06 – The Control Processing Unit

- Architecture and Components
- Instruction Coding and Microprogramming
- Control words of the CPU
- Registers – definition, use
  - Data,
  - Address
  - Status

The Embedded Board

- In embedded devices, all the electronics hardware resides on a board, also referred to as a printed wiring board (PW) or printed circuit board (PCB).
- All of the hardware on an embedded board is located in the hardware layer of the Embedded Systems Model.

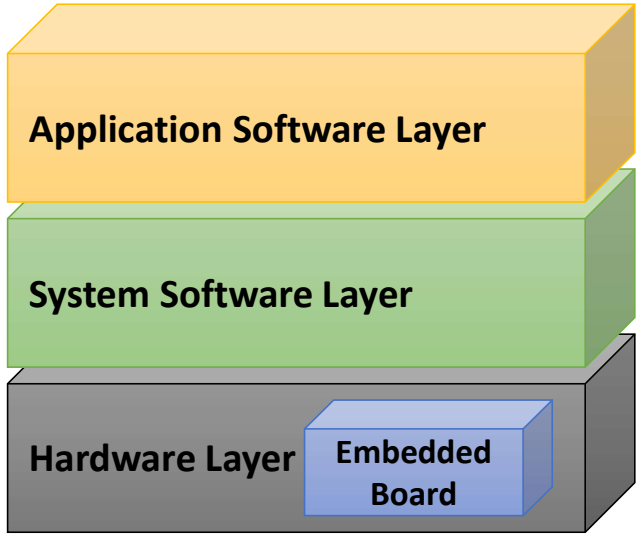


Figure 1: Embedded board and the Embedded Systems Model

Embedded System Board Organization

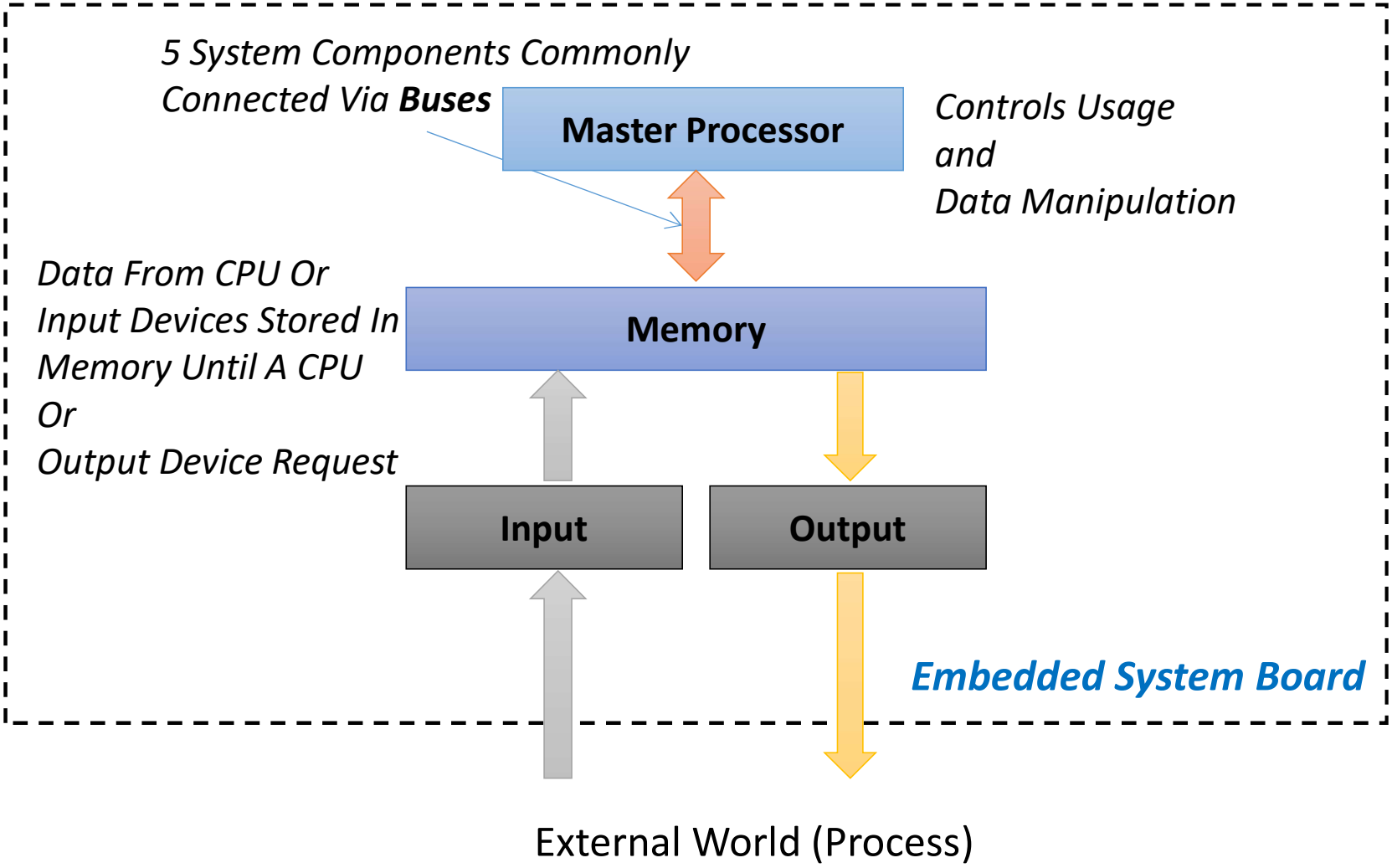


Figure 2: Embedded system board organization

ISA Architecture Models

- An ISA implementation is a determining factor in defining important characteristics of an embedded design, such as
  - performance,
  - design time,
  - available functionality
  - cost.

Operations

- Operations are made up of one or more instructions that execute certain commands.
  - Different processors can execute the exact same operations using a different number and different types of instructions.
- Operations are commonly referred to simply as **instructions**
- An ISA typically defines the types and formats of operations.

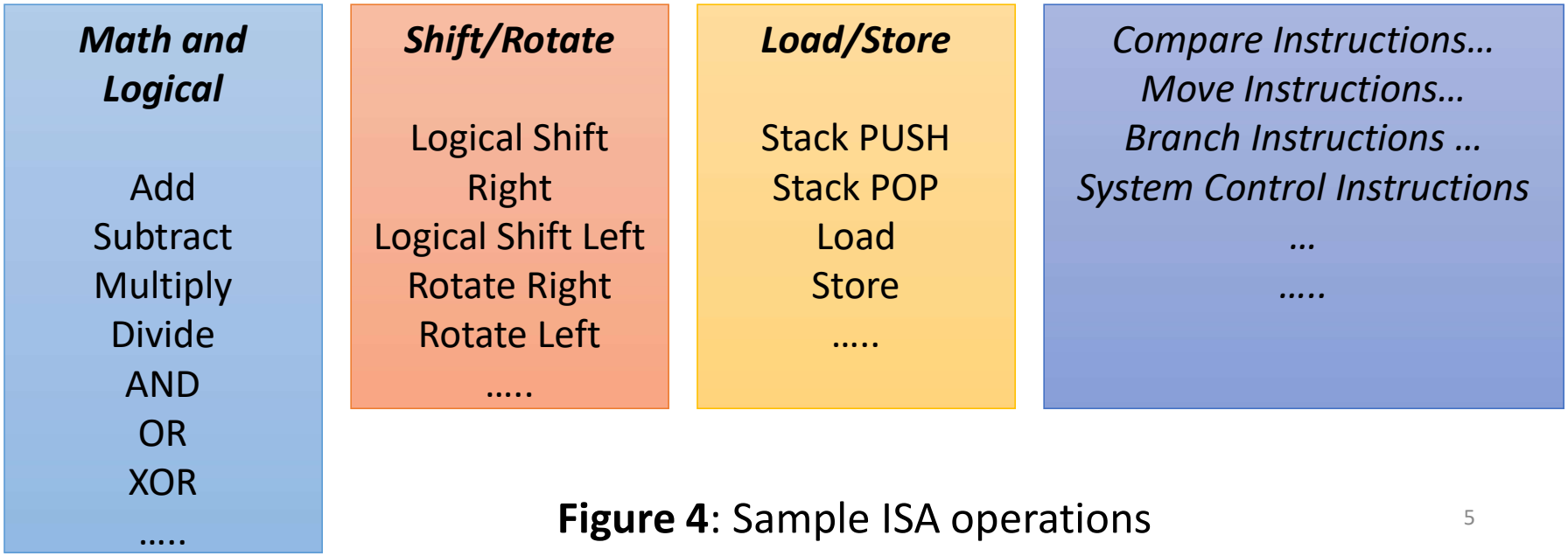


Figure 4: Sample ISA operations

Types of Operations

- Operations typically include
  - **computations** (math operations),
  - **movement** (moving data from one memory location/register to another),
  - **branches** (conditional/unconditional moves to another area of code to process),
  - **input/output operations** (data transmitted between I/O components and master processor), and
  - **context switching operations** (where location register information is temporarily stored when switching to some routine to be executed and after execution, by the recovery of the temporarily stored information, there is a switch back to executing the original instruction stream).

Operands

- Operands are the data that operations manipulate.
- An ISA defines the types and formats of operands for a particular architecture.
  - For MPC823 (Motorola/Freescale PowerPC), SA-1110 (Intel StrongARM), the ISA defines simple operand types of bytes (8 bits), halfwords (16 bits), and words (32 bits).
- More complex data types such as integers, characters, or floating point are based on the simple types shown.

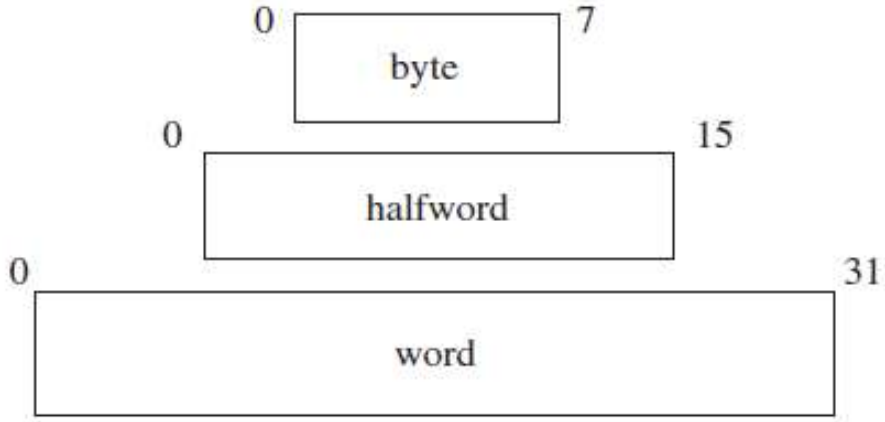


Figure 5: Simple operand types

Operands

- An ISA also defines the operand formats (how the data looks) that a particular architecture can support, such as
  - binary,
  - decimal and
  - hexadecimal.
- Below is an example showing how an architecture can support various operand formats.

```
MOV registerX, 10d        ; Move decimal value 10 into register X
MOV registerX, $0Ah       ; Move hexadecimal value A (decimal 10) to register X
MOV registerX, 00001010b  ; Move binary value 00001010 (decimal 10 ) to register X

.....
```

Figure 6: Operand formats pseudocode example

Storage

- The ISA specifies the features of the programmable storage used to store the data being operated on, primarily:
  A. The organization of memory used to store operands
  B. Register Set
  C. How Registers Are Used

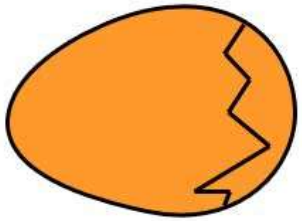
Address Space

- An ISA defines specific characteristics of the address space, such as whether it is:
  - **Linear**.
  - **Segmented**.
  - Containing any **special address regions**.
  - **Limited** in any way.

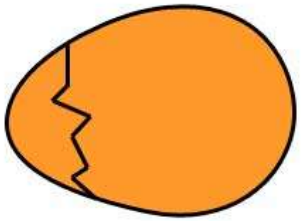
Byte Ordering

- Different ISAs define also how data is stored in memory—specifically in what order the bits (or bytes) that make up the data is stored, or byte ordering.
- The two byte-ordering approaches are:
  - **big-endian**, in which the most significant byte or bit is stored first, and
  - **little-endian**, in which the least significant bit or byte is stored first.
- For example:
  - 68000 and SPARC are big-endian
  - x86 is little-endian
  - ARM, MIPS and PowerPC can be configured as either big-endian or little-endian using a bit in their machine state registers

Byte Ordering

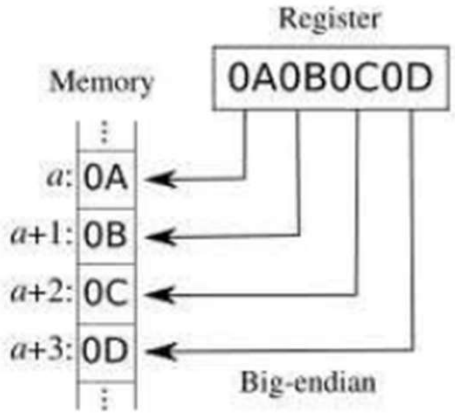


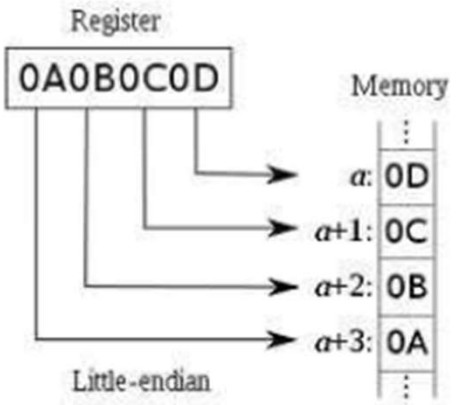
BIG ENDIAN - The way people always broke their eggs in the Lilliput land



LITTLE ENDIAN - The way the king then ordered the people to break their eggs

Big Endian vs. Little Endian





Architecture Examples

- Commonly used embedded processors support 4-bit, 8-bit, 16-bit, 32-bit, and/or 64-bit processing.
  - Some processors can process larger amounts of data and can access larger memory spaces in a single instruction, such as 128-bit architectures, but they are not commonly used in embedded designs.

Table 1: "x-bit" architecture examples

"x"-Bit	Architecture
4	Intel 4004, …
8	Mitsubishi M37273, 8051, 68HC08, Intel 8008/8080/8086,…
16	ST ST10, TI MSP430, Intel 8086/286,…
32	68K, PowerPC, ARM, x86 (386+), MIPS32,…
64	Intel 64 – Sandy Bridge / Ivy Bridge
128	Some CPUs have 128 bit registers; (IPv6)

Central Processing Unit (CPU)

- The semantics of this section can be a little confusing, because processors themselves are commonly referred to as CPUs
  - it is actually the **processing unit within a processor that is the CPU**.
- The CPU is responsible for executing the cycle of
  - **fetching**,
  - **decoding**, and
  - **executing instructions** (see next figure).
- This three-step process is commonly referred to as **a three stage pipeline**, and most recent CPUs are pipelined designs.

Fetch, decode and execution cycle of CPU

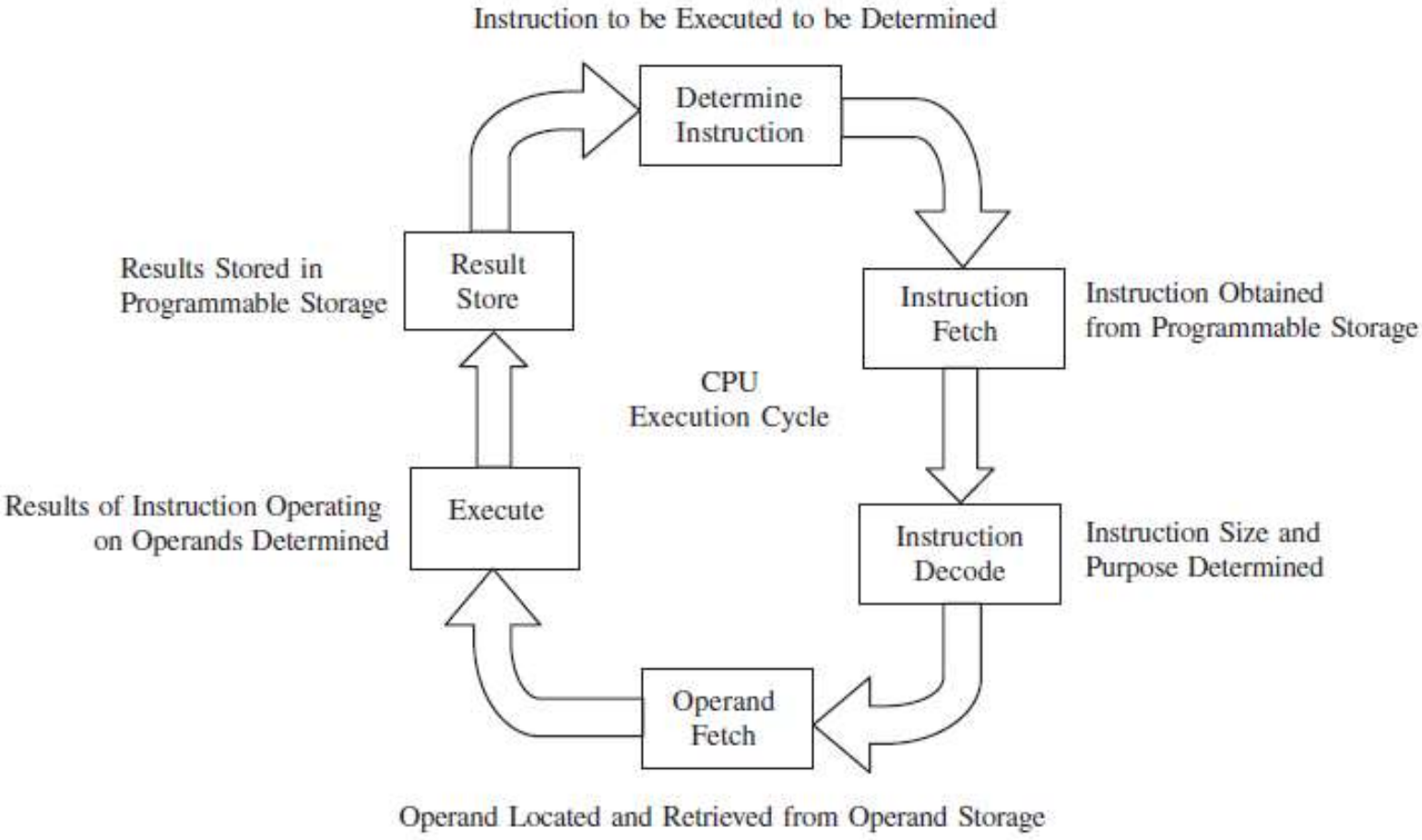
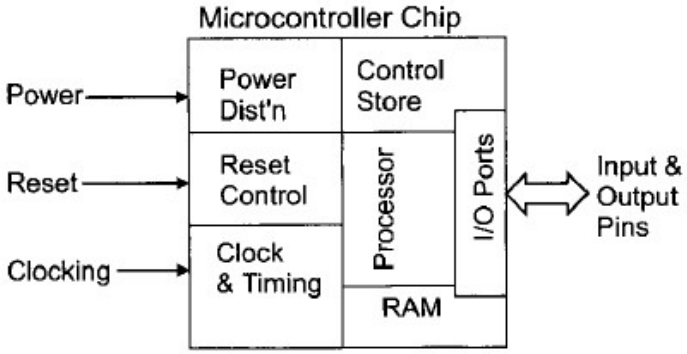
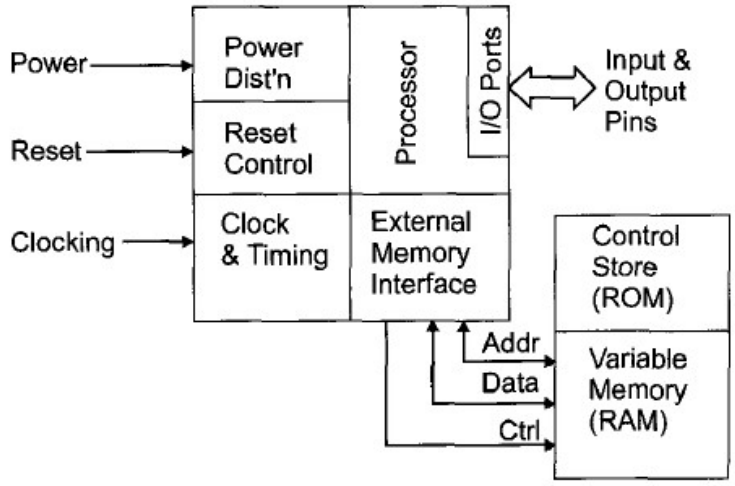


Figure 12: Fetch, decode and execution cycle of CPU

A Typical Microcontroller



No external Memory



External Memory Added

Figure 18: Microcontroller with built-in and external memory

Microchip 16-bit Product Families

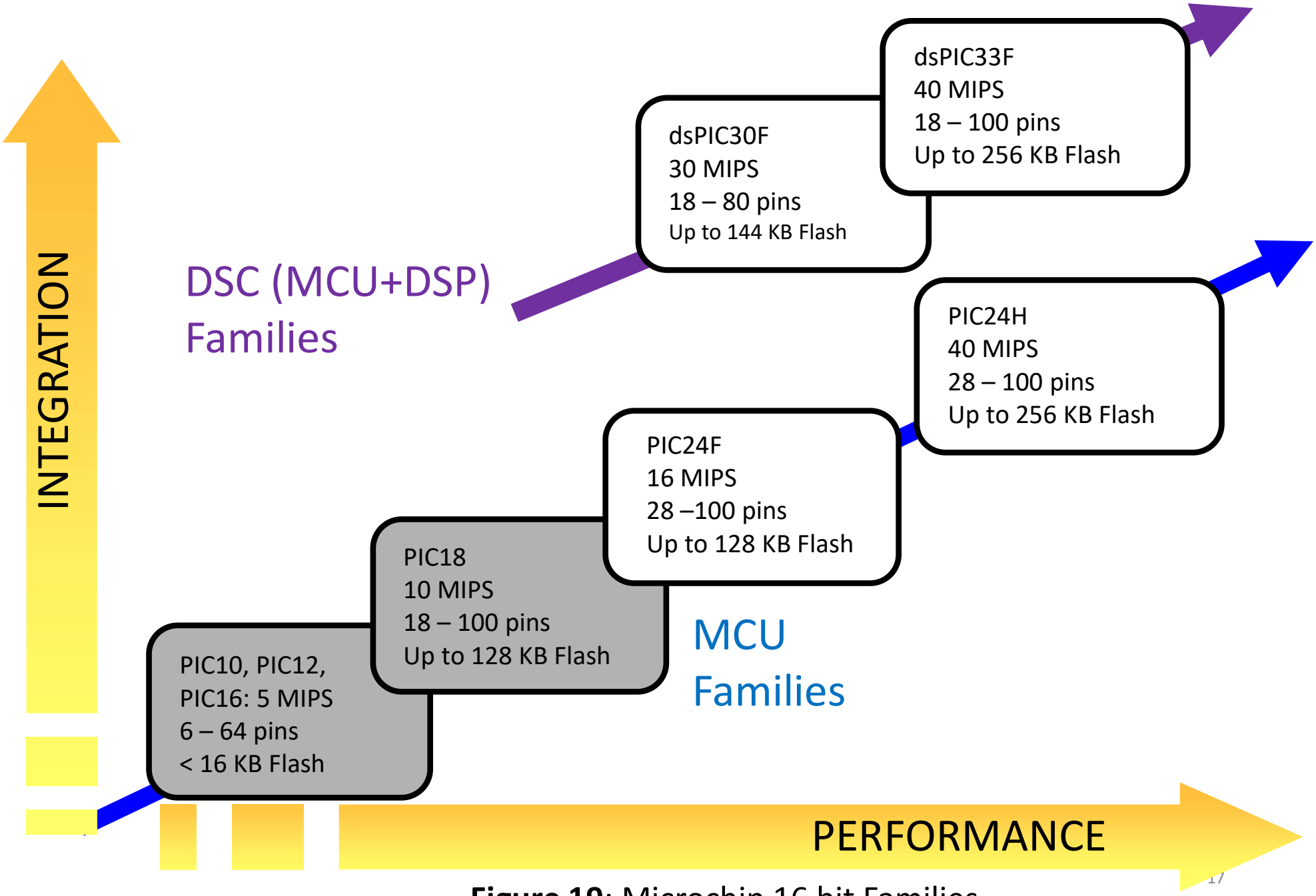
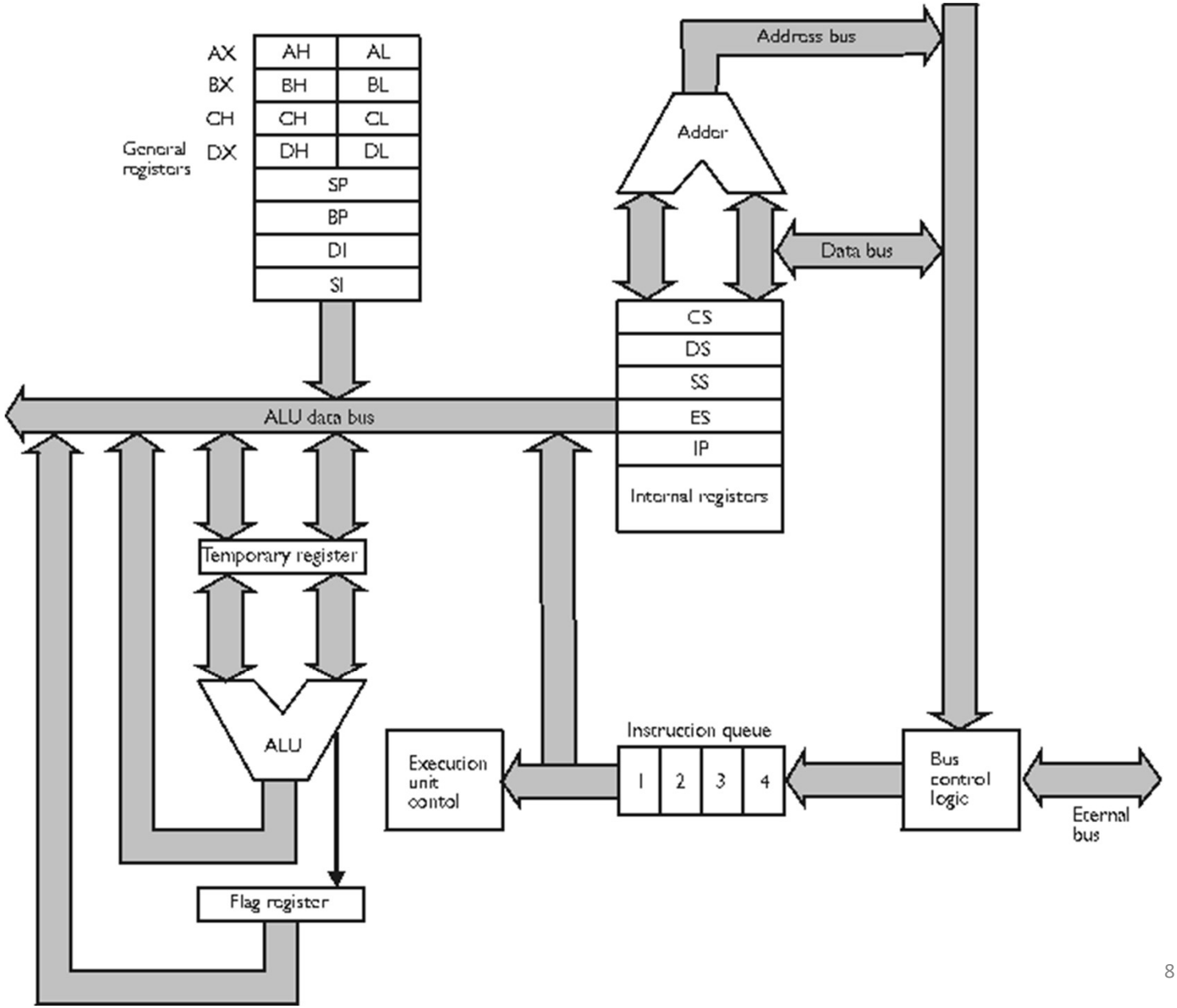
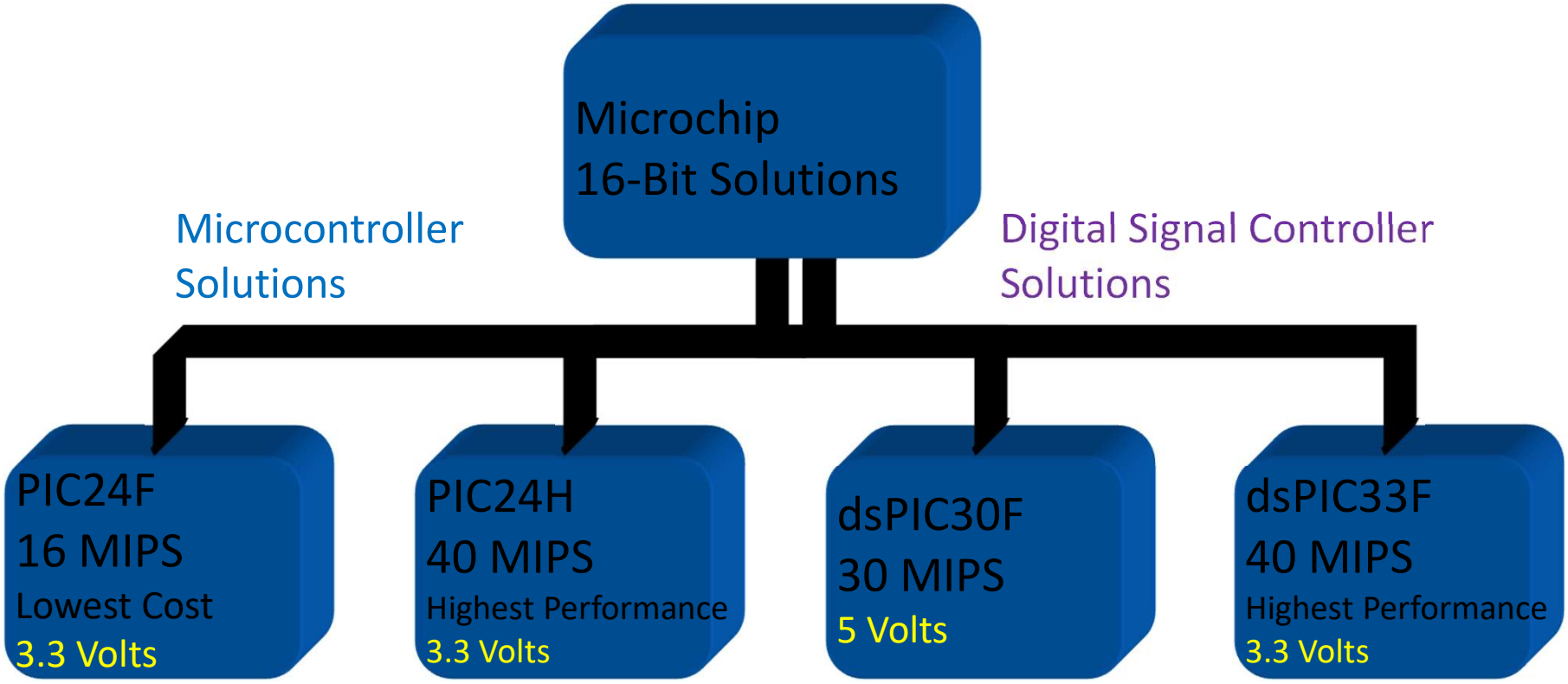


Figure 19: Microchip 16 bit Families

8086 Programming Model



Four 16-Bit Performance Options



PIC18 Migration
More Memory
More Peripherals

32-bit Competitor
DMA
More Performance
Than ARM7

DSP Instructions
5-Volt Analog

DSP Instructions
Improved Analog
Fast Math
DMA

Scale Performance in MIPS & Add Computational Math Across 4 Code-compatible Families of Products

PIC24F Family