Lecture 2

Introduction to Microcontrollers

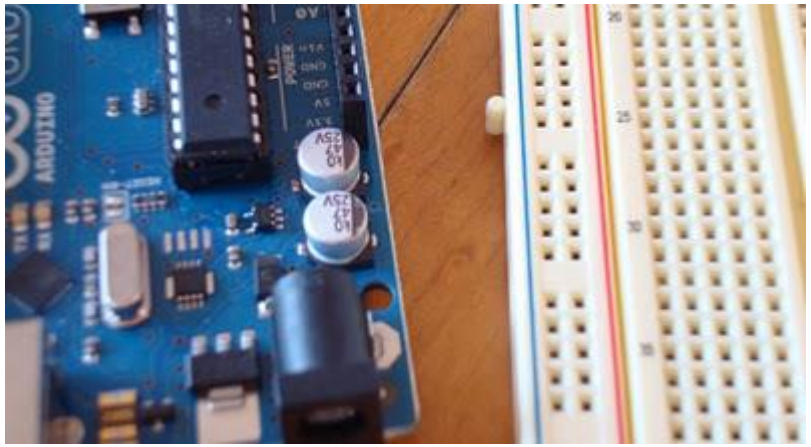
Microcontrollers

- Microcontroller
  - CPU + ++++++++++
- Microprocessor
  - CPU (on single chip)

What is a Microcontroller

- Integrated chip that typically contains integrated CPU, memory (RAM ROM), I/O ports on a single Chip.
- System on a single Chip/ small computer on a single chip
- Designed to execute a specific task to control a single system
- Smaller & Specified (design cost)
- Differs from Microprocessor
  - general-purpose chip
  - Used to design multi purpose computers or devices
  - Require Multiple chips to handle various tasks
- Typically Microcontroller embedded inside some device
- Microcontrollers are important part of Embedded systems

What is a Development Board

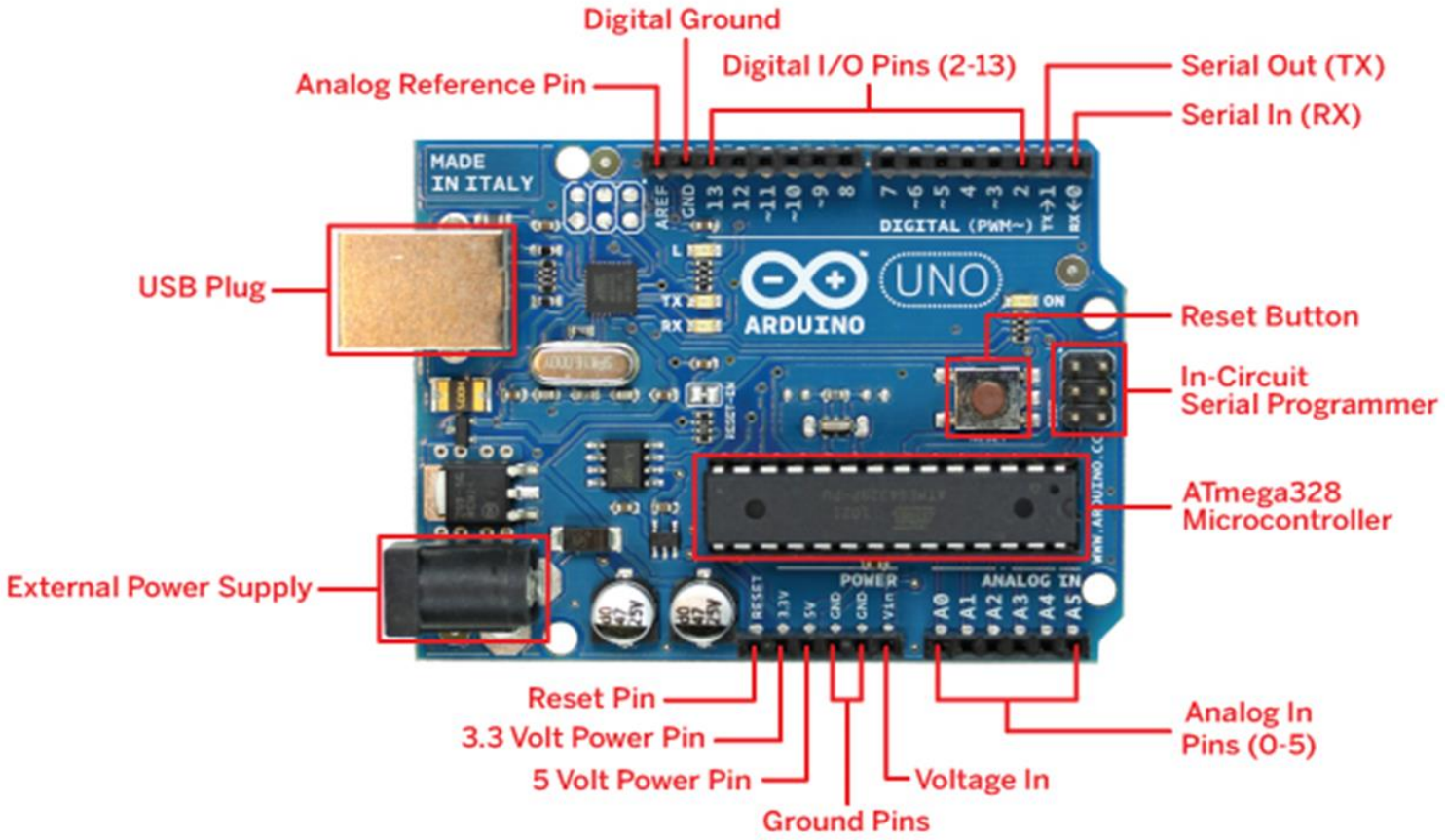


- A printed circuit board designed to facilitate work with a particular microcontroller.

- Typical components include:
  - power circuit
  - programming interface
  - basic input; usually buttons and LEDs
  - I/O pins

The Arduino Development Board

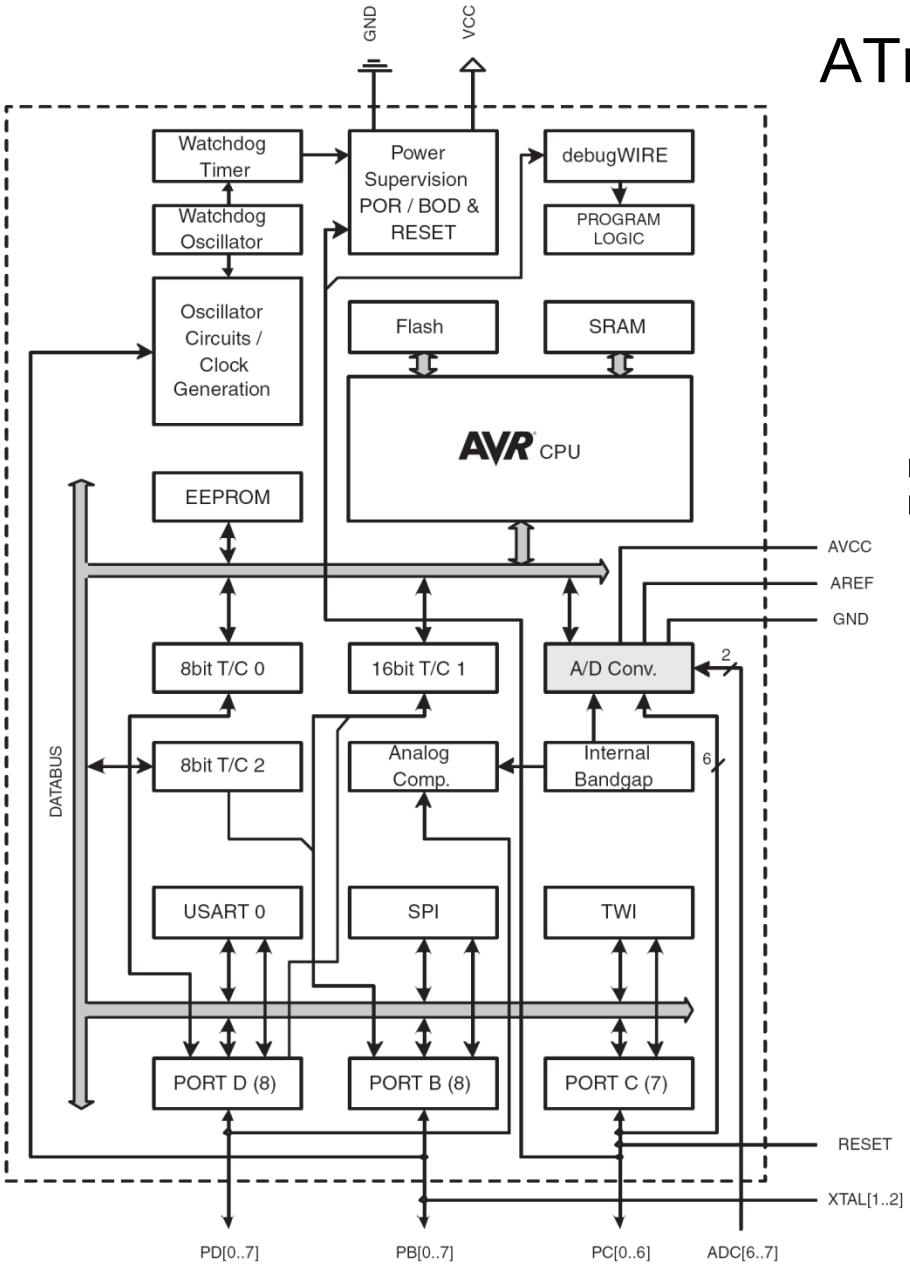
Arduino (The name is an Italian , meaning "strong friend") is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

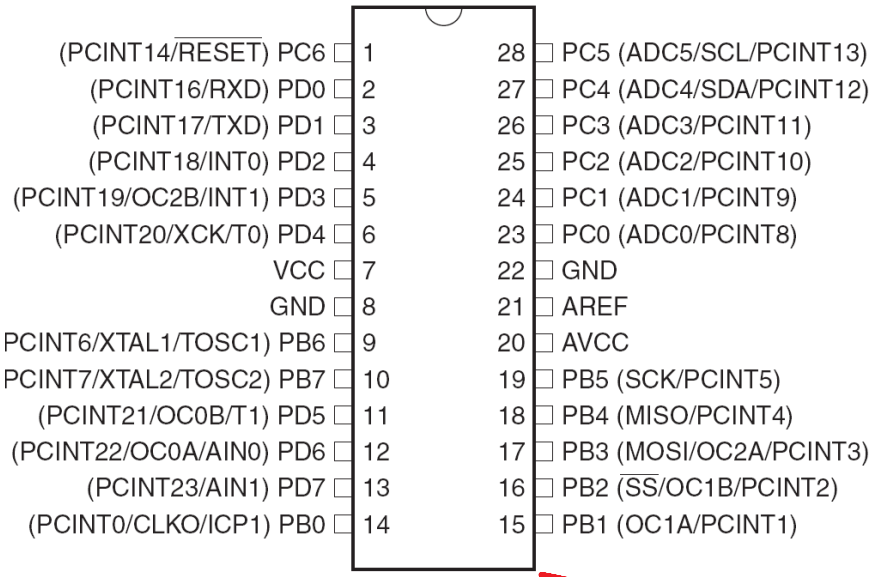


Arduino Microcontroller Boards

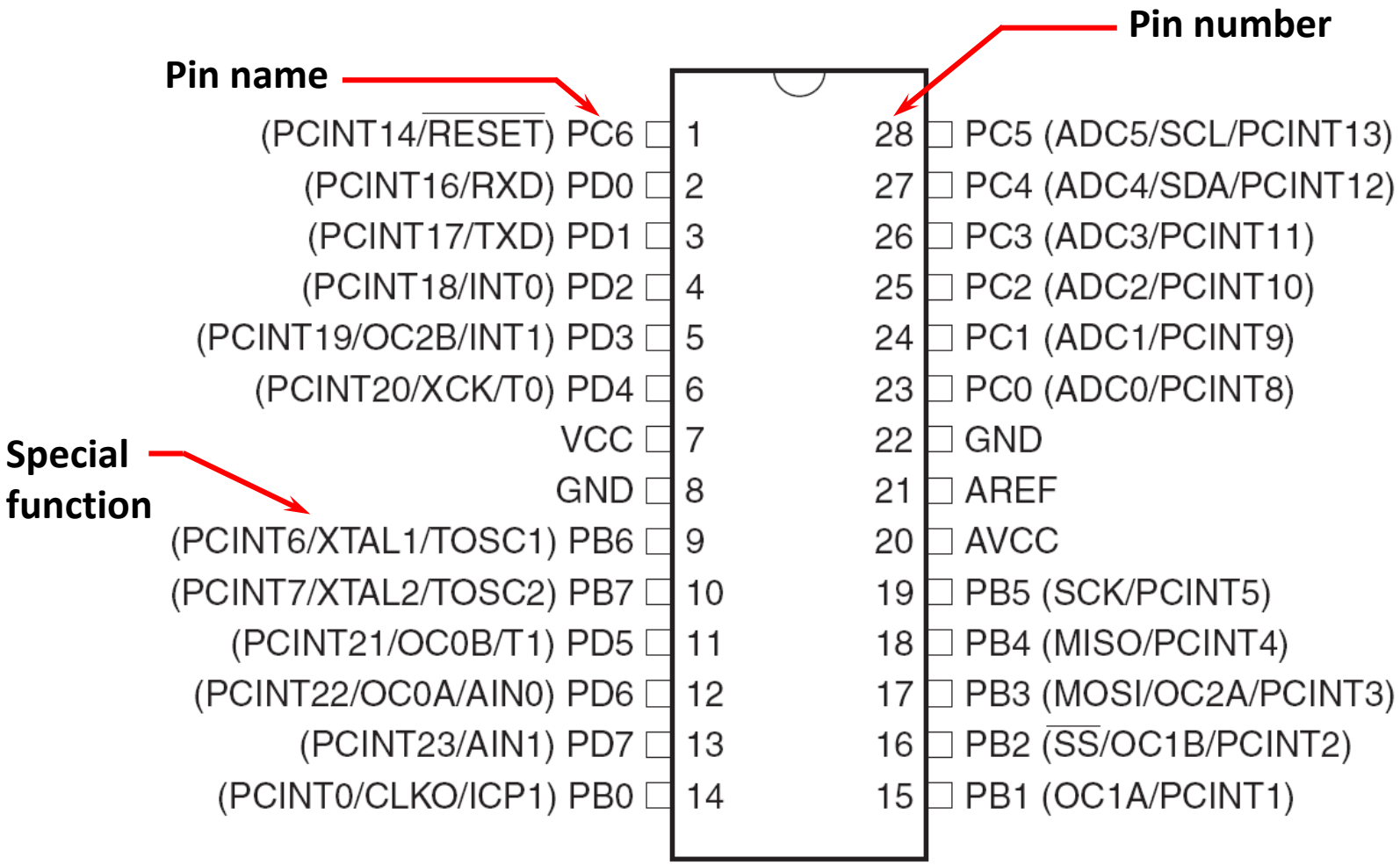
Microcontroller	ATmega328
Operating Voltage	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

ATmega328 Internal Architecture



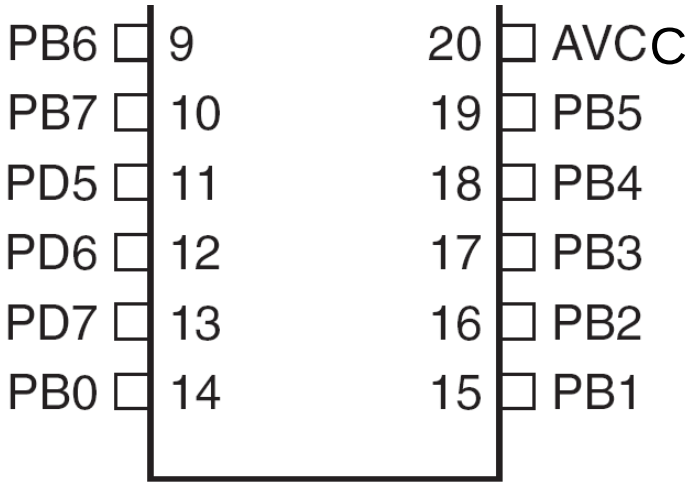


ATmega328 Microcontroller

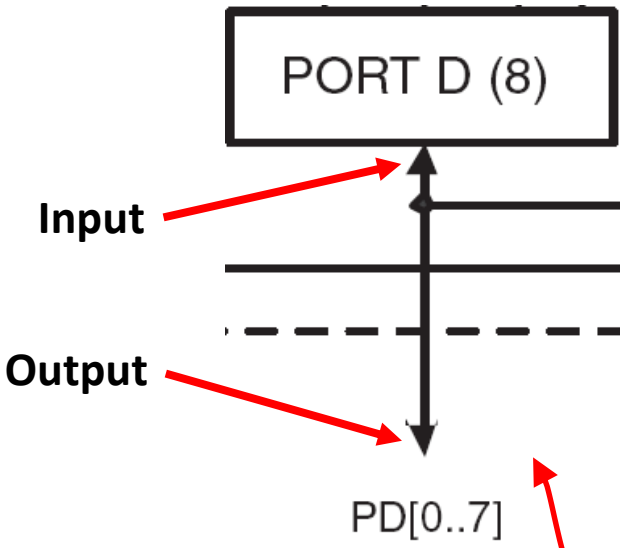


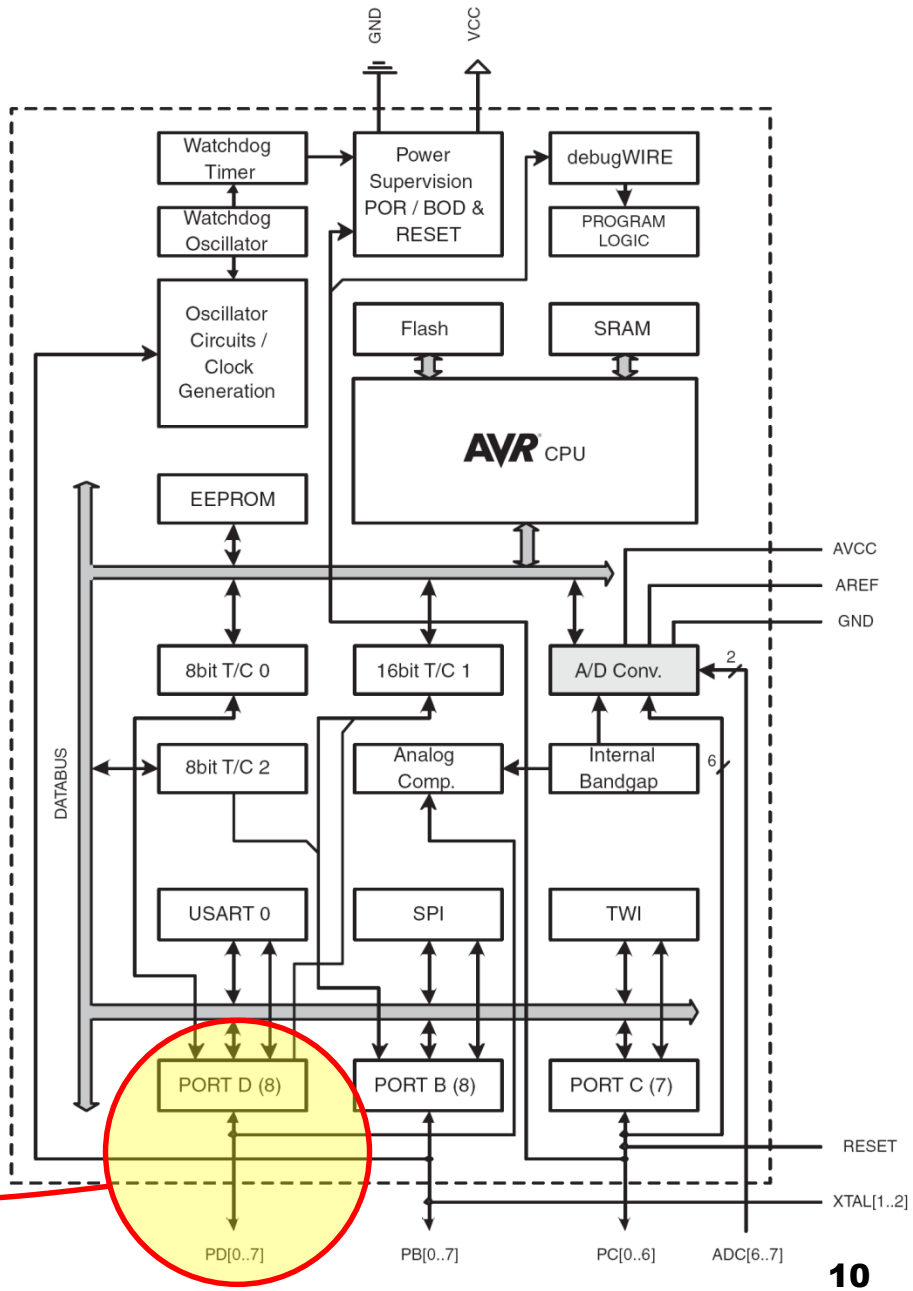
Microcontroller Ports and Pins

- The communication channels through which information flows into or out of the microcontroller
  - Ex. PORTB
    - Pins PB0 – PB7
      - May not be contiguous
      - Often bi-*directional*



ATmega328 Block Diagram

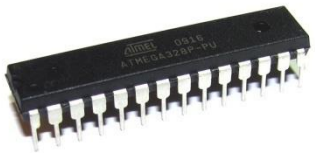


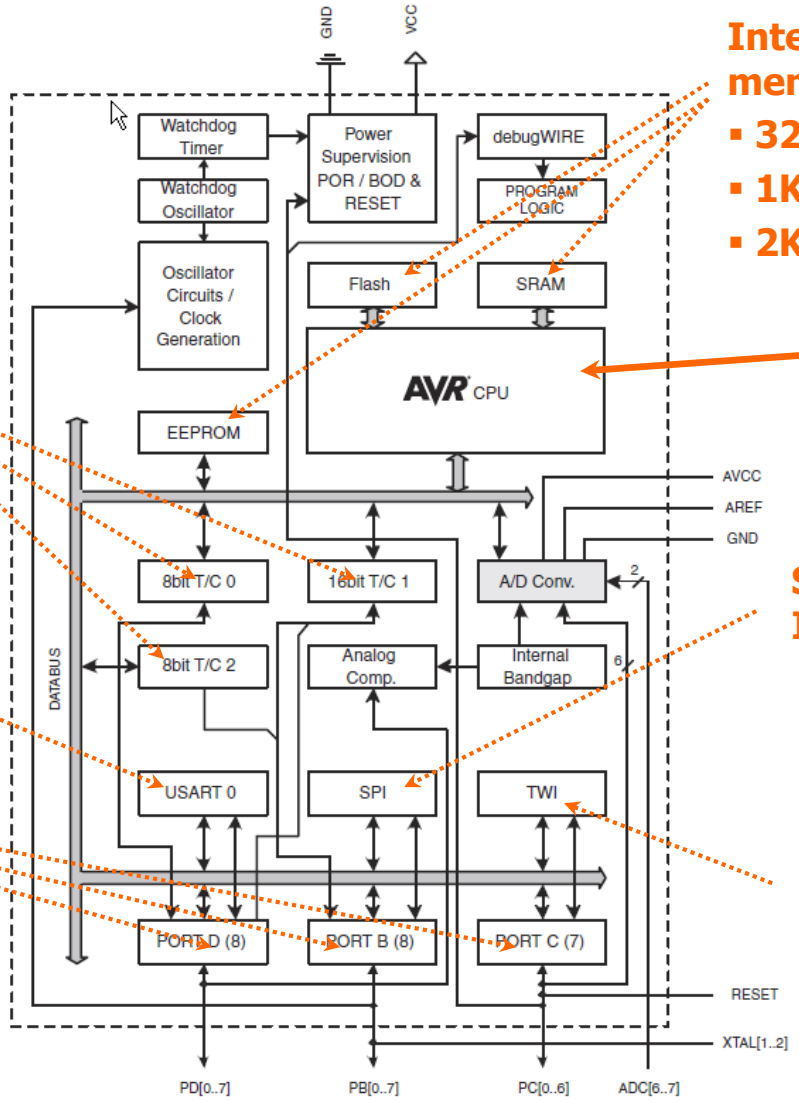


Setting the Pin Data Direction

- Arduino
  - pinMode(*pin_no.*, *dir*)
    - Ex. Make Arduino pin 3 (PD3) an *output*
      - `pinMode(3, OUTPUT);`
  - Note: one pin at a time
    - Suppose you wanted Arduino pins 3, 5, and 7 (PD3, PD5, and PD7) to be outputs?
    - Is there a way to make them all outputs at the same time?
      - Yes! Answer coming later…

Atmega328 Overview





Internal memories

- 32KB Flash
- 1KB EEPROM
- 2KB SRAM

8-bit CPU

Timer/Counter

Universal Synchronous and Asynchronous serial Receiver and Transmitter (Serial)

Serial Peripheral Interface

GPIO

2-wire Serial Interface

AVR Microcontroller

AVR stand for?

- Advanced Virtual RISC,
  the founders are Alf Egil Bogen Vegard Wollan RISC

- AVR architecture was conceived by two students at Norwegian Institute of Technology (NTH) and further refined and developed at **Atmel Norway (**Atmel AVR).

AVR Microcontroller

- AVR Micro controllers is Family of RISC Microcontrollers from Atmel.
- There are multiple architectures

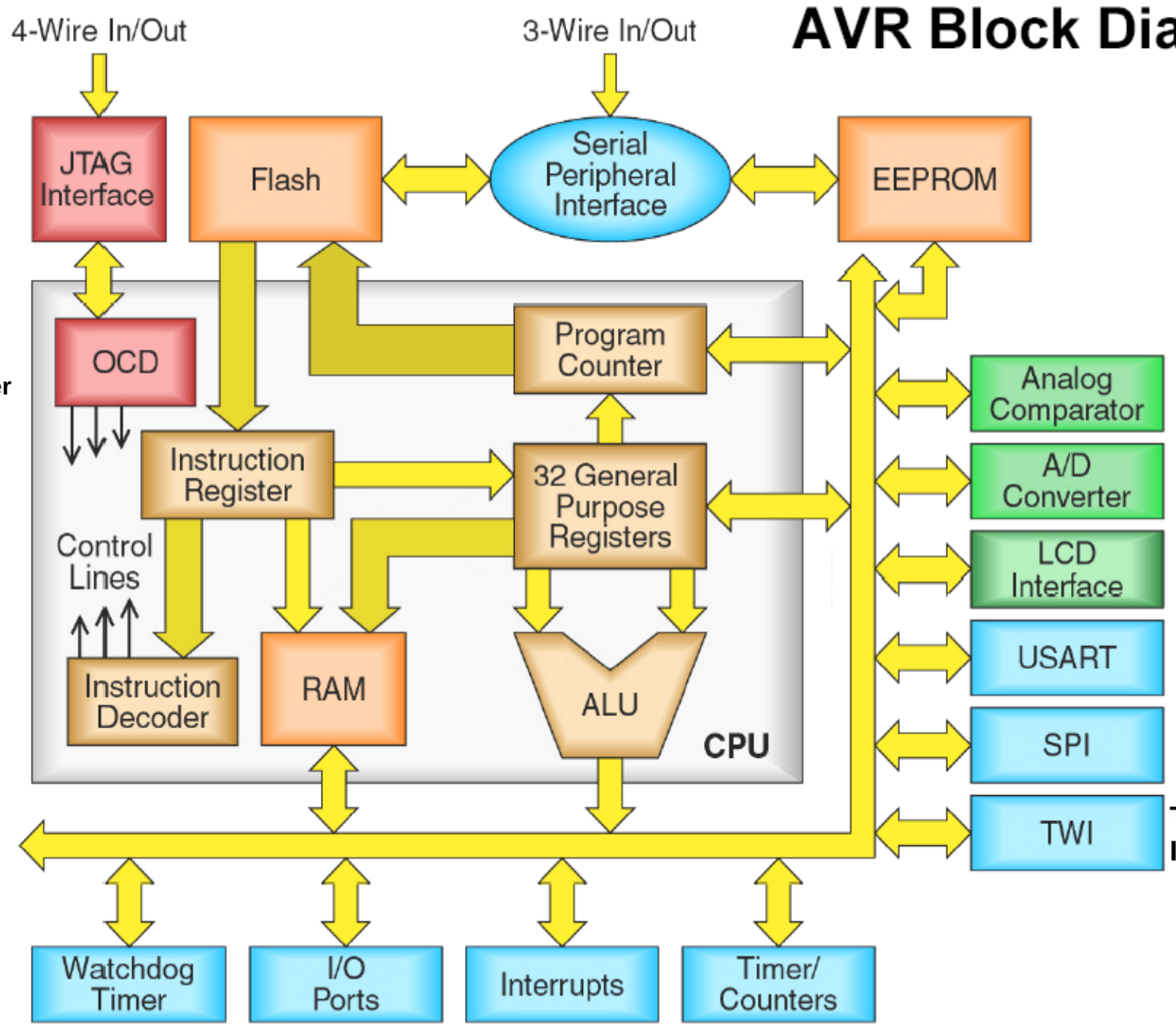
  RISC (Reduced Instruction Set Computer)
  CISC (Complex Instruction Set Computer)

RISC Microcontroller

Reduced Instruction Set Computers Advantages

- Fast Execution of Instructions due to simple instructions for CPU.
- RISC chips require fewer transistors, which makes them cheaper to design and produce.
- Emphasis on software
- Single-clock , reduced instruction only
- Register to register: “LOAD" and "STORE“ are independent instructions
- Spends more transistors on memory registers

AVR Block Diagram



On Chip Debugger

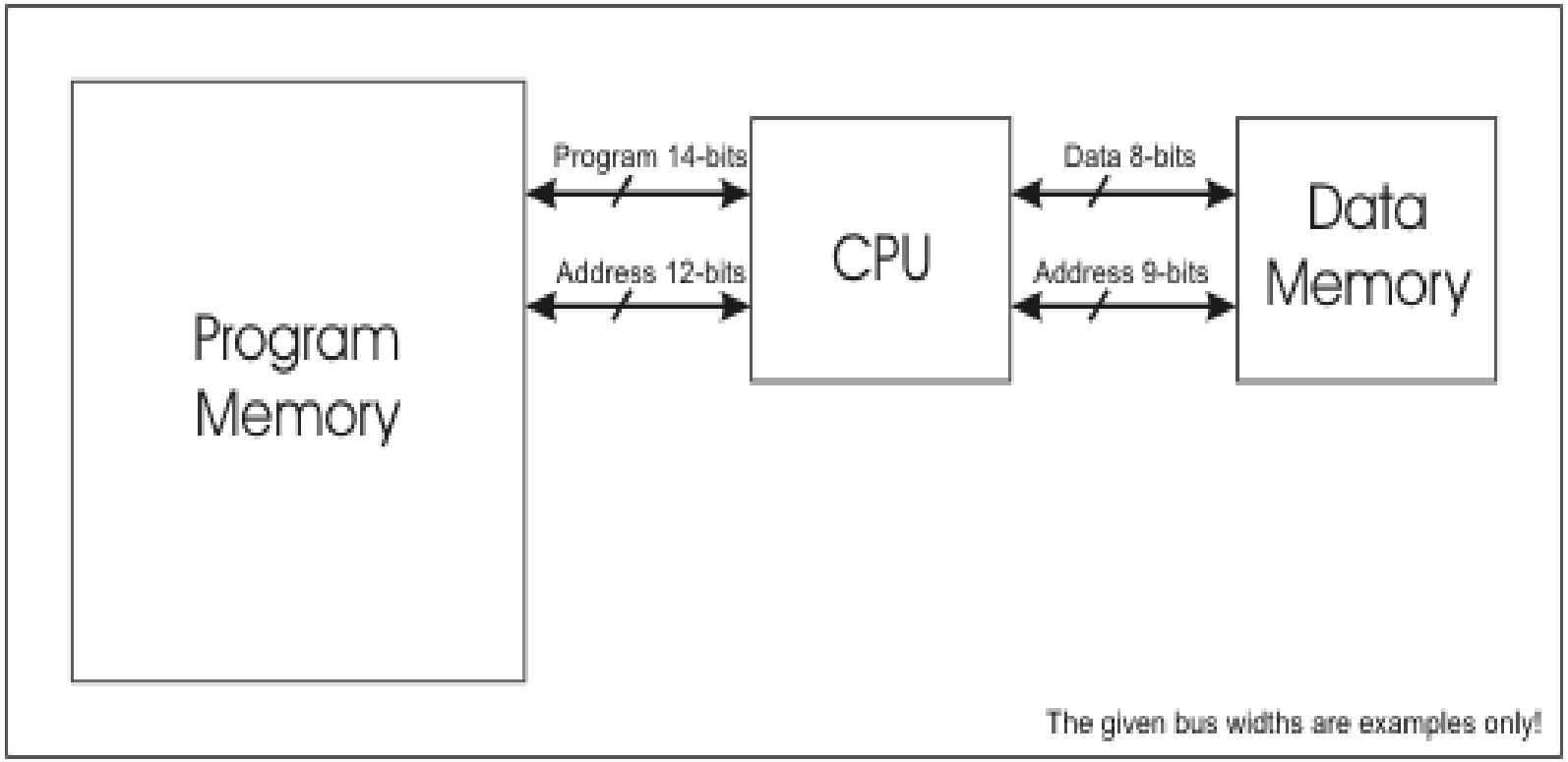
Two Wire Interface

AVR Microcontroller

The AVR is a Harvard architecture CPU.

- ***Harvard Architecture***
  - Computer architectures that used physically separate storage and signal pathways for their instructions and data.
  - CPU can read both an instruction and data from memory at the same time that makes it faster.
- ***von Neumann architecture***
  - CPU can Read an instruction or data from/to the memory.
  - Read, Write can`t occur at the same time due to same memory and signal pathway for data and instructions.

AVR Microcontroller



Harvard Architecture diagram

AVR Microcontroller

- AVR is a family of 8-bit microntrollers with a large range of variants differing in:

  - size of program-memory (flash)
  - size of EEPROM memory
  - number of I/O pins
  - number of on-chip features such as UART and ADC

- Smallest microcontroller is the ATTiny11 with 1k flash ROM, no RAM and 6 I/O pins.

- Large such as the ATMEGA128 with 128k flash, 4KB RAM, 53 I/O pins and lots of on-chip features.

Part Number	Pins	Flash	EEPROM	RAM
90S1200	20	1K	64 Bytes	0
90S2313	20	2K	128	128
90S2323	8	2K	128	128
90S2333	28	2K	128	128
90S4433	28	4K	256	128
90S4414	40	4K	256	256
90S8515	40	8K	512	512
90S4434	40	4K	256	256
90S2343	8	2K	128	128
Mega103	64	128K	4096	4096
Mega603	64	64K	2048	4096
Tiny10	8	1K	64	0
Tiny12	8	1K	64	0
Tiny13	8	2K	128	128

AVR Architecture

- **Registers**
- **Instruction Set**
- **I/O ports**
- **Memory (flash & RAM & ROM)**
- **CPU**

AVR Architecture

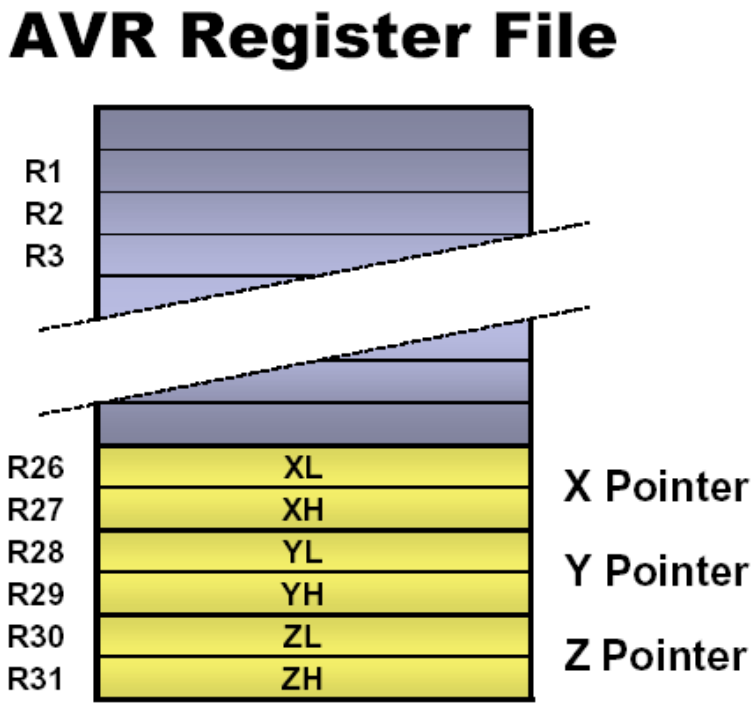
- **Registers: Two types of registers**
  GERNEL purpose & SPECIAL purpose registers

- **GERNEL purpose**
  32 general purpose registers having storage capacity of 8-Bits
  Named as R0,R1,R2 to R31.
  Register 0 to 15 & 16 to 31 are different.
  Can store both Data & Addresses.

- **SPECIAL purpose: Three registers**
  Program counter
  Stack Pointer
  Status Register

AVR Memory Space

- Program Flash
  - Vectors, Code, and (Unchangeable) Constant Data
- Working Registers
  - Includes X, Y, and Z registers.
- I/O Register Space
  - Includes “named” registers
- SRAM – Data Space
  - Runtime Variables and Data
  - Stack space
- EEPROM space
  - For non-volatile but alterable data

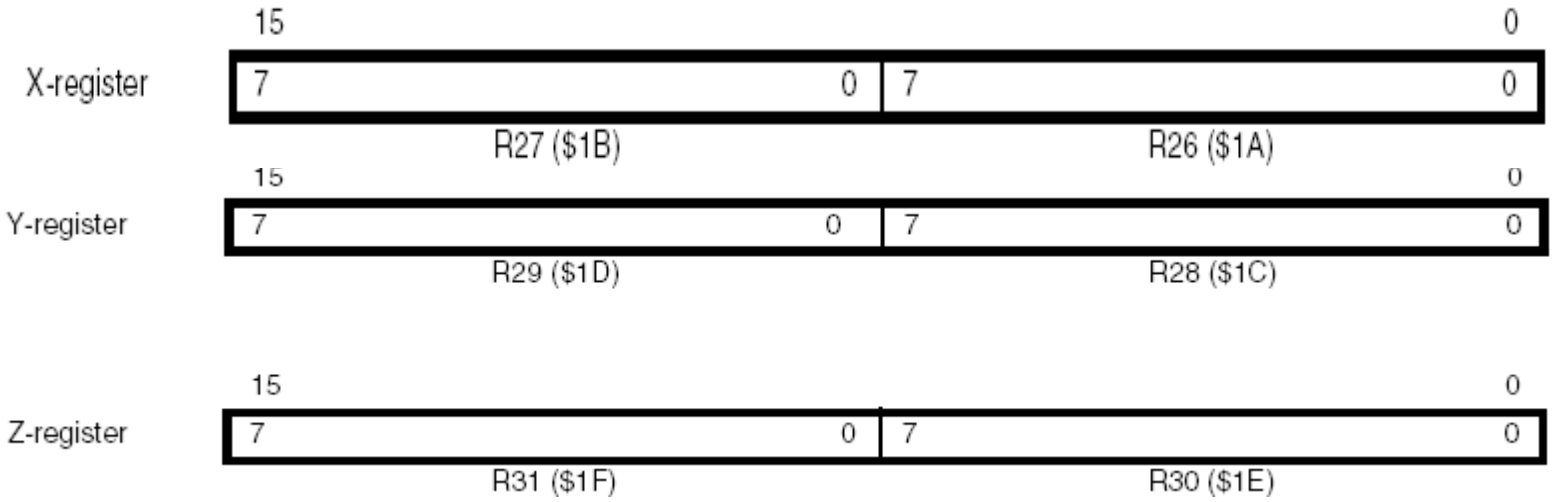


AVR Architecture

Pointer Register

Three 16-bit address registers pairs of registers 26 to 31 have extra meaning in AVR assembly.
X (r27:r26), y (r29:r28), z (r31:r30).

pointer	Sequence
X	**Read/Write from address X, don't change the pointer**



AVR Architecture

status register
(SREG) It is 8-bit long each bit has a different meaning.

I	T	H	S	V	N	Z	C

I: Global Interrupt Enable/Disable Flag, SREG7
T: Transfer bit used by BLD and BST instructions, SREG6
H: Half Carry Flag, SREG5
S: For signed tests Instruction Set, SREG4
V: Two's complement overflow indicator, SREG3
N: Negative Flag, SREG2
Z: Zero Flag, SREG1
C: Carry Flag, SREG0

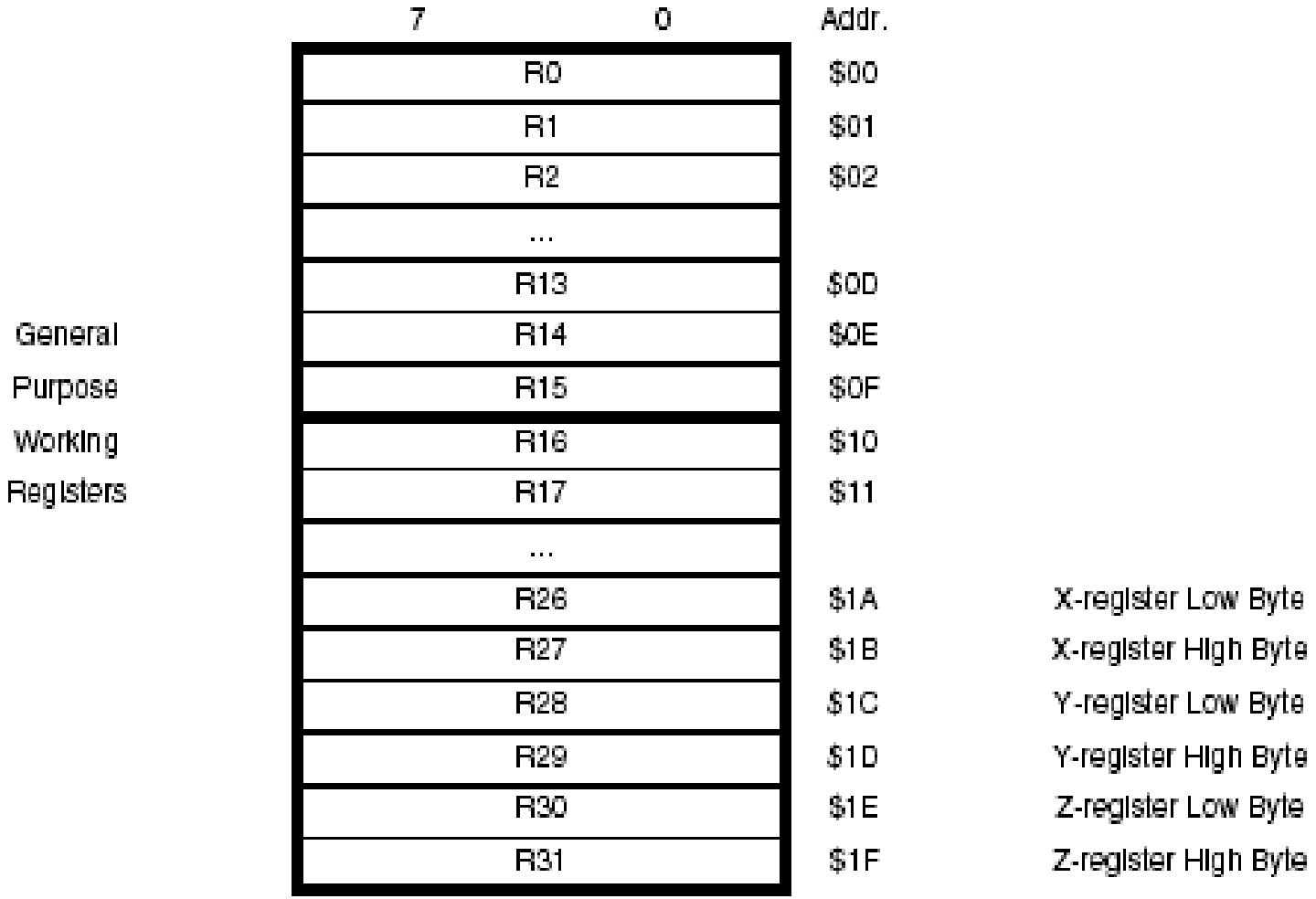
AVR Architecture

Stack Pointer (SP)

16-bit stack pointer (SP) holds address in data space of area to save function call information.

AVR

Register Architecture

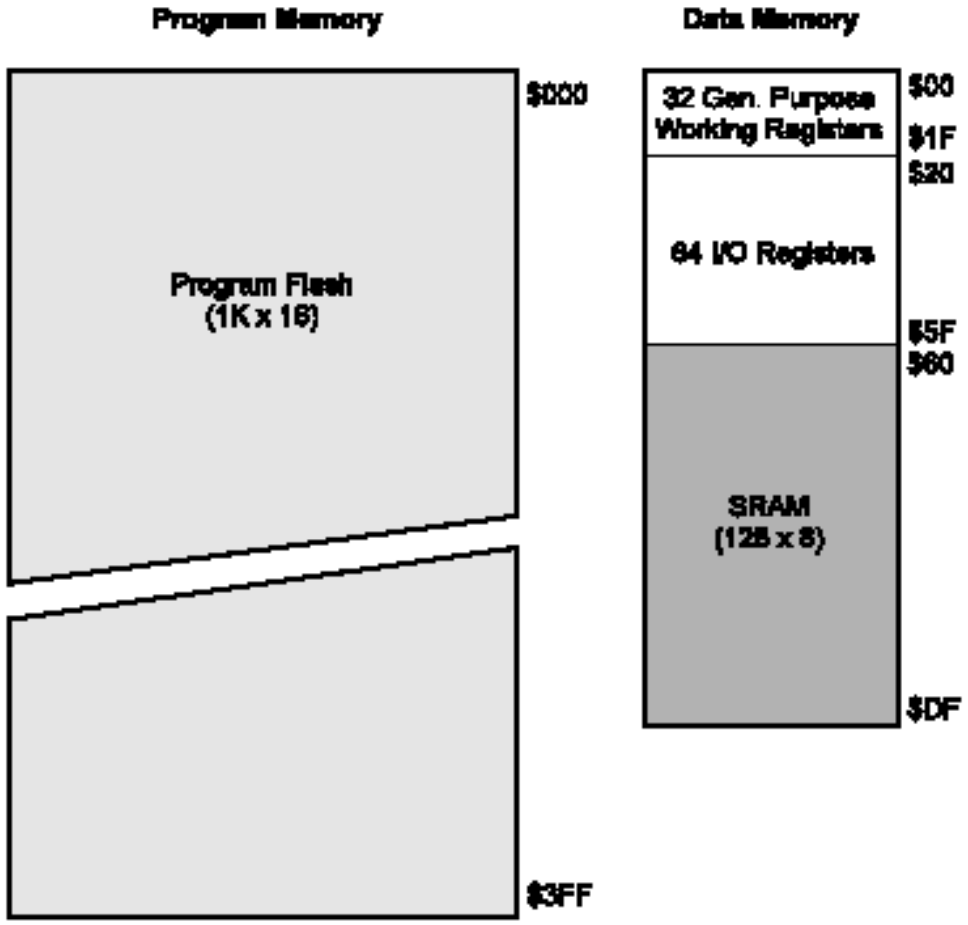


AVR Architecture

Memory:

There are two separate memories

Program Memory
Data Memory



AVR Studio

- Integrated Development Environment (IDE) for writing and debugging AVR applications for windows environments.

- AVR Studio provides a project management tool, source file editor, chip simulator and In-circuit emulator interface for the powerful AVR 8-bit RISC family of microcontrollers.