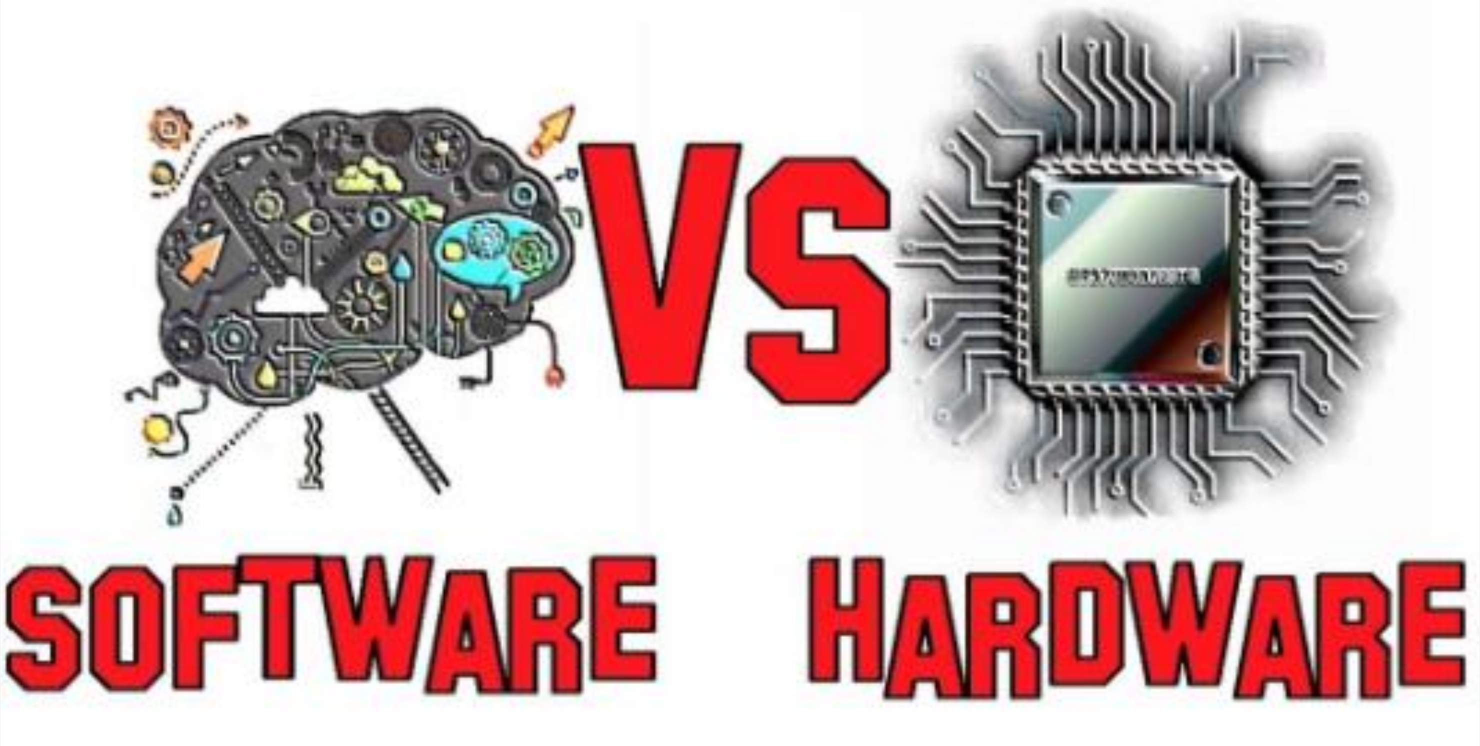
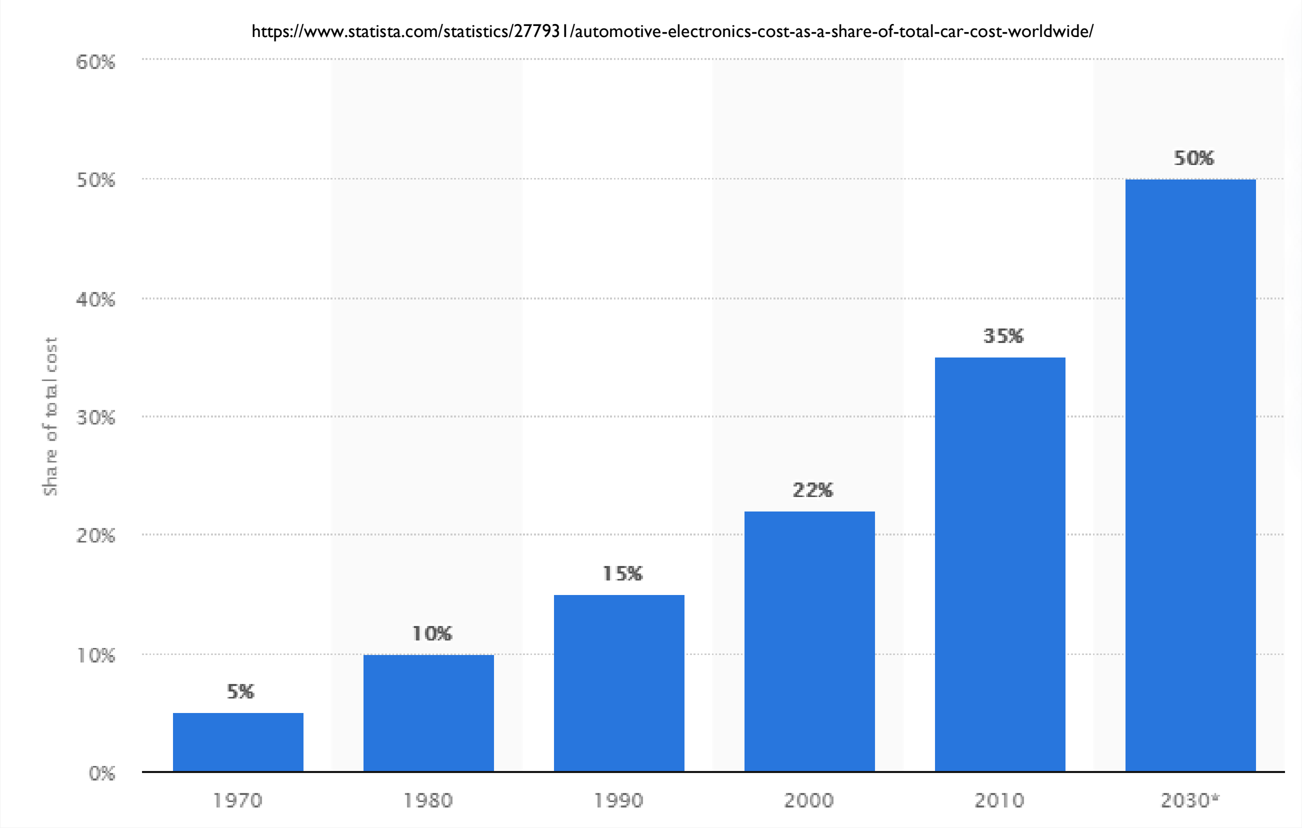
REAL-TIME

EMBEDDED SYSTEMS

EMBEDDED SOFTWARE OR HARDWARE?



EMBEDDED SOFTWARE OR HARDWARE?


https://www.statista.com/statistics/277931/automotive-electronics-cost-as-a-share-of-total-car-cost-worldwide/

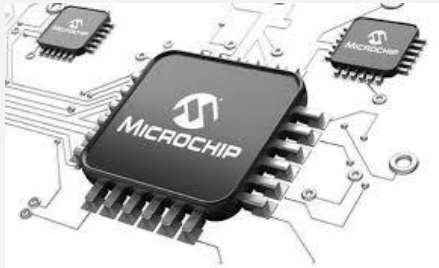
Automotive electronics cost as a percentage of total car cost worldwide from 1950 to 2030

The value added to the final product by embedded software is much higher than the cost of the embedded device itself.

LET'S TALK HW

Infineon technologies

SAMSUNG



TOSHIBA

TEXAS INSTRUMENTS

NXP

MARVELL

arm

LET'S TALK HW

- Microcontroller (MCU)
- Microprocessor (MPU) (CSEN601- CSEN602!)
- Power Management Unit (PMU/ PMIC)
- Field Programmable Gate Arrays (FPGA) (CSEN605)
- Digital Signal Processor (DSP) (does the stuff from THAT OTHER COURSE and 1rst tuto)
- ASIC

SOUBRA©2022

PMIC / PMU

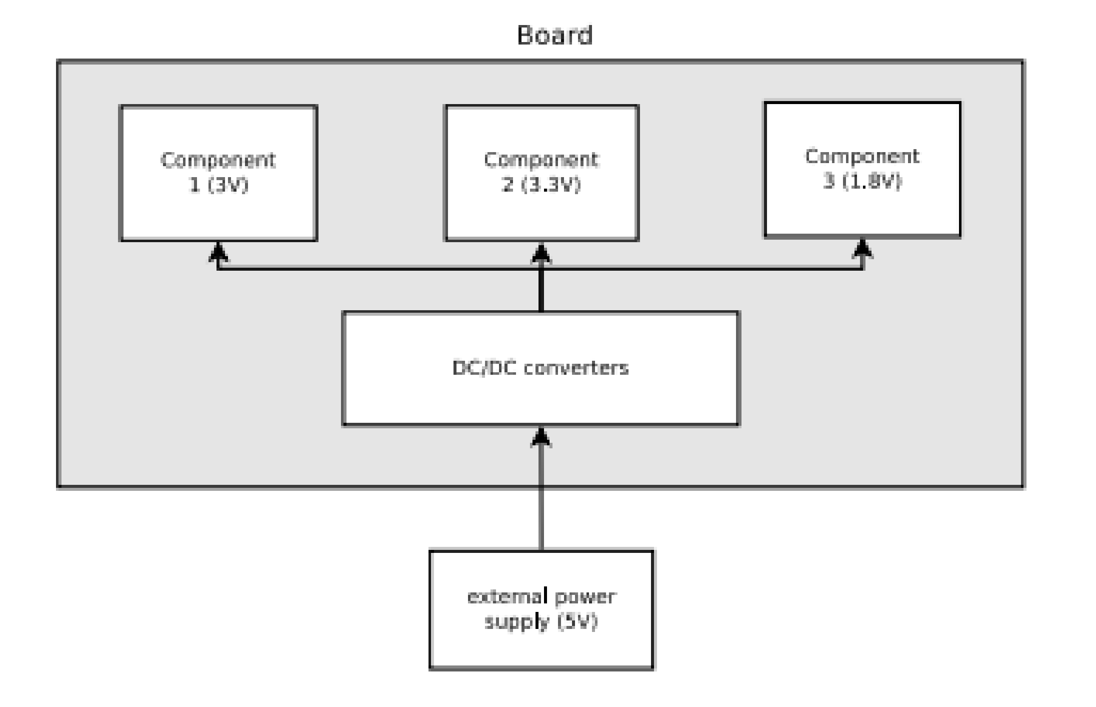
Power Management Integrated Circuit,

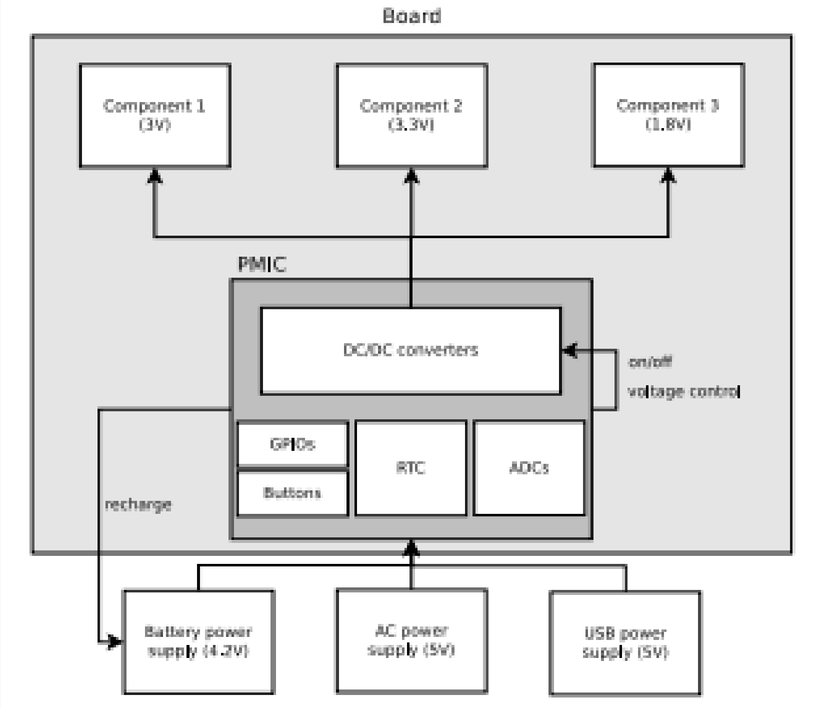
- handles the power sequence of the board,
- supplies power to the different components inside the board,
- protects the board from unsupported overvoltage and undervoltage,
- might handle different external power supplies,
- can provide other misc. features (GPIO, ADC, ...),
- is usually software-controllable (often as an i2c device),
- is not mandatory

range pi

PMIC / PMU

AXP209
C6032CA 67K1


Board
Component 1 (3V)
Component 2 (3.3V)
Component 3 (1.8V)
DC/DC converters
external power supply (5V)


Board
Component 1 (3V)
Component 2 (3.3V)
Component 3 (1.8V)
PMIC
DC/DC converters
on/off
voltage control
GPIOs
Buttons
RTC
ADCs
recharge
Battery power supply (4.2V)
AC power supply (5V)
USB power supply (5V)

ASIC

Application Specific Integrated Circuit

- Custom designed for a specific application.
- It is normally designed by a company for a specific use or for a specific customer alone.
- WHY?

=> optimized power and performance of that specific application.

ASICs have great utility in aerospace applications
Because they are not field programmable => they are more radiation tolerant.

FPGA

- Array of Programmable blocks with a programmable interconnect which can be used to create programmable hardware designs.

- Designs are typically captured in an HDL and can be synthesized for the FPGA technology and downloaded into an FPGA and then used along with other devices in a system.

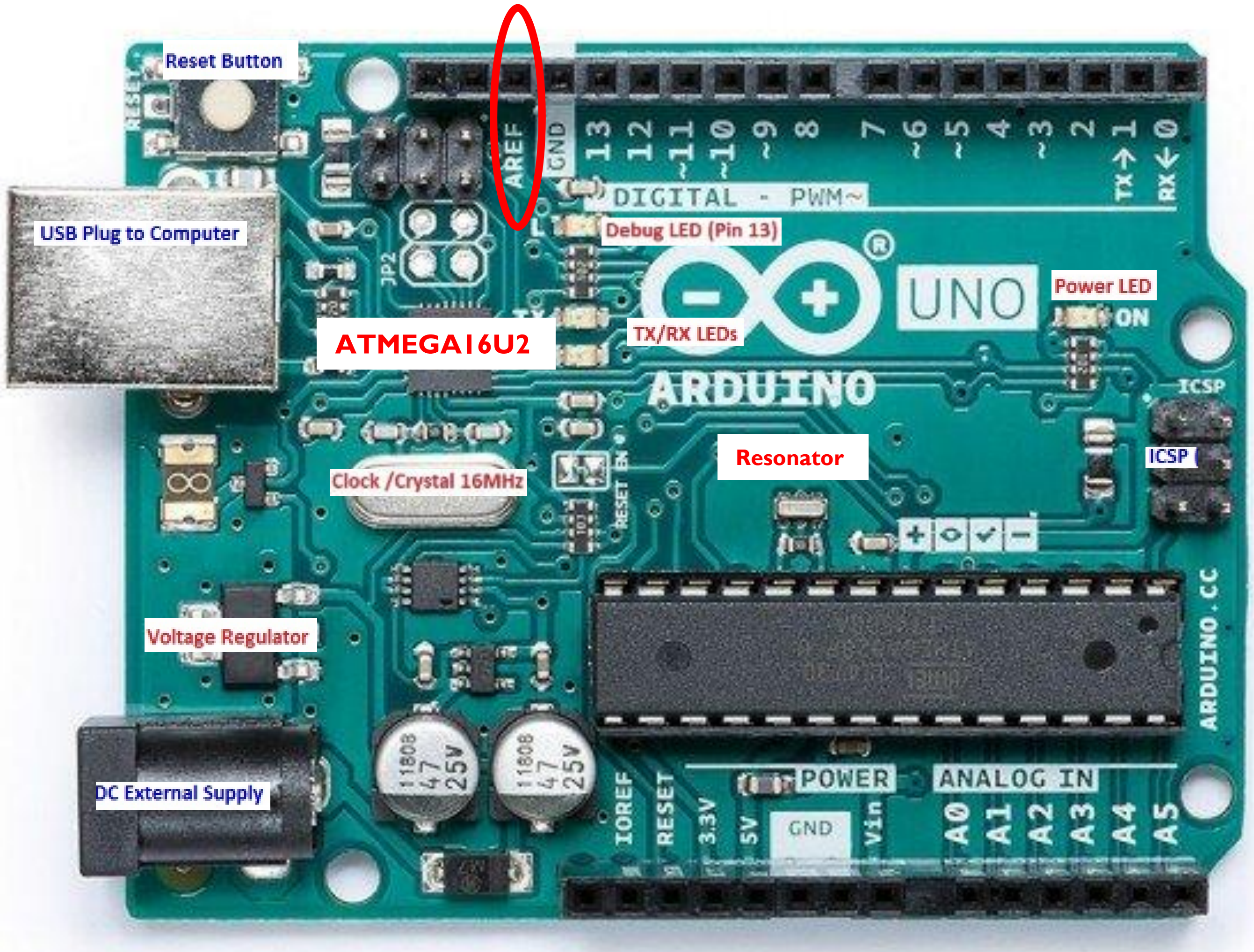
- Reconfiguration is possible by changing the design and downloading to same FPGA which makes this very useful for prototyping as well as making changes unlike an ASIC.

Tutorial 2! NIOS II

ARDUINO

Arduino Uno R3

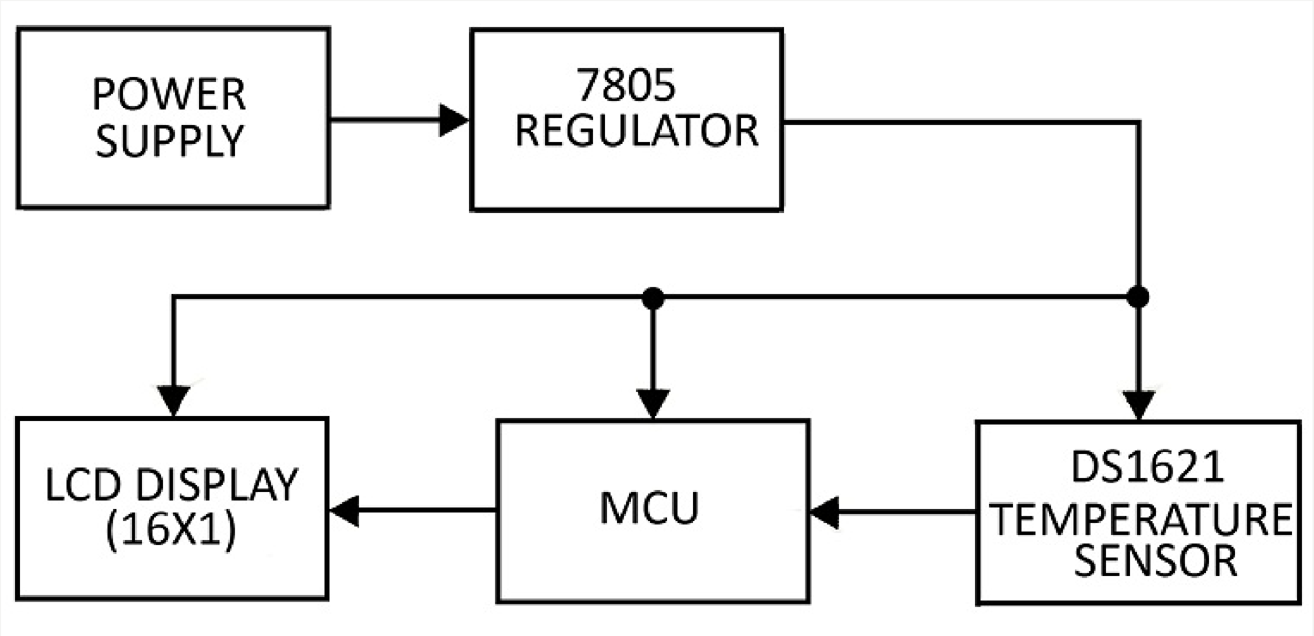
- LEDs?
- Clock?
- MISC?
- MCU?



AREF PIN

- ATMEGA comes with a 10bit ADC (Analog-Digital-Converter), which converts incoming voltages between 0V and 5V to integer values between what and what?
  - 0 and 1023.

- This results in a resolution of what?
  - 5/1023= roughly 4.88 mV.

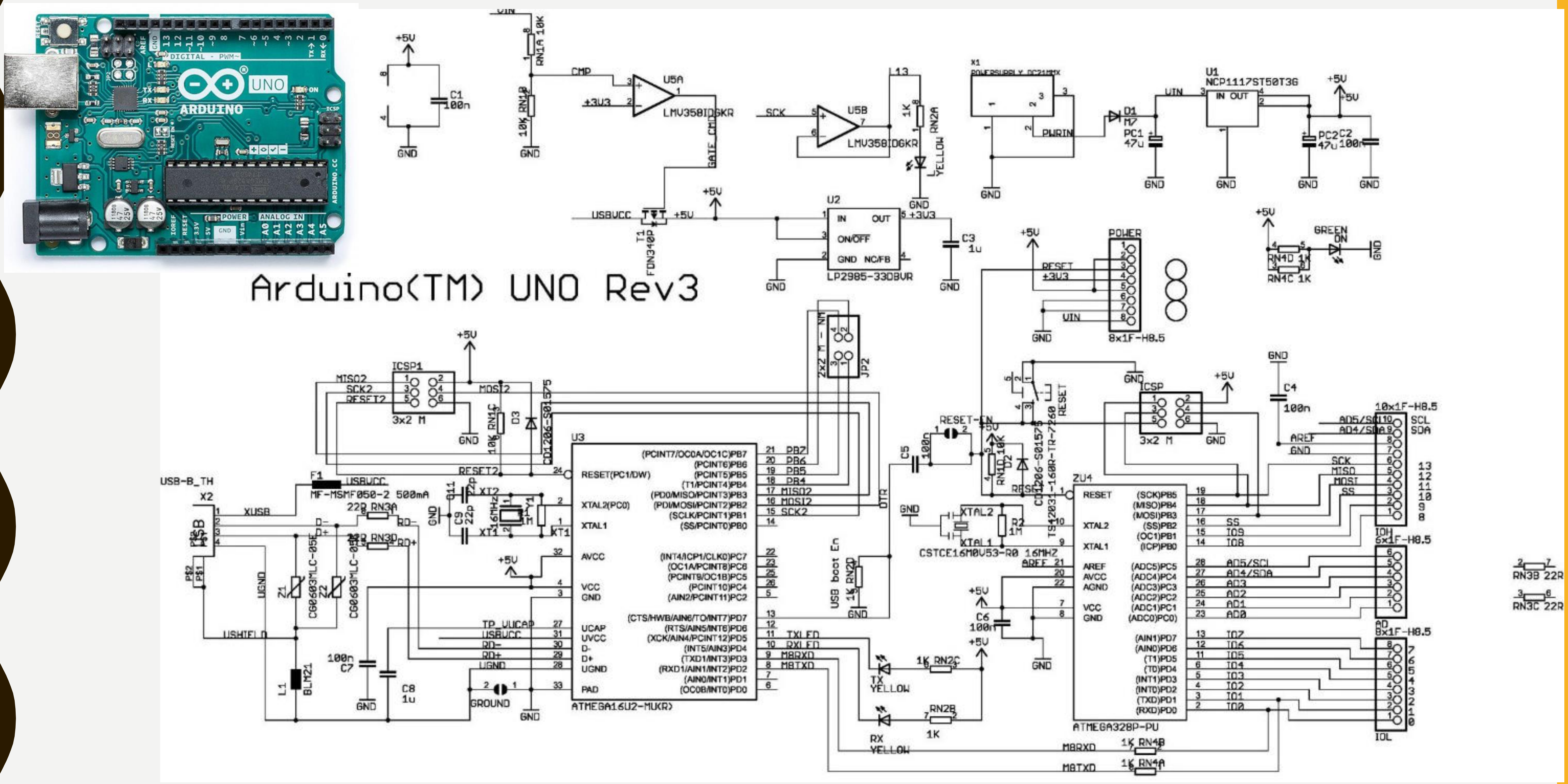


AREF PIN

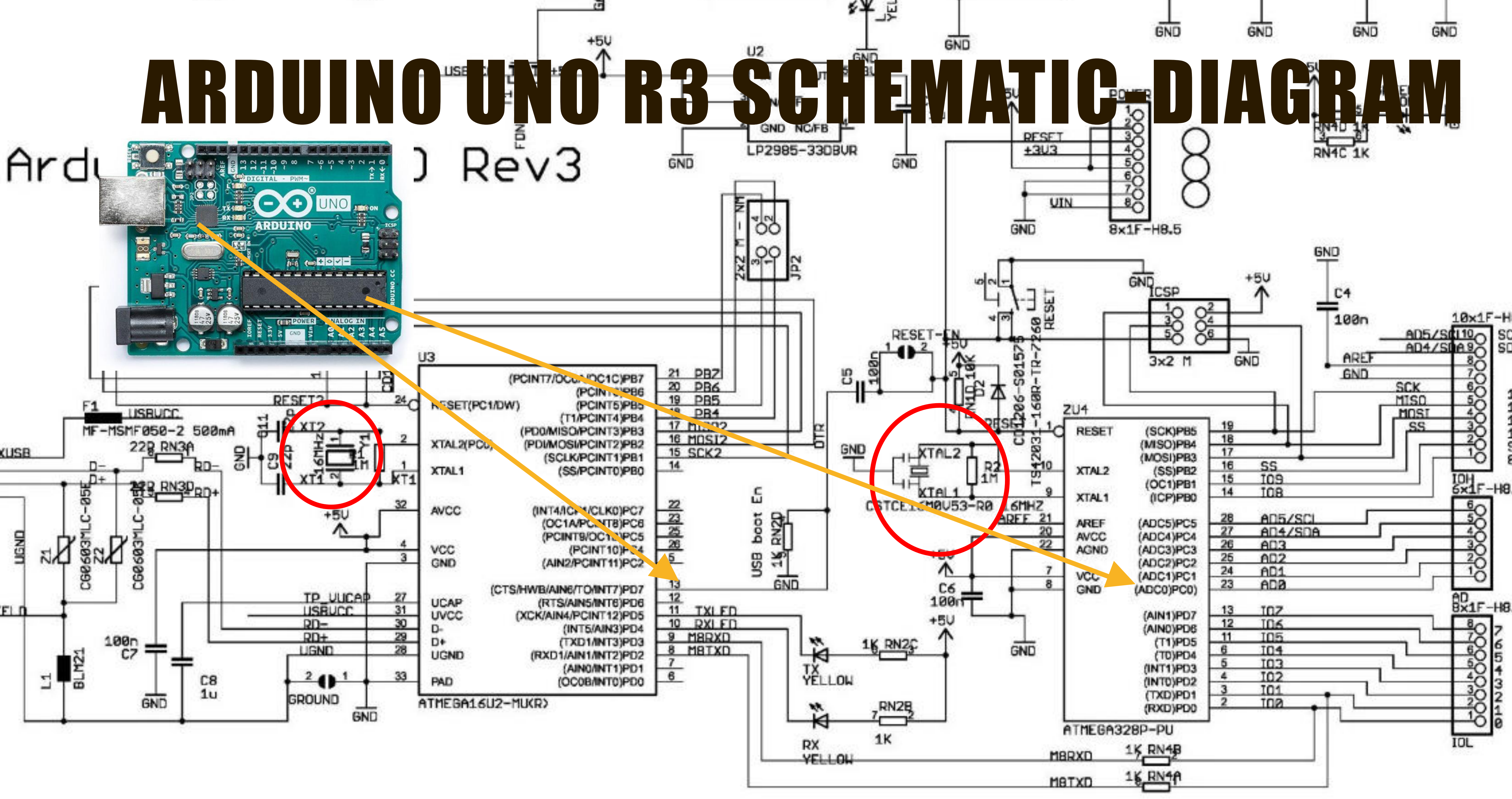
- Imagine that a sensor only outputs from 0-3 volts. This means what?
  - analog reading of max 1023*3/5=(approx.)614 out of 1024. With a resolution of 4.88 mV
- Can we improve the resolution?
- Yes!: use AREF Pin!
- To increase the resolution of that measurement, the microcontroller can take a reference voltage, *usually supplied by a 'voltage divider‘*

- Feed it a 3v AREF,
  - you get 3/1023 = 2.93mV.
  - That's ~40% more precise
  - without using a more precise sensor (more expensive).

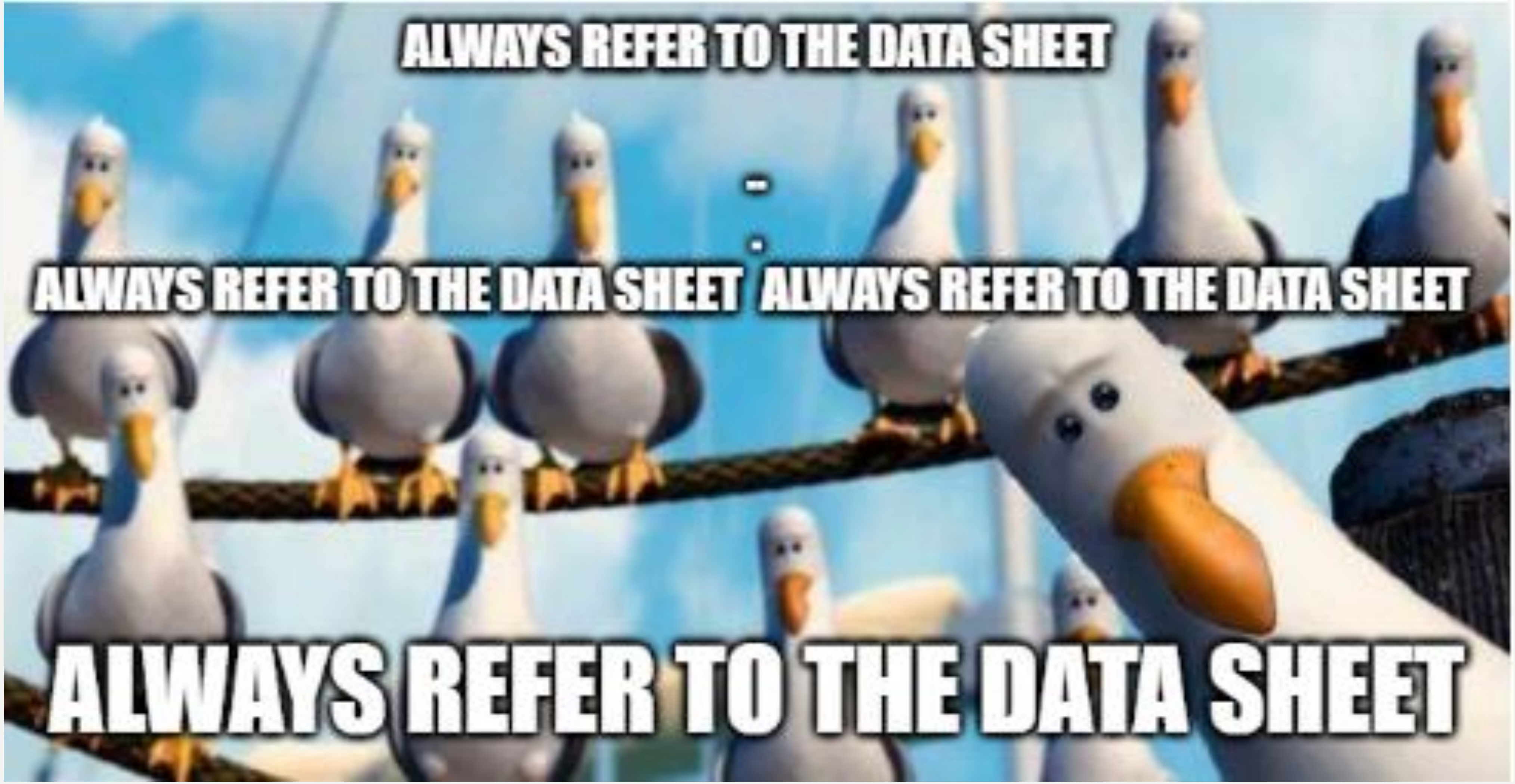
ARDUINO UNO R3 SCHEMATIC-DIAGRAM



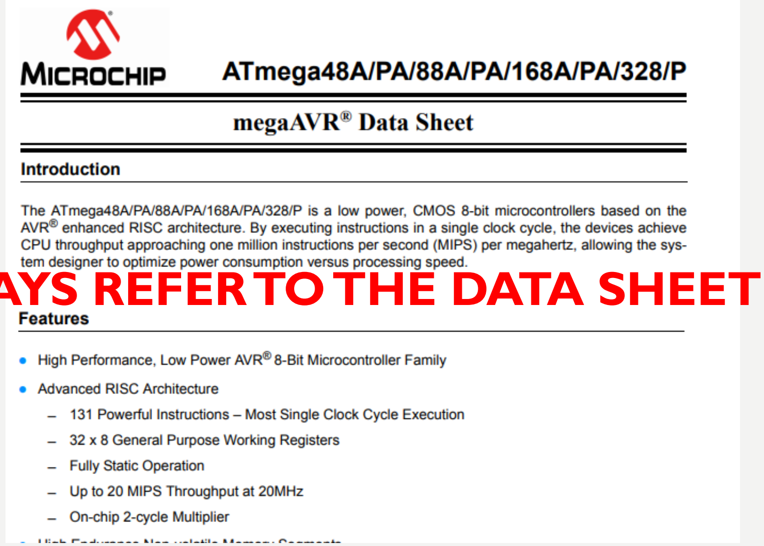
ARDUINO UNO R3 SCHEMATIC DIAGRAM



DATASHEET


ALWAYS REFER TO THE DATA SHEET
ALWAYS REFER TO THE DATA SHEET
ALWAYS REFER TO THE DATA SHEET
ALWAYS REFER TO THE DATA SHEET

ATMEGA328



MICROCHIP **ATmega48A/PA/88A/PA/168A/PA/328/P**

megaAVR® Data Sheet

Introduction

The ATmega48A/PA/88A/PA/168A/PA/328/P is a low power, CMOS 8-bit microcontrollers based on the AVR® enhanced RISC architecture. By executing instructions in a single clock cycle, the devices achieve CPU throughput approaching one million instructions per second (MIPS) per megahertz, allowing the system designer to optimize power consumption versus processing speed.

Features

- High Performance, Low Power AVR® 8-Bit Microcontroller Family
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments

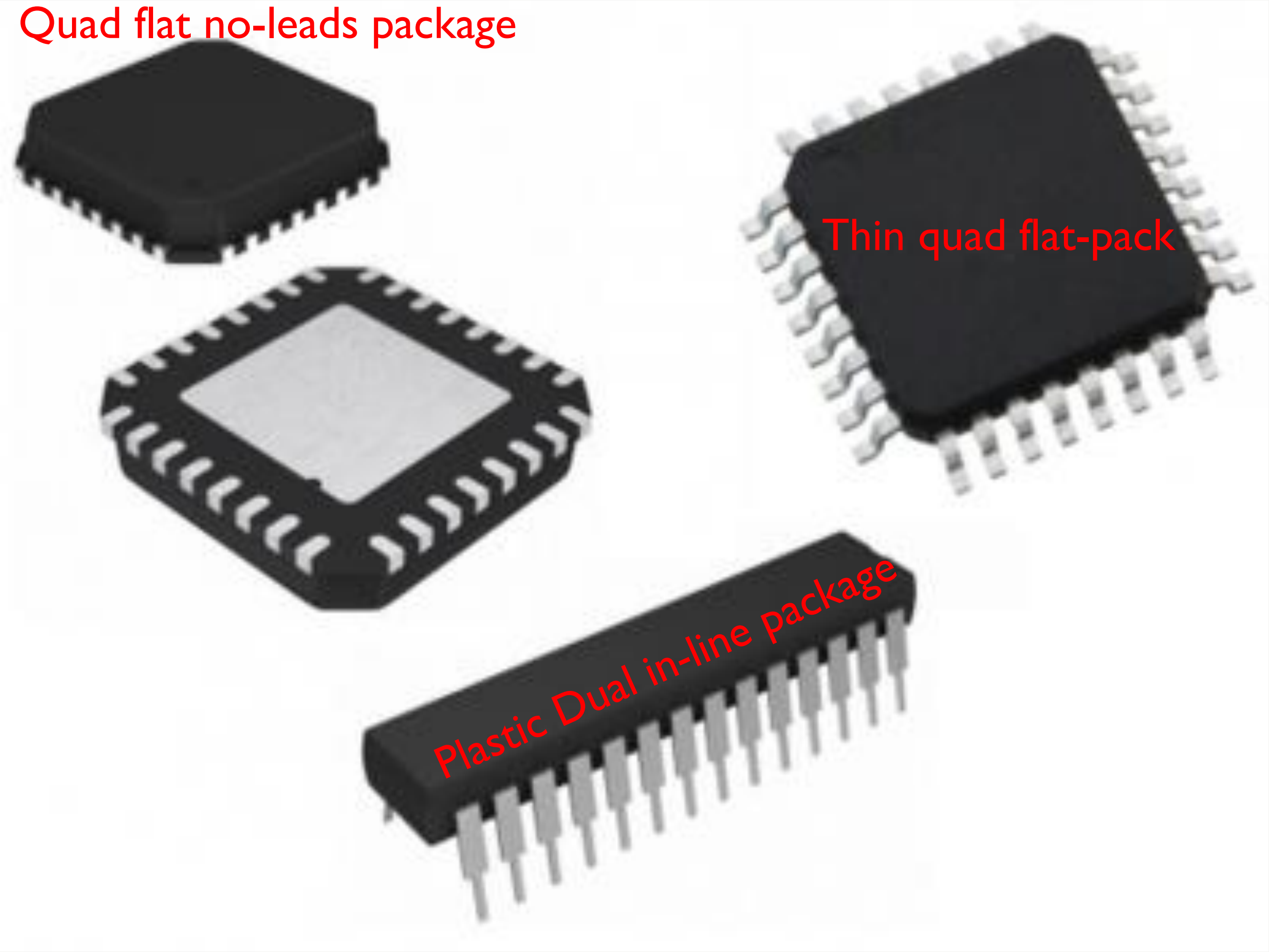
ALWAYS REFER TO THE DATA SHEET!

- Some Advanced Virtual RISC (AVR) micro-controller.

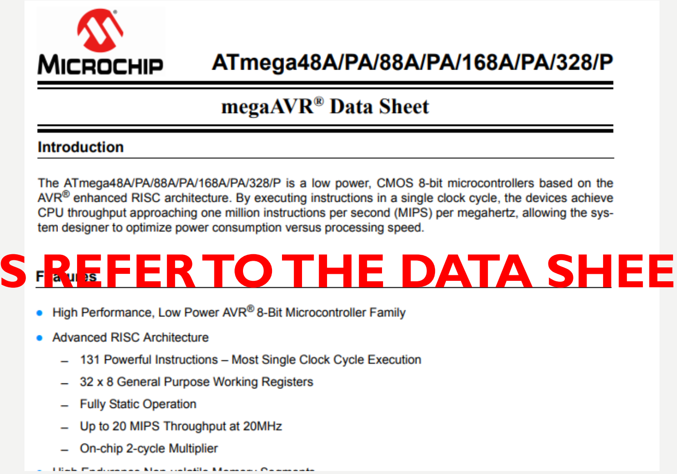
No. of Pins	28
CPU	RISC 8-Bit
Operating Voltage	1.8 to 5.5 V
Program Memory	32KB
Program Memory Type	Flash
SRAM	2048 Bytes
EEPROM	1024 Bytes
ADC	10-Bit

Number of ADC Channels	8
PWM Pins	6
Comparator	1
Packages (4)	28-pin PDIP 32-lead TQFP 28-pad QFN/MLF 32-pad QFN/MLF
Oscillator	up to 20 MHz
Timer (3)	8-Bit x 2 & 16-Bit x 1

IC PACKAGING



ATMEGA



MICROCHIP ATmega48A/PA/88A/PA/168A/PA/328/P

megaAVR® Data Sheet

Introduction

The ATmega48A/PA/88A/PA/168A/PA/328/P is a low power, CMOS 8-bit microcontrollers based on the AVR® enhanced RISC architecture. By executing instructions in a single clock cycle, the devices achieve CPU throughput approaching one million instructions per second (MIPS) per megahertz, allowing the system designer to optimize power consumption versus processing speed.

Features

- High Performance, Low Power AVR® 8-Bit Microcontroller Family
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments

ALWAYS REFER TO THE DATA SHEET!

- Some Advanced Virtual RISC (AVR) micro-controller.

Enhanced Power-on-Reset	Yes
Power Up Timer	Yes
I/O Pins	23
Manufacturer	Microchip
SPI	Yes
I2C	Yes
Watchdog Timer	Yes

Brown out detect (BOD)	Yes
Reset	Yes
USI (Universal Serial Interface)	Yes
Operating Temperature	-40 C to +85 C

AVR MCU

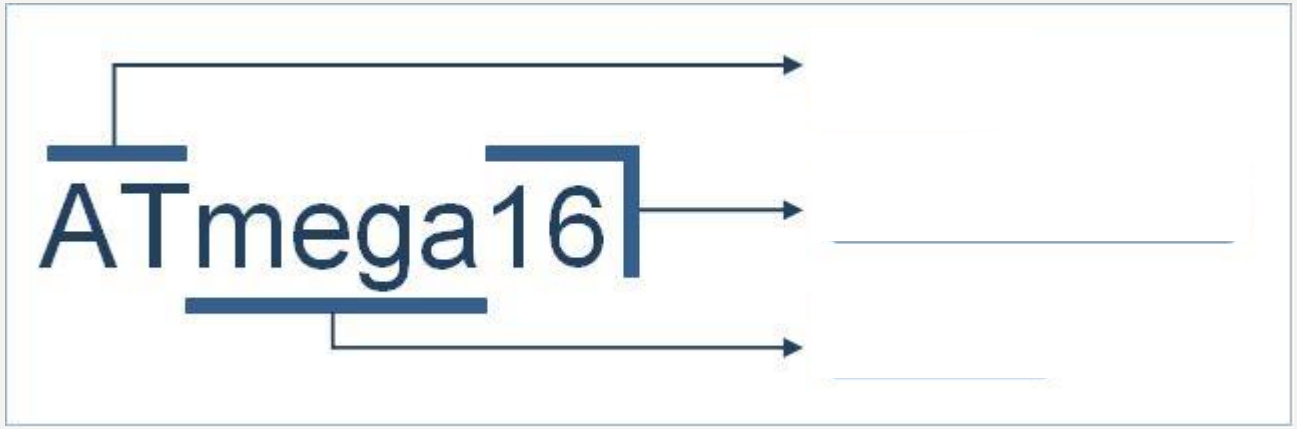
AVR microcontrollers are available in three (+2 new) categories:

1. **TinyAVR** – Less memory, small size, suitable for simple applications
2. **MegaAVR** – the most popular ones: good amount of memory (upto 256 KB), higher number of inbuilt peripherals and suitable for moderate to complex applications.
3. **XmegaAVR** – Used commercially for complex applications, which require large program memory and high speed.

GOOD TO KNOW

4. **Application-specific AVR** – megaAVRs with special features not found on the other members of the AVR family, such as LCD controller, USB controller, advanced PWM, CAN, etc.
5. **FPSLIC (AVR with FPGA)** –FPGA 5k to 40k gates + SRAM for the AVR program code+ can run at up to 50 MHz

NAMING CONVENTION



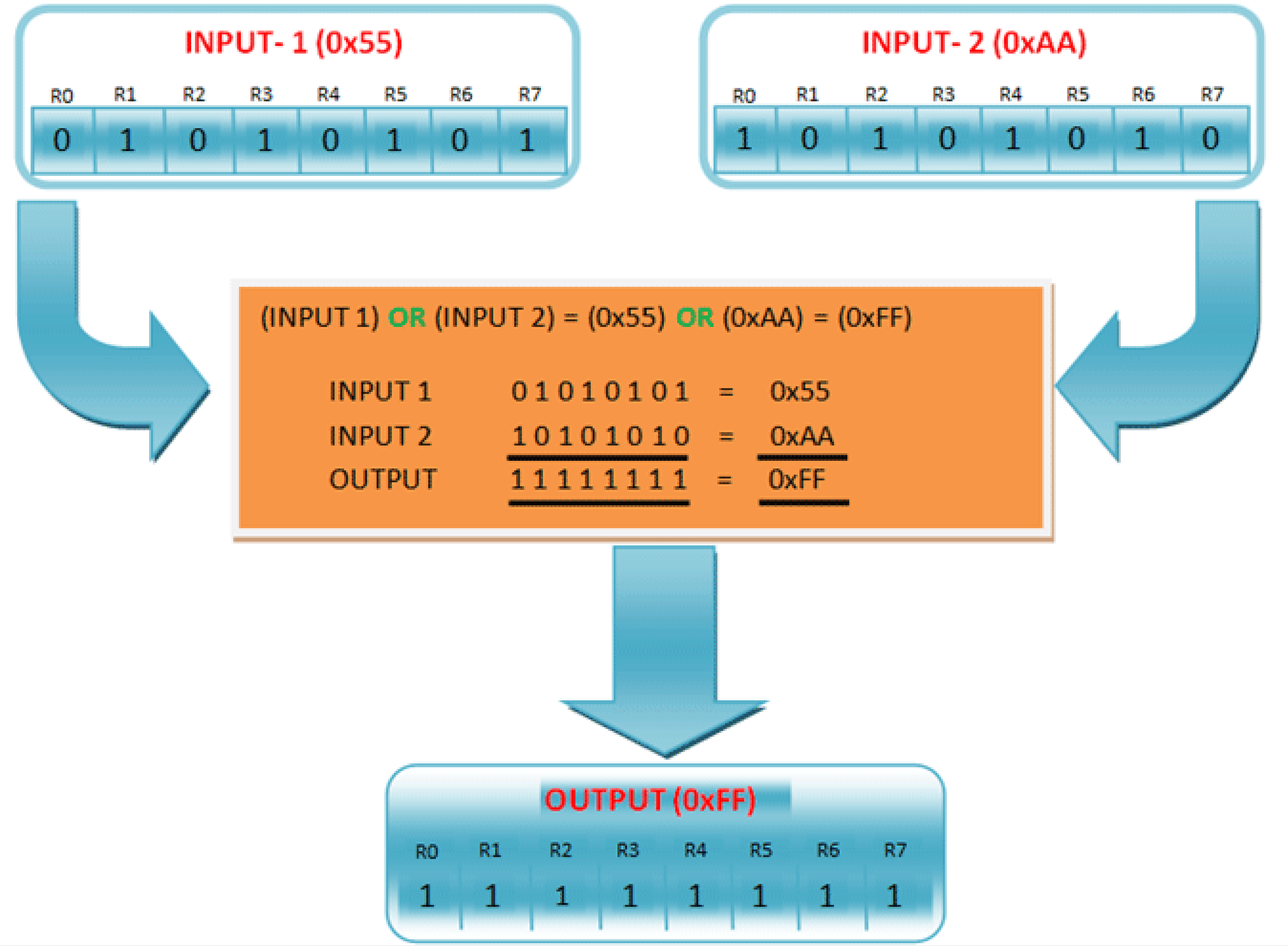
IMPORTANCE OF AVR

What's special about AVR?

- Execute most of the instructions in single execution cycle.
- consume less power and can be operated in different power saving modes.
- 8-bit RISC microcontroller.

What is the other type of ISA categorization?

WHAT IS HAPPENING HERE?



INPUT- 1 (0x55)

R0	R1	R2	R3	R4	R5	R6	R7
0	1	0	1	0	1	0	1

INPUT- 2 (0xAA)

R0	R1	R2	R3	R4	R5	R6	R7
1	0	1	0	1	0	1	0

(INPUT 1) OR (INPUT 2) = (0x55) OR (0xAA) = (0xFF)

INPUT 1	0 1 0 1 0 1 0 1	=	0x55
INPUT 2	1 0 1 0 1 0 1 0	=	0xAA
OUTPUT	1 1 1 1 1 1 1 1	=	0xFF

OUTPUT (0xFF)

R0	R1	R2	R3	R4	R5	R6	R7
1	1	1	1	1	1	1	1

How many cycles?

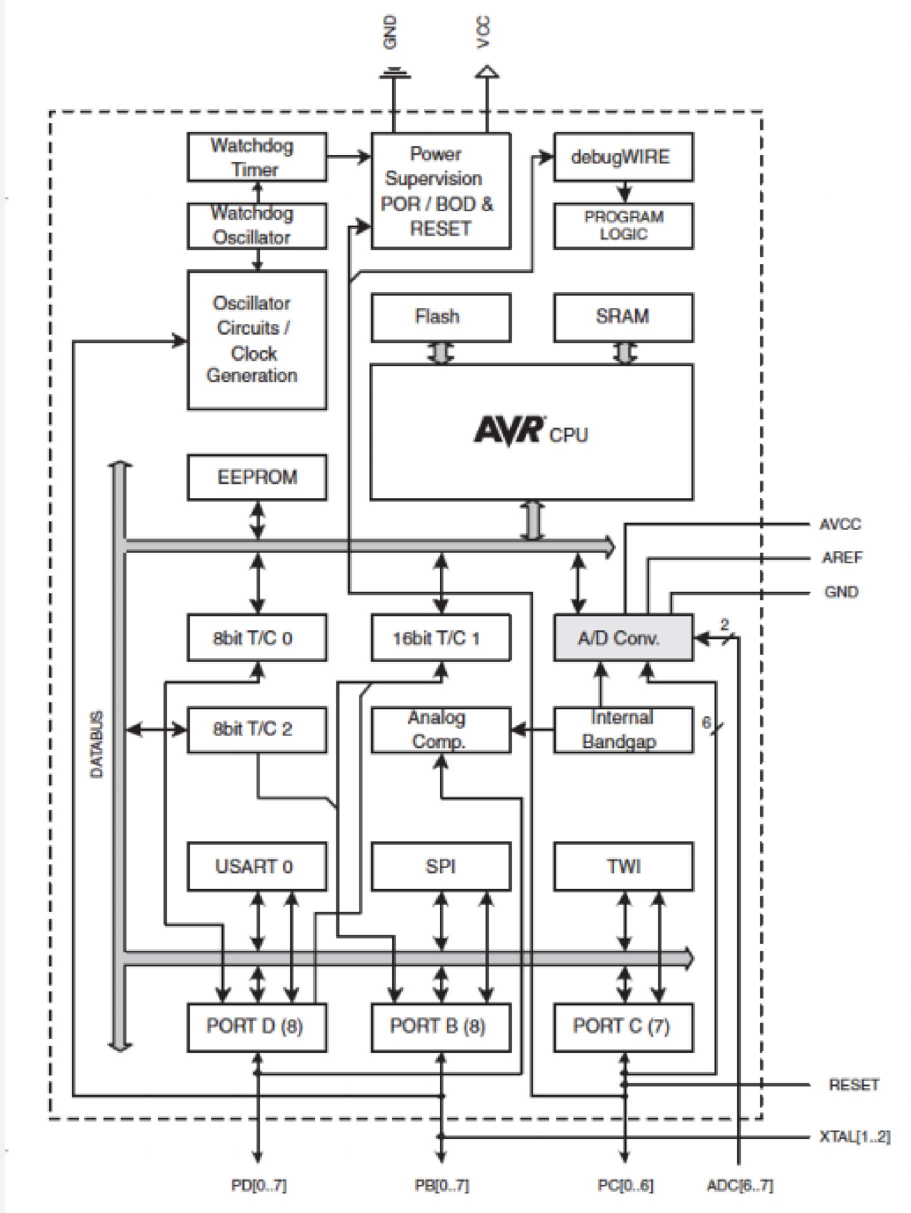
AVR ARCHITECTURE

- The AVR microcontrollers = RISC architecture and consist of 32 x 8-bit general purpose registers.

- Within one single clock cycle, AVR can take inputs from two general purpose registers and put them to ALU for carrying out the requested operation, and transfer back the result to an arbitrary register.

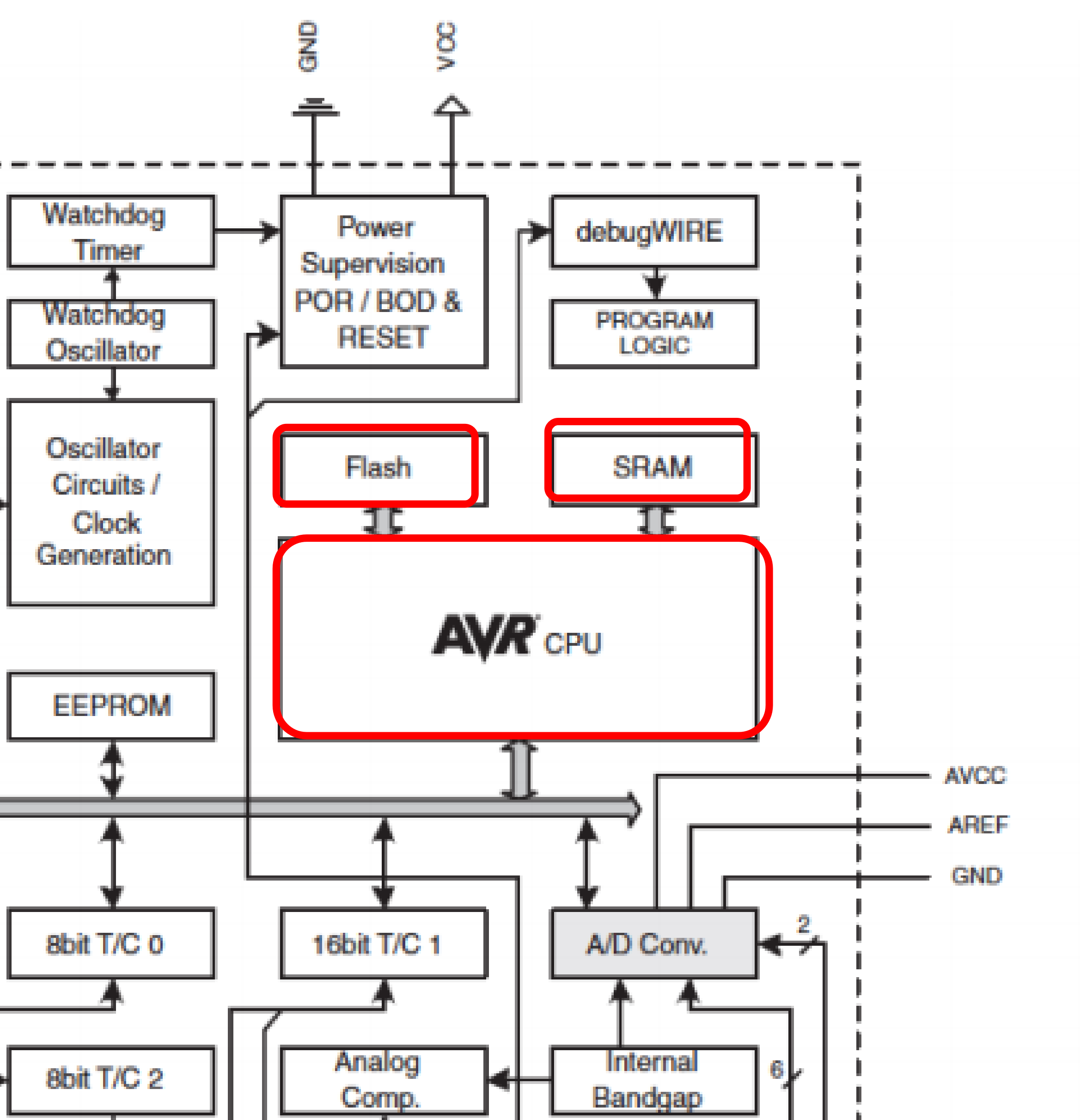
- **How many instructions per second can AVR execute if cycle frequency is 1MHz? Prove it!**

AVR MICROARCHITECTURE

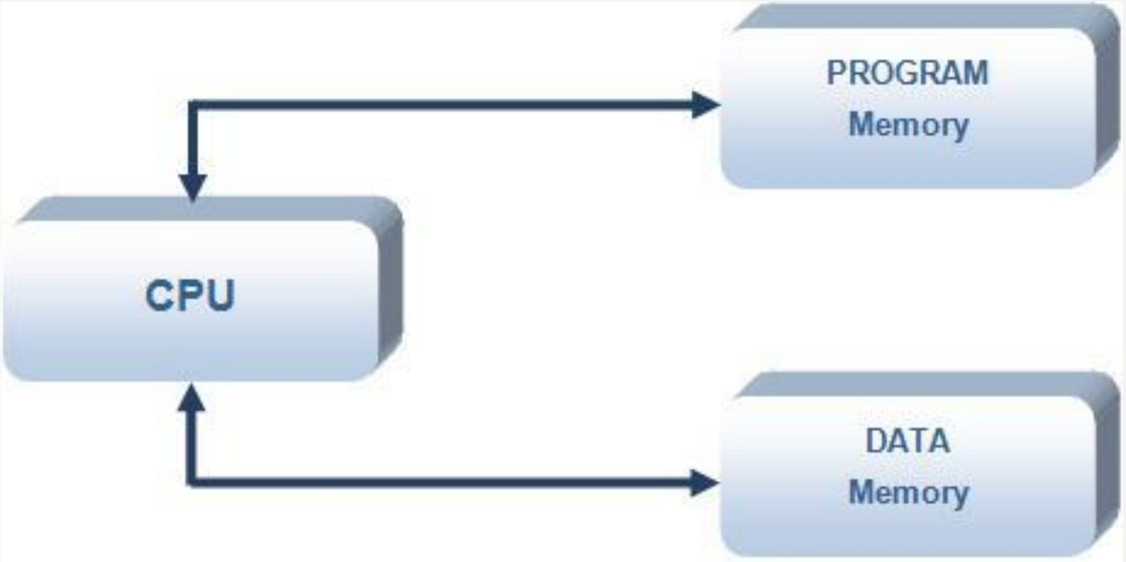


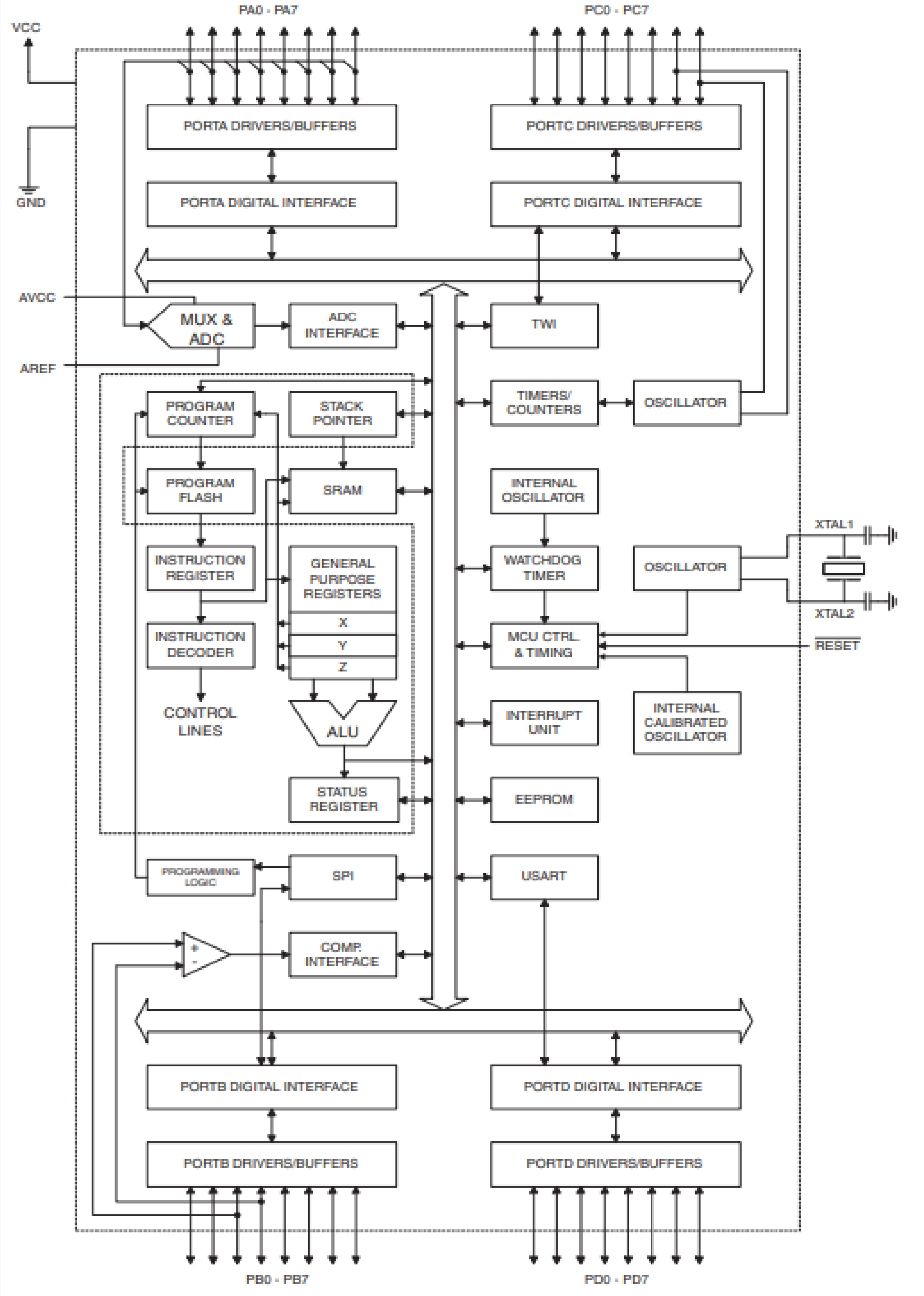
- Do you recognize the microarchitecture type?

AVR MICROARCHITECTURE



- Do you recognize the microarchitecture type?





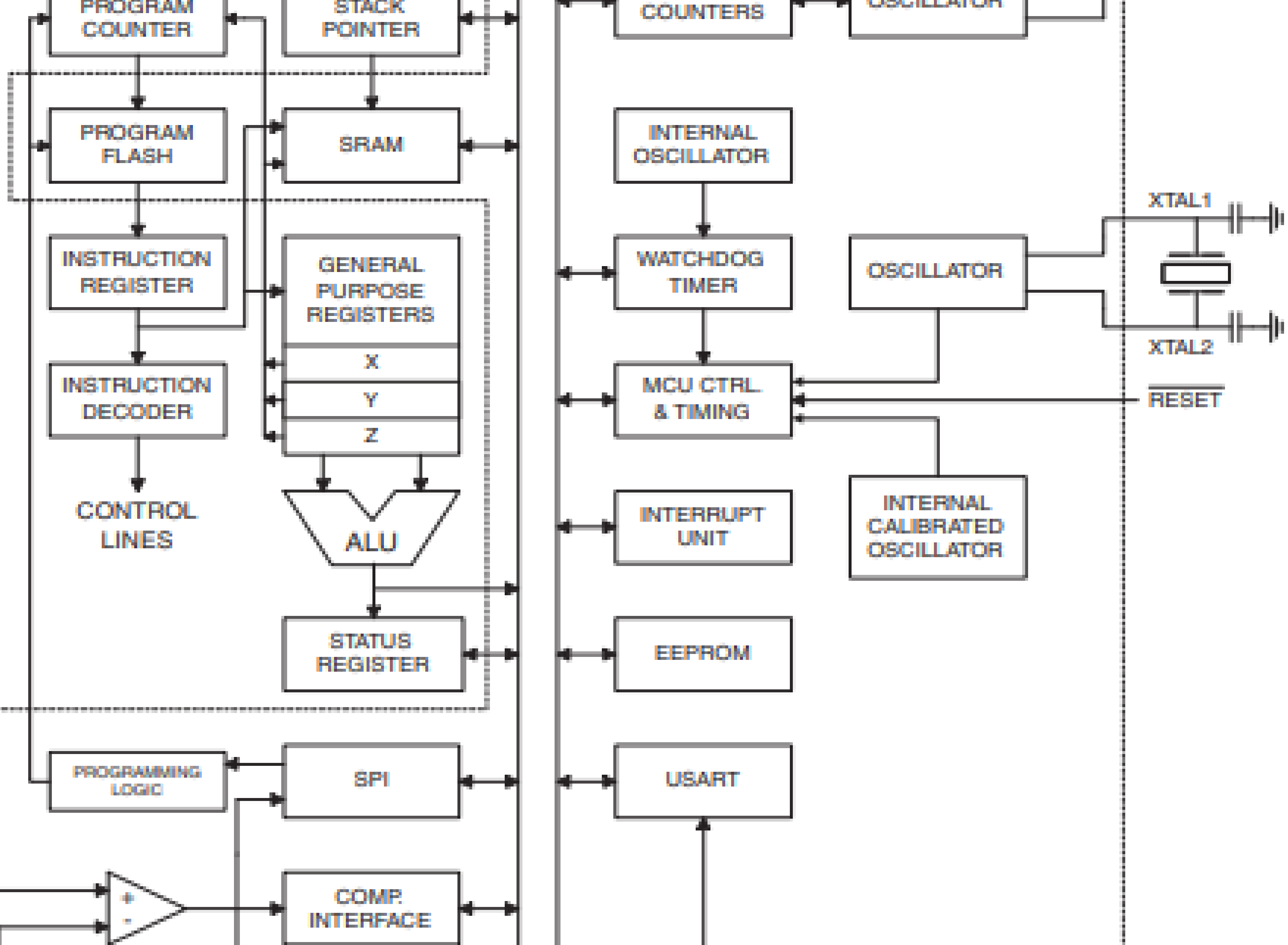
PINS?

CPU?

ALWAYS REFER TO THE DATA SHEET!

ATMEGA328 BLOCK DIAGRAM

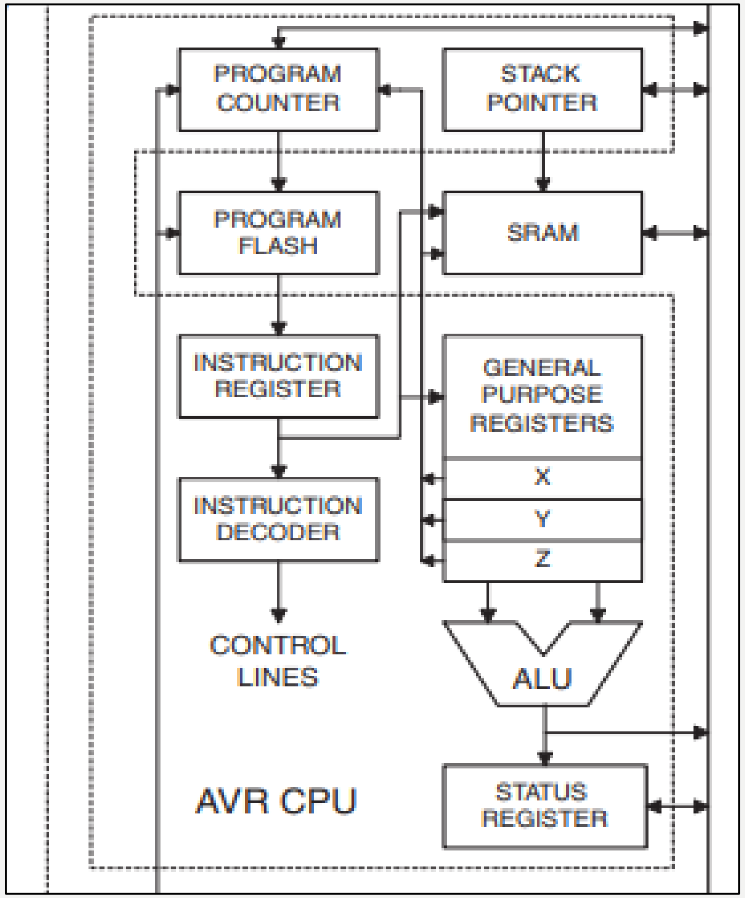
- 32 general purpose working registers directly connected to the ALU
- 32Kbytes of In-System Programmable Flash Program memory
- 1024bytes EEPROM
- 2Kbyte SRAM
- 32 general purpose I/O lines
- JTAG interface



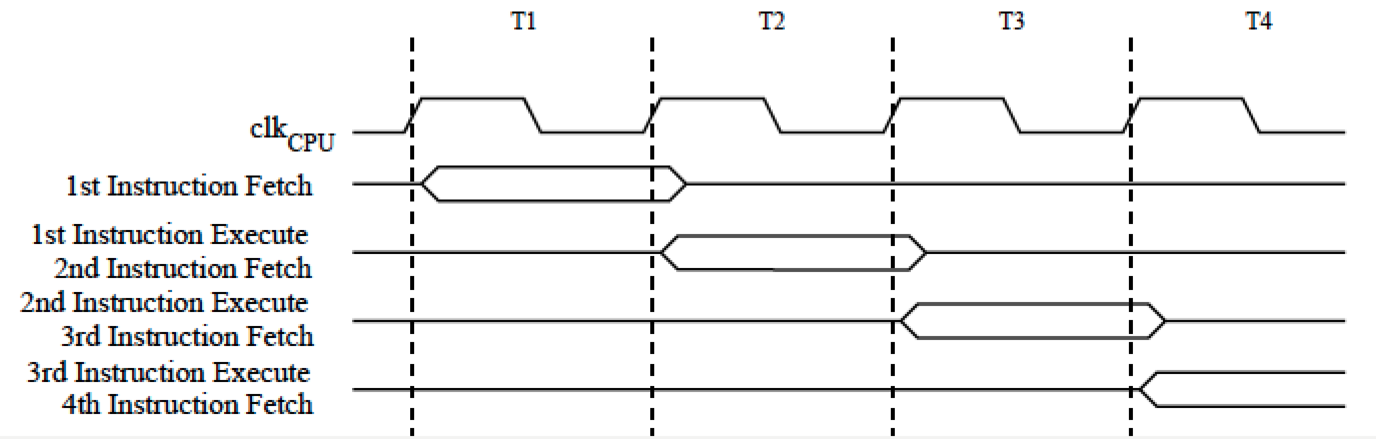
ALWAYS REFER TO THE DATA SHEET!

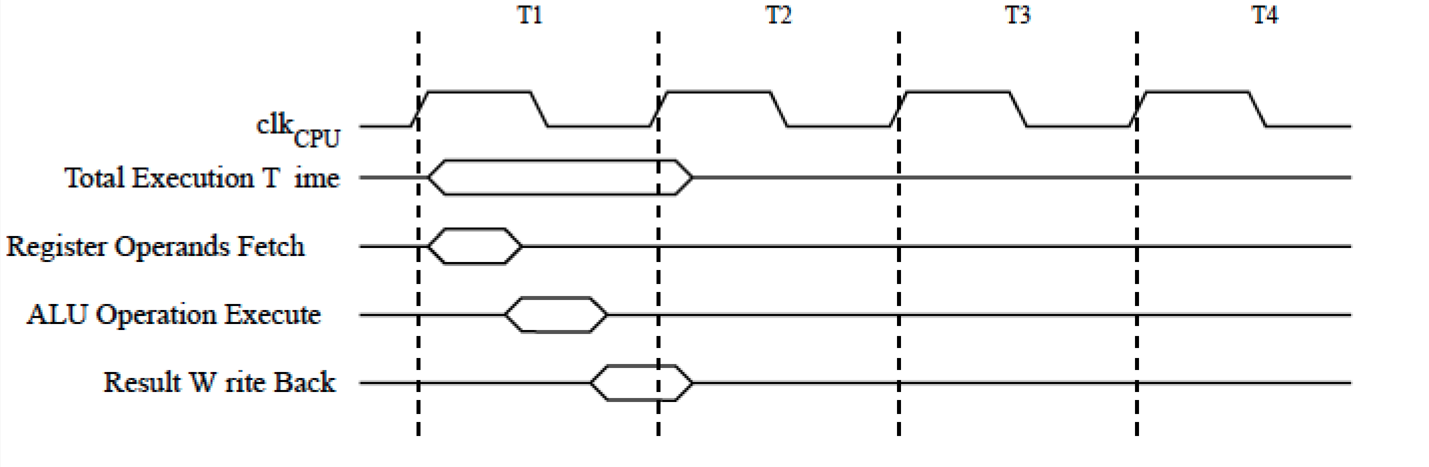
AVR CPU CORE

- Harvard architecture: with separate memories and buses for program and data.
- Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory.
- This concept enables instructions to be executed in every clock cycle.

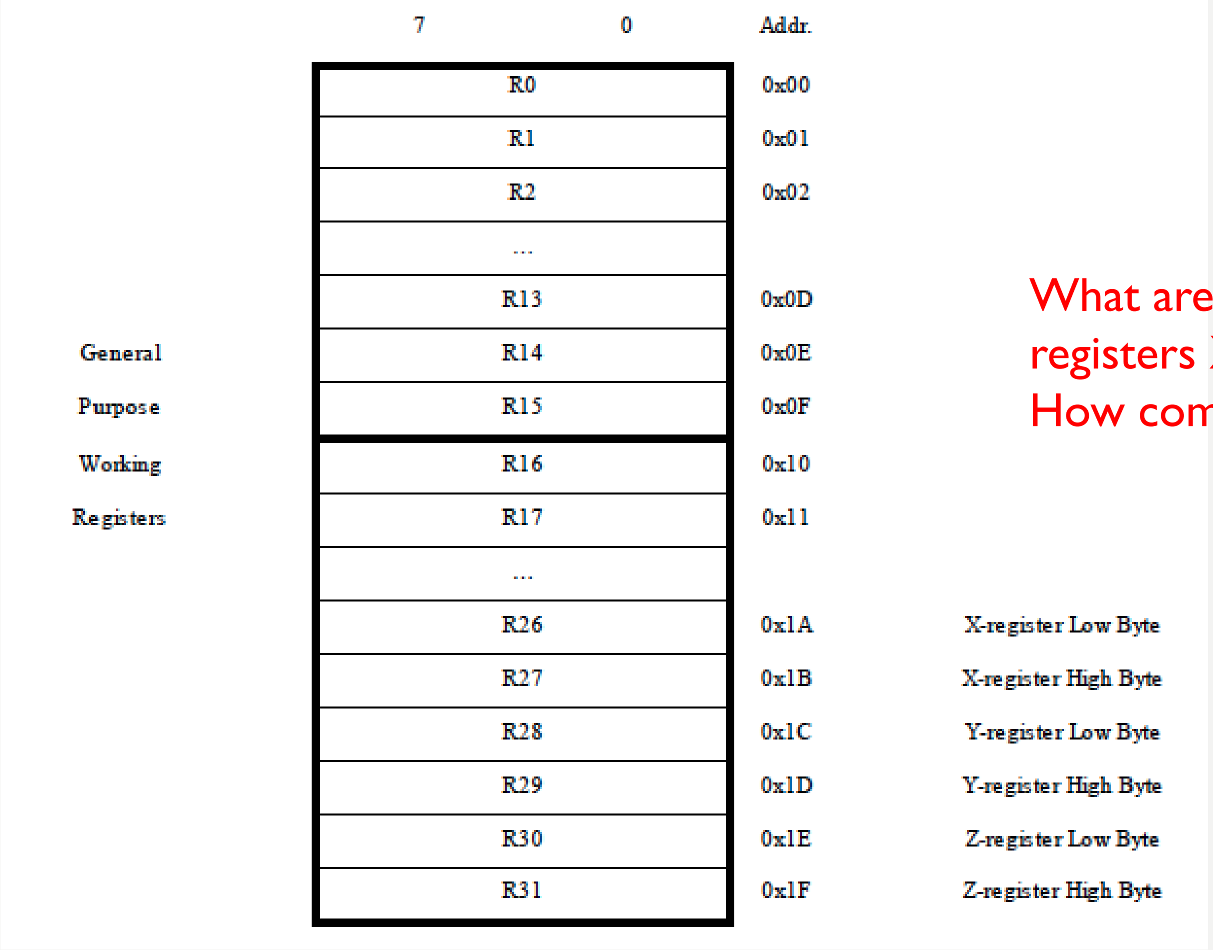


AVR INSTRUCTION TIMING



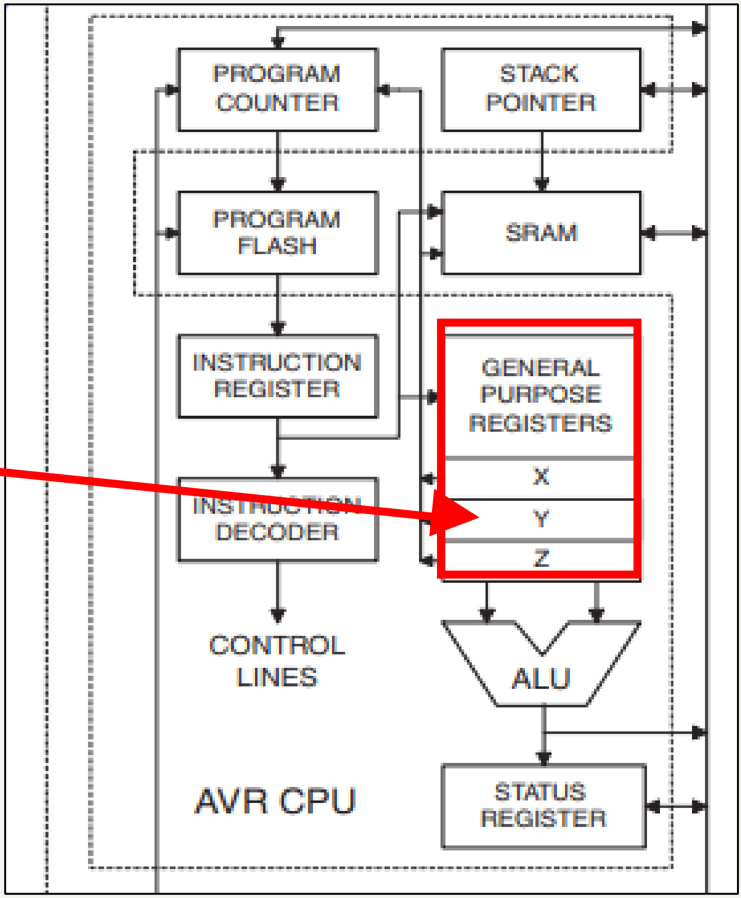


AVR REGISTERS

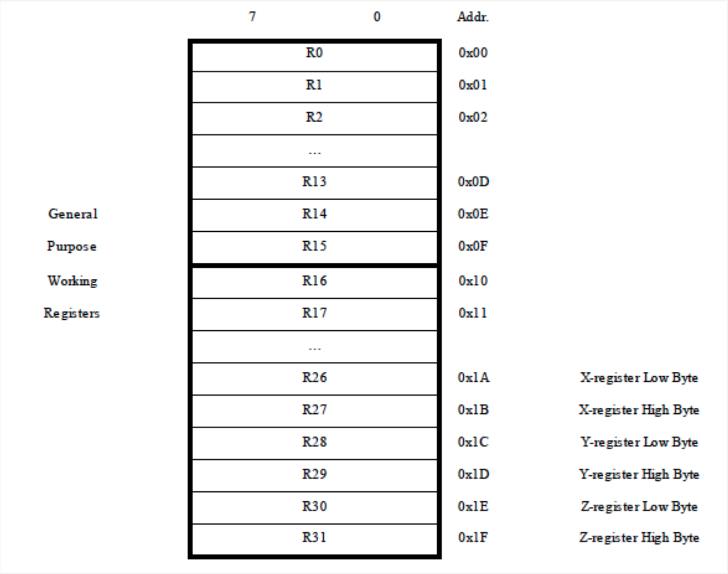


	7 0	Addr.	
General Purpose Working Registers	R0	0x00	
	R1	0x01	
	R2	0x02	
	…		
	R13	0x0D	
	R14	0x0E	
	R15	0x0F	
	R16	0x10	
	R17	0x11	
	…		
	R26	0x1A	X-register Low Byte
	R27	0x1B	X-register High Byte
	R28	0x1C	Y-register Low Byte
	R29	0x1D	Y-register High Byte
	R30	0x1E	Z-register Low Byte
	R31	0x1F	Z-register High Byte

What are the sizes of registers X,Y and Z? How come?



AVR REGISTERS



	7 0	Addr.	
General Purpose Working Registers	R0	0x00	
	R1	0x01	
	R2	0x02	
	...		
	R13	0x0D	
	R14	0x0E	
	R15	0x0F	
	R16	0x10	
	R17	0x11	
	...		
	R26	0x1A	X-register Low Byte
	R27	0x1B	X-register High Byte
	R28	0x1C	Y-register Low Byte
	R29	0x1D	Y-register High Byte
	R30	0x1E	Z-register Low Byte
	R31	0x1F	Z-register High Byte

- The fast-access Register File contains 32 × 8-bit general purpose working registers with a single clock cycle access time: This allows single-cycle Arithmetic Logic Unit (ALU) operation.

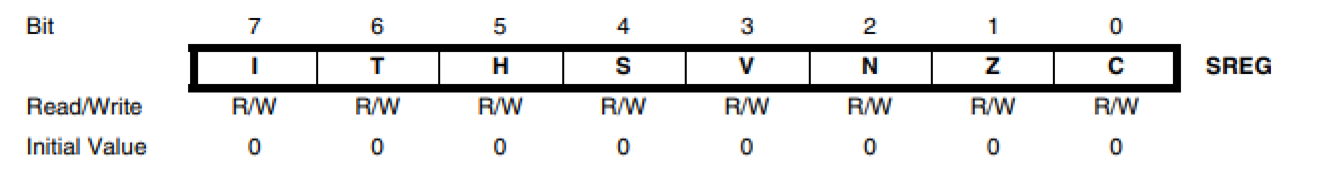
- In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

- Six of the 32 registers can be used as three 16-bit indirect address register pointers (X-Y-Z) for Data Space addressing – enabling efficient address calculations. One of the these address pointers can also be used as an address pointer for look up tables in Flash Program memory.

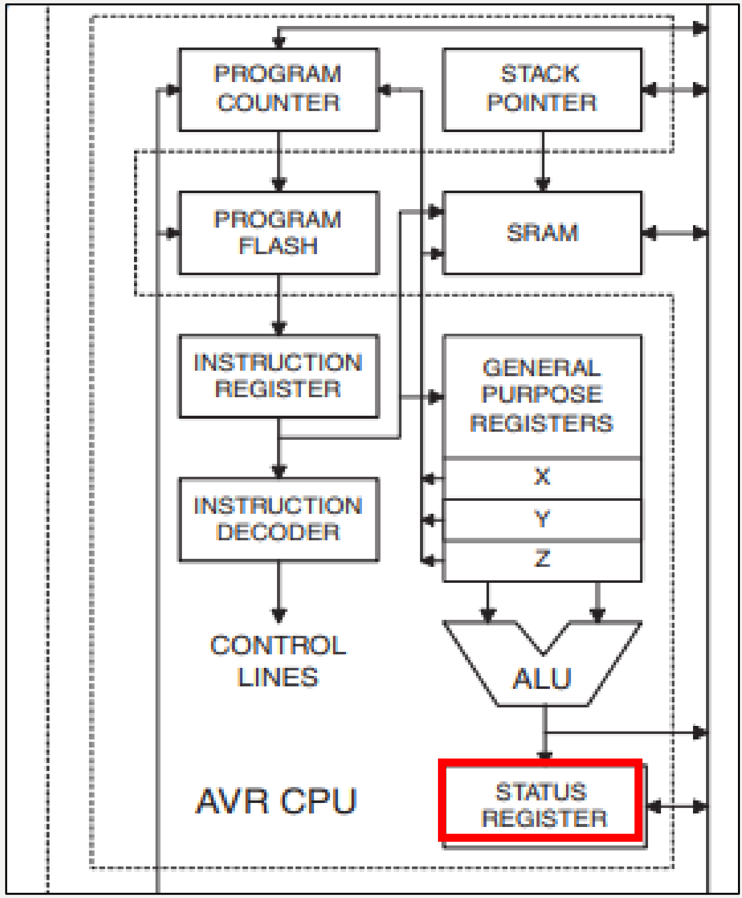
STATUS REGISTER

- The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations.

- SREG is updated after all ALU operations….SO WHAT?
- This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

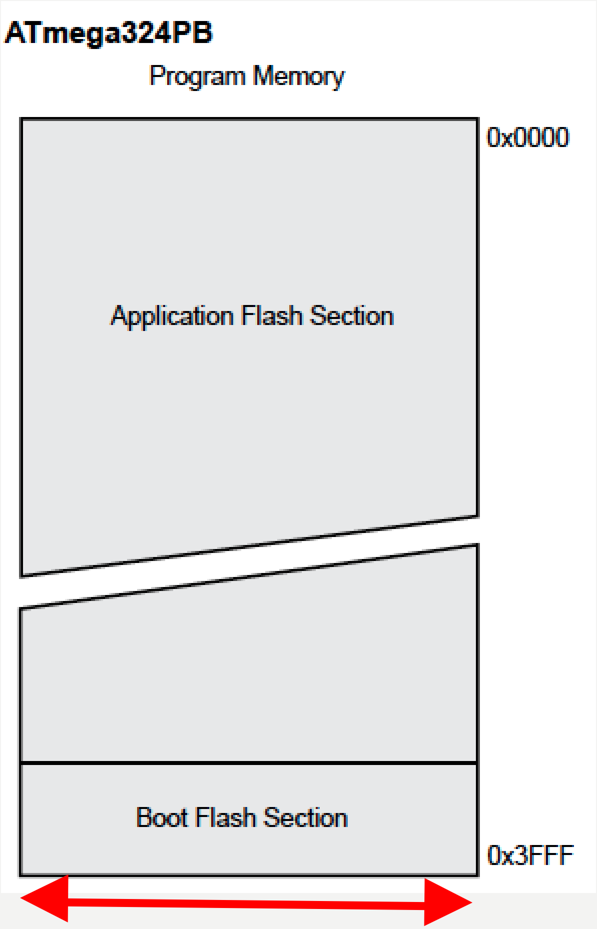


Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

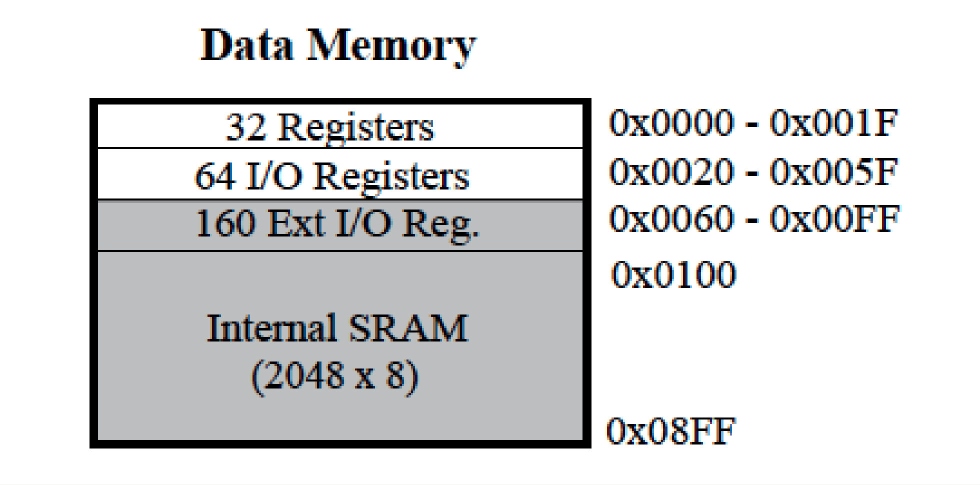


Details in tutorial!!

AVR MEMORIES



Word size?



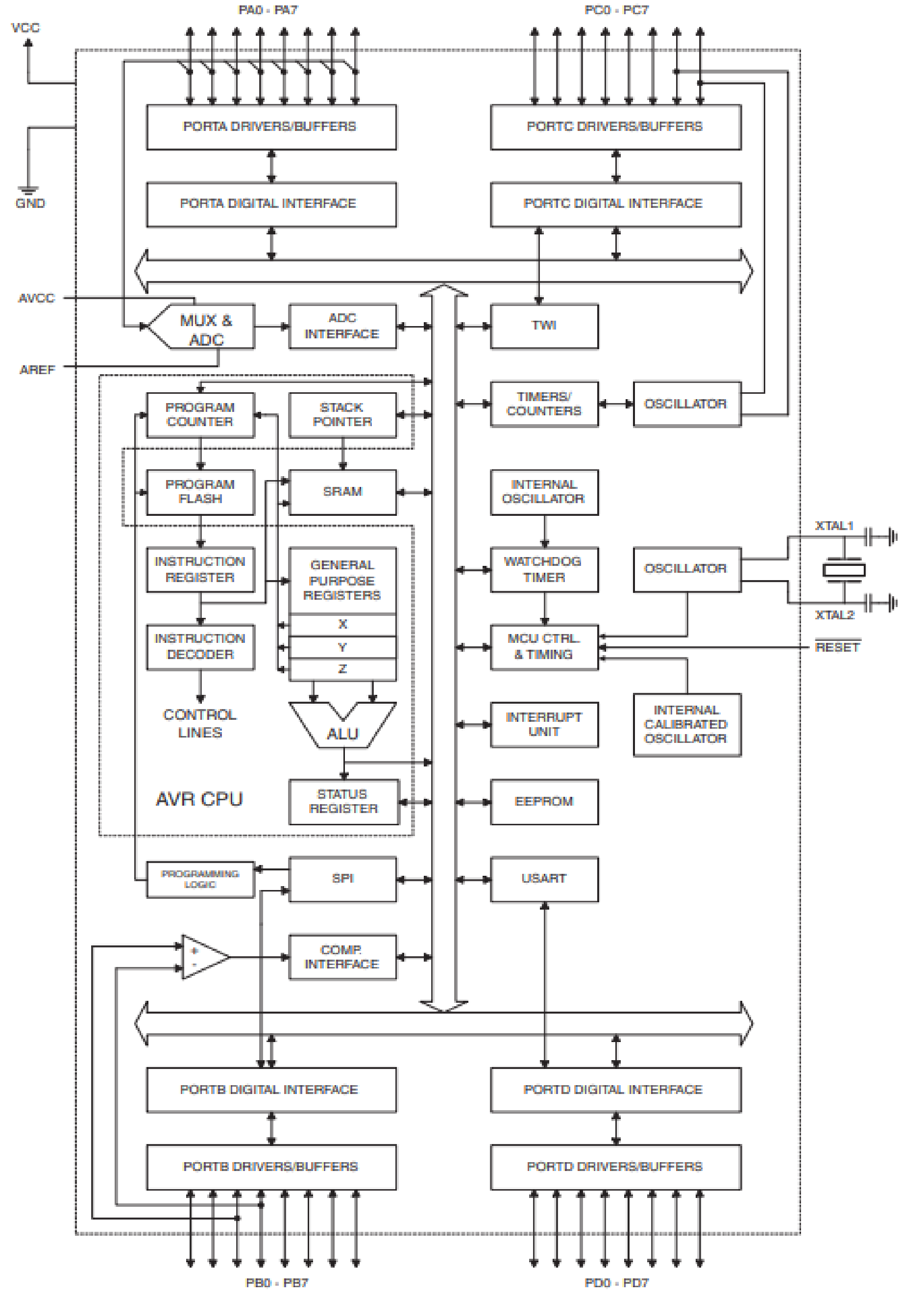
Data Memory

32 Registers	0x0000 - 0x001F
64 I/O Registers	0x0020 - 0x005F
160 Ext I/O Reg.	0x0060 - 0x00FF
Internal SRAM (2048 x 8)	0x0100 0x08FF

What's missing?

PIN DESCRIPTIONS

1. VCC Digital supply voltage.
2. GND Ground.
3. RESET’ : Reset Input. A low level on this pin will generate a reset, even if the clock is not running.
4. XTAL1 Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
5. XTAL2 Output from the inverting Oscillator amplifier.
6. AVCC the supply voltage pin for Port A and the A/D Converter.
7. AREF is the analog reference pin for the A/D Converter.



PIN DESCRIPTIONS

1. Port A (PA7..PA0) serves as the analog inputs to the A/D Converter. Port A also serves as an 8-bit bi-directional I/O port with internal pull-up resistors, if the A/D Converter is not used.

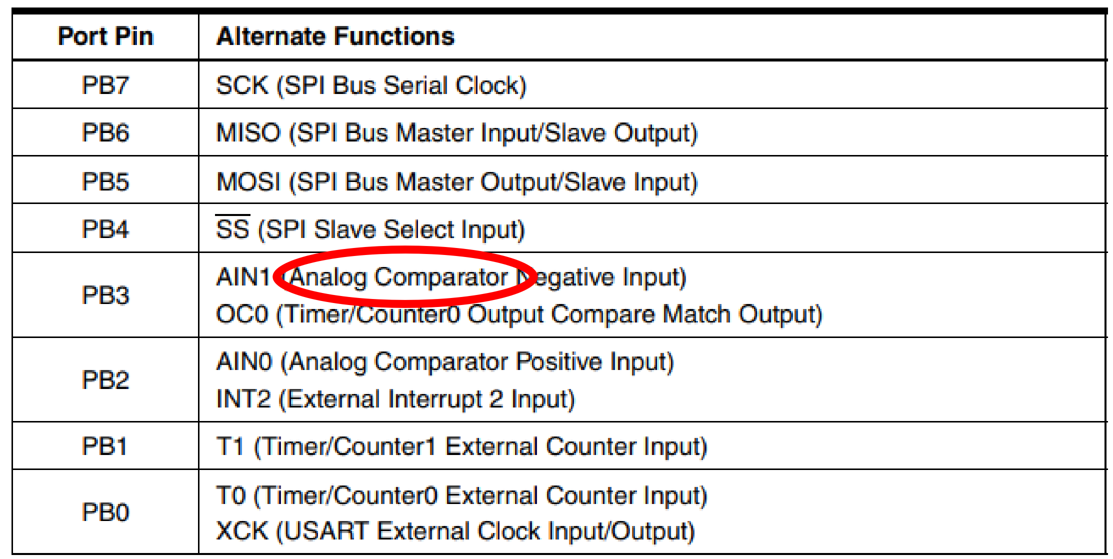
2. Port B (PB7..PB0) is an 8-bit bi-directional I/O port with internal pull-up resistors. +*

3. Port C (PC7..PC0) is an 8-bit bi-directional I/O port with internal pull-up resistors. Port C also serves the functions of the JTAG interface +*

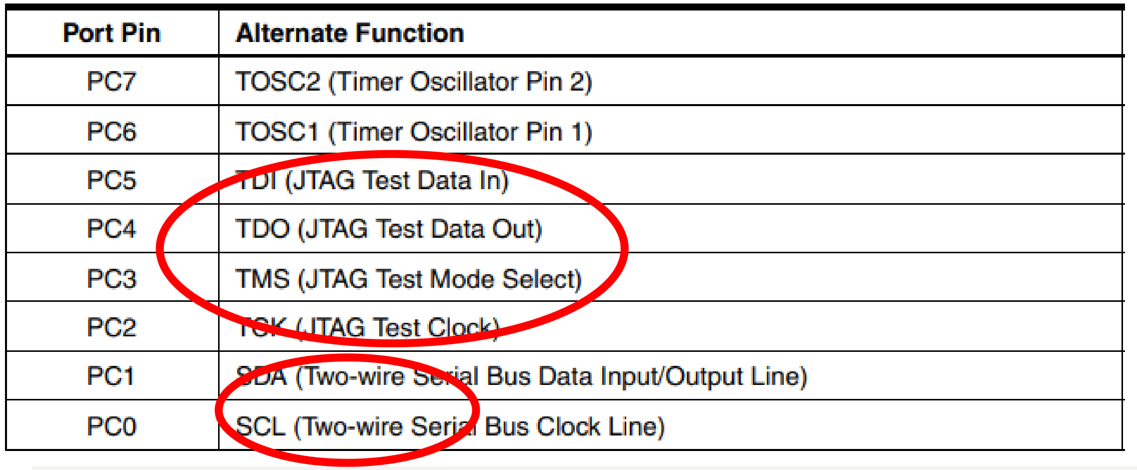
4. Port D (PD7..PD0) Port D is an 8-bit bi-directional I/O port with internal pull-up resistors +*

*+ various special features of the ATmega32

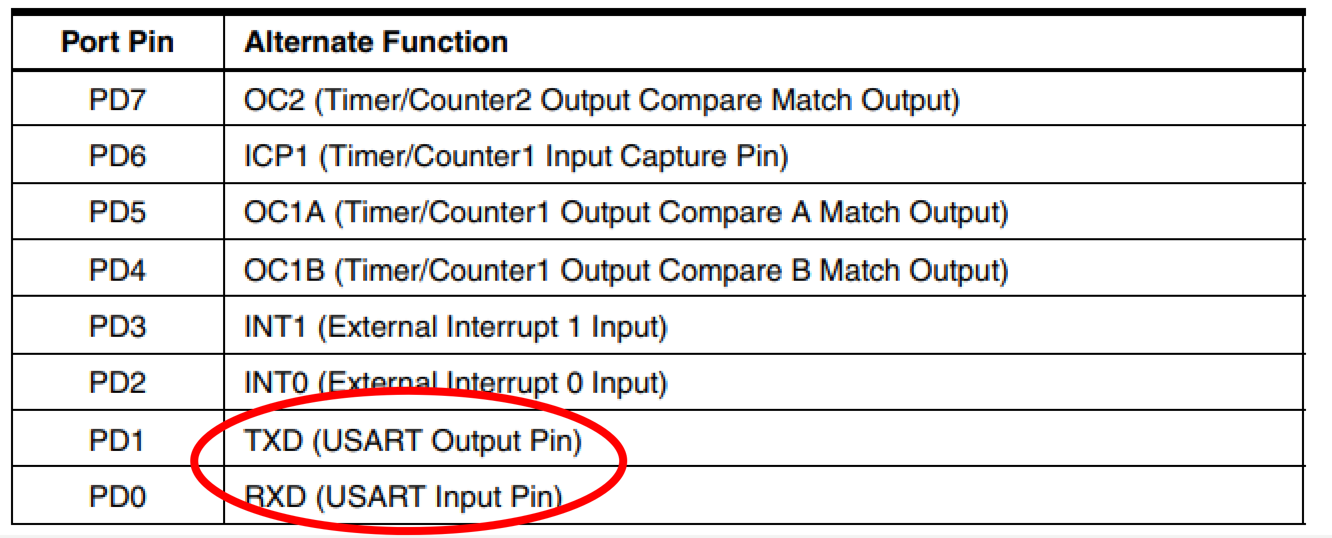
PORTS' SPECIAL FEATURES



Port Pin	Alternate Functions
PB7	SCK (SPI Bus Serial Clock)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB5	MOSI (SPI Bus Master Output/Slave Input)
PB4	$\overline{\text{SS}}$ (SPI Slave Select Input)
PB3	AIN1 (Analog Comparator Negative Input) OC0 (Timer/Counter0 Output Compare Match Output)
PB2	AIN0 (Analog Comparator Positive Input) INT2 (External Interrupt 2 Input)
PB1	T1 (Timer/Counter1 External Counter Input)
PB0	T0 (Timer/Counter0 External Counter Input) XCK (USART External Clock Input/Output)



Port Pin	Alternate Function
PC7	TOSC2 (Timer Oscillator Pin 2)
PC6	TOSC1 (Timer Oscillator Pin 1)
PC5	TDI (JTAG Test Data In)
PC4	TDO (JTAG Test Data Out)
PC3	TMS (JTAG Test Mode Select)
PC2	TCK (JTAG Test Clock)
PC1	SDA (Two-wire Serial Bus Data Input/Output Line)
PC0	SCL (Two-wire Serial Bus Clock Line)



Port Pin	Alternate Function
PD7	OC2 (Timer/Counter2 Output Compare Match Output)
PD6	ICP1 (Timer/Counter1 Input Capture Pin)
PD5	OC1A (Timer/Counter1 Output Compare A Match Output)
PD4	OC1B (Timer/Counter1 Output Compare B Match Output)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)