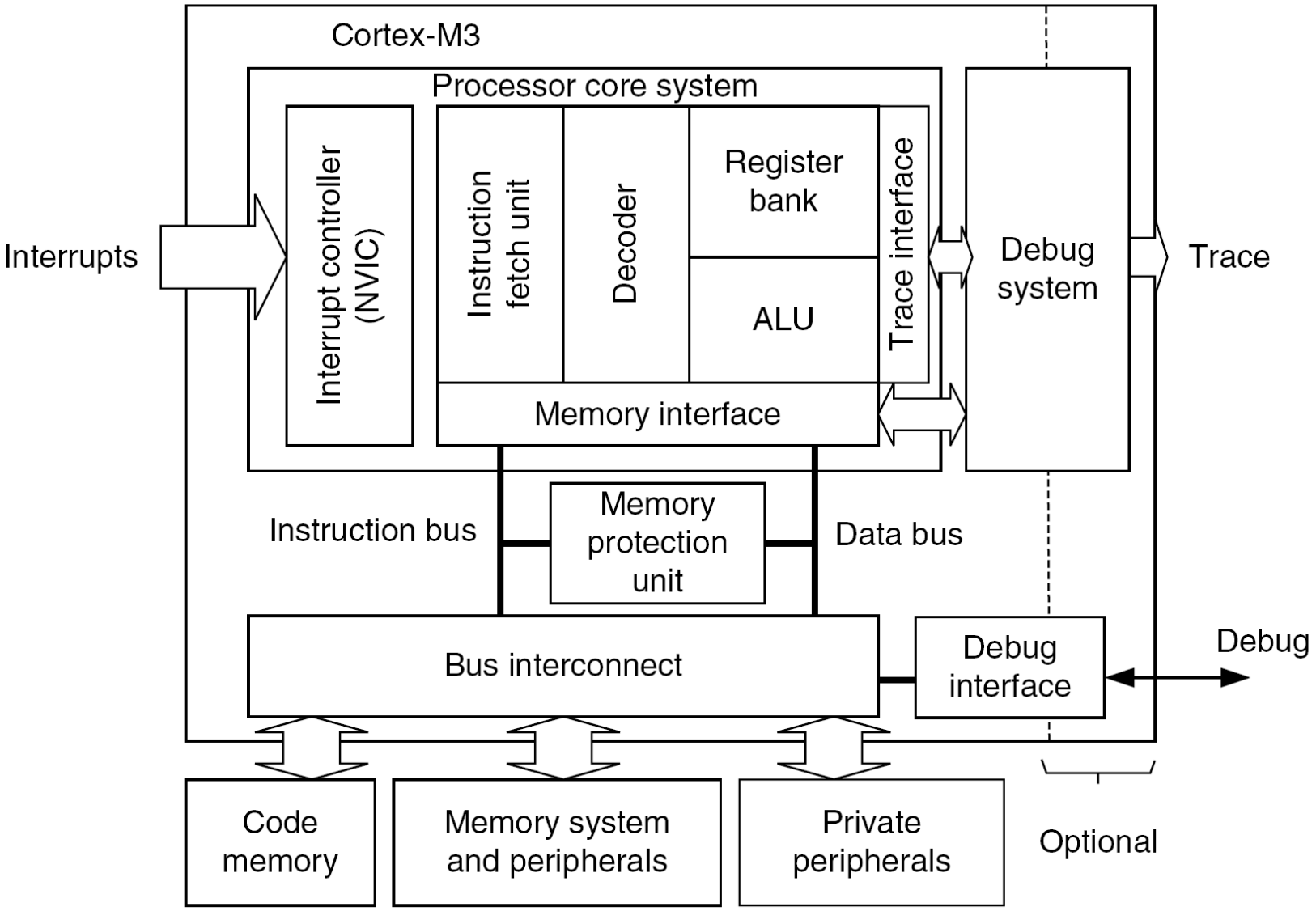
- Two different instruction sets are supported (ARM & Thumb)
- Arm instructions are 32 bits
- Thumb instruction are 16 bits (later Thumb2 contains both 16-bits and 32-bits instructions)
  - Subset of Arm instructions
  - Provide higher code density
- Processor can be dynamically switched between Arm state and Thumb state

Reference Manuals

Further information

- *The Cortex-M3 Technical Reference Manual* (*TRM*)
  - provides detailed information about the processor, including programmer's model, memory map, and instruction timing.
- *The ARMv7-M Architecture Application Level Reference Manual*
  - contains detailed information about the instruction set and the memory model.
- *Cortex-M3 User Guides*
  - contains a programmer's model for the ARM Cortex-M3 processor, and instruction set details, and is customized by each MCU vendors to match their microcontroller implementations.
- *ARM Application Note 179: Cortex-M3 Embedded Software Development*
  - C programming tips for Cortex-M3

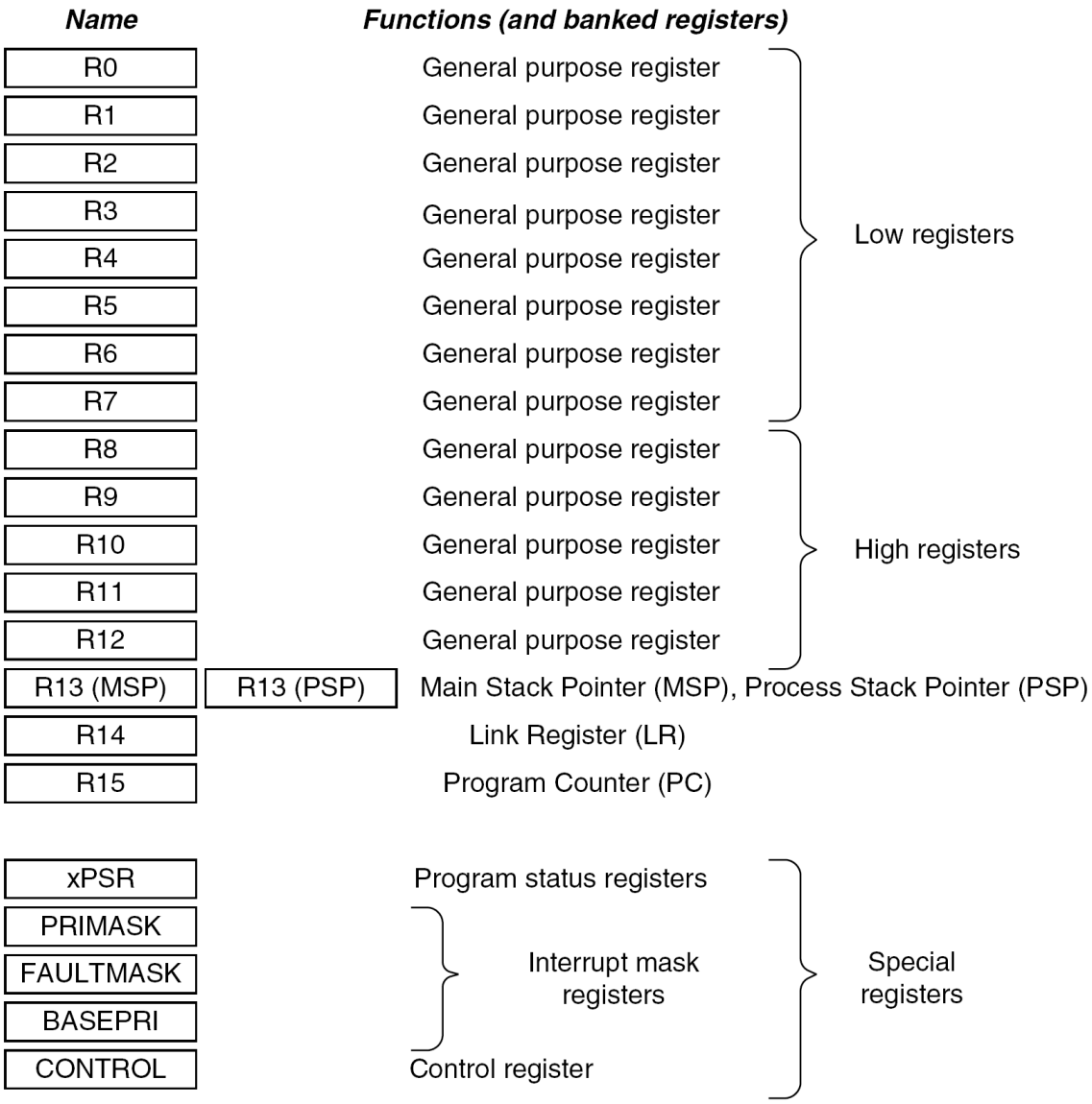
Cortex-M3: Simplify View



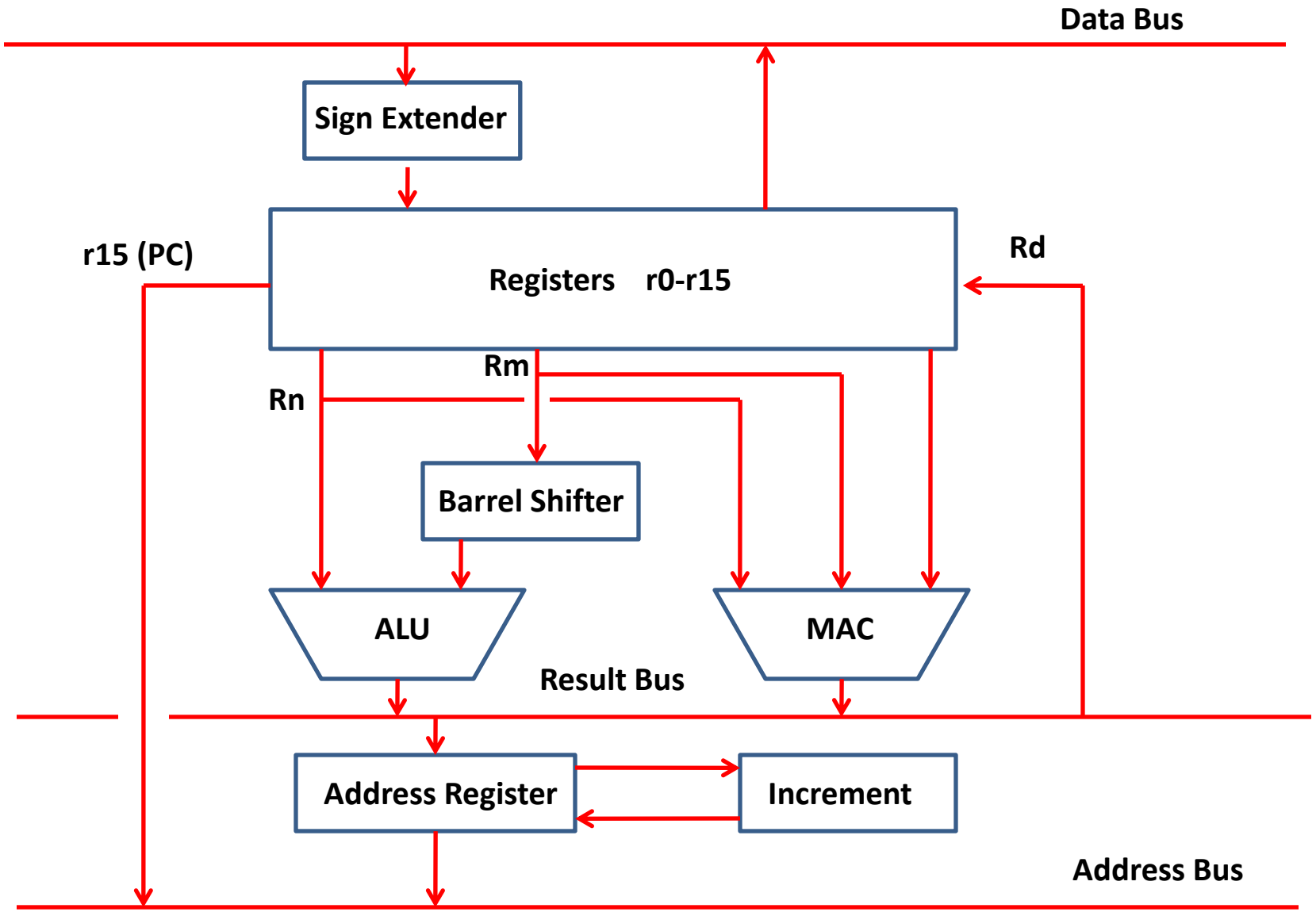
Cortex-M3: Features

- 32-bit microprocessor
- 32-bit data path
- 32-bit register bank
- 32-bit memory interface
- Harvard architecture i.e. separate instruction bus and data bus
  - Multiple bus interfaces : optimized usage and used simultaneously
- Support both **little endian** and **big endian** memory systems
- Debugging:
  - Fixed internal components such as **breakpoints** and **watch points**
  - Optional components: such as **instruction trace**

Cortex-M3 : Registers



ARM Cortex-M3 : Internal Organization



ARM Cortex-M3 : Internal Organization

Internal organization

- All data inside CPU are always **32-bit wide**
  - A hardware sign extender convert 8-bit, 16-bit operand to full 32-bit
- Use **load/store** architecture
  - Instruction can only operate on the contents of register, not on the contents of the memory location
- A **barrel shifter** allows operand in Rm to be shifted arithmetically before being combined with in Rn simplifies the calculation of certain multiplication.
- **Memory Address Calculator** prepare subscripted and relative address reference for memory access

Cortex M3 : Instruction Pipelining

Increasing instruction throughput