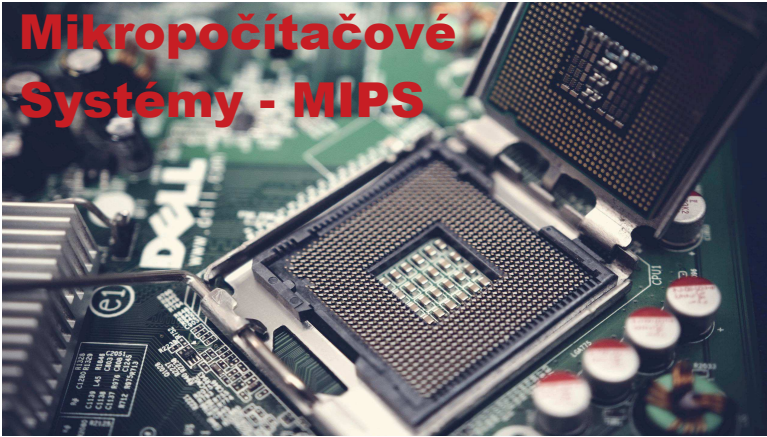
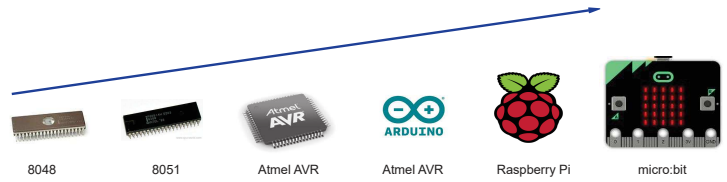


Mikropočítačové Systémy - MIPS



Richard Balogh
Štefan Chamraz



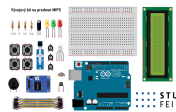
Cvičenia



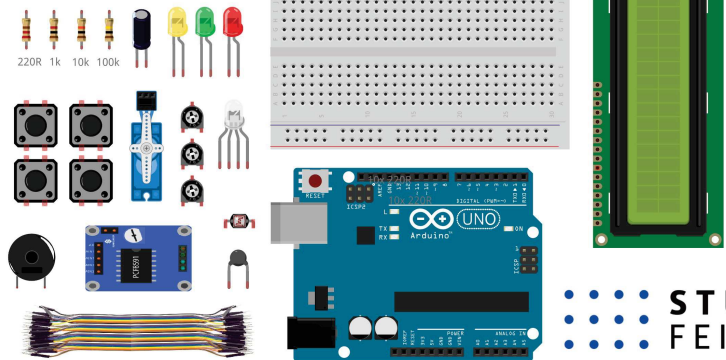
Google Classroom



SENSOR



Vývojový kit na predmet MIPS



Podmienky na absolvovanie predmetu:

Cvičenia: X bodov za úlohy z cvičení (2021: 43b.) **TERMÍN?**
a záverečný projekt (20b.)

Skúška : 100 - X bodov (2021: 37b)

3



arduino

embedded c

MICROCHIP Products Solutions Tools and Resources Support Education About Order Now

Overview Features Development Environment Similar Devices RoHS Information Purchase

ATmega328PB ☆

Status: In Production

Download Data Sheet Documentation Symbols

The high-performance, low-power Microchip 8-bit AVR® RISC-based microcontroller combines 64 KB SPI Flash memory with read-while-write capabilities, 2 KB EEPROM, 4 KB SRAM, 27 general purpose I/O lines, 32 general purpose working registers, two flexible timer/counters with compare modes and PWM, one UART with HW LIN, an 11-channel 10-bit A/D converter with two differential programmable gain input stages, a 16-bit D/A converter, a programmable watchdog timer with an internal individual oscillator, SPI serial port, an on-chip debug system, and four software selectable power saving.

Purchase Options

Sampling Options

Programming Services

Read More

Product Features

- Recommended for Automotive Design
- Functional Safety Ready
- 131 Powerful Instructions
- Most Single Clock Cycle Execution
- 21 x 8 General Purpose Working Registers
- Fully Static Operation
- Up to 20 MIPS Throughput at 20 MHz
- On-Chip 2-Cycle Multiplier
- 22 KB of In-System Self-Programmable Flash program memory
- 1 KB EEPROM

Parameters

Click on a property to perform a parametric search for other products with that property.

Max ADC Resolution (bits)	10
Program Memory Size (KB)	32

Physical / Virtual Computing



Jean Jennings, Marlyn Wescoff a Ruth Lichterman programujú počítač ENIAC na Pensylvánskej univerzite v USA, 1946.

Physical / Virtual Computing

```

# generate 2d classification dataset
x, y = make_circles(n_samples=1000, noise=0.1, random_state=1)

# scatter plot for each class value
for class_value in range(2):
    # select indices of points with the class label
    row_ix = np.where(y == class_value)
    # scatter plot for points with a different color
    pyplot.scatter(x[row_ix, 0], x[row_ix, 1])
# show plot
pyplot.show()
    
```



Vývoj HW



4+1

1 MB pamäte
550 W
29 kg
1,2 MIPS

6000 x
100 x
150 x
8000 x

4+4

6 GB RAM + 64 GB sd
5 W
191 g
10 000 MIPS

Vývoj SW

```

.ORG 0030h
DB 03h,01h,02h,03h
.ORG 0000

LDA 0030H ; Initialize C-counter
MOV C,A ; Initialize C-counter
SUB A ; sum = 0
LXI H,0031H ; Initialize pointer
ADD M ; SUM = SUM + data
INX H ; increment pointer
DCR C ; Decrement counter
JNZ BACK ; if counter 0 repeat
STA 040H ; Store sum
    
```

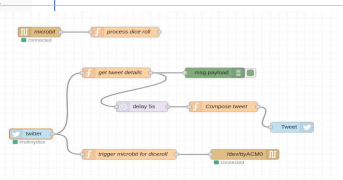
```

for (;;)
{
    c = (char)rand();
    printf("%d,%d\n", (int) (100*(sin(x)+1)), c);
    if (c>245)
        printf("Alarm! Temperature HIGH!!!\n");
    if (c<5)
        printf("Alarm! Temperature LOW!!!\n");
    delay_ms(300);
    x=x+0.1;
}
return 0;
/* End of Main */
    
```

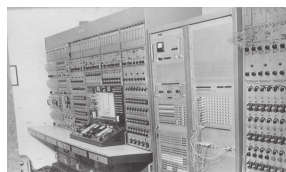
```

from microbit import *
display.show(Image.YES)

while True:
    if button_a.is_pressed() and button_b.is_pressed():
        display.show(Image.HEART)
        sleep(500)
    elif button_a.is_pressed():
        display.show(Image.HAPPY)
    elif button_b.is_pressed():
        display.show(Image.SAD)
    
```



Miniaturizácia



Vývoj programovacích jazykov: **Asembler**, **jazyk C**, **Python** a **Node-Red**

Miniaturizácia



Moorov zákon

Ak by sa rovnako rozvíjal automobilový priemysel:

300 000 km/h

850 000 km / liter

0,04 \$ / ks



Embedded Computing



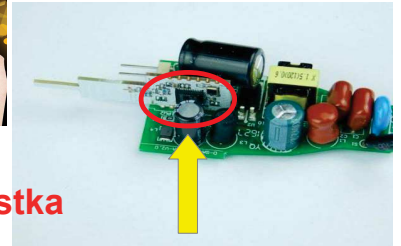
Play Games With the Entire Family!



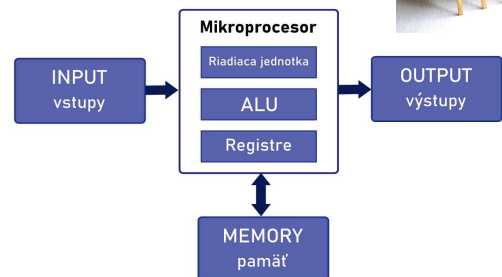
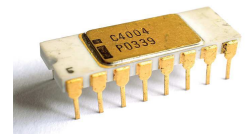
Embedded Computing / vnorené systémy



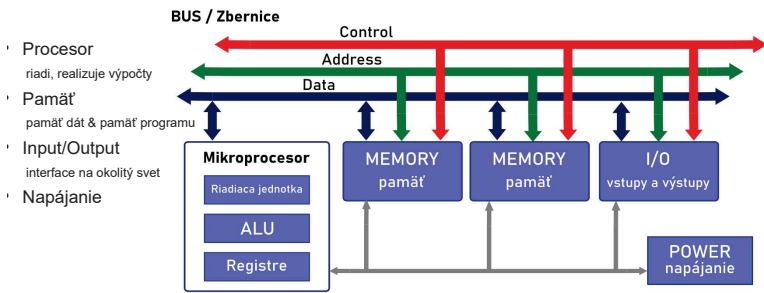
počítač = súčiastka



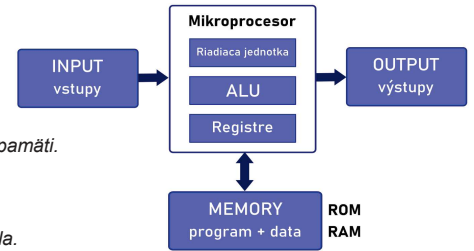
Mikroprocesor



Štruktúra počítača



Architektúra Von Neumann (Princetonská)



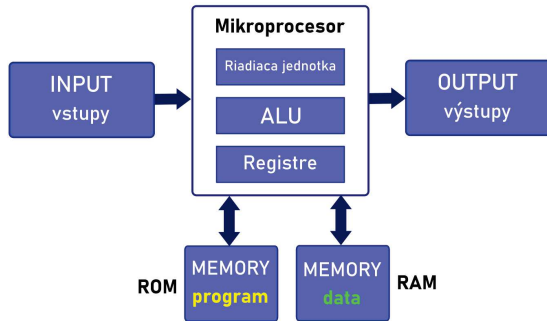
Program a Dáta: sú uložené v pamäti.

- Inštrukcie,
 - dáta (čísla, znaky, ...),
 - adresy,
- sú ukladané ako dvojkové čísla.

Požadujeme: rýchly procesor a veľkú pamäť =>

Presúvanie informácií: „von Neumannov bottleneck“ – John Backus 1977.

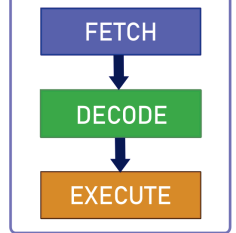
Harvardská architektúra (Aiken)



Inštrukčný cyklus

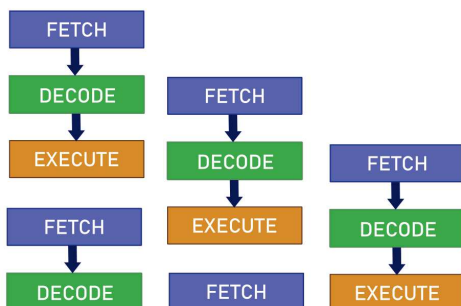
- Výber inštrukcie z pamäte
- Dekódovanie inštrukcie
- Výber operandov
- Vykonanie požadovanej operácie
- Zápis výsledkov do pamäte

Inštrukčný cyklus



Paralelizmus: pipeline

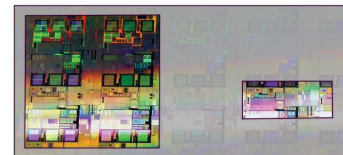
3-stupňová pipeline



CISC



MULT A, B



RISC



LOAD r1, A
LOAD r2, B
MULT r1, r2
STORE

CISC: málo (4-100)

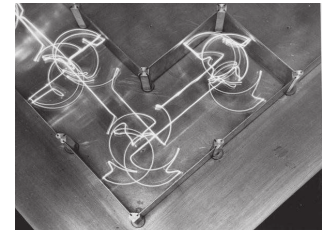
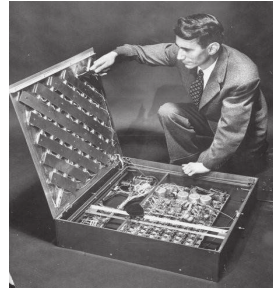
execution time = počet inštrukcií x CPI x frekvencia

RISC: veľa 1

Historické okienko

1952

C. Shannon - Theseus: Maze-Solving Mouse (1952)



Source: <http://cyberneticzoo.com>

1974

From CPU to software, the 8080 Microcomputer is here.

Intel's new 8080 8-channel microcomputer is tremendously easy to interface, simple to program and with up to 100 times the performance of p-channel MOS microcomputers.

Best of all, the 8080 is real—in production at low and available in volume quantities today. It's also available through distributors along with a growing list of peripheral circuits and a new version of the Intel 808 program and hardware development system for the 8080.

all supported with software packages, design documentation and manuals, and backed by more than 100 man years of microcomputer expertise.

The 8080 is the inevitable successor to complex custom MOS and many large discrete logic subsystems. It is the industry's first general purpose 8-channel microcomputer and the first high performance single-chip CPU with extremely simple interface requirements and straightforward programming. It runs a full instruction cycle in 2 microseconds.

As such, the 8080 extends the economic benefits of Intel's p-channel microcomputers to a new universe of systems that need fast, multipoint controllers and processors. These systems include intelligent terminals, point of sale systems, process and numeric controllers, advanced calculations, word processors, self-calibrating instruments, data loggers, communications controllers, and many more.

You can use 256 input and 256 output channels, handle almost unlimited interrupt levels, directly access 64 Kbytes of memory and put many satellite 8080 processors around a single memory.

Interfacing is minimal and design is easy with the 8080 because all controls are fully decided on the CPU chip itself and inputs and outputs are TTL compatible. There are separate data, address and control buses.

The 8080 microcomputer has 76 basic instructions, including the 8088 plus new ones that make possible such features as vectored multi-level interrupt, unlimited subroutine nesting and very fast decimal and binary arithmetic.

Program development for the 8080 can be done either on a large computer using the Intel software cross products (P1, P2) system language compiler, macro-assembler and simulator, or on an Intel 8080 development system with a resident machine test editor and macro-assembler.

The new 8080 product family includes performance matched peripheral and memory circuits configured to maximize design effort and maximize system performance. Large, low cost RAMs, ROMs, EPROMs and I/O devices are available now and we will soon announce other 8080 I/O support circuits.

The 8080 is easier to use and more economical than any high performance microcomputer in its class. It's here now in volume, from the inventors of the microcomputer and the people who lead the industry in production and design support.

Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051, (408) 246-7501.

intel Microcomputers. First from the beginning.

Intel 8080

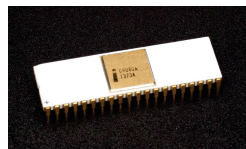
Produced April 1974; 43 years ago(!)

CPU clock: 2 MHz to 3.125 MHz (8080-A)
Package: 40-pin DIP
Power: +5, -5 and +12 V, ~ 1,3 Watt
Military: - 55 °C to +125 °C
Technology: 6 μm, 6 000 transistors, die size 20 mm²

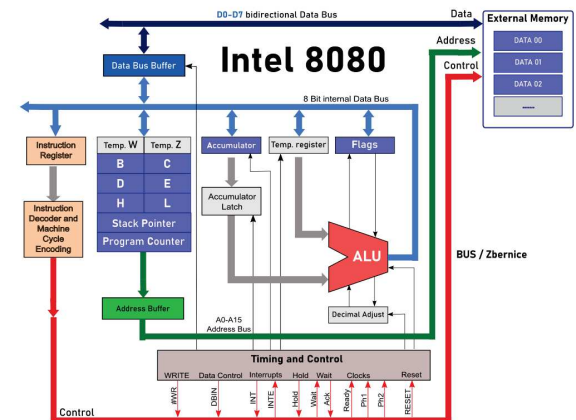
Common manufacturer(s)

Intel, AMD, NEC, National Semiconductor, OKI, Siemens, Texas Instruments, Sovietskij zvez KR580VM80A, Pofsko MCY7880, ČSSR Tesla Piešťany MHB8080A

r. 1979 – 500 000 ks / mesačne za cca \$3 – 4 / ks



By Konstantin Lanzet - CPU collection Camera <https://commons.wikimedia.org/w/index.php?curid=7020599>



Inštrukčná sada procesora 8080

- Skladá sa z
 - 74 základných inštrukcií, napr. MOV
 - 246 konkrétnych inštrukcií, napr. MOV A,B

• 8080 inštrukcie delíme na

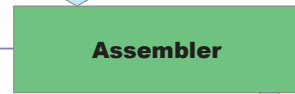
1. Data Transfer (Copy)
2. Arithmetic
3. Logical and Bit manipulation
4. Branch
5. Machine Control

Presunové
Aritmetické
Logické
Skokové
Riadiace

Strojový kód vs. jazyk symbolických adries

```

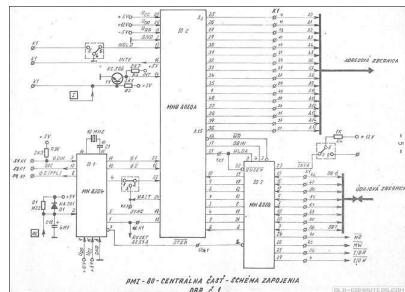
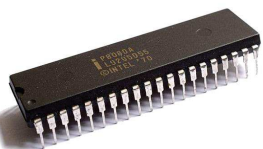
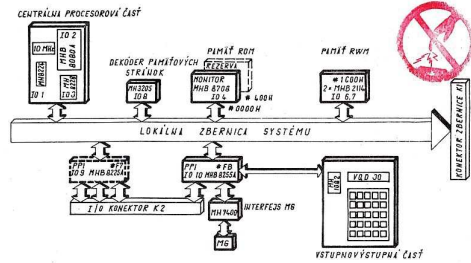
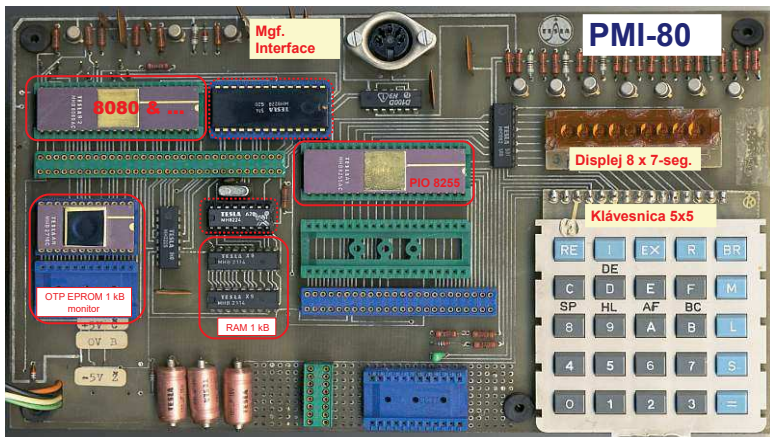
; sucet dvoch cisel A a B
START: MVI A, 01h ; vloz 01 do A
       MVI B, 02h ; vloz 02 do B
       ADD B      ; A = A + B
    
```



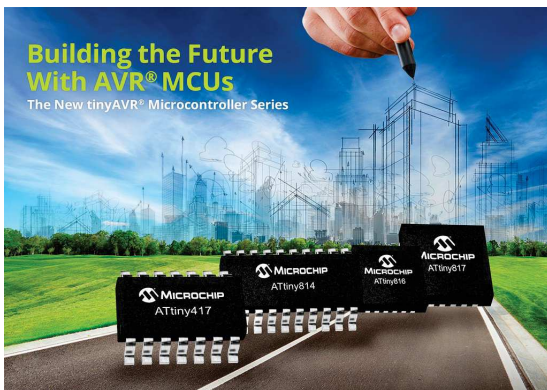
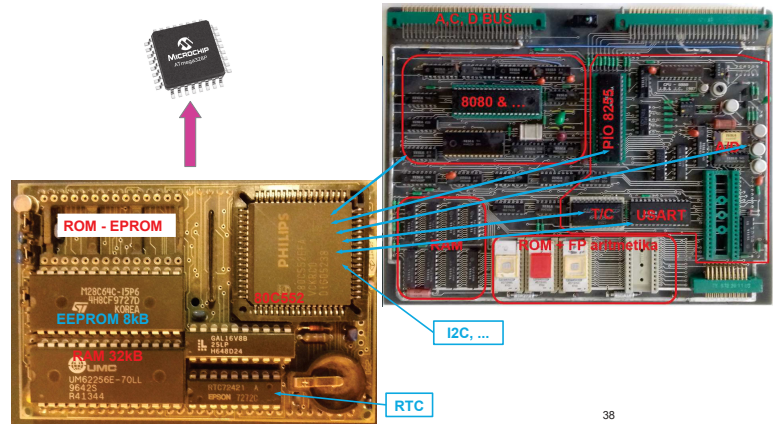
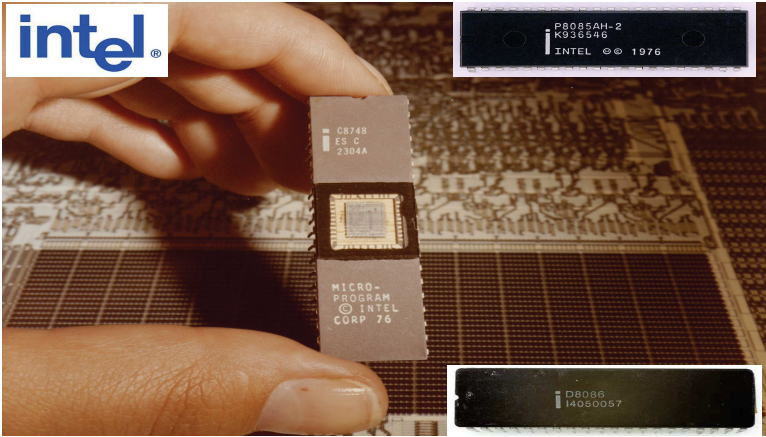
```

0000 3A 0D 00 47 3A 0E 00 80 0008
32 0F 00 76
    
```

Mikroprocesor rozumie len strojovému kódu!



1976



MICROCHIP Products Solutions Tools and Resources Support Education About Order Now

Overview Features Development Environment Similar Devices RoHS Information Purchase

ATmega328PB

Status: In Production.

Download Data Sheet Documentation Symbols

The high-performance, low-power Microchip 8-bit AVR® RISC-based microcontroller combines 64 KB ISP Flash memory with read-while-write capabilities, 2 KB EEPROM, 4 KB SRAM, 20 general purpose I/O lines, 32 general purpose working registers, two flexible timer/counters with compare modes and PWM, one UART with HW LIN, an 11-channel 10-bit A/D converter with two differential programmable gain input stages, a 10-bit D/A converter, a programmable watchdog timer with an internal individual oscillator, SPI serial port, an on-chip debug system, and four software selectable power saving modes.

Product Features

- Recommended for Automotive Design
- Functional Safety Ready
- 131 Powerful Instructions
- Most Single Clock Cycle Execution
- 32 x 8 General Purpose Working Registers
- Fully-Static Operation
- Up to 20 MIPS Throughput at 20 MHz
- On-Chip 2-Cycle Multiplier
- 32 KB of In-System Self-Programmable Flash program memory
- 1 KB EEPROM

Parametrics

Click on a property to perform a parametric search for other products with that property.

Max ADC Resolution (bits)	10
Program Memory Size (KB)	32

AVR



1995 AVR architektúra: **Alf-Egil Bogen a Vegard Wollan**, študenti na *Norwegian Institute of Technology v Trondheim*

Alf and Vegard's RISC processor
Advanced Virtual RISC processor

original AVR MCU – local ASIC house in Trondheim, Norway (now *Nordic Semiconductor*)

1996 technology was sold to Atmel from Nordic VLSI further developed by Bogen and Wollan at *Atmel Norway*

jeden z prvých bol AT90S8515, 40-pin DIP pinout as an 8051 (polarity of the RESET line was opposite).

1997 začala výroba

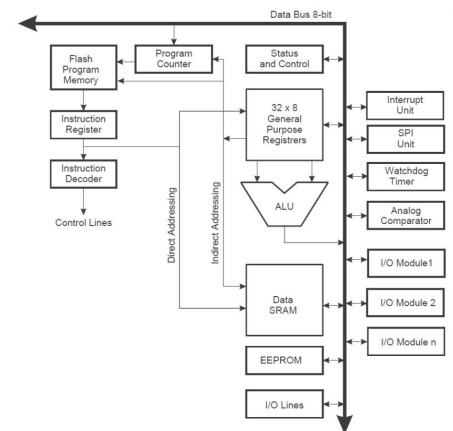
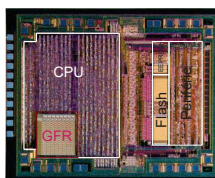
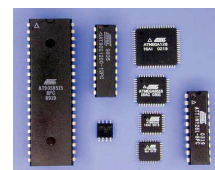
len do r. 2003, Atmel had shipped *500 million AVR*

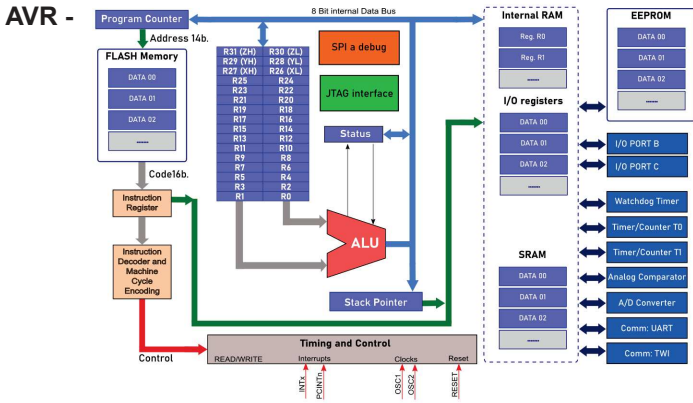
2019 Atmel -> *Microchip Technology*

Processor optimalizovaný pre jazyk C

- + RISC: väčšina inštrukcií trvá 1SC
- + FLASH: rapidné zrychlenie a zlacnenie vývoja

AVR - Architektúra

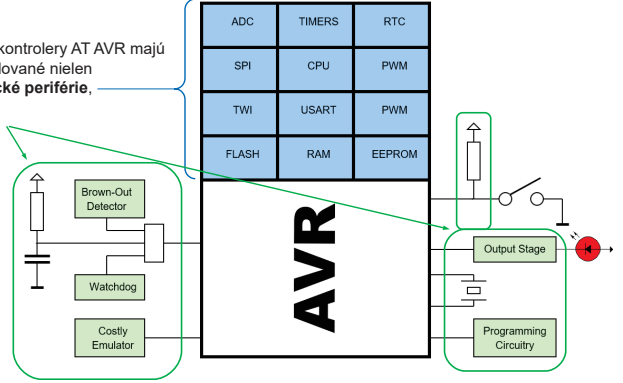




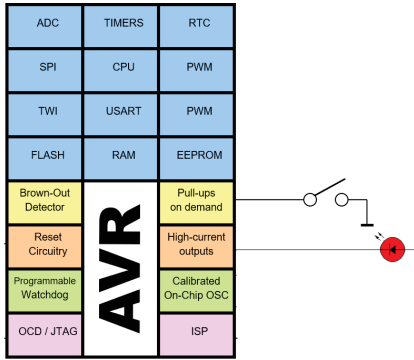
Pomocou AT AVR mikroprocesorov možno realizovať jednočipové riešenie

Mikrokontrolery AT AVR majú zabudované nielen klasické periférie,

ale aj



Pomocou AT AVR mikroprocesorov možno realizovať jednočipové riešenie



AVR families

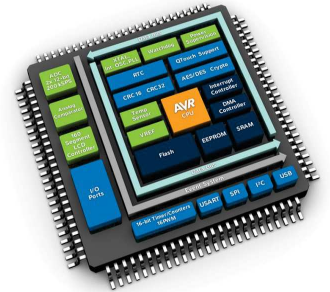
tinyAVR

megaAVR

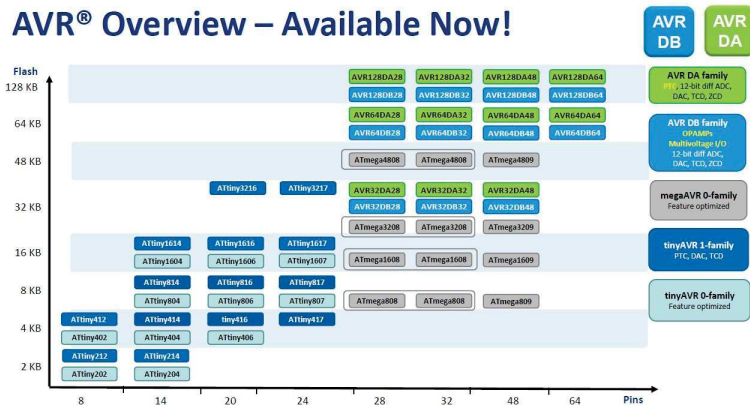
XMEGA (real time, USB, comm)

AVR UC3 L3 (battery, low power)

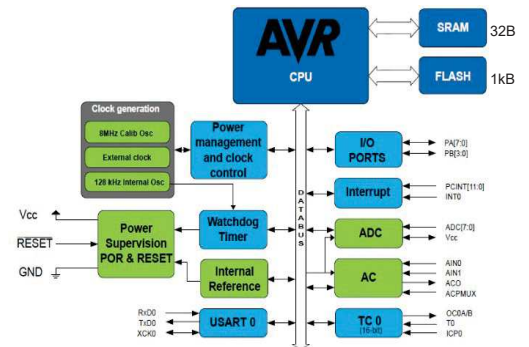
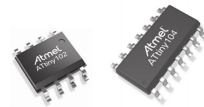
avr32



AVR® Overview – Available Now!



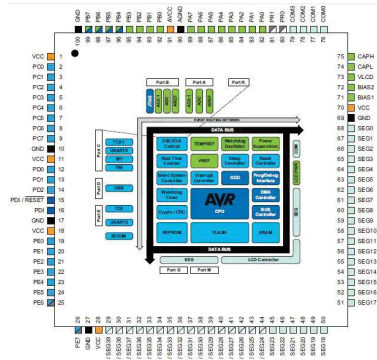
ATtiny



Microchip Technology ATtiny102/104 AVR 8-Bit Microcontrollers

XMEGA

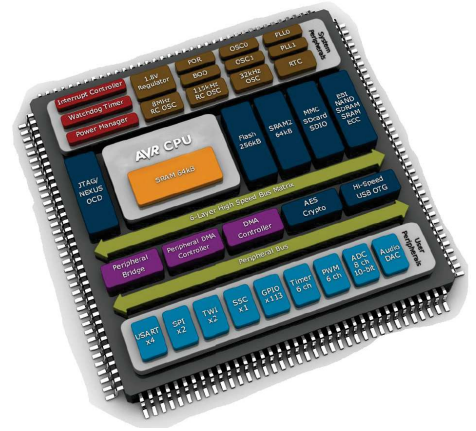
128 kB FLASH
8 kB boot
8 kB SRAM
2 kB EEPROM



Microchip Technology AVR® XMEGA® 8/16-bit Microcontroller 128B1

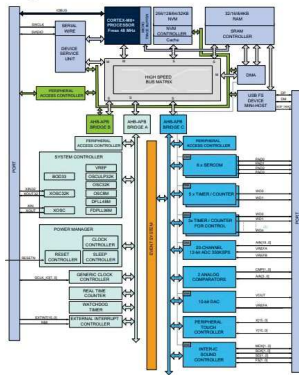
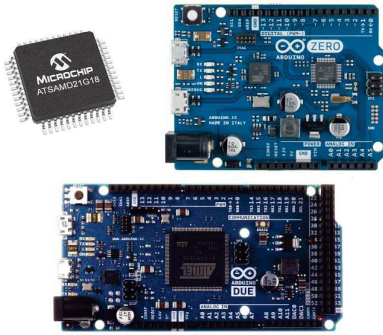
AVR32

256 kB FLASH
64 kB SRAM
SDcard



Microchip AVR® 32-bit AVR UC3

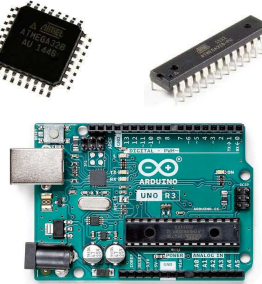
SAM -> nie je AVR!



Microchip Technology ATSAM21

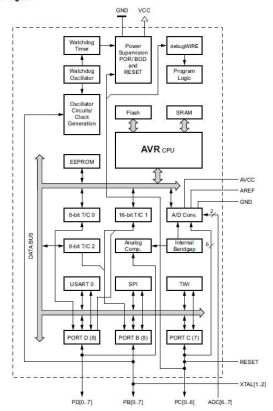
ATmega328

32B
1KB

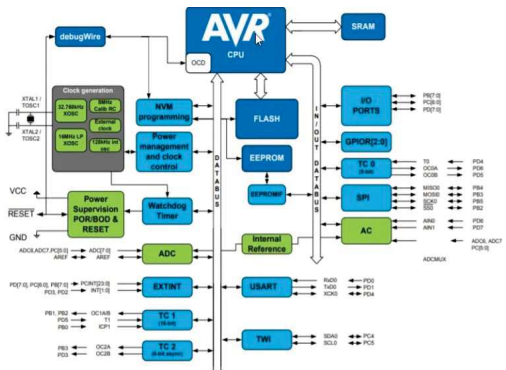


Microchip Technology ATmega328 AVR 8-Bit Microcontrollers

Figure 2-1. Block Diagram

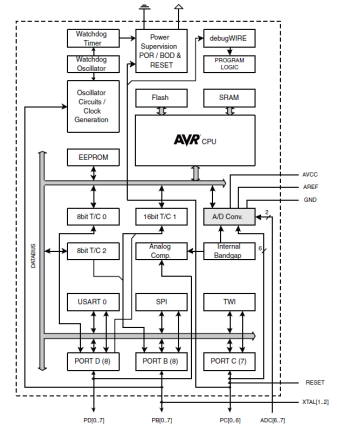
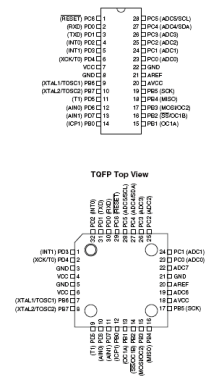


ATmega328

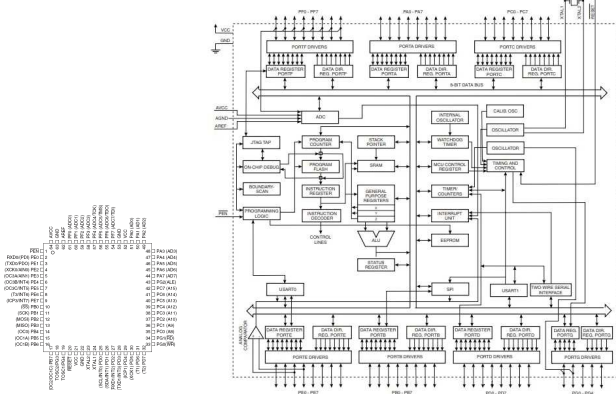


Microchip Technology ATmega328 AVR 8-Bit Microcontrollers

ATMEGA8

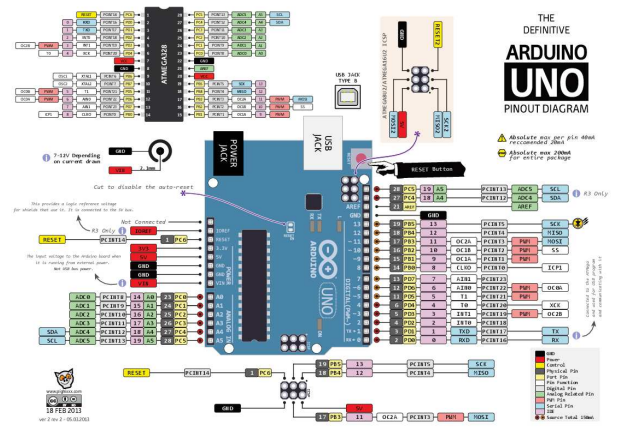
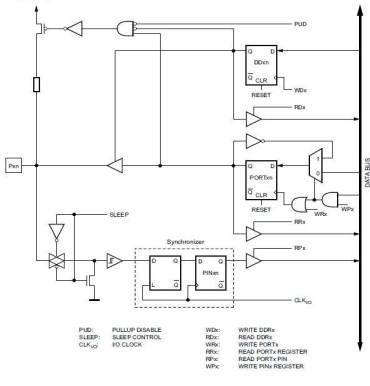


ATMEGA128



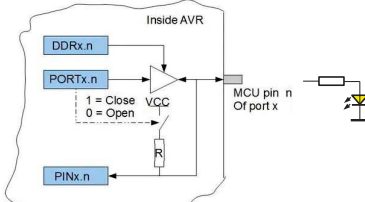
Vstupy a výstupy

Figure 13-2. General Digital I/O⁽¹⁾



Vstupy a výstupy (I/O)

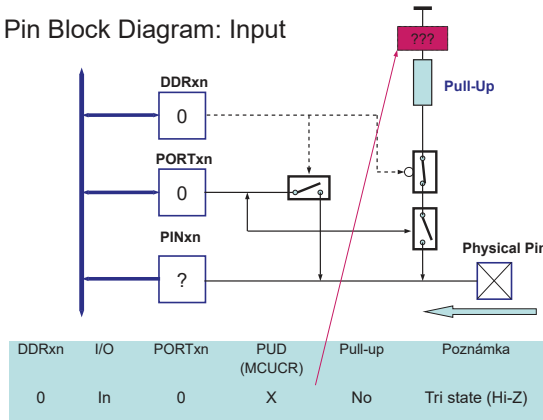
- PORT A
- PORT B
- PORT C
- PORT D
- PORT E
- PORT F
- PORT B.0
- PORT B.1
- PORT B.2
- PORT B.3
- PORT B.4
- PORT B.5
- PORT B.6
- PORT B.7



13.4.2 PORTB – The Port B Data Register

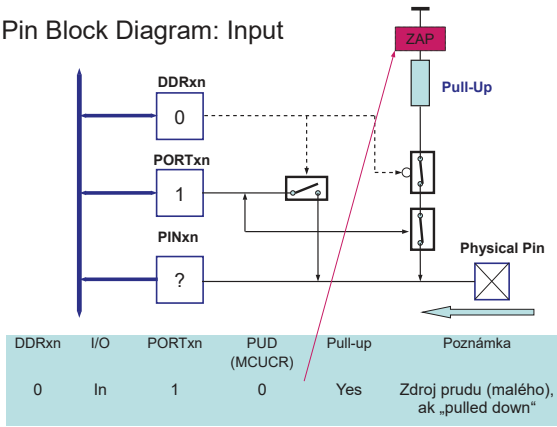
Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

I/O Pin Block Diagram: Input

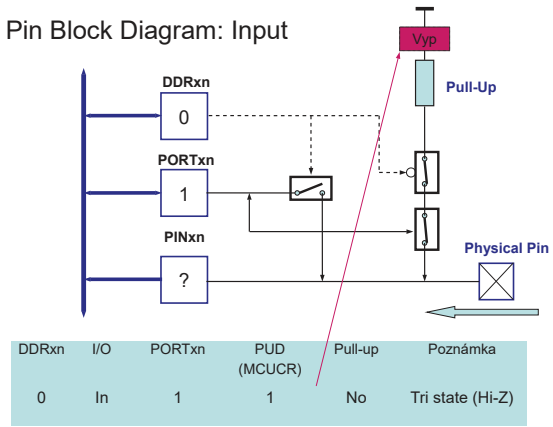


DDRxn	I/O	PORTxn	PUD (MCUCR)	Pull-up	Poznámka
0	In	0	X	No	Tri state (Hi-Z)

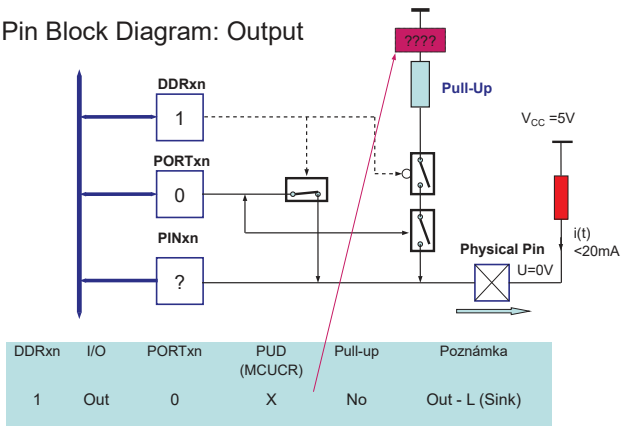
I/O Pin Block Diagram: Input



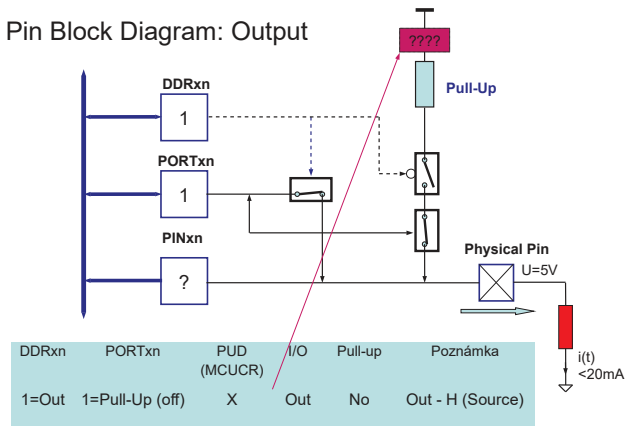
I/O Pin Block Diagram: Input



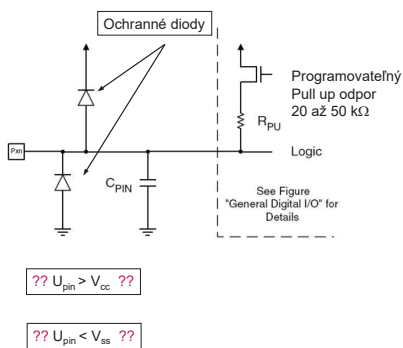
I/O Pin Block Diagram: Output



I/O Pin Block Diagram: Output

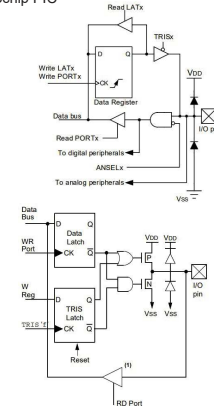


I/O Port – zapojenie vstupného pinu



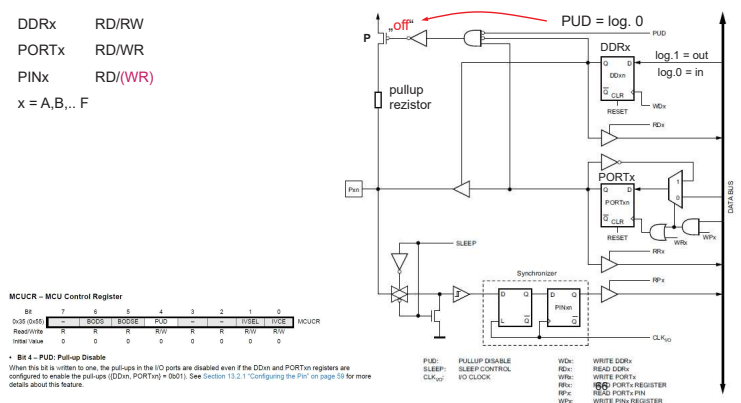
„ATMEL AVR“

„Microchip PIC“



I/O Port – novšia architektúra portu:

DDRxn RD/RW
 PORTxn RD/WR
 PINx RD/(WR)
 x = A,B,... F

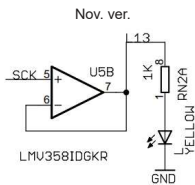
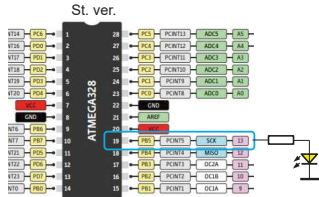


Programovanie I/O bitov v Arduino C

```
// initialize digital pin LED_BUILTIN as an output.
pinMode(LED_BUILTIN, OUTPUT);

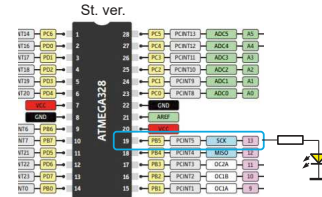
// turn the LED on (HIGH is the voltage level)
digitalWrite(LED_BUILTIN, HIGH);

// turn the LED off by making the voltage LOW
digitalWrite(LED_BUILTIN, LOW);
```



Programovanie I/O bitov v assembleri

```
START:      SBI 0x04,5      ; DDRB.5 = 1 (t.j. Output)
LOOP:      SBI 0x05,5      ; PORTB.5 = 1 (t.j. High, rozsviet LED)
           CBI 0x05,5      ; PORTB.5 = 0 (t.j. Low, zhasni LED)
```



13.4.2 PORTB – The Port B Data Register

Bit	7	6	5	4	3	2	1	0
Output Value	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
Input Value	0	0	0	0	0	0	0	0

13.4.3 DDRB – The Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0
Output Value	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Input Value	0	0	0	0	0	0	0	0

13.4.4 PINB – The Port B Input Pins Address

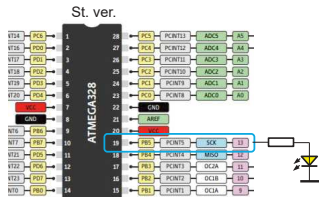
Bit	7	6	5	4	3	2	1	0
Output Value	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
Input Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Programovanie I/O bitov v jazyku C

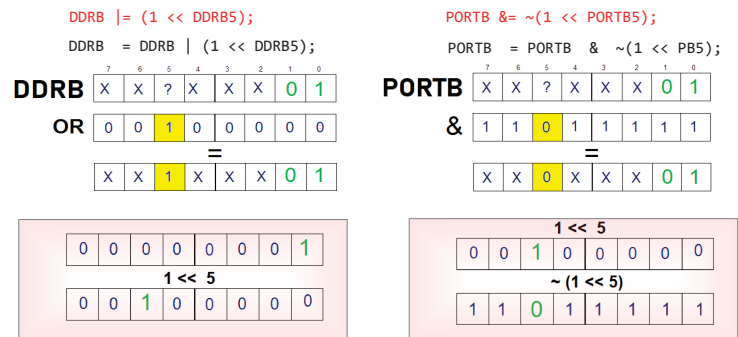
```
// initialize digital pin LED_BUILTIN as an output.
pinMode(LED_BUILTIN, OUTPUT);      DDRB |= (1 << DDRB5);

// turn the LED on (HIGH is the voltage level)
digitalWrite(LED_BUILTIN, HIGH);    PORTB |= (1 << PORTB5);

// turn the LED off by making the voltage LOW
digitalWrite(LED_BUILTIN, LOW);     PORTB &= ~(1 << PORTB5);
```



Programovanie I/O bitov v jazyku C



https://senzor.robotika.sk/sensorwiki/index.php/AVR_Bit_Magic