

# Systemy wbudowane - wykład 8

Przemek Błaśkiewicz

25 kwietnia 2019

Komponenty dyskretne: ?

Komponenty dyskretne:

- jednostki obliczeniowe (CPU);

Komponenty dyskretne:

- jednostki obliczeniowe (CPU);
- jednostki pomocnicze (ko-procesory);

Komponenty dyskretne:

- jednostki obliczeniowe (CPU);
- jednostki pomocnicze (ko-procesory);
- akceleratory;

## Komponenty dyskretne:

- jednostki obliczeniowe (CPU);
- jednostki pomocnicze (ko-procesory);
- akceleratory;
- pamięci;

## Komponenty dyskretne:

- jednostki obliczeniowe (CPU);
- jednostki pomocnicze (ko-procesory);
- akceleratory;
- pamięci;
- kontrolery magistrali, mostki (bridge);

## Komponenty dyskretne:

- jednostki obliczeniowe (CPU);
- jednostki pomocnicze (ko-procesory);
- akceleratory;
- pamięci;
- kontrolery magistrali, mostki (bridge);
- magistrale;



## Komponenty dyskretne:

- jednostki obliczeniowe (CPU);
- jednostki pomocnicze (ko-procesory);
- akceleratory;
- pamięci;
- kontrolery magistrali, mostki (bridge);
- magistrale;
- kontrolery we/wy (I/O);

# (Główne) jednostki obliczeniowe

- funkcjonalność procesora
- technologia wykonania układu procesora

- funkcjonalność procesora
  - ogólnego przeznaczenia;
- technologia wykonania układu procesora

- funkcjonalność procesora
  - ogólnego przeznaczenia;
  - programowalno-dedykowany;
- technologia wykonania układu procesora

- funkcjonalność procesora
  - ogólnego przeznaczenia;
  - programowalno-dedykowany;
  - specjalizowany (custom, single-operation);
- technologia wykonania układu procesora

- funkcjonalność procesora
  - ogólnego przeznaczenia;
  - programowalno-dedykowany;
  - specjalizowany (custom, single-operation);
- technologia wykonania układu procesora
  - CMOS

- funkcjonalność procesora
  - ogólnego przeznaczenia;
  - programowalno-dedykowany;
  - specjalizowany (custom, single-operation);
- technologia wykonania układu procesora
  - CMOS
  - FPGA



- funkcjonalność procesora
  - ogólnego przeznaczenia;
  - programowalno-dedykowany;
  - specjalizowany (custom, single-operation);
- technologia wykonania układu procesora
  - CMOS
  - FPGA
  - CPLD

- funkcjonalność procesora
  - ogólnego przeznaczenia;
  - programowalno-dedykowany;
  - specjalizowany (custom, single-operation);
- technologia wykonania układu procesora
  - CMOS
  - FPGA
  - CPLD
  - ...

Charakterystyka:

## Charakterystyka:

- uniwersalny procesor o szerokim zakresie typowych operacji;
- zazwyczaj architektura von Neumana (pamięć programu, pamięć danych);
- w oparciu o rejestry i dużą pamięć (zatem cache, zatem zarządca pamięci);
- potokowość, przepustowość, szeroka magistrala → szybkość działania;

## Charakterystyka:

- uniwersalny procesor o szerokim zakresie typowych operacji;
- zazwyczaj architektura von Neumana (pamięć programu, pamięć danych);
- w oparciu o rejestry i dużą pamięć (zatem cache, zatem zarządca pamięci);
- potokowość, przepustowość, szeroka magistrala → szybkość działania;

## Zalety:

## Charakterystyka:

- uniwersalny procesor o szerokim zakresie typowych operacji;
- zazwyczaj architektura von Neumana (pamięć programu, pamięć danych);
- w oparciu o rejestry i dużą pamięć (zatem cache, zatem zarządca pamięci);
- potokowość, przepustowość, szeroka magistrala → szybkość działania;

## Zalety:

- szybka produkcja (łatwiejsze programowanie);
- łatwa dostępność;
- wysoka elastyczność - jedna aplikacja, wiele możliwości.

## Charakterystyka:

- uniwersalny procesor o szerokim zakresie typowych operacji;
- zazwyczaj architektura von Neumana (pamięć programu, pamięć danych);
- w oparciu o rejestry i dużą pamięć (zatem cache, zatem zarządca pamięci);
- potokowość, przepustowość, szeroka magistrala → szybkość działania;

## Zalety:

- szybka produkcja (łatwiejsze programowanie);
- łatwa dostępność;
- wysoka elastyczność - jedna aplikacja, wiele możliwości.

koszt...



- 4-bit dane
- 8-bit instr.
- 12-bit adr.
- zegar 740kHz
- 2300 tranzystorów
- 8 cykli/instr.
- arch. harwardzka

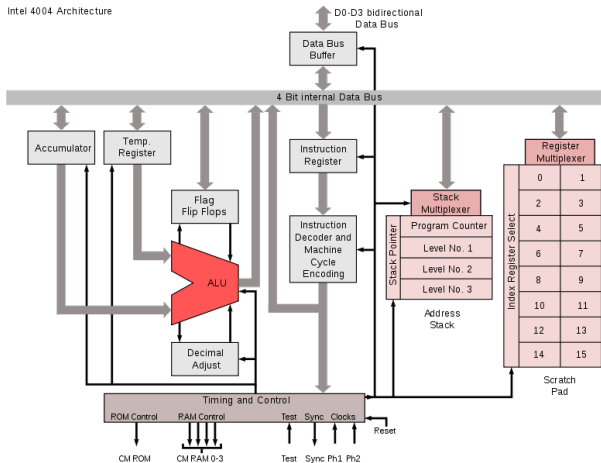


# Intel 4004



- 4-bit dane
- 8-bit instr.
- 12-bit adr.
- zegar 740kHz
- 2300 tranzystorów
- 8 cykli/instr.
- arch. harwardzka

Intel 4004 Architecture



Pewne operacje są typowe dla pewnych dziedzin/zastosowań.

Pewne operacje są typowe dla pewnych dziedzin/zastosowań.

Filtry to często obliczenie:

$$y = \sum_{k=0}^{M-1} A_k X_k$$

Operacja MAC → *multiply and add*.

Pewne operacje są typowe dla pewnych dziedzin/zastosowań.

Filtry to często obliczenie:

$$y = \sum_{k=0}^{M-1} A_k X_k$$

Operacja MAC  $\rightarrow$  *multiply and add*.

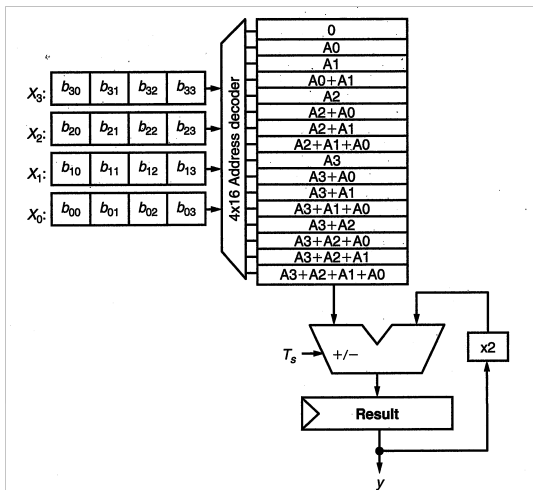
Niech  $X_m = b_{m(N-1)}b_{m(N-2)} \dots b_m 1 b_m 0$ :

$$\begin{aligned} y &= \sum_{k=0}^{M-1} A_k \left[ b_{k(N-1)} 2^{N-1} + \sum_{n=0}^{N-2} b_{kn} 2^n \right] \\ &= \sum_{k=0}^{M-1} A_k b_{k(N-1)} 2^{N-1} + \sum_{n=0}^{N-2} \left[ \sum_{k=0}^{M-1} A_k b_{kn} \right] 2^n \end{aligned}$$

$$y = \sum_{k=0}^{M-1} A_k b_{k(N-1)} 2^{N-1} + \sum_{n=0}^{N-2} \left[ \sum_{k=0}^{M-1} A_k b_{kn} \right] 2^n$$

# Procesor programowalno-dedykowany

$$y = \sum_{k=0}^{M-1} A_k b_{k(N-1)} 2^{N-1} + \sum_{n=0}^{N-2} \left[ \sum_{k=0}^{M-1} A_k b_{kn} \right] 2^n$$



Procesory dedykowane, np. DSP (*digital signal processing*).

- przyspieszone operacje (np. MAC, wektorowe) – dedykowane układy mnożące oraz operacje w asemblerze; specjalne tryby adresowania;

Procesory dedykowane, np. DSP (*digital signal processing*).

- przyspieszone operacje (np. MAC, wektorowe) – dedykowane układy mnożące oraz operacje w asemblerze; specjalne tryby adresowania;
- architektura harwardzka i potokowość;



Procesory dedykowane, np. DSP (*digital signal processing*).

- przyspieszone operacje (np. MAC, wektorowe) – dedykowane układy mnożące oraz operacje w asemblerze; specjalne tryby adresowania;
- architektura harwardzka i potokowość;
- **trudniej programować, wolniejsze w ogólnych zastosowaniach...**

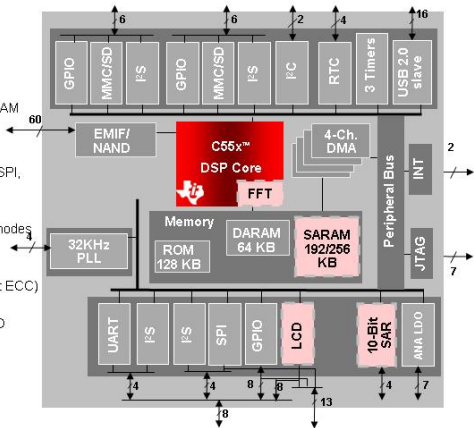
## TMS320C5504/5 Block Diagram and Deltas

### C5504 Features

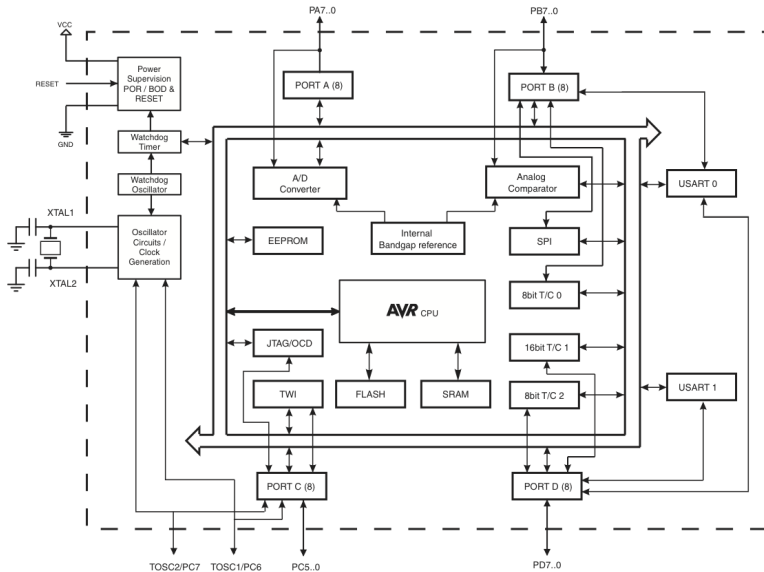
- **Core**
  - Dual MAC, C55x CPU with JTAG disable option
  - 1.05V @ 60MHz, 1.3V @ 100MHz(120MHz available)
  - Dynamic Voltage and Frequency Scaling
- **Memory**
  - 256-KB On-Chip Memory: 64-KB DARAM, 192-KB SARAM
  - 128-KB ROM
- **Peripherals**
  - Four serial busses offering combinations of I<sup>2</sup>S, UART, SPI, MMC/SD, and GPIO
  - High speed USB2.0
  - Multi-master and Slave I<sup>2</sup>C with 7 or 10-bit addressing modes
  - Three 32-bit timers with watchdog functionality
  - Four 4-Channel DMAs
  - 16-bit EMIF with asynchronous SRAM, NAND (with 4-bit ECC)
  - Low power PLL (0.7mA) with 32KHz crystal oscillator
  - Real-time clock with 32-KHz crystal input, 1 analog LDO
- **I/O**
  - 1.8V, 2.5V, 2.8V, 3.3V
  - Separate I/O supplies for EMIF and serial interface
- **Package:** 196-pin 10x10mm BGA with 0.65mm pitch

### C5505 Additional Features:

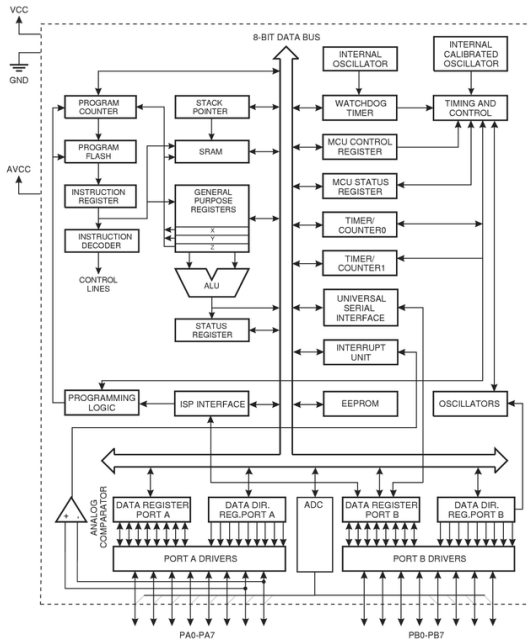
- 320-KB On-Chip Memory: 64-KB DARAM, 256-KB SARAM
- 1024-point FFT Coprocessor
- LCD Bridge muxed with I<sup>2</sup>S, SPI, UART, and GPIO
- 4ch 10-bit SAR ADC



# Mikrokontrolery AVR – ATtiny, ATmega, X-Mega...



# Mikrokontrolery AVR – ATtiny, ATmega, X-Mega...



Mały układ wykonujący jedną, (prostą) czynność.

Mały układ wykonujący jedną, (prostą) czynność.

- timer/watchdog/counter;

Mały układ wykonujący jedną, (prostą) czynność.

- timer/watchdog/counter;
- UART/USART

Mały układ wykonujący jedną, (prostą) czynność.

- timer/watchdog/counter;
- UART/USART
- kontroler LCD, silnika krokowego...

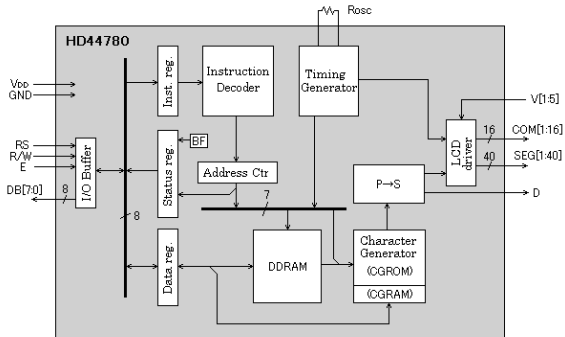


Mały układ wykonujący jedną, (prostą) czynność.

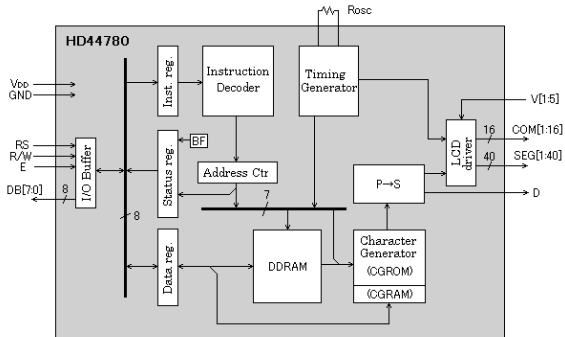
- timer/watchdog/counter;
- UART/USART
- kontroler LCD, silnika krokowego...

Gotowe procesory vs. budowane własnym sumptem.

# HD44780 - kontroler LCD 2x16 znaków



# HD44780 - kontroler LCD 2x16 znaków



MakeSystem.net

# HD44780 - kontroler LCD 2x16 znaków

| Instruction             | Code |     |     |     |     |     |     |     |     |     | Description | Execution Time<br>(max) (when $f_{cp}$ or<br>$f_{osc}$ is 270 kHz)  |            |
|-------------------------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-------------|---|------------|
|                         | RS   | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |             |   |            |
| Clear display           | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   |             | Clears entire display and sets DDRAM address 0 in address counter.  |            |
| Return home             | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | —           | Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged. | 1.52 ms    |
| Entry mode set          | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | I/D | S           | Sets cursor move direction and specifies display shift. These operations are performed during data write and read.                      | 37 $\mu$ s |
| Display on/off control  | 0    | 0   | 0   | 0   | 0   | 0   | 1   | D   | C   | B   |             | Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).                                       | 37 $\mu$ s |
| Cursor or display shift | 0    | 0   | 0   | 0   | 0   | 1   | S/C | R/L | —   | —   |             | Moves cursor and shifts display without changing DDRAM contents.  | 37 $\mu$ s |
| Function set            | 0    | 0   | 0   | 0   | 1   | DL  | N   | F   | —   | —   |             | Sets interface data length (DL), number of display lines (N), and character font (F).   | 37 $\mu$ s |
| Set CGRAM address       | 0    | 0   | 0   | 1   | ACG | ACG | ACG | ACG | ACG | ACG |             | Sets CGRAM address. CGRAM data is sent and received after this setting.   | 37 $\mu$ s |
| Set DDRAM address       | 0    | 0   | 1   | ADD | ADD | ADD | ADD | ADD | ADD | ADD |             | Sets DDRAM address. DDRAM data is sent and received after this setting.   | 37 $\mu$ s |

## COordinate Rotation Dlgital Computer

Zestaw parametryzowanych algorytmów aproksymujących różne funkcje matematyczne (sin, cos, cosh, atan, pierwiastek...)

## Coordinate Rotation Digital Computer

Zestaw parametryzowanych algorytmów aproksymujących różne funkcje matematyczne (sin, cos, cosh, atan, pierwiastek...)

- operacje dodawania i przesunięcia, prosta arytmetyka

## COordinate Rotation Dlgital Computer

Zestaw parametryzowanych algorytmów aproksymujących różne funkcje matematyczne (sin, cos, cosh, atan, pierwiastek...)

- operacje dodawania i przesunięcia, prosta arytmetyka
- efektywny dla wieloelementowych wektorów  $w_x/w_y$

- $$\begin{aligned}x_f &= x_s \cos \theta - y_s \sin \theta \\y_f &= x_s \sin \theta + y_s \cos \theta\end{aligned}$$



- $x_f = x_s \cos \theta - y_s \sin \theta$   
 $y_f = x_s \sin \theta + y_s \cos \theta$
- $\theta = \sum_{i=0}^{\text{inf}} \alpha_i$

- $x_f = x_s \cos \theta - y_s \sin \theta$   
 $y_f = x_s \sin \theta + y_s \cos \theta$
- $\theta = \sum_{i=0}^{\text{inf}} \alpha_i$
- $x_{i+1} = x_i \cos \alpha_i - y_i \sin \alpha_i$   
 $y_{i+1} = y_i \sin \alpha_i + x_i \cos \alpha_i$

- $x_f = x_s \cos \theta - y_s \sin \theta$   
 $y_f = x_s \sin \theta + y_s \cos \theta$
- $\theta = \sum_{i=0}^{\infty} \alpha_i$
- $x_{i+1} = x_i \cos \alpha_i - y_i \sin \alpha_i$   
 $y_{i+1} = y_i \sin \alpha_i + x_i \cos \alpha_i$
- $x_{i+1} = \cos(\alpha_i)(x_i - y_i \tan(\alpha_i))$   
 $y_{i+1} = \cos(\alpha_i)(y_i + x_i \tan(\alpha_i))$

- $x_f = x_s \cos \theta - y_s \sin \theta$   
 $y_f = x_s \sin \theta + y_s \cos \theta$
- $\theta = \sum_{i=0}^{\infty} \alpha_i$
- $x_{i+1} = x_i \cos \alpha_i - y_i \sin \alpha_i$   
 $y_{i+1} = y_i \sin \alpha_i + x_i \cos \alpha_i$
- $x_{i+1} = \cos(\alpha_i)(x_i - y_i \tan(\alpha_i))$   
 $y_{i+1} = \cos(\alpha_i)(y_i + x_i \tan(\alpha_i))$
- $\alpha_i = \tan^{-1}(\sigma_i 2^{-i})$

- $x_f = x_s \cos \theta - y_s \sin \theta$   
 $y_f = x_s \sin \theta + y_s \cos \theta$
- $\theta = \sum_{i=0}^{\infty} \alpha_i$
- $x_{i+1} = x_i \cos \alpha_i - y_i \sin \alpha_i$   
 $y_{i+1} = y_i \sin \alpha_i + x_i \cos \alpha_i$
- $x_{i+1} = \cos(\alpha_i)(x_i - y_i \tan(\alpha_i))$   
 $y_{i+1} = \cos(\alpha_i)(y_i + x_i \tan(\alpha_i))$
- $\alpha_i = \tan^{-1}(\sigma_i 2^{-i})$
- $x_{i+1} = \cos(\alpha_i)(x_i - \sigma_i y_i 2^{-i})$   
 $y_{i+1} = \cos(\alpha_i)(y_i + \sigma_i x_i 2^{-i})$

# CORDIC – obrót wektora $(x_s, y_s)$ o $\theta$

- $x_f = x_s \cos \theta - y_s \sin \theta$   
 $y_f = x_s \sin \theta + y_s \cos \theta$

- $\theta = \sum_{i=0}^{\text{inf}} \alpha_i$

- $x_{i+1} = x_i \cos \alpha_i - y_i \sin \alpha_i$

- $y_{i+1} = y_i \sin \alpha_i + x_i \cos \alpha_i$

- $x_{i+1} = \cos(\alpha_i)(x_i - y_i \tan(\alpha_i))$

- $y_{i+1} = \cos(\alpha_i)(y_i + x_i \tan(\alpha_i))$

- $\alpha_i = \tan^{-1}(\sigma_i 2^{-i})$

- $x_{i+1} = \cos(\alpha_i)(x_i - \sigma_i y_i 2^{-i})$

- $y_{i+1} = \cos(\alpha_i)(y_i + \sigma_i x_i 2^{-i})$

$$i = 0$$

$$x_0 = x_s$$

$$y_0 = y_s$$

$$z_0 = \theta$$

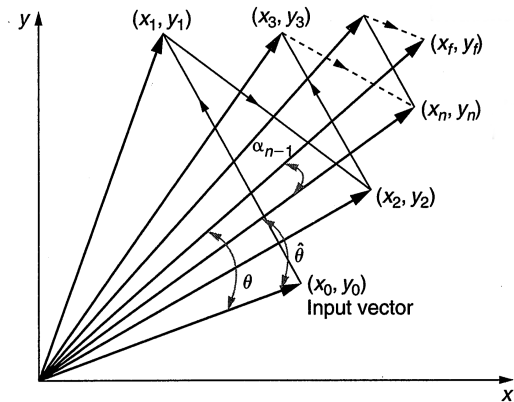
$$x_{i+1} = x_i - \sigma_i y_i 2^{-i}$$

$$y_{i+1} = y_i + \sigma_i x_i 2^{-i}$$

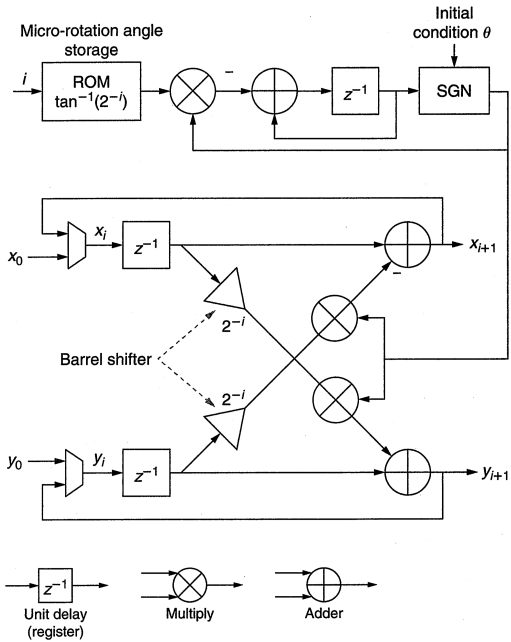
$$z_{i+1} = z_i - \sigma_i \tan^{-1}(2^{-i})$$

$$\sigma_i = \begin{cases} 1 & \text{if } z_i \geq 0 \\ -1 & \text{if } z_i < 0 \end{cases}$$

# CORDIC



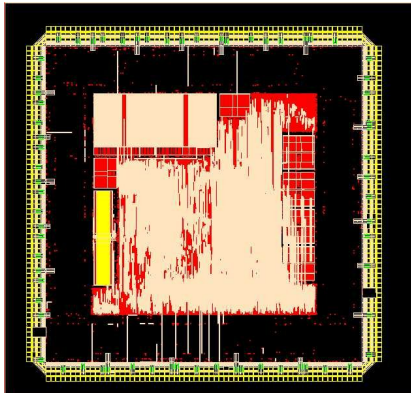
# CORDIC



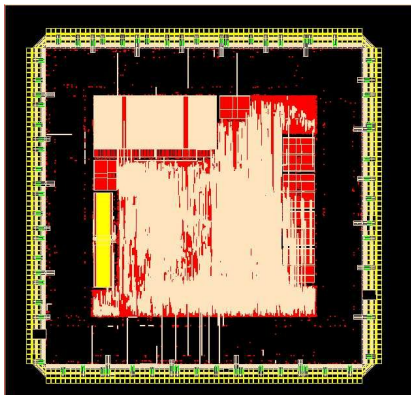


Procesory “programowalne”

– modyfikowalny hardware: FPGA/CPLD

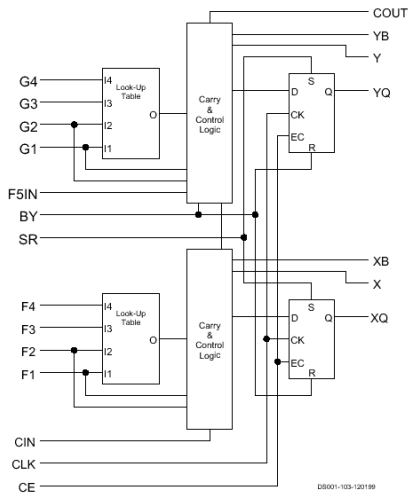


Leon – a new recipe for chips - [www.ESA.int](http://www.ESA.int)

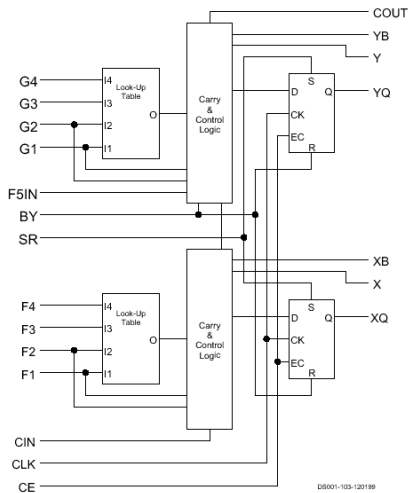


*“The next stage was then to physically translate that code to create a test board hosting a programmable chip called a ‘field programmable gate array’ (FPGA) where the LEON design could be put to work and tested.”*

Leon – a new recipe for chips - [www.ESA.int](http://www.ESA.int)



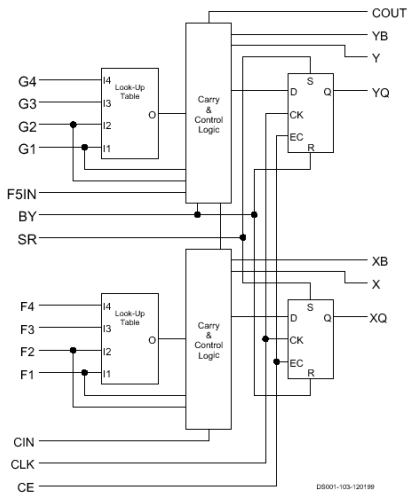
# SLICE



SLICE

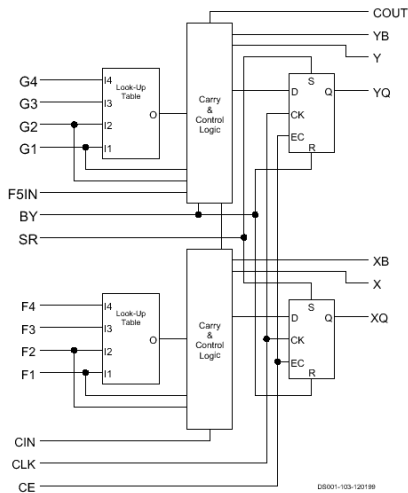
D8001-103-120189

- wchodzi w skład CLB (configurable logic block)



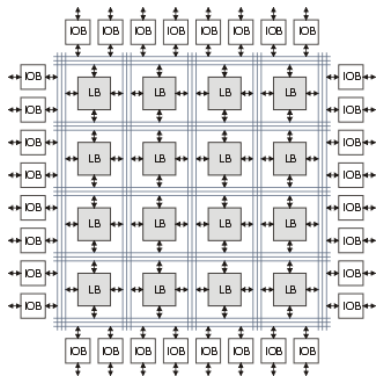
SLICE

- wchodzi w skład CLB (configurable logic block)
- dodatkowa "drobnica" (carry chain, OR, XOR ...)

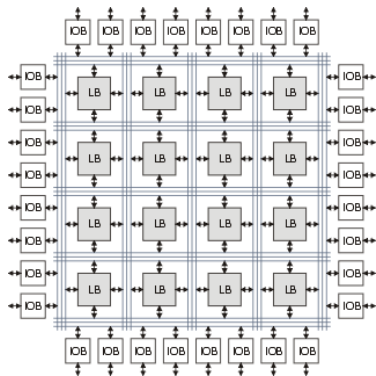


SLICE

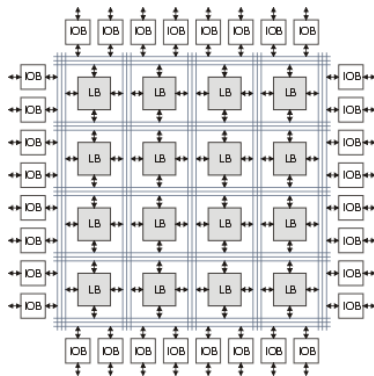
- wchodzi w skład CLB (configurable logic block)
- dodatkowa "drobnica" (carry chain, OR, XOR ...)
- CLB mają szybkie połączenia z sąsiadami



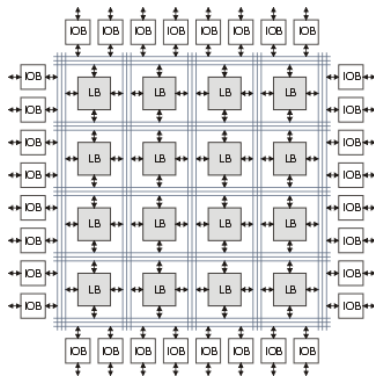




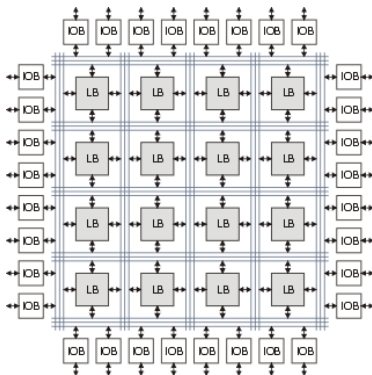
- łączę co dwa, co 4, co 8, wszystkie w koło



- łączę co dwa, co 4, co 8, wszystkie w koło
- układ mnożący (w Virtexie 18x18 bitów)

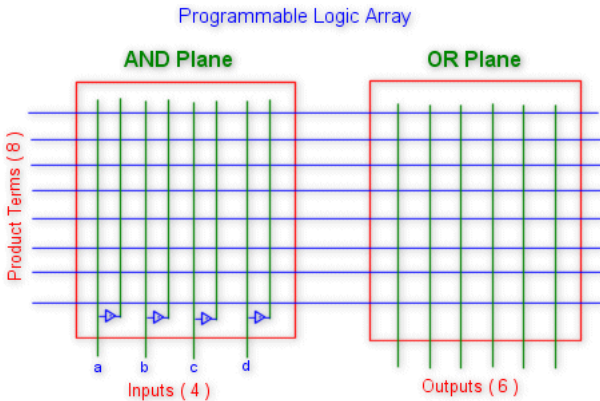


- łączę co dwa, co 4, co 8, wszystkie w koło
- układ mnożący (w Virtexie 18x18 bitów)
- rozsznane bloki pamięci RAM (po np. 18kBit)



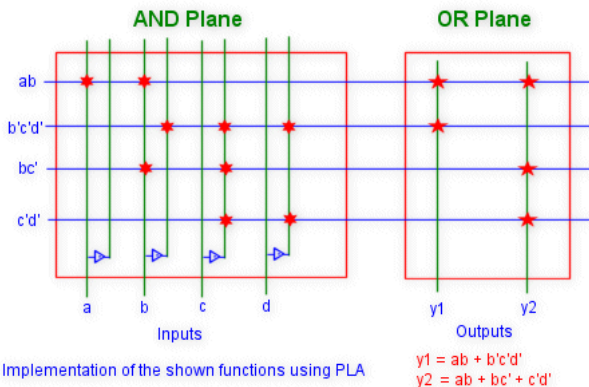
- łączę co dwa, co 4, co 8, wszystkie w koło
- układ mnożący (w Virtexie 18x18 bitów)
- rozsiane bloki pamięci RAM (po np. 18kBit)
- peryferia plus układ kompensacji zegara

# SPLD - dwie płaszczyzny, AND, OR



# SPLD - dwie płaszczyzny, AND, OR

Programmable Logic Array (4 x 4 x 2)



- stałe bramki, konfigurowalne połączenia

- stałe bramki, konfigurowalne połączenia
- FPLA, GAL – reprogramowalne

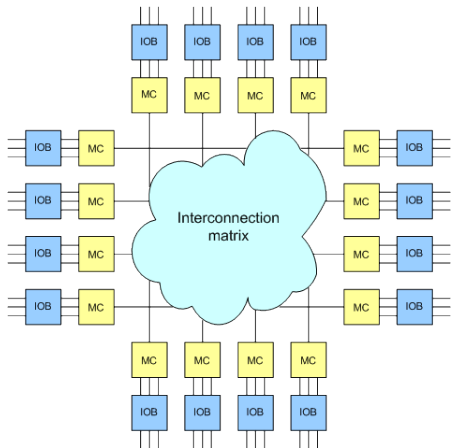


- stałe bramki, konfigurowalne połączenia
- FPLA, GAL – reprogramowalne
- GAL, PAL – stałe łączenia bramek OR

- stałe bramki, konfigurowalne połączenia
- FPLA, GAL – reprogramowalne
- GAL, PAL – stałe łączenia bramek OR
- dysjunkcyjna postać normalna

- stałe bramki, konfigurowalne połączenia
- FPLA, GAL – reprogramowalne
- GAL, PAL – stałe łączenia bramek OR
- dysjunkcyjna postać normalna
- dobre gdy iloczyny wielu we “kompresowane” do alternatywy

# CPLD – Complex Programmable Logic Device



- LB - logic blocks (kilka)
- microcells -  $\sim 8-16$  w LB
- MC połączone w LB
- LB połączone w CPLD

## PLD

- nieulotnie programowalne
- bloki we/wy bezpośrednio do LB
- dalej podstawą suma-produktów

## FPGA

- zwiększona złożoność operacji
- dużo bramek
- podzespoły ogólnego przeznaczenia

- przewidywalne opóźnienia czasowe **krótsze niż FPGA**

- przewidywalne opóźnienia czasowe **krótsze niż FPGA**
  - krytyczne, “high-performance” aplikacje nadzorcze

- przewidywalne opóźnienia czasowe **krótsze niż FPGA**
  - krytyczne, “high-performance” aplikacje nadzorcze
- **niedrogie i oszczędne** energetycznie



- **przewidywalne** opóźnienia czasowe **krótsze niż FPGA**
  - krytyczne, “high-performance” aplikacje nadzorcze
- **niedrogie i oszczędne** energetycznie
  - wymagające finansowo, bateryjne systemy mobilne (sensory)

- **przewidywalne** opóźnienia czasowe **krótsze niż FPGA**
  - krytyczne, “high-performance” aplikacje nadzorcze
- **niedrogie i oszczędne** energetycznie
  - wymagające finansowo, bateryjne systemy mobilne (sensory)
  - proste aplikacje np. 1-z-N, dekodowanie adresów etc.

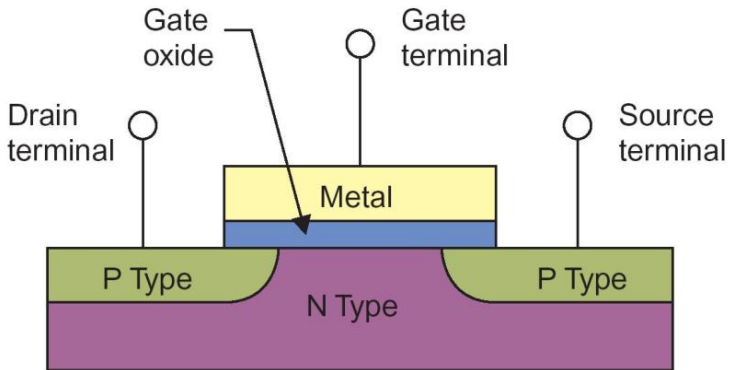
- **przewidywalne** opóźnienia czasowe **krótsze niż FPGA**
  - krytyczne, “high-performance” aplikacje nadzorcze
- **niedrogie i oszczędne** energetycznie
  - wymagające finansowo, bateryjne systemy mobilne (sensory)
  - proste aplikacje np. 1-z-N, dekodowanie adresów etc.
- **mniej wszechstronna** struktura

- **przewidywalne** opóźnienia czasowe **krótsze niż FPGA**
  - krytyczne, “high-performance” aplikacje nadzorcze
- **niedrogie i oszczędne** energetycznie
  - wymagające finansowo, bateryjne systemy mobilne (sensory)
  - proste aplikacje np. 1-z-N, dekodowanie adresów etc.
- **mniej wszechstronna** struktura
- **ograniczona** ilość wejść dla microcell

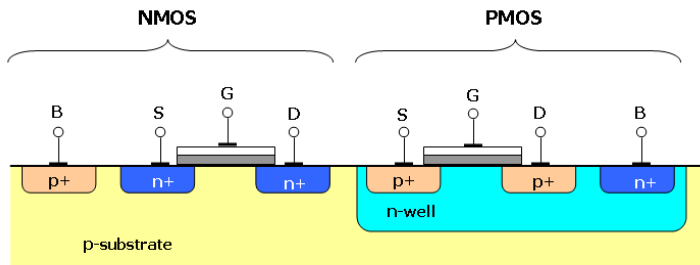
Metal-On-Silicon, Field-Effect, Transistor → MOSFET

# Tranzystor polowy - MOSFET

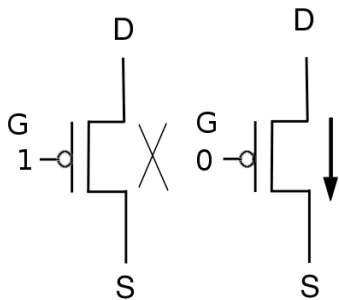
Metal-On-Silicon, Field-Effect, Transistor → MOSFET



Metal-On-Silicon, Field-Effect, Transistor → MOSFET



## pMOS

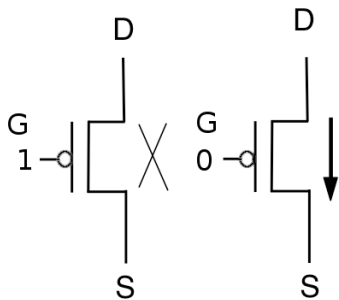


$G = 0 \rightarrow$  przewodzenie

$G = 1 \rightarrow$  brak przewodzenia

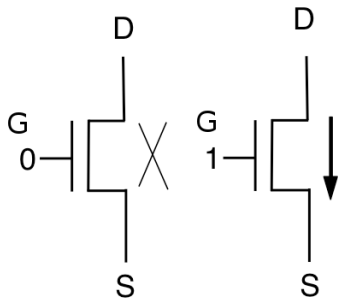


## pMOS



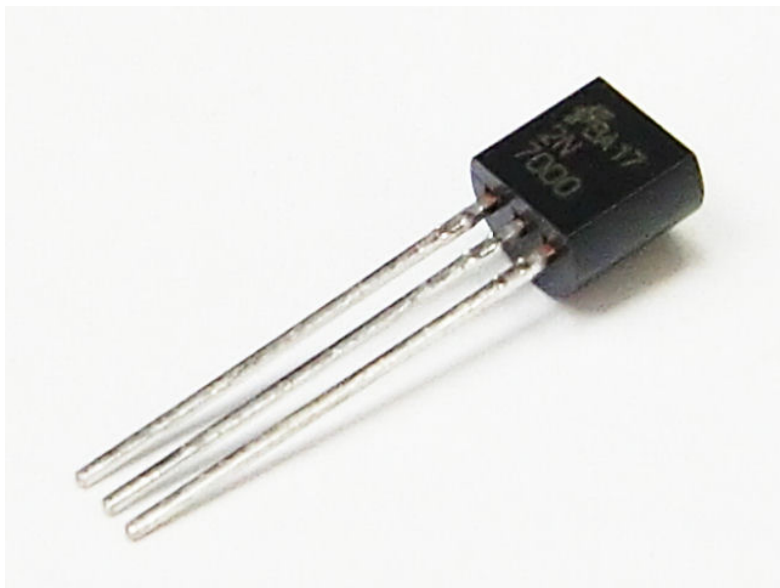
$G = 0 \rightarrow$  przewodzenie  
 $G = 1 \rightarrow$  brak przewodzenia

## nMOS



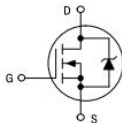
$G = 0 \rightarrow$  brak przewodzenia  
 $G = 1 \rightarrow$  przewodzenie

# MOSFET 2N7000



# MOSFET 2N7000

## N-Channel



TO-92  
CASE 29  
STYLE 22

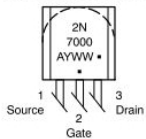


STRAIGHT LEAD  
BULK PACK



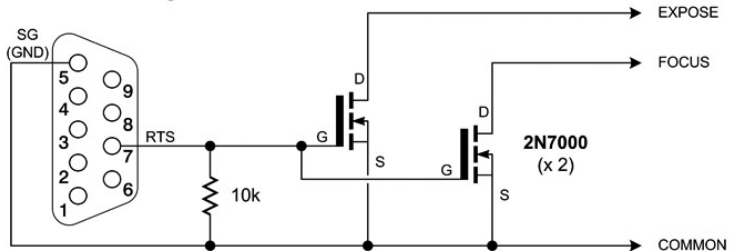
BENT LEAD  
TAPE & REEL  
AMMO PACK

## MARKING DIAGRAM AND PIN ASSIGNMENT



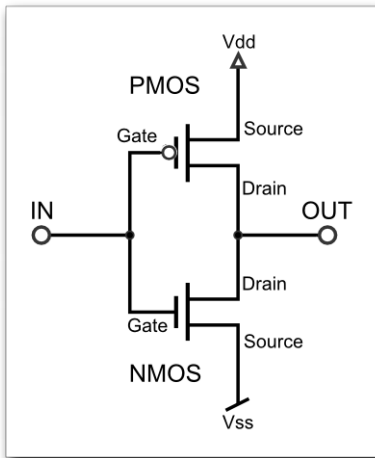
# MOSFET 2N7000

Serial cable for Canon EOS 40D  
Michael A. Covington 2007

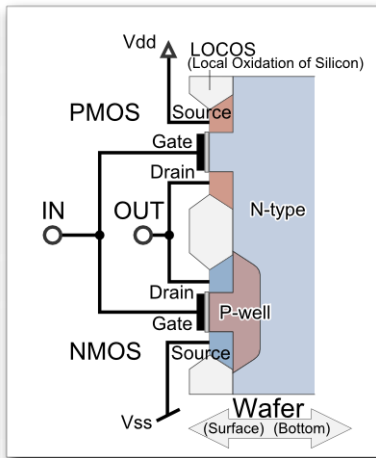


## CMOS inverter

Model chart

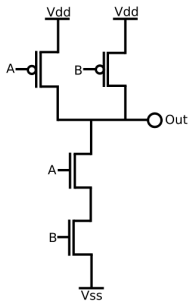


Silicon wafer

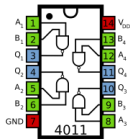
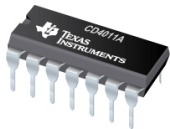
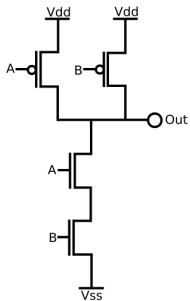




# Bramki logiczne CMOS - NAND, NOR

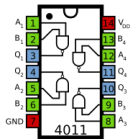
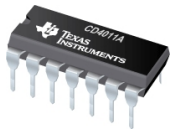
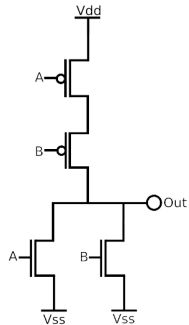
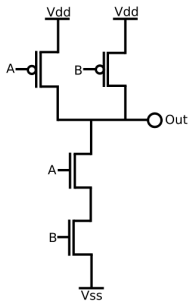


# Bramki logiczne CMOS - NAND, NOR

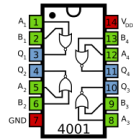
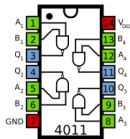
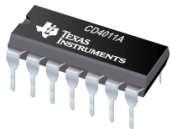
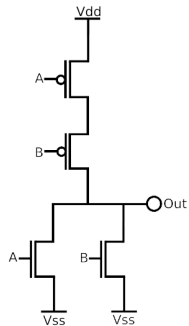
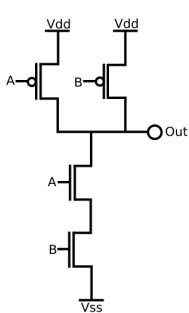


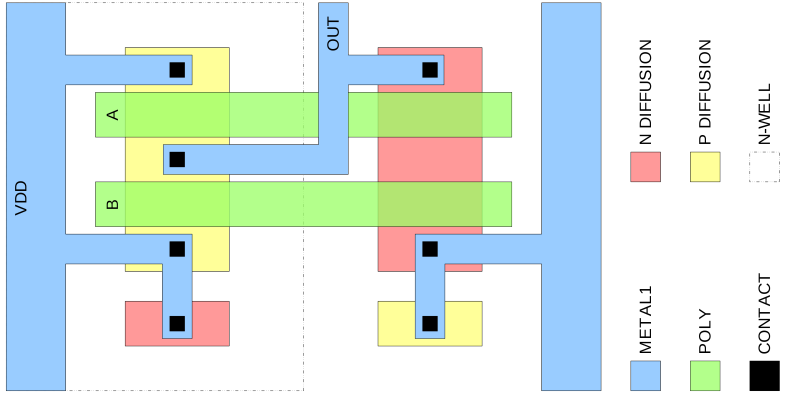


# Bramki logiczne CMOS - NAND, NOR



# Bramki logiczne CMOS - NAND, NOR





## Rzeczy do zapamiętania

- procesory specjalizowane vs. programowalne vs. programowalne-dedykowane;
- CORDIC;
- budowa, porównanie FPGA, (S/C)PLD;
- technologia CMOS: bramki INV, NAND, NOR;

## Do przeczytania

- [http://www.esa.int/Our\\_Activities/](http://www.esa.int/Our_Activities/) ⇒ LEON – a new recipe for chips.