

Procesory i logika programowalna

Wykład 4

Technologie na urządzenia mobilne

Mgr inż. Łukasz Kirchner

Lukasz.kirchner@cs.put.poznan.pl

<http://www.cs.put.poznan.pl/lkirchner>

Bibliografia

- ▶ Sztuka Elektroniki – P. Horowitz, W.Hill
- ▶ Układy półprzewodnikowe – U.Tietze, Ch. Schenk
- ▶ Projektowanie układów analogowych – Robert A.Rease
- ▶ Układy Cyfrowe – Wojciech Głocki – WSiP
- ▶ Filtry analogowe i cyfrowe – Jacek Izydorczyk, Jacek Konopacki – PAN
- ▶ Lokalne interfejsy szeregowo w systemach cyfrowych – Jacek Bogusz – BTC
- ▶ Zasilacze urządzeń elektronicznych – Joseph J.Carr – BTC
- ▶ Mikrokontrolery w systemach zdalnego sterowania – Zbigniew Hajduk – BTC
- ▶ Pamięci masowe w systemach mikroprocesorowych – Paweł Marks – BTC
- ▶ Cyfrowe przetwarzanie sygnałów – Tomasz P. Zieliński – WKŁ
- ▶ Moduły GSM w systemach mikroprocesorowych – J Bogusz – BTE
- ▶ Systemy GPS – Cezary Specht – BERNARDIUM
- ▶ GPS i inne satelitarne systemy nawigacyjne – Janusz Narkiewicz – WKŁ
- ▶ Technologie i materiałoznawstwo dla elektroników – Zbigniew Szczepiński , Stefan Okoniewski WSiP

Procesory

- ▶ Procesor, mikroprocesor, mikrokontroler
- ▶ Procesor – urządzenie cyfrowe sekwencyjne potrafiące pobierać dane z pamięci, interpretować je i wykonywać jako rozkazy. Wykonuje on bardzo szybko ciąg prostych operacji (rozkażów) wybranych ze zbioru operacji podstawowych określonych zazwyczaj przez producenta procesora jako lista rozkażów procesora.

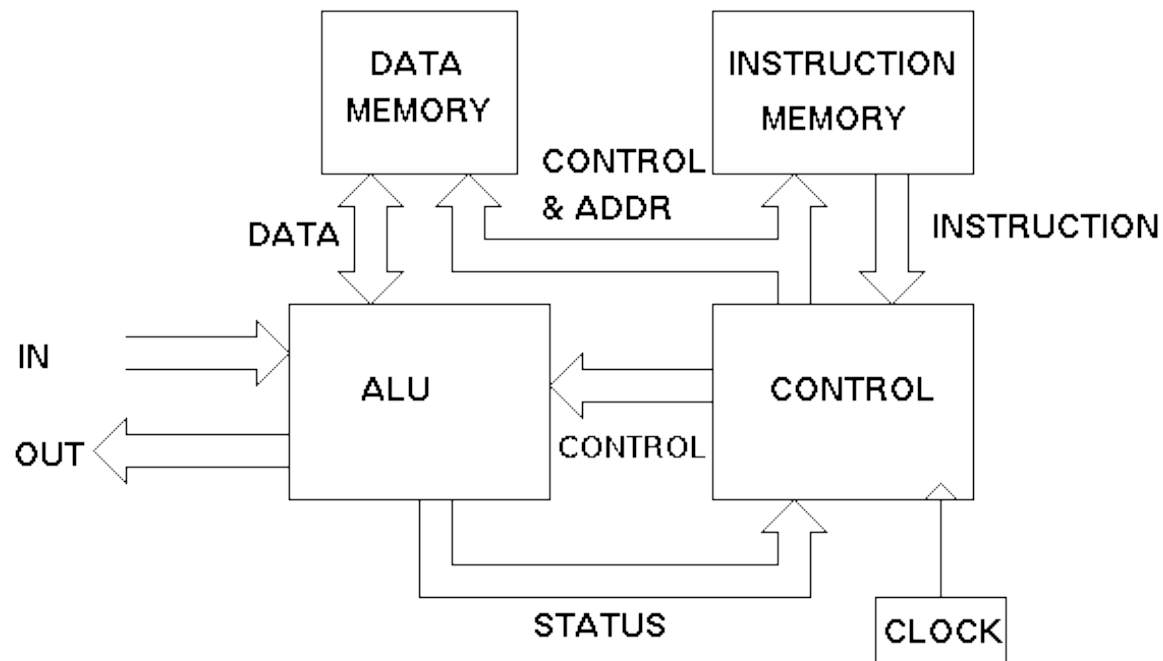
Procesory

- ▶ Integracja dużej ilości elementów cyfrowych (ALU, pamięć CACHE, DMA, ...) na małej powierzchni
- ▶ Aktualnie mikrokontrolery mogą mieć dołączone do swojej struktury układy odpowiedzialne za: Reset, Zasilanie, Komunikacje, Programowanie, ...)

Procesory – Architektury

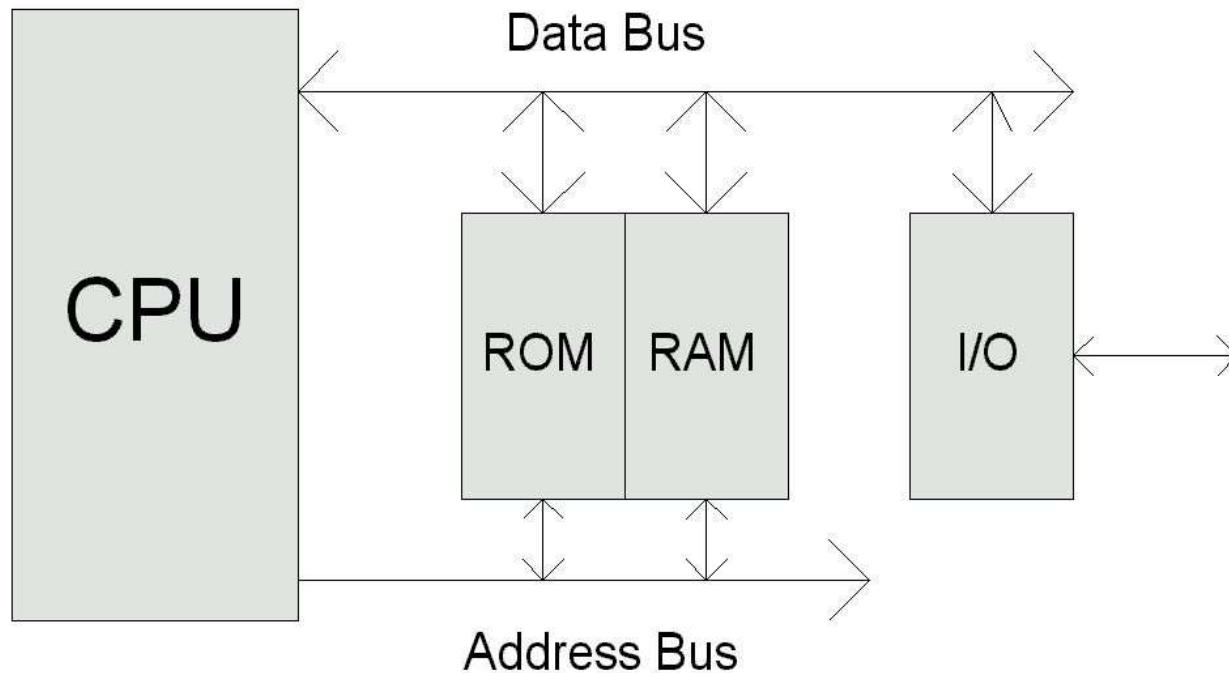
HARVARD ARCHITECTURE

MICROPROCESSOR



Procesory – Architektura

John von Neumann Architecture



Procesory – Architektura

Zmodyfikowana architektura Harvard:

- ▶ Możliwość odczytu pamięci rozkazów tak jakby były danymi
- ▶ Przykłady: AVR, ZILOG, TMS320

RISC – CISC

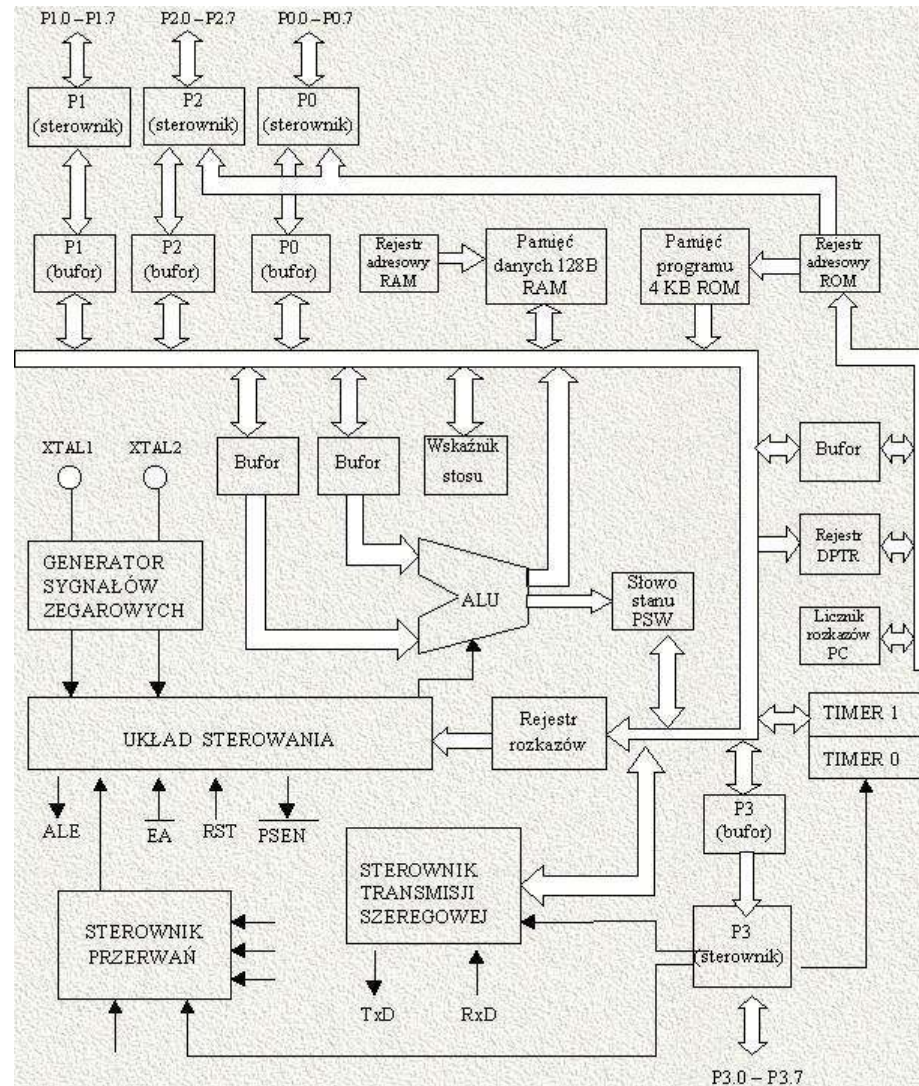
CISC – Complex instruction set computer

- ▶ Duża liczba rozkazów
- ▶ Mała optymalizacja
- ▶ Duża liczba trybów adresowania
- ▶ Duża liczba rozkazów odwołujących się do pamięci
- ▶ Powolne działanie dekodera rozkazów

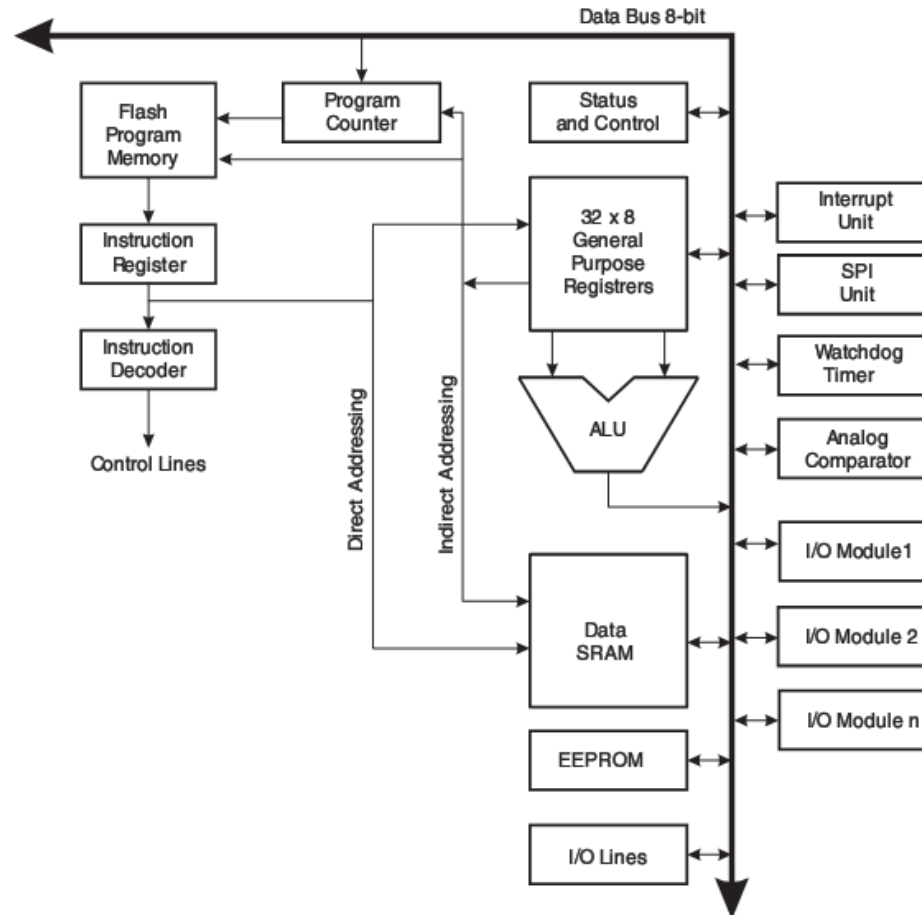
RISC – Reduced instruction set computer

- ▶ Większa od CISC częstotliwość taktowania

Architektura 8051



Architektura AVR



MIKROKONTROLERY –



Seria ADuC MicroConverter (ADC, DAC, VREF, Temp Sense, Flash)

- ▶ ADuC800 (8052 –12clk, 1clk)
- ▶ ADuC7000 (ARM7TDMI)

- ▶ ADSP–21xx (160MHz, speech, voice–modems)
- ▶ SHARC (Floating–point, 319MFLOPS/1\$)
- ▶ TigerSHARC
- ▶ Blackfin (16/32bit,

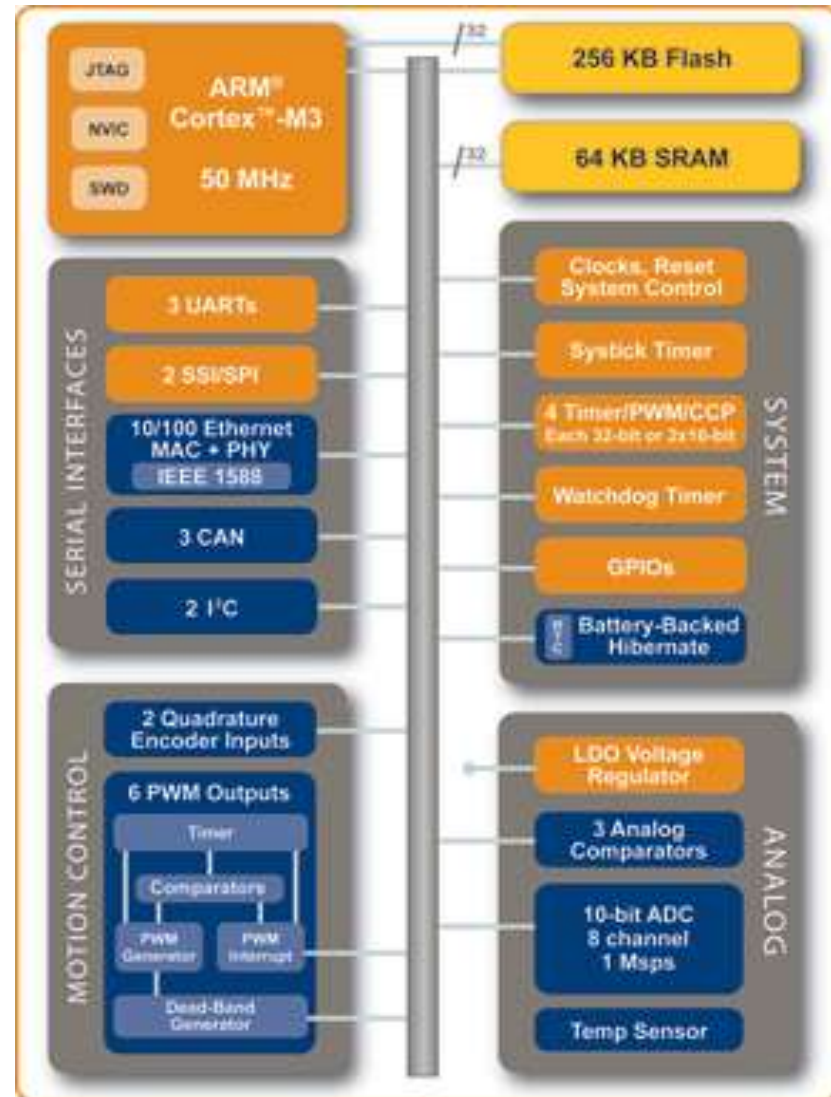
Mikrokontrolery –



- ▶ 8051 – 8bit, CISC
- ▶ AVR – 8bit, RISC
- ▶ AVR32 – 32bit
- ▶ MARC4 – 4bit
- ▶ FPSLIC – AVR + FPGA
- ▶ AT91SAM – ARM7, ARM9
- ▶ Advanced DSP – 1GFLOPS + ARM7

LuminaryMicro Inc.

▶ ARM Cortex-M3



MIKROKONTROLERY –




MICROCHIP

- ▶ PIC10, PIC12, PIC16, PIC18, PIC24F, PIC24H, dsPIC30, dsPIC33
- ▶ 8,16,32 bit, 384B – 256K, 40 MIPS
- ▶ Obudowy 6 – 100 wyprowadzeń
- ▶ rfPIC
- ▶ CAN, USB, Ethernet, nanoWatt, Motor Control, LCD
- ▶ ROM/ROMless, OTP, Flash

MIKROKONTROLERY – founded by Philips

- ▶ NXP Semiconductor
- ▶ 16/32 bit ARM7 (LPC2103, LH79520)
- ▶ 16/32 bit ARM9 (LH7A4000, LH7A404)
- ▶ 8-bit 80C51 (P89C51, P80C51, P87C51)
- ▶ 16-bit XA (PXAC..., PXAG..., PXAH..., PXAS...)

MIKROKONTROLERY – Everywhere you imagine

- ▶ Produkcja dla telefonii komórkowej, motoryzacja, największa na świecie fabryka mikrokontrolerów
 - ▶ LCD drivers, RF IC, Mixed-signal, System-on-a-chip
 - ▶ SuperH RISC
- 

MIKROKONTROLERY –



- ▶ SGS–Thomson (www.st.com)
- ▶ 8bit ST6, ST7 (DISEqC), uPSD
- ▶ 16bit ST10
- ▶ 32bit ARM (STM32 – CortexM3, STR7 – ARM7TDMI, STR9 – ARM966E–S)

MIKROKONTROLERY — TEXAS INSTRUMENTS

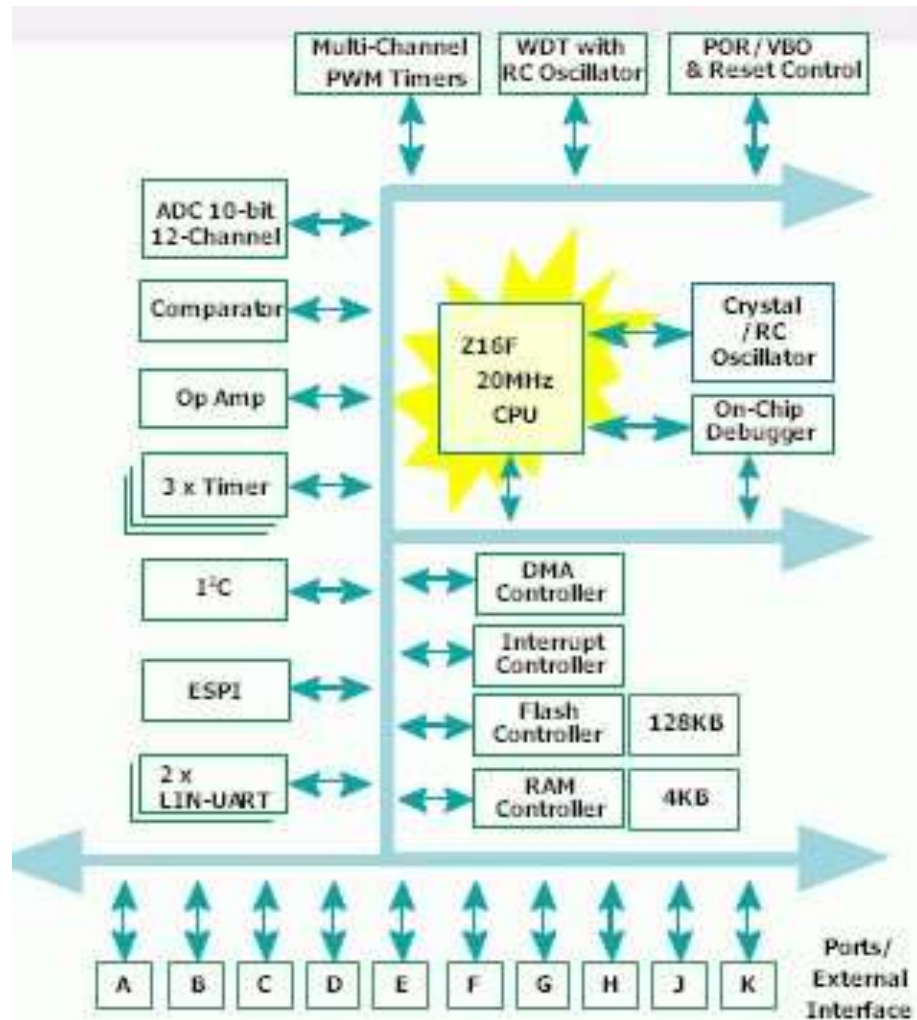
- ▶ TI – Texas Instrument
- ▶ MSP430 – Ultra–Low Power – 16–bit RISC, mixed–signal,
- ▶ C2000™ – 32 bit – high performance, 16/32 bit, 150 DSP–quality MIPS, 300 MFLOPS (MSP320F – controller)
- ▶ C5000, C6000 (TMS320C – DSP)
- ▶ DaVinci (Digital Media Processor – TMS320DM)
- ▶ TMS470 ARM7TDMI – Industrial
- ▶ TMS470 ARM7TDMI – Automotive

MIKROKONTROLERY –




- ▶ Z80, Z8000, Z8, Z800, Z80000, Z280, Z180, Z380, eZ80, eZ8, Z16
- ▶ 8,16,24,32 bit
- ▶ Nintendo, ZX80, ZX Spectrum
- ▶ IrDA, Single Chip modem, PCMCIA adaptor, wireless, DSP, TV,
- ▶ Zilog ZNEO (Single-cycle CISC)

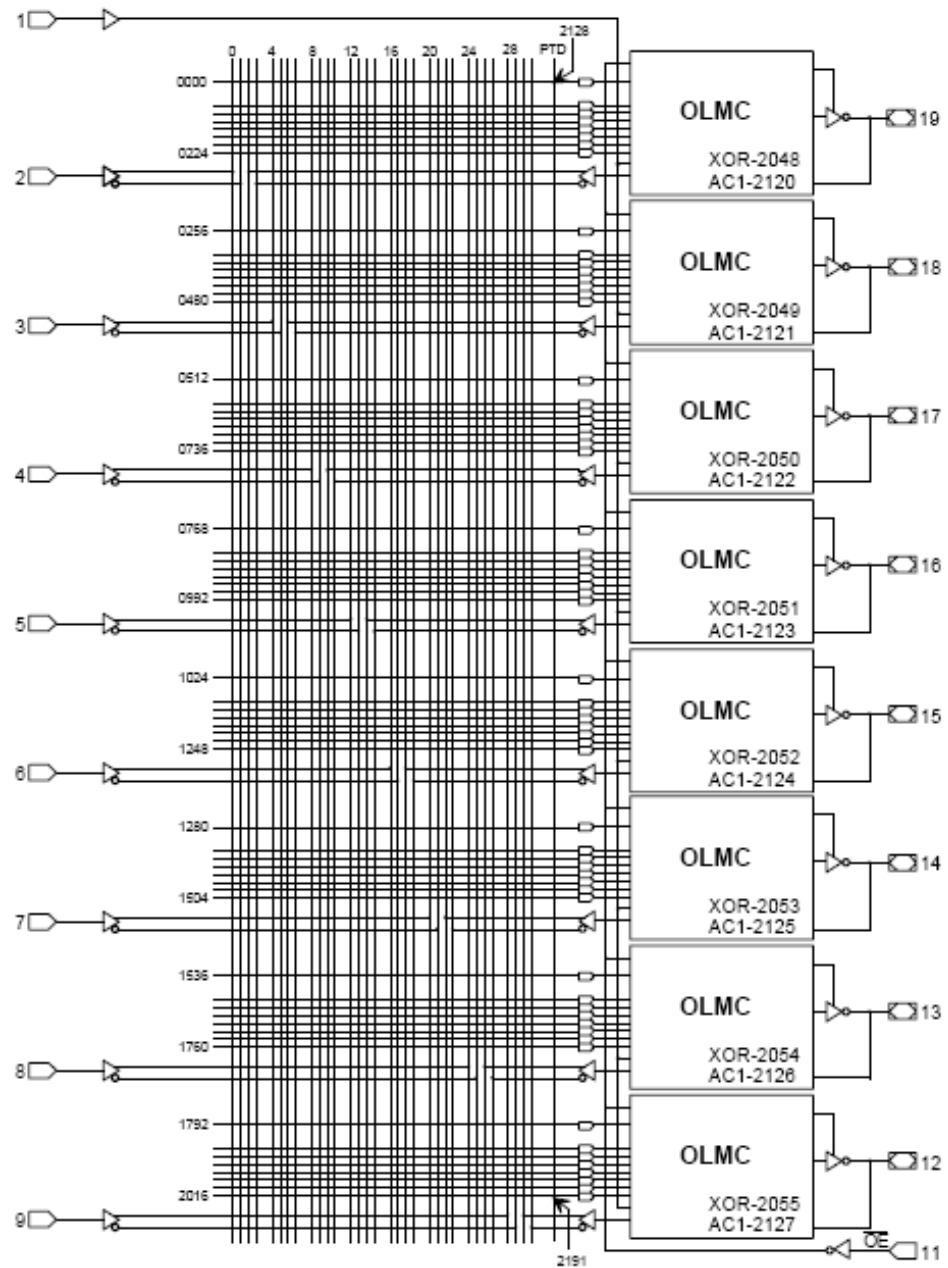
ZILOG ZNEO



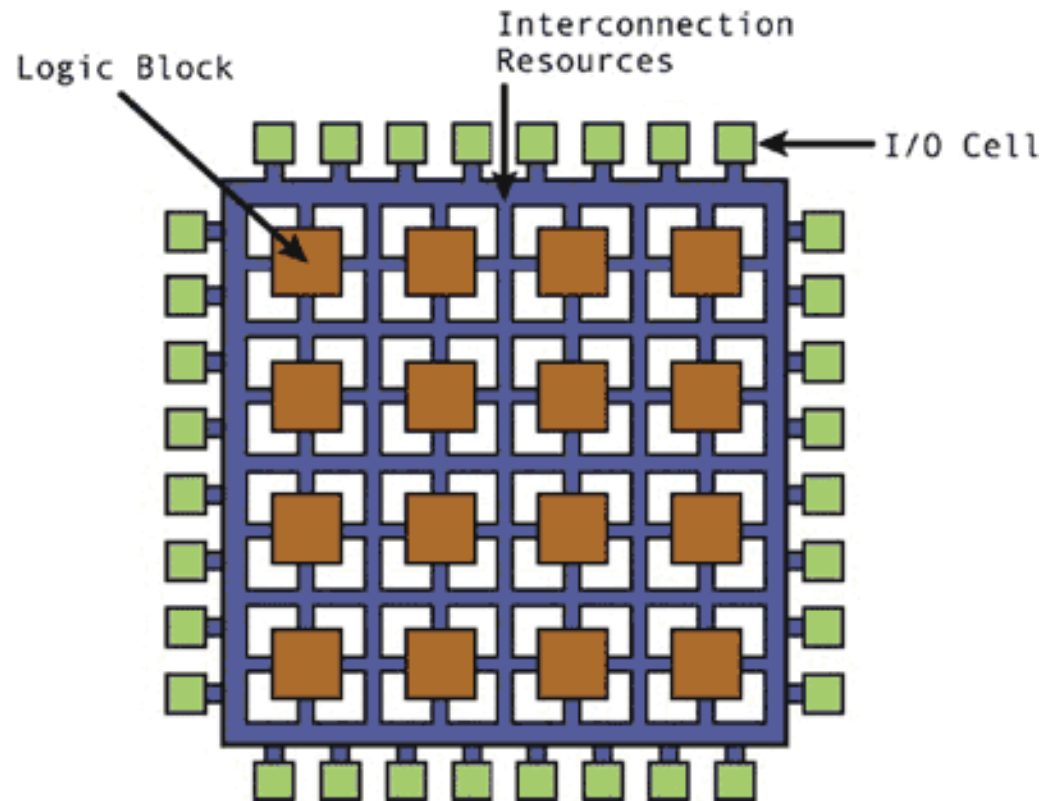
Logika Programowalna

- ▶ PLD – Programmable Logic Device
 - ▶ PLA – Programmable Logic Arrays (1975)
 - ▶ PAL – Programmable Array Logic (1978)
 - ▶ SPLD (Simple Programmable Logic Device)
 - ▶ CPLD (Complex Programmable Logic Device)
 - ▶ FPGA – Field Programmable Gate Arrays (1985)
- 


GAL16V8



FPGA



Technologia ASIC

- ▶ **Application-specific integrated circuit**
 - ▶ Układ z góry przeznaczony do wykonywania określonego zadania
 - ▶ Zastępują szereg układów ogólnego przeznaczenia realizując tą samą funkcję
 - ▶ Mniejszy układ, tańszy koszt produkcji, większa szybkość, mniejsza energia, bardziej niezawodne,
 - ▶ SoC – System-On-Chip
- 

Technologia ASIC

Wady:

- ▶ Wysoki koszt jednostkowy przy małej produkcji
- ▶ Mniejsza uniwersalność niż mikroprocesory
- ▶ Długi czas projektowania 6–24 miesięcy

- ▶ Zastosowanie: komputery, telefony, szyfrowanie, deszyfrowanie, szybkie przetwarzanie audio, DSP

PLD – Altera

CPLD:

- ▶ EEPROM (MAX7000A/B/S, MAX3000, MAX II)

FPGA

- ▶ SRAM (Stratix III, Cyclone III, Arria GX)

PLD – Xilinx

CPLD:

- ▶ Flash (CoolRunner II, CoolRunner XPLA3, XC9500/XC9500XV/XL)

FPGA:

- ▶ SRAM (Virtex/ Virtex E ... Virtex-5, Spartan 3A)

PLD – Lattice

SPLD:

- ▶ EEPROM (GAL)

CPLD:


- ▶ EEPROM (ispLSI2000, ispMACH, ispXPLD, ispGAL, MachXO)

FPGA:

- ▶ SRAM (LatticeSC, LatticeXP, ispXPGA, MachXO)

PLD – Języki opisu

- ▶ ABEL (Advanced Boolean Expression Language) – Xilinx
 - ▶ AHDL (Altera HDL)
 - ▶ CUPL
 - ▶ VHDL (VHSIC HDL) (VHSIC: Very-High-Speed Integrated Circuits)
 - ▶ Verilog

 - ▶ Biblioteki IP Core
- 

VHDL – Przykład

```
-- (this is a VHDL comment)

-- import std_logic from the IEEE library
library IEEE;
use IEEE.std_logic_1164.all;

-- this is the entity
entity ANDGATE is
  port (
    IN1 : in std_logic;
    IN2 : in std_logic;
    OUT1: out std_logic);
end ANDGATE;
architecture RTL of ANDGATE is


begin

  OUT1 <= IN1 and IN2;

end RTL;
```

Verilog – Przykład #1

```
module main;  
  initial  
    begin  
      $display("Hello world!");  
      $finish;  
    end  
endmodule
```



Verilog – Przykład #2

```
module toplevel(clock,reset);
input clock;
input reset;


reg flop1;
reg flop2;

always @ (posedge reset or posedge clock)
if (reset)
begin
flop1 <= 0;
flop2 <= 1;
end
else
begin
flop1 <= flop2;
flop2 <= flop1;
end
endmodule
```


ABEL – przykład

- ▶ module my_first_circuit;
- ▶ title 'ee200 assignment 1'
- ▶ EE200XY device 'XC4003E';
- ▶
- ▶
- ▶ " input pins
- ▶ A, B pin 3, 5;
- ▶ " output pins
- ▶ SUM, Carry_out pin 15, 18 istype 'com';
- ▶ equations
- ▶ $SUM = (A \& !B) \# (!A \& B) ;$
- ▶ Carry_out = A & B;
- ▶ end my_first_circuit;

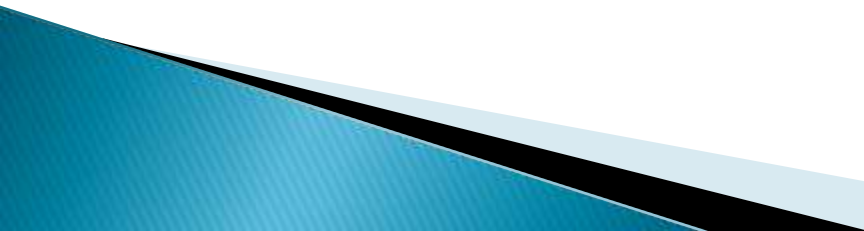
PLD – Oprogramowanie

- ▶ ISE WebPack 9.2i (Xilinx) (FREE)
 - ▶ ISE Foundation (Xilinx)
 - ▶ Quartus II (Altera)
 - ▶ MAX+PLUS II (Altera)
 - ▶ ispLEVER 7.0 (Lattice)
- 

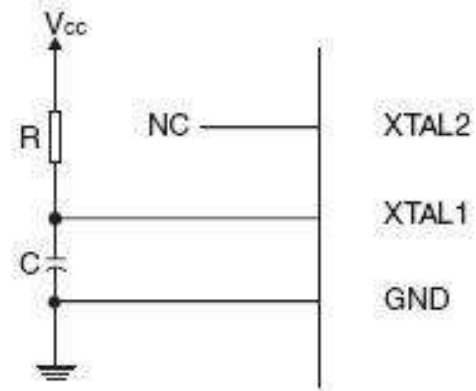
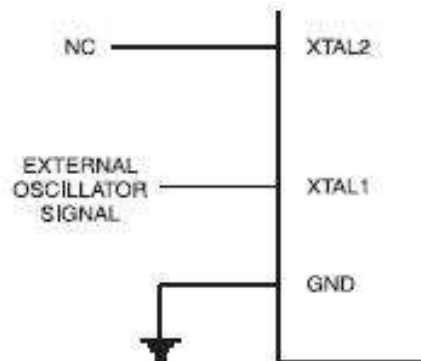
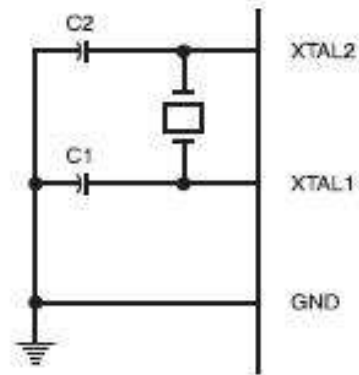
Taktowanie Procesora

- ▶ Procesory – urządzenie synchroniczne – wymaga zegara do określania prędkości pracy

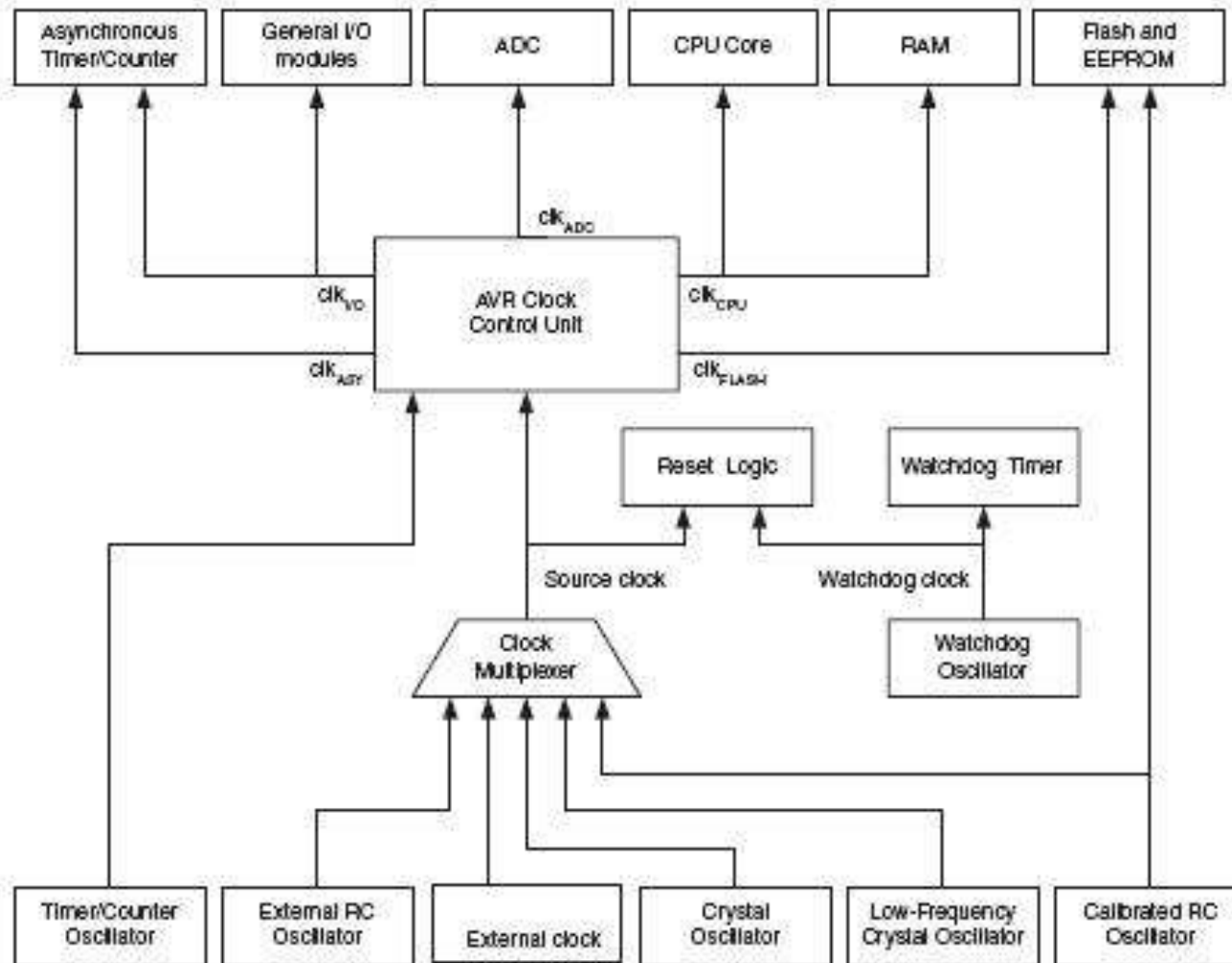
Procesory taktowane mogą być z:

- ▶ Wewnętrzny układ oscylacji RC
 - ▶ Zewnętrzny układ oscylatora RC
 - ▶ Zewnętrzny układ oscylatora kwarcowego
 - ▶ Zewnętrzne źródło zegarowe
 - ▶ PLL
- 

Taktowanie

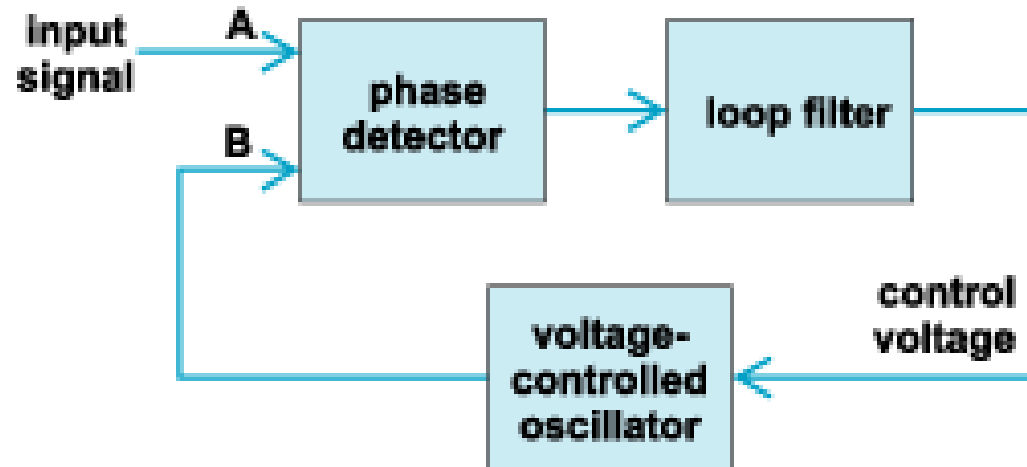


Taktowanie – AVR

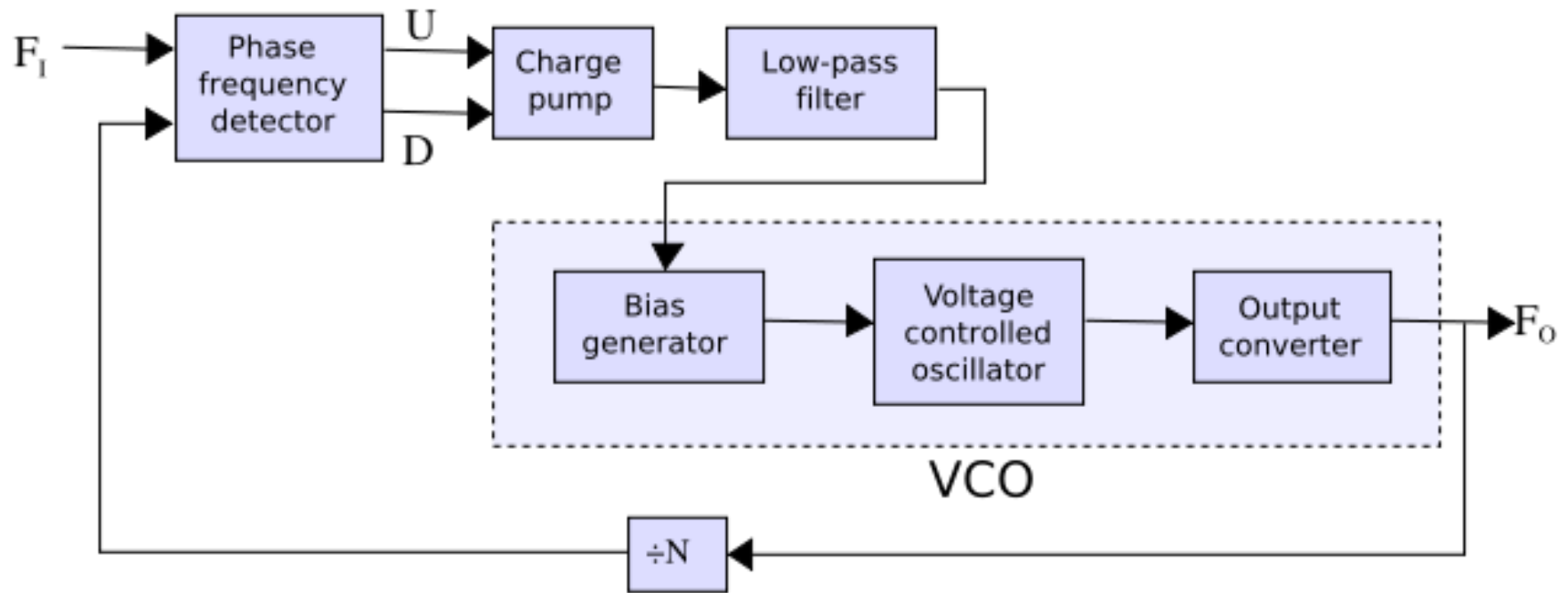


Taktowanie procesora – PLL

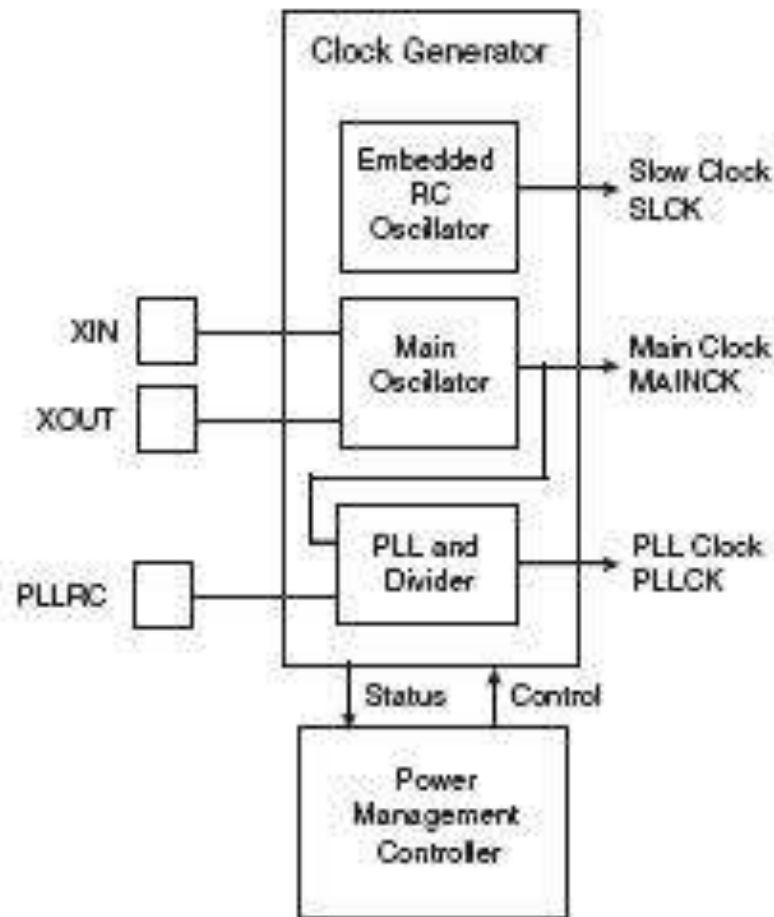
- ▶ Wejściowy sygnał zegarowy może być pomniejszany (preskaler, DIVider)
- ▶ Częstotliwość może być podnoszona za pomocą układu PLL
- ▶ PLL (Phase-Locked Loop)



PLL



Taktowanie – ARM



Taktowanie – Cortex

