Introduction Memory Organization

Embedded Software Essentials

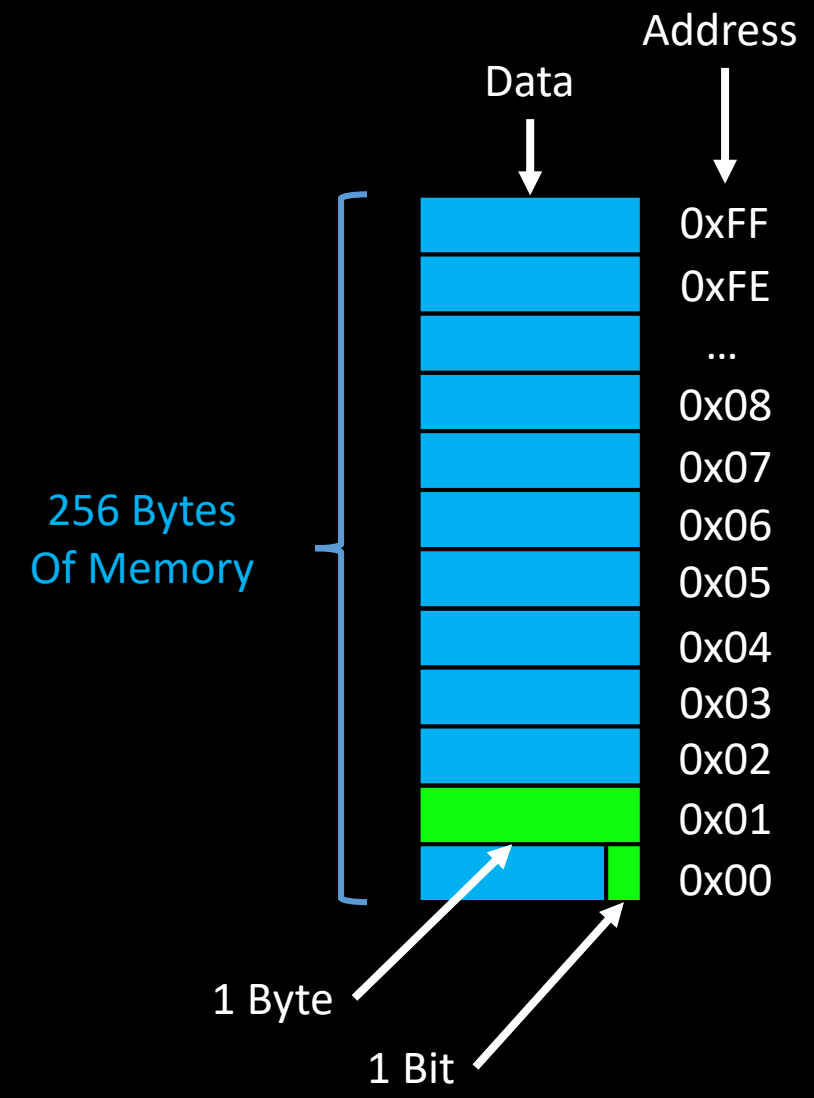
C1M3V1

Memories

- Bit – Building Block of Memory
  - Stores 1 piece of Boolean information (0 or 1)
- Byte – 8 Bits
  - Usually minimum unit for access

Memory Scale Examples:

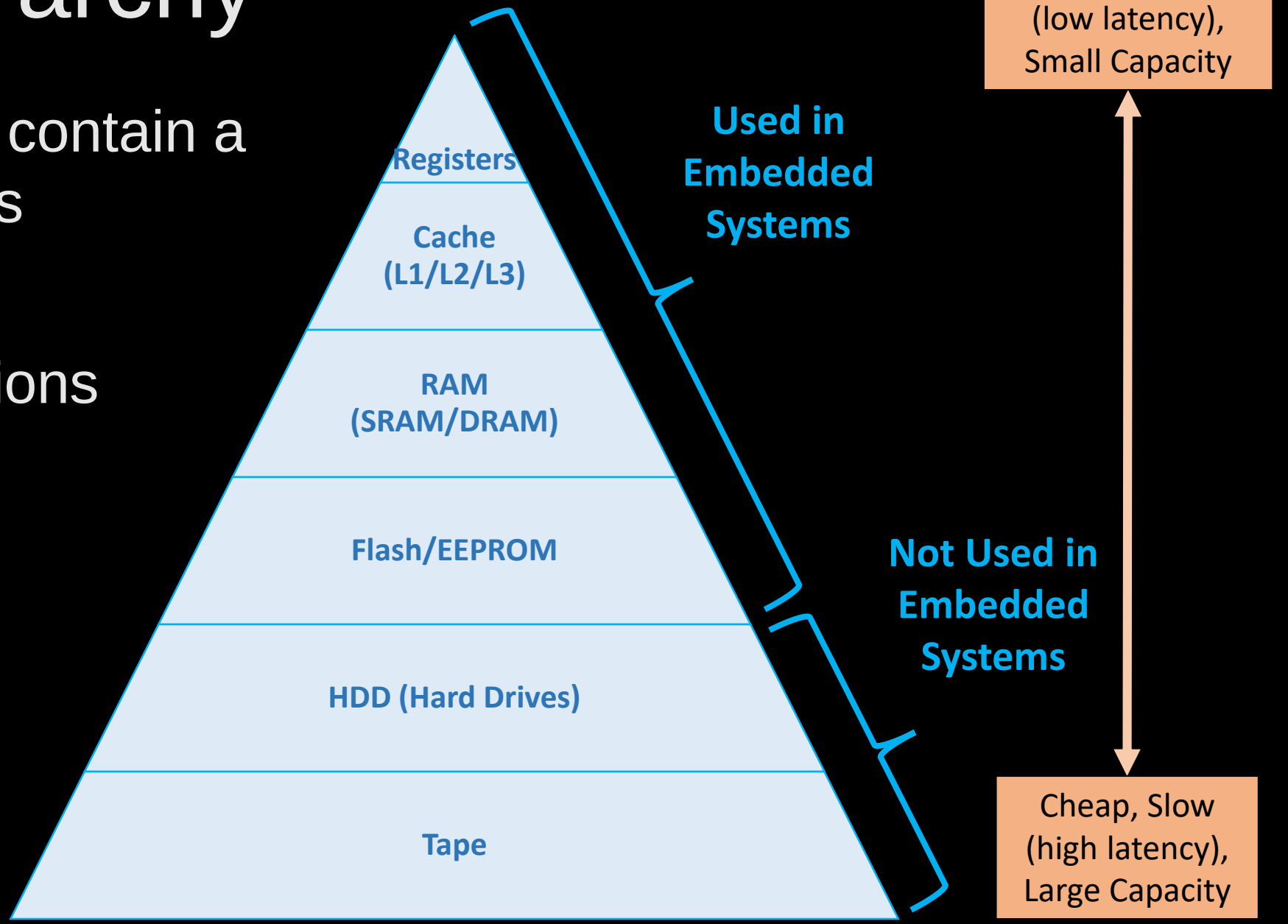
- Data Centers → PetaBytes
- Personal Computers → MegaBytes – TeraBytes
- Embedded Systems → KiloBytes - MegaBytes



Memory Hierarchy

- Computer Systems contain a mixture of memories

- Memory considerations Technology
  - Capacity
  - Power
  - Speed/Latency
  - Price



Memory Sizes

- Manufacturers release multiple chips from a family with varying features and memory sizes
  - Flash & SRAM
- Accommodates different applications with a different chip cost

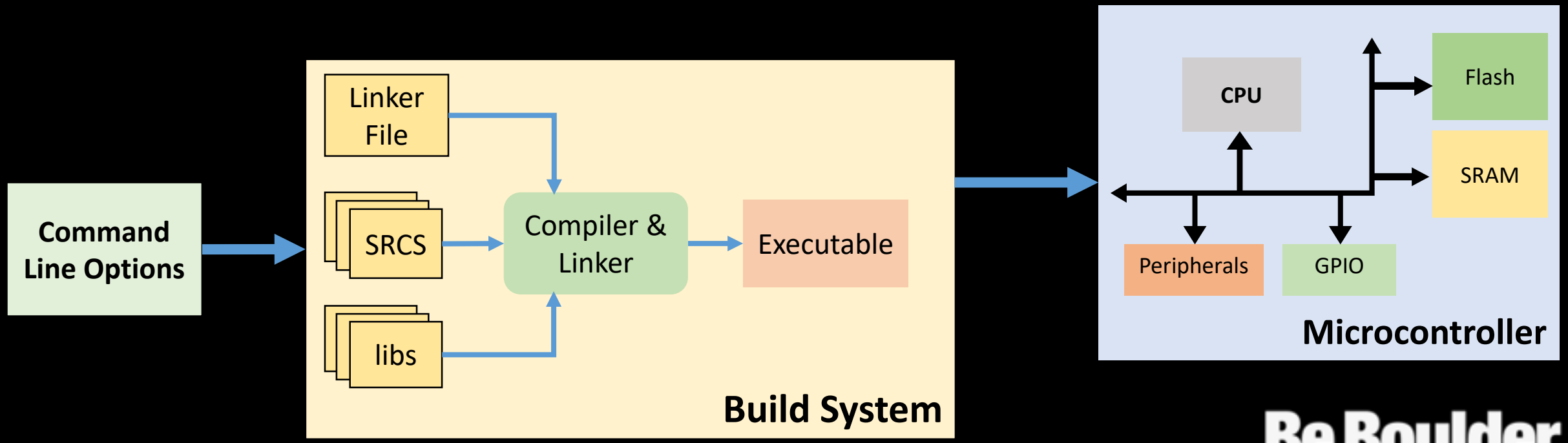
Different KL2x Chips have:

- **SRAM Ranges from 4 KB – 32 KB**
- **FLASH Ranges from 32 KB – 256 KB**

Sub-Family	Part Number	CPU (MHz)	Memory		Features												
			Flash (KB)	SRAM (KB)	DMA	Low-Power UART	UART	ISO7 816-3	SPI	I²C	TSI	I²S	Flex IO	RTC	12-bit DAC	16-bit ADC w/DP Ch.	12-bit ADC	Total I/Os
KL24	MKL24Z32xxx4	48 MHz	32	4	√	1	2		2	2				√			√	23~66
	MKL24Z64xxx4	48 MHz	64	8	√	1	2		2	2				√			√	23~66
KL25	MKL25Z32xxx4	48 MHz	32	4	√	1	2		2	2	√			√	√	√		23~66
	MKL25Z64xxx4	48 MHz	64	8	√	1	2		2	2	√			√	√	√		23~66
	MKL25Z128xxx4	48 MHz	128	16	√	1	2		2	2	√			√	√	√		23~66
KL26	MKL26Z32xxx4	48 MHz	32	4	√	1	2		2	2	√	√		√	√	√		23~50
	MKL26Z64xxx4	48 MHz	64	8	√	1	2		2	2	√	√		√	√	√		23~50
	MKL26Z128xxx4	48 MHz	128	16	√	1	2		2	2	√	√		√	√	√		23~80
	MKL26Z256xxx4	48 MHz	256	32	√	1	2		2	2	√	√		√	√	√		50~80
KL27	MKL27Z128xxx4	48 MHz	128	32	√	2	1	1	2x 16b	2		√	√	√	√	√		23~50
	MKL27Z256xxx4	48 MHz	256	32	√	2	1	1	2x 16b	2		√	√	√	√	√		23~50
	MKL27Z32xxx4	48 MHz	32	8	√	2	1	1	2x 16b	2			√	√		√		23~50
	MKL27Z64xxx4	48 MHz	64	16	√	2	1	1	2x 16b	2			√	√		√		23~50

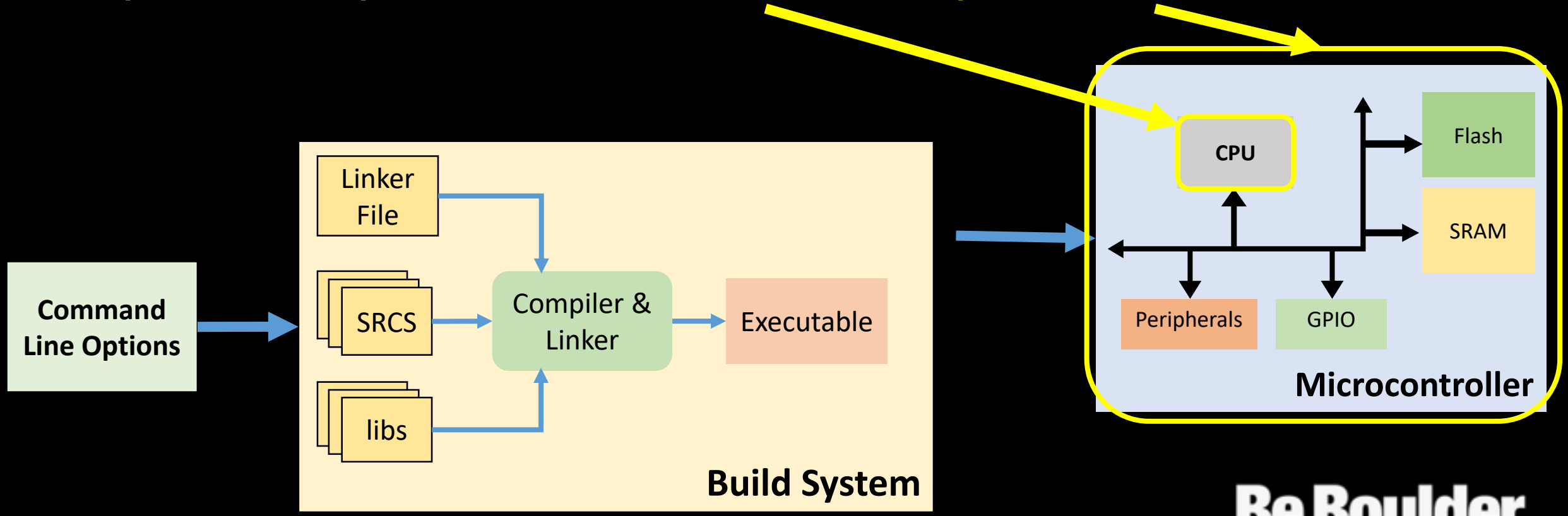
Choosing a Platform

- Executable Program consists of program code and program data compiled for a particular architecture and platform



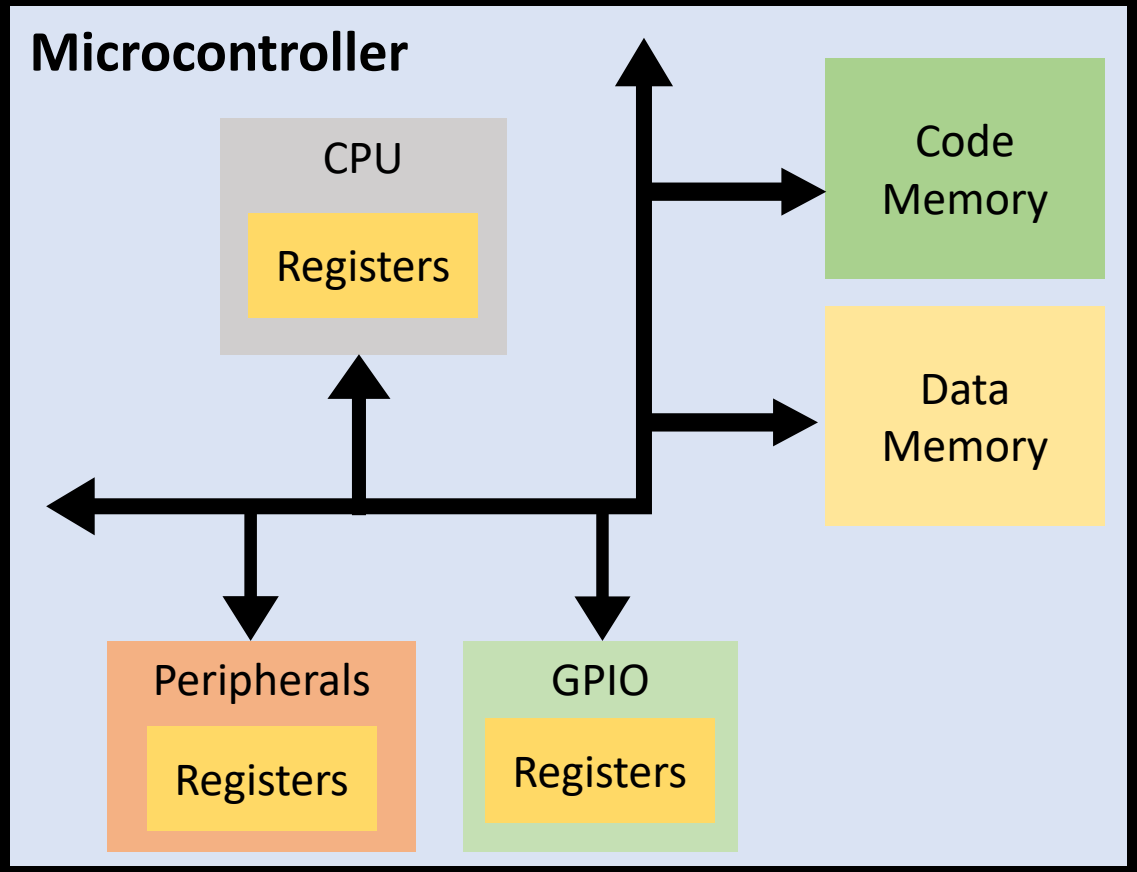
Choosing a Platform

- Executable Program consists of program code and program data compiled for a particular architecture and platform



Memory

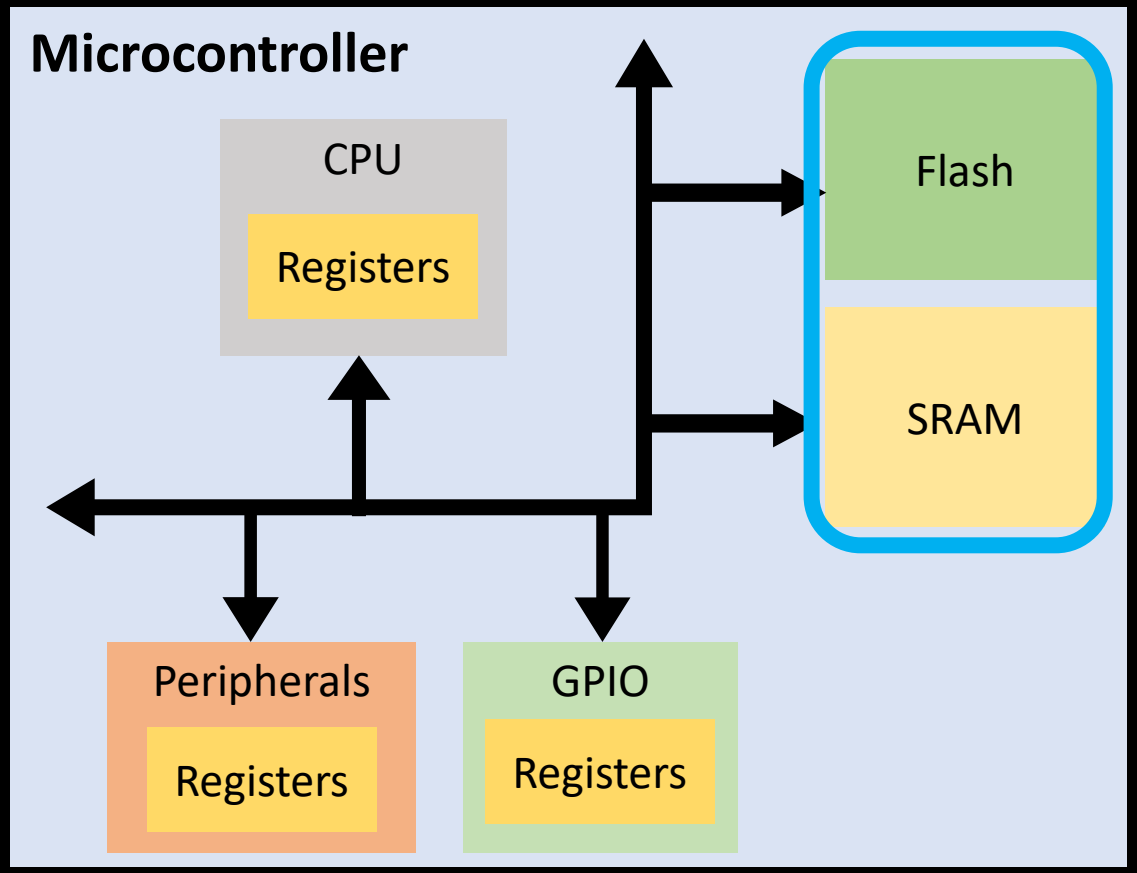
- Three types of stoage needed for a program
  - Code Memory
  - Data Memory
  - Runtime State of Program



Memory

- Three types of stoage needed for a program
  - Code Memory
  - Data Memory
  - Runtime State of Program

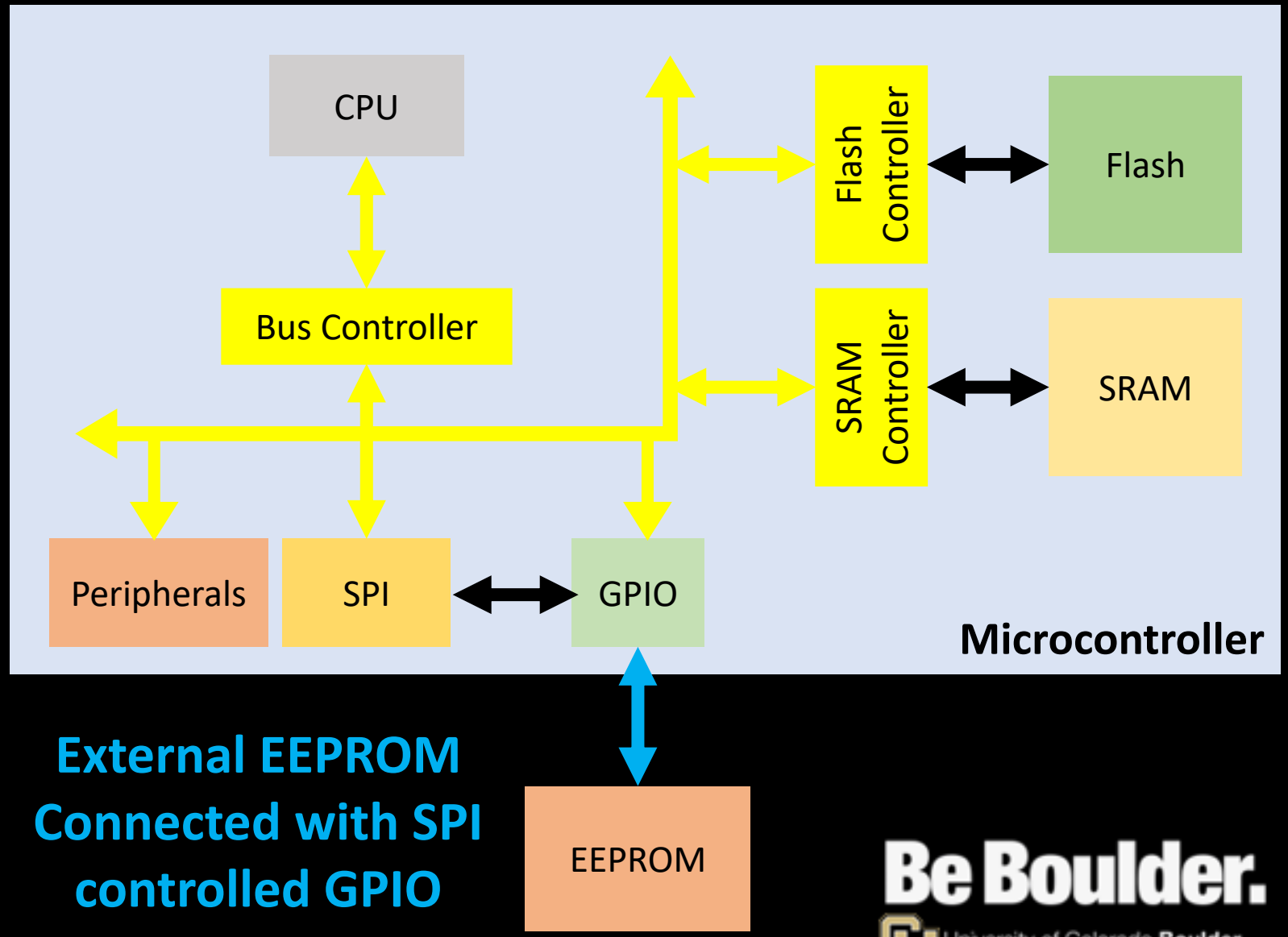
- Compilation tracks and maps memory from program code and program data into segments
  - Code Segment (Flash)
  - Data Segment (SRAM)



Specified in the Linker File

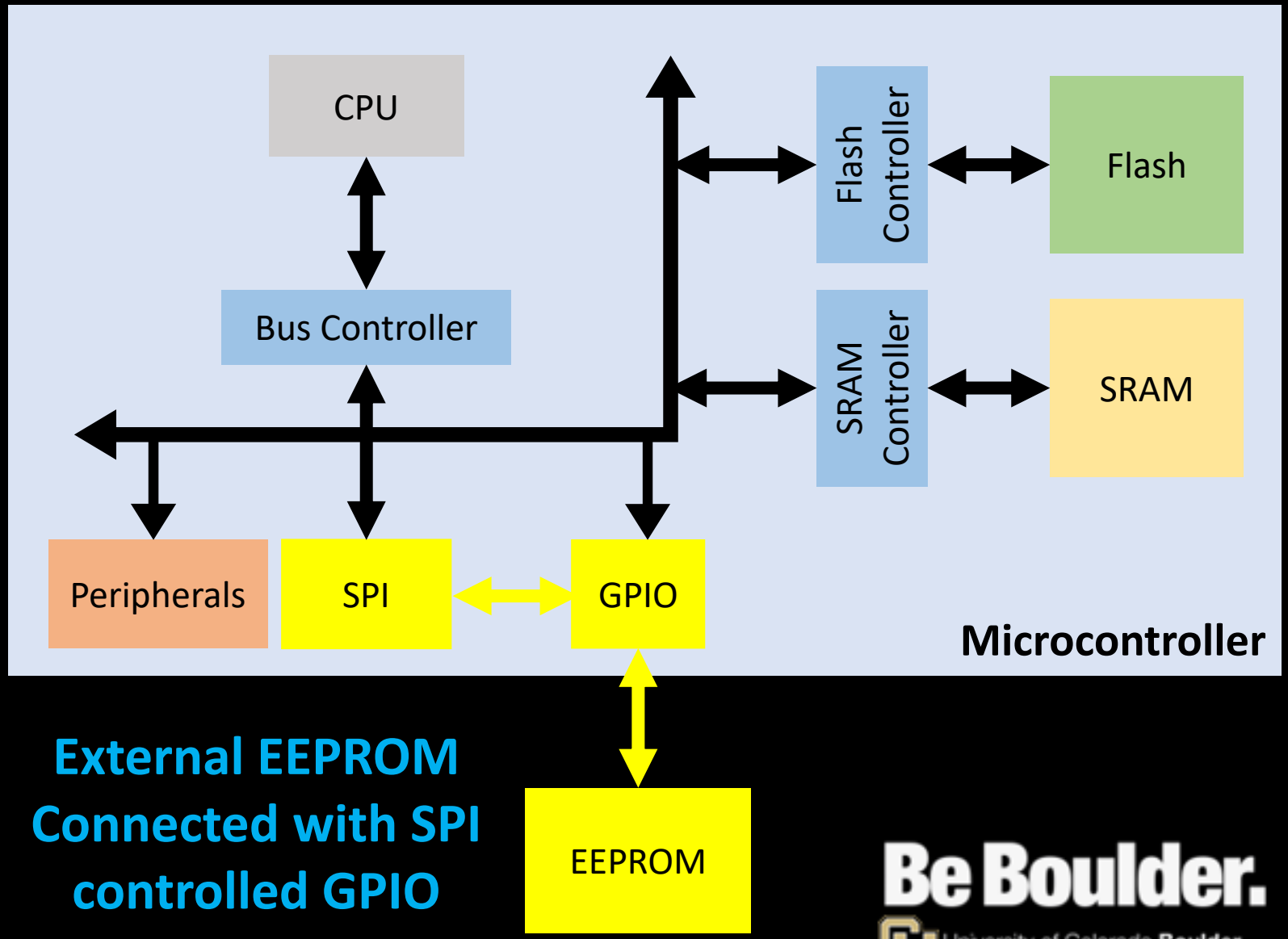
Memory

- Flash and SRAM memory require a controller to manage interface to CPU
- Internal Bus also has a controller
- External memory can be added if more memory is needed
  - EEPROM



Memory

- Flash and SRAM memory require a controller to manage interface to CPU
- Internal Bus also has a controller
- Additional external memory can be connected through a I/O pins if more memory is needed
  - EEPROM

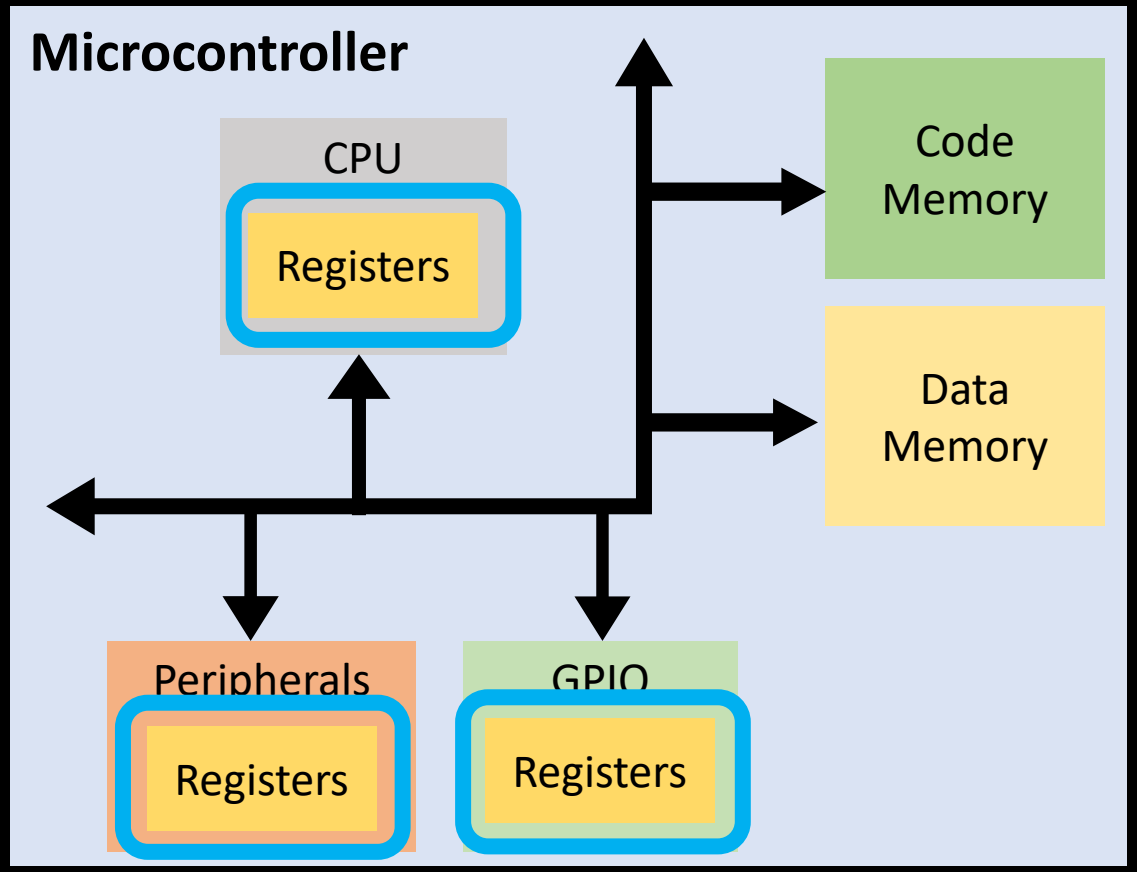


External EEPROM Connected with SPI controlled GPIO

Memory

- Three types of stoage needed for a program
  - Code Memory
  - Data Memory
  - Runtime State of Program

- The CPU and peripherals contain register memory
  - CPU Registers
    - General Purpose
    - Special Purpose
  - Peripheral Registers

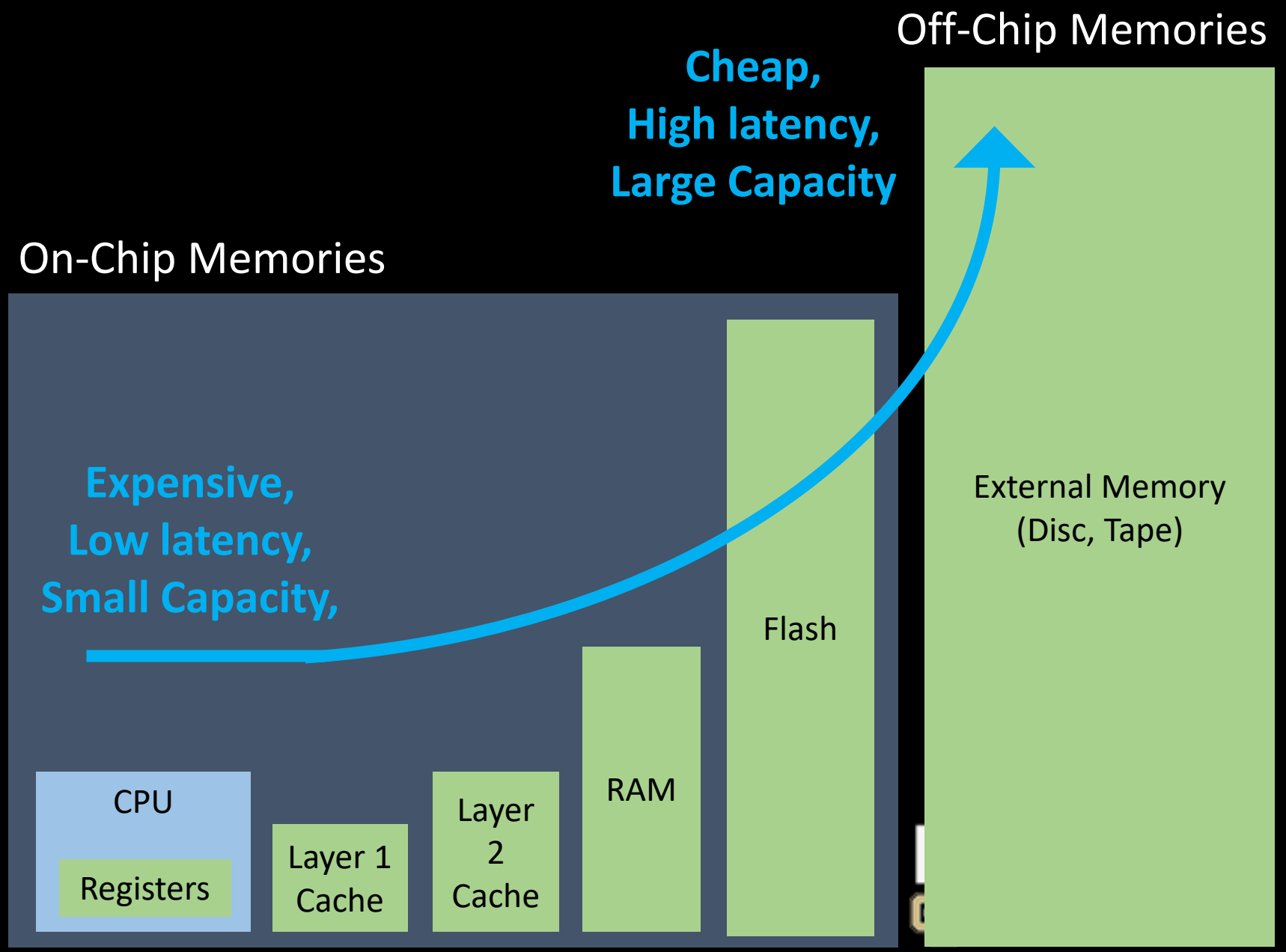


Tradeoffs

- Want memory with:
  - High Capacity
  - Low Latency
  - Low Power
  - High Reliability

…but

Usually cannot get all of these without high cost!!!



Outcomes

- *At the end of this module, students will be able to…*
  - Understand an embedded system memory organization
  - Write C-programs to define code and data in different memory regions
  - Analyze the memory space of a program