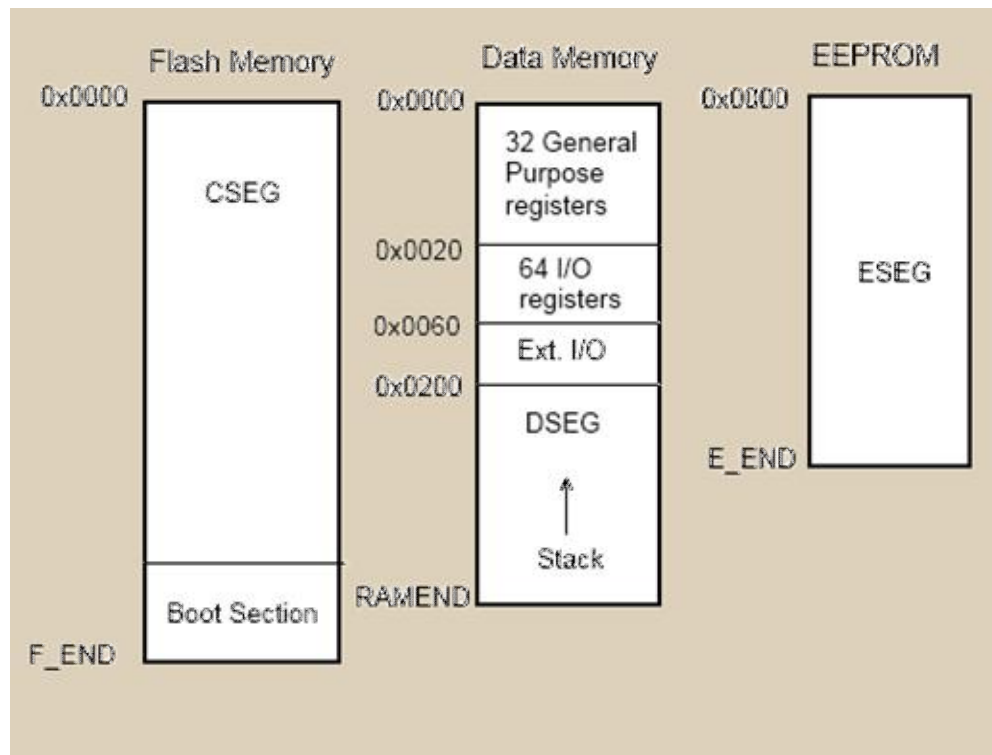


# AVR Memory Architecture

# Overview

- Harvard architecture



# Overview

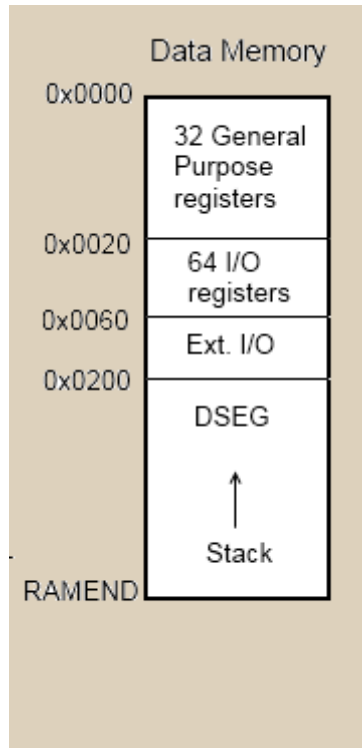


- Flash
  - ❑ Bootloader
  - ❑ CSEG

# Overview

- EEPROM(Electrically Erasable Programmable Read-Only Memory)
  - ESEG

# Overview



- SRAM
  - ❑ General-purpose Registers
  - ❑ I/O Registers
  - ❑ DSEG

# AVR Register

- The first 32 bytes of RAM are general-purpose registers

- 8 bits capacity, the least significant bit starts with 0

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

- Can be used directly in assembly commands

e.g. `MOV R20,R18`

- Only load and store instructions access RAM

# AVR Register

- 32 general-purpose Registers:

R0                      Temporary register - use in interrupts not recommended.

R1                      Zero register - can be used for temporary data but must be set to zero after use.

R18-R27,  
R30-R31                These are general purpose registers and don't need to be saved when using in conjunction with 'C' code.

R2-R17,  
R28-R29                These are general purpose registers but do need to be saved when using in conjunction with 'C' code.

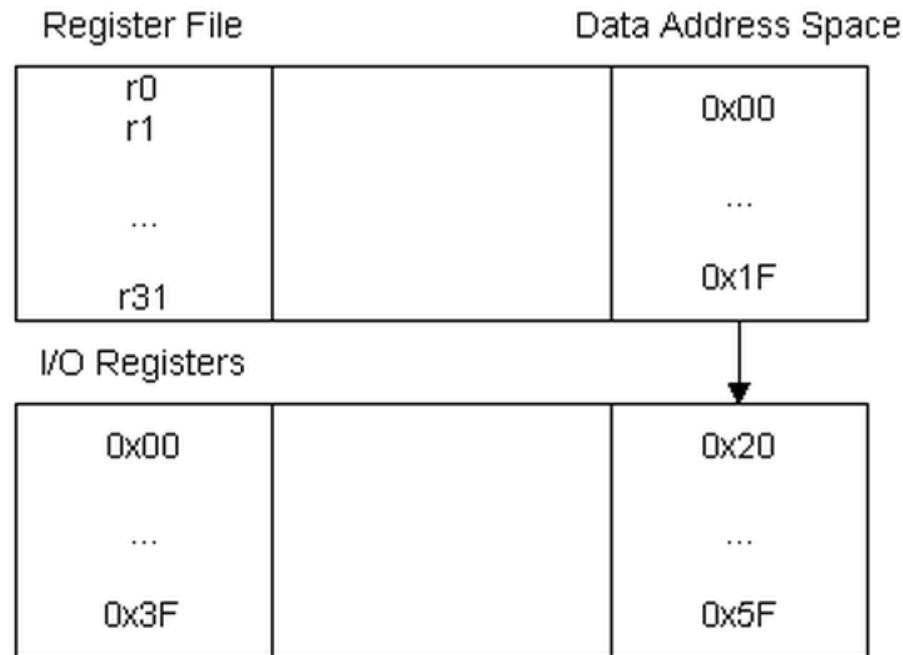
# AVR Register

- Special registers :
  - R25:R24 : return value of a function call
  - R29:R28 : Stack Frame Pointer (Y pointer)



# AVR Register

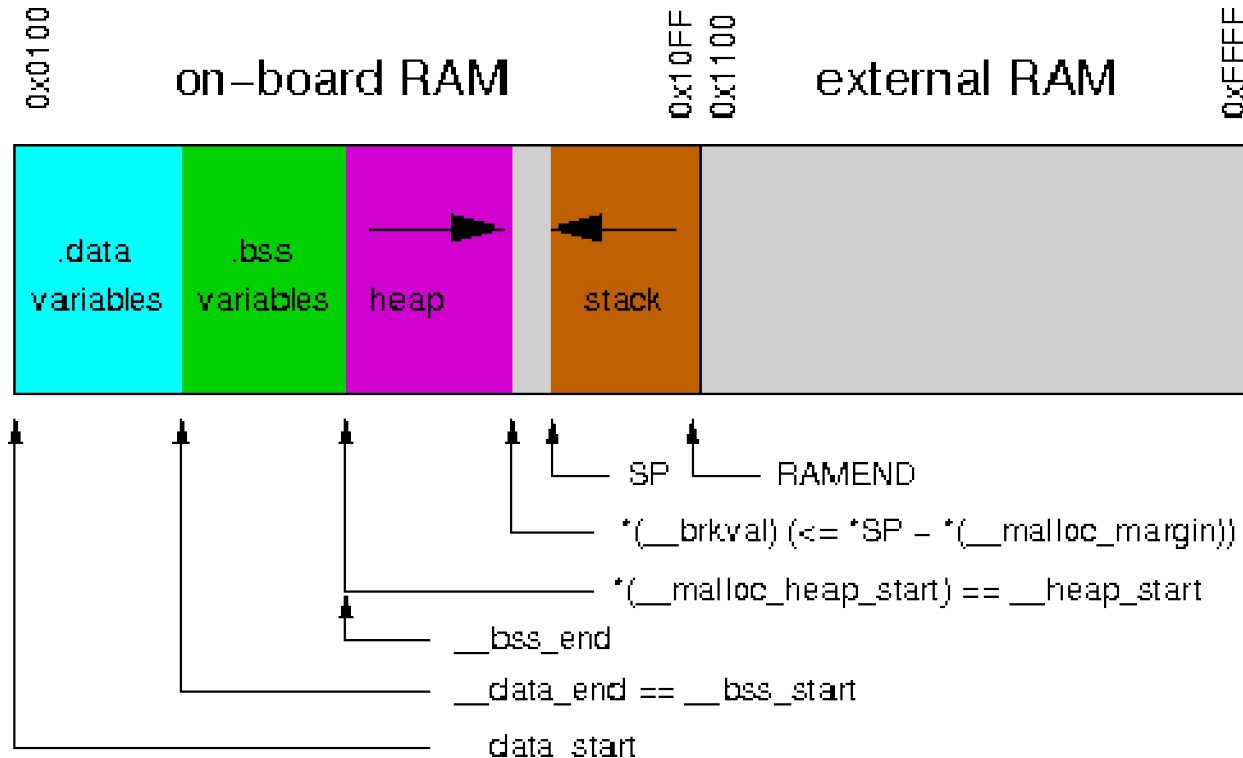
- The next 64 bytes of RAM are I/O registers
  - written as "0x00 (0x20)" through "0x3F (0x5F)"



# AVR Register

- Special I/O registers :
  - 0x3E:0x3D (0x5E:0x5D): Stack Pointer (SP, which indicates the current top of the stack)

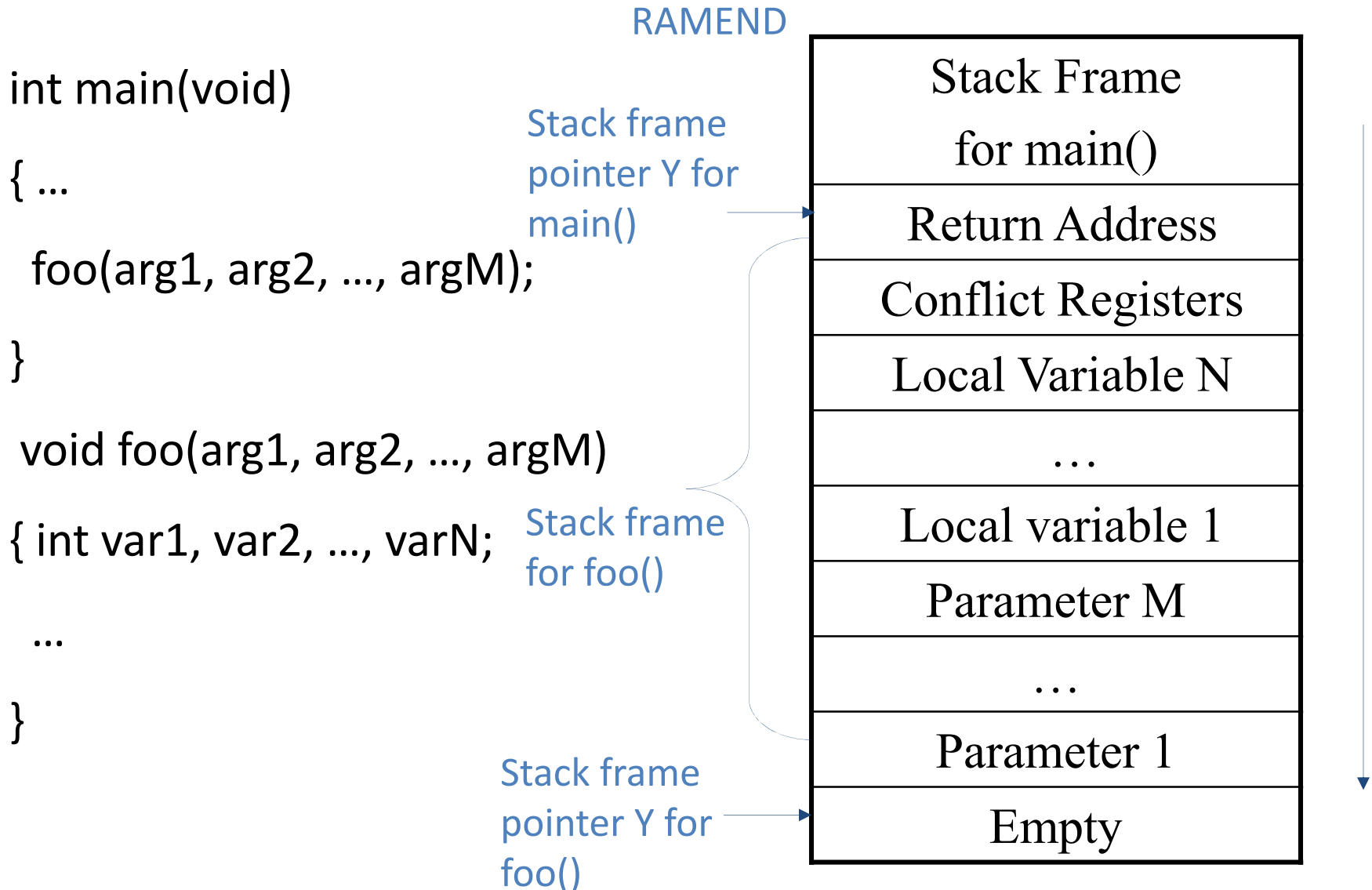
# AVR Memory Model



# AVR Stack Structure

- A stack consists of stack frames
- A Stack frame is a region in the stack used by a function
- Created whenever a function is called
- Freed whenever the function returns.
- What's inside a stack frame?

# AVR Stack Structure



# Why Stack Frame?