

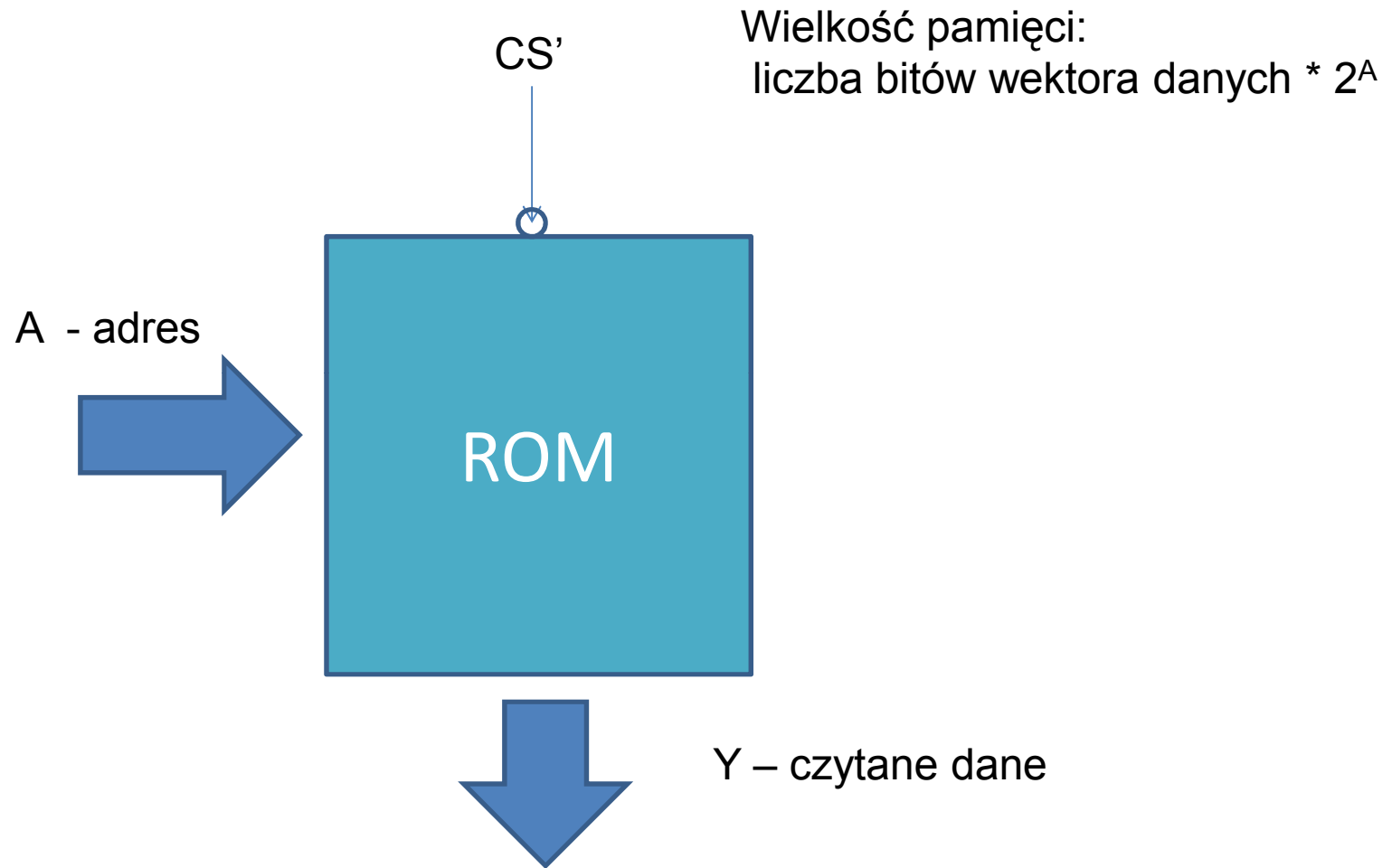
# Układy programowalne

Wykład z ptc część 5

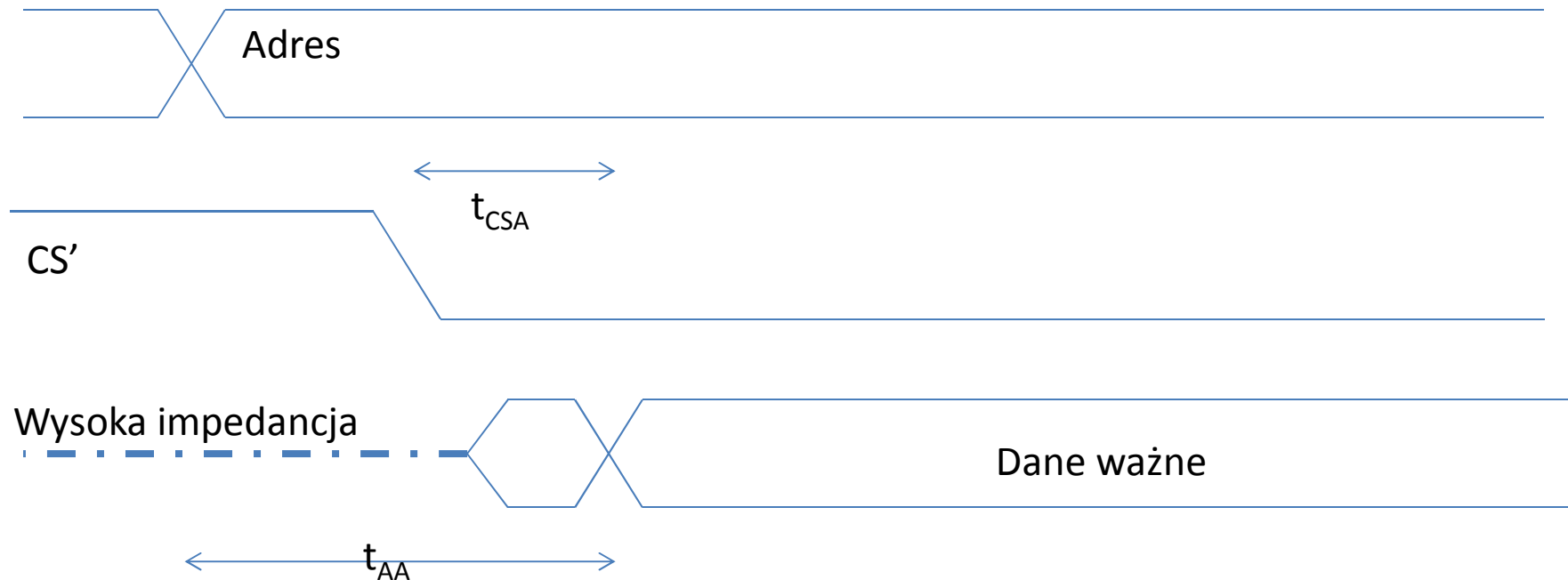
# Pamięci ROM

- Pamięci stałe typu ROM (Read only memory) umożliwiają jedynie odczytanie informacji zawartej w strukturze pamięci.
- Działanie:  $Y = X_j * cs'$  gdzie  $j = \text{Linia}(A)$ .
- W wyniku wybrania układu  $cs' = 0$  na wyjściach  $Y$  pojawia się wektor  $X_j$  zapisany w linii o numerze  $j$  o adresie określonym przez wejścia adresowe  $A$ .
- Wejście  $cs$  wprowadza wyjścia układu w stan wysokiej impedancji umożliwiając łączenie pamięci w bloki o większej pojemności i podłączanie do magistral.

# Schemat pamięci



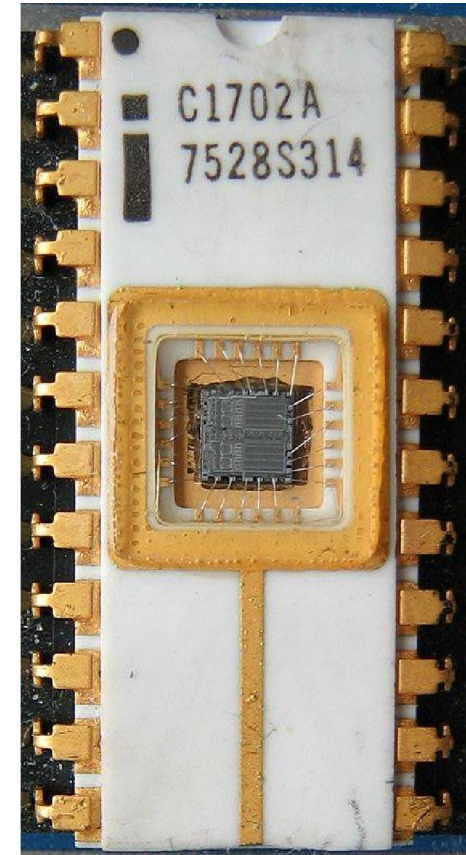
# Wykres dostępu do pamięci ROM



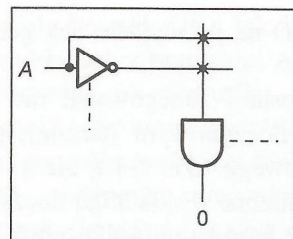
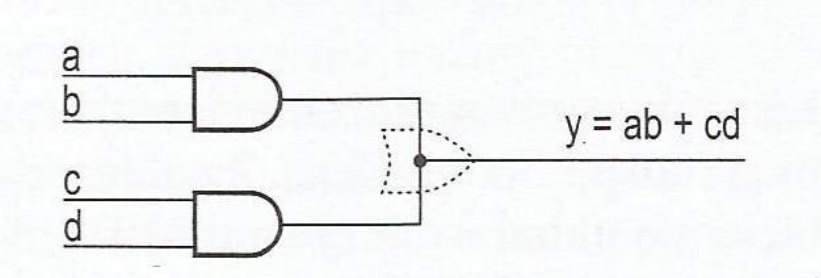
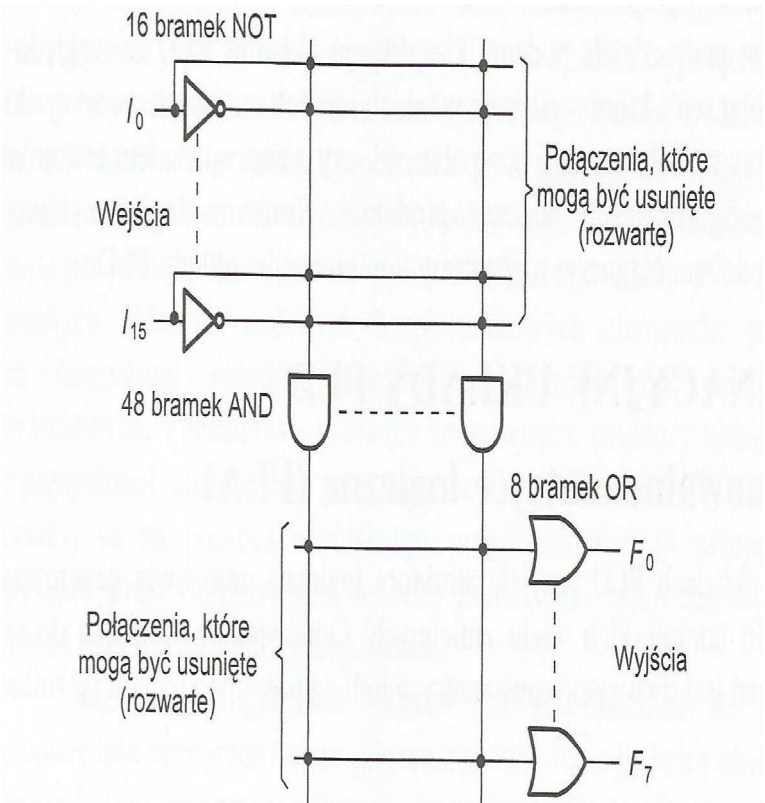
- Dla pojawiania się danych ważnych musi upłynąć co najmniej:
  - $t_{CSA}$  (czas dostępu od wybrania układu CS'=0) i
  - $T_{AA}$  (czas dostępu od adresu)

# Układy programowalne PLD

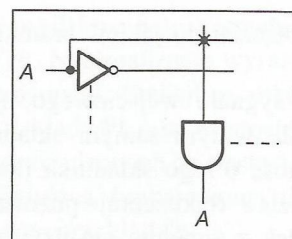
- Układy pamięci ROM reprogramowalne –
  - EEPROM – wymazywalne elektrycznie
  - EPROM – wymazywalne za pomocą światła UV
- ROM – PROM – PLD (Programmable Logic Devices)
- Układy programowane przez użytkownika PLD
- Wykorzystanie wyjść z otwartym kolektorem do realizacji – montażowych sum lub montażowych iloczynów (zmniejszenie liczby bramek - zwarcie wyjść) porównaj bramka z OC (wykład układy kombinacyjne)
- **Układy PLA** – (Programmable Logic Array) zbudowane z matrycy bramek iloczynów i matrycy bramek sum, możliwość określenia sygnałów wejściowych bramek – programowanie przez użytkownika.



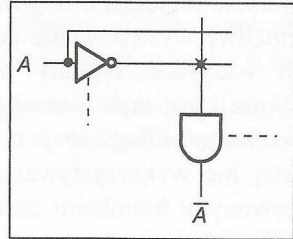
# Matryca PLA, jej struktura i programowanie



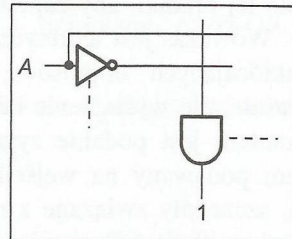
Stan nie zaprogramowany



Wybrano zmienną prostą

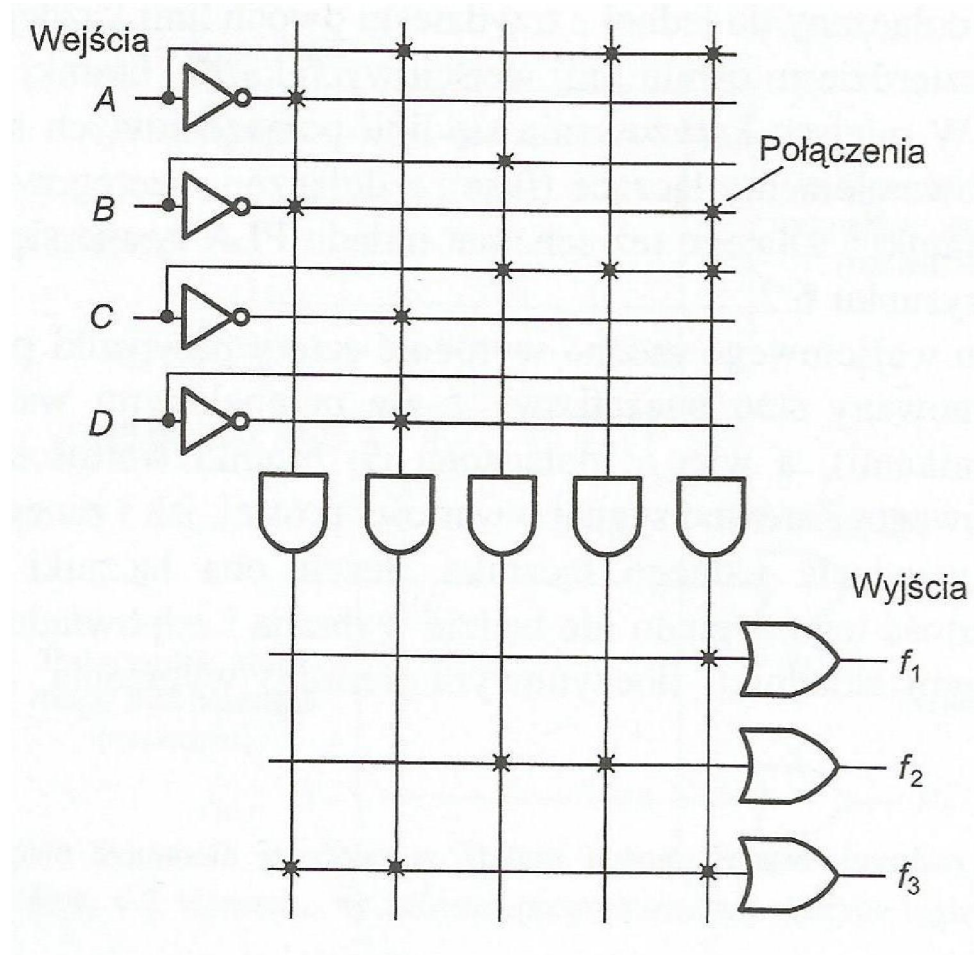


Wybrano zmienną zanegowaną



Nie wybrano żadnej zmiennej

# PLA zaprogramowany



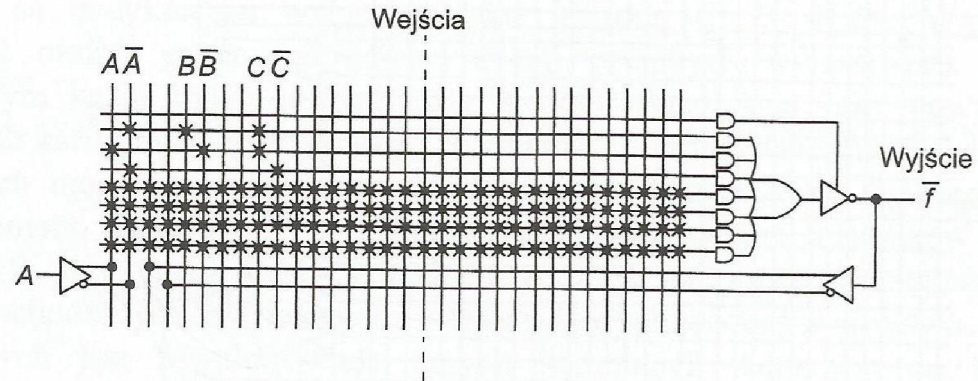
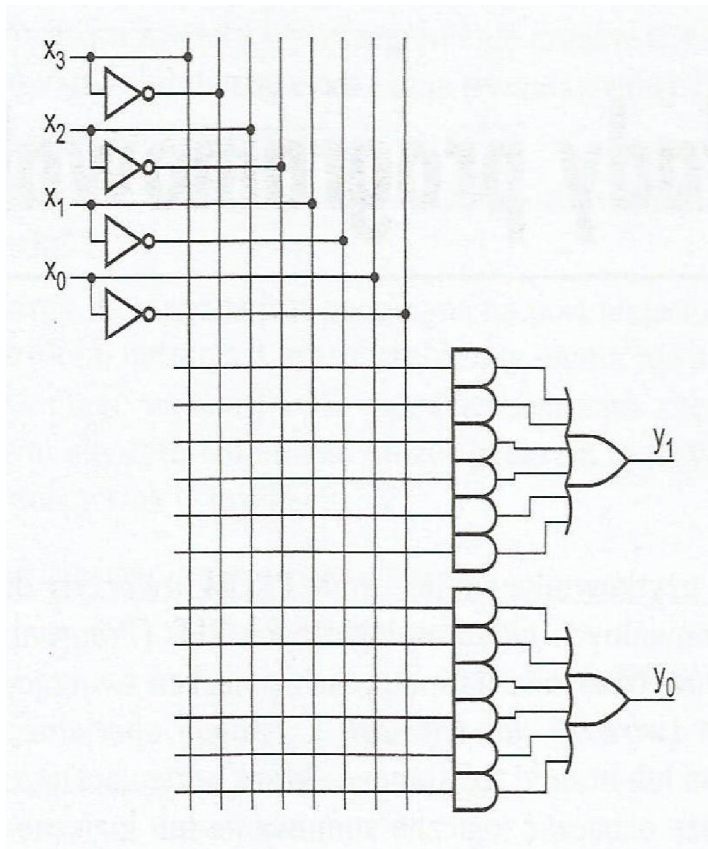
# Układy PAL

Brak wykorzystania w praktyce możliwości struktur układów PLA prowadzi do prostszych realizacji PLD – układów PAL (Programmable Array Logic) – typy układów posiadają:

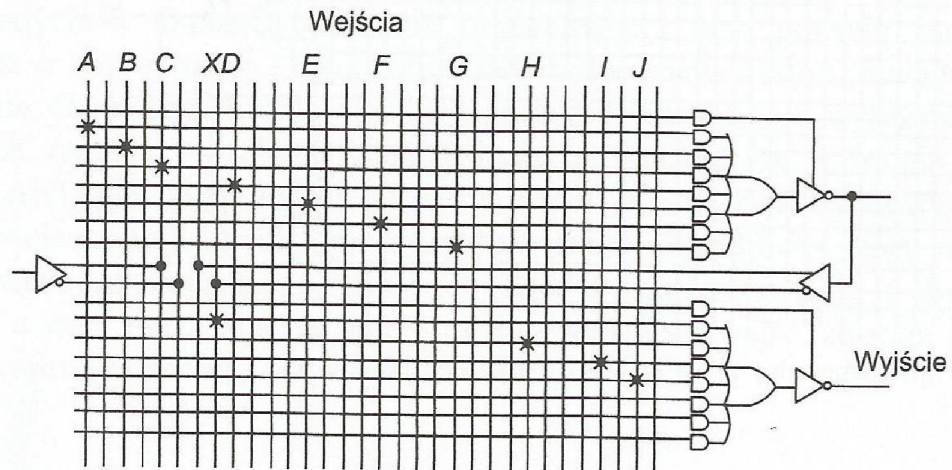
- ustaloną liczbę wyjść – bramki OR o **ustalonej liczbie wejść** – bramki AND i
- liczbę wejść – sygnały przez programowanie wykorzystane w iloczynach.
- Nazwy: np. GAL 16V8 pierwsza liczba to liczba wyprowadzeń wej/wyj, druga liczba wyjść



# PAL STRUKTURA i zastosowanie



Rys. 6.6 Implementacja funkcji  $\bar{f} = \bar{A}BC + A\bar{B}C + \bar{A}\bar{B}C$  w układzie PAL



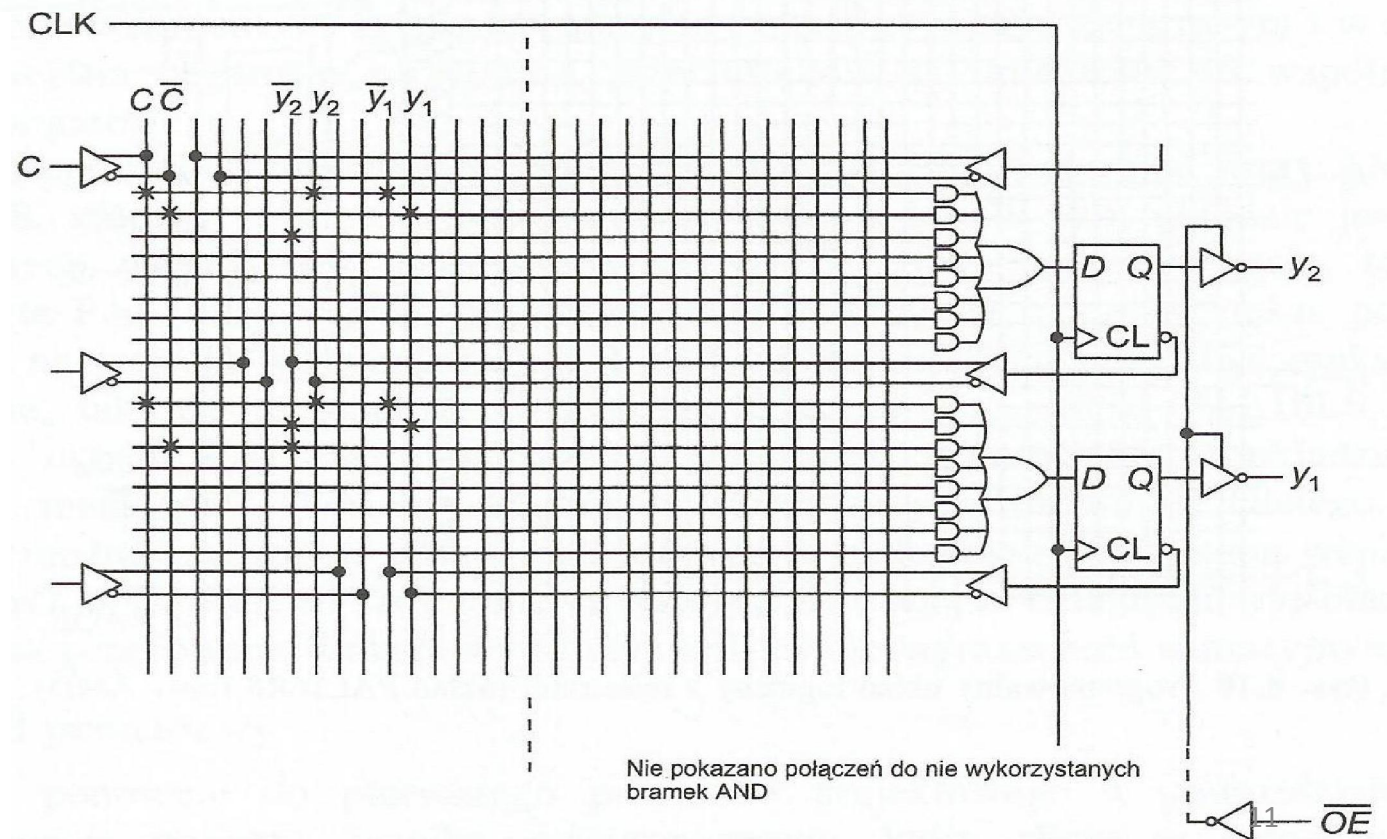
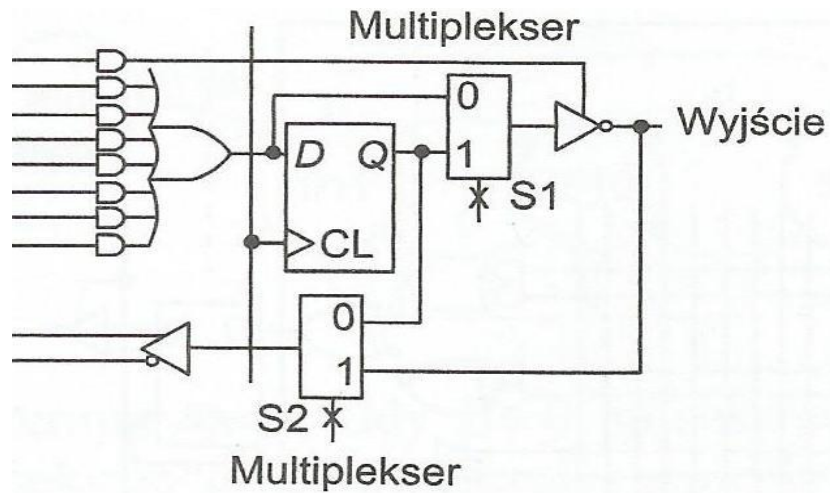
Nie pokazano połączeń do nie wykorzystanych bramek AND

Rys. 6.7 Implementacja wielopoziomowego wyrażenia sumacyjnego w układzie PAL

# Rodzaje układów PAL

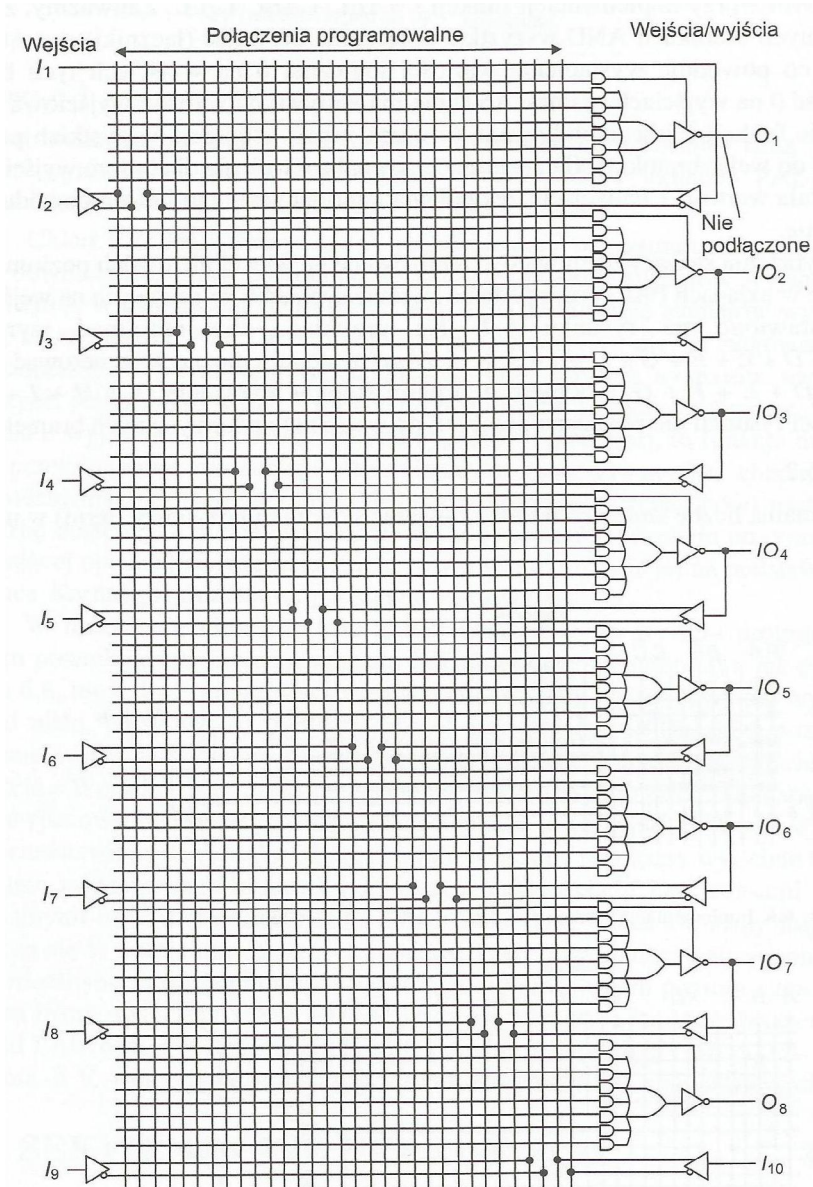
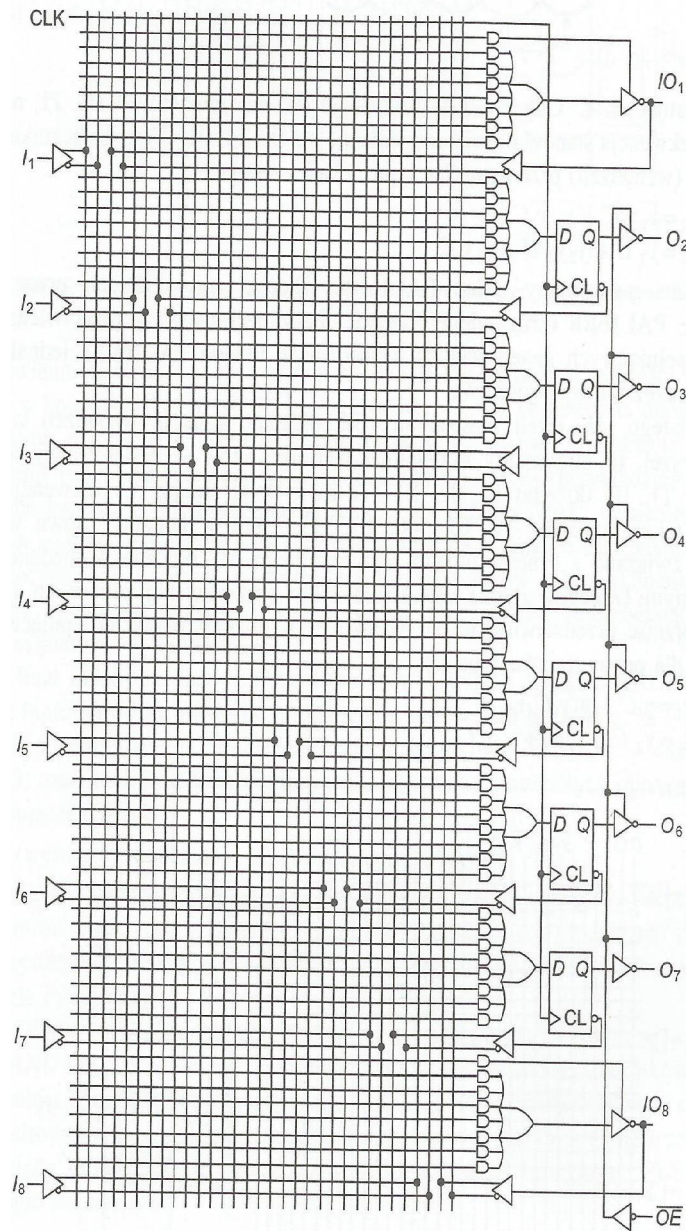
- Sprzężenie zwrotne – układy asynchroniczne
- Wejścia-wyjścia
- Rejestry synchroniczne – automaty synchroniczne
- Rejestry z programowanym sygnałem zegara
- Rozbudowa układów PAL w kierunku bardziej złożonych jakościowo i ilościowo:
  - Układów CPLD (complex ...) – elementy programowane – komórki logiczne
  - Układów FPGA (Field Programmable Gate Array) – złożone strukturalnie makrokomórki

# Makrokomórka PAL i projekt licznika





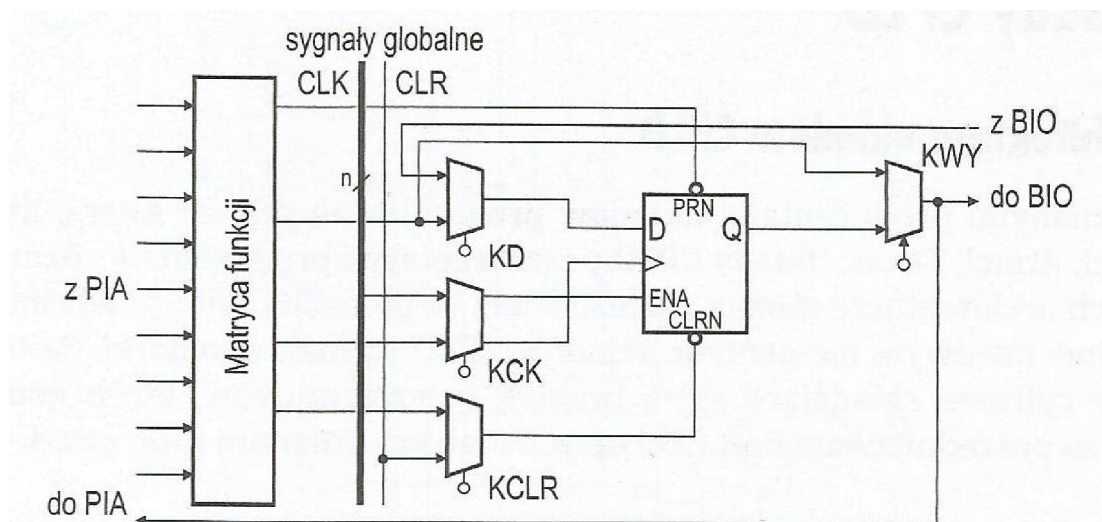
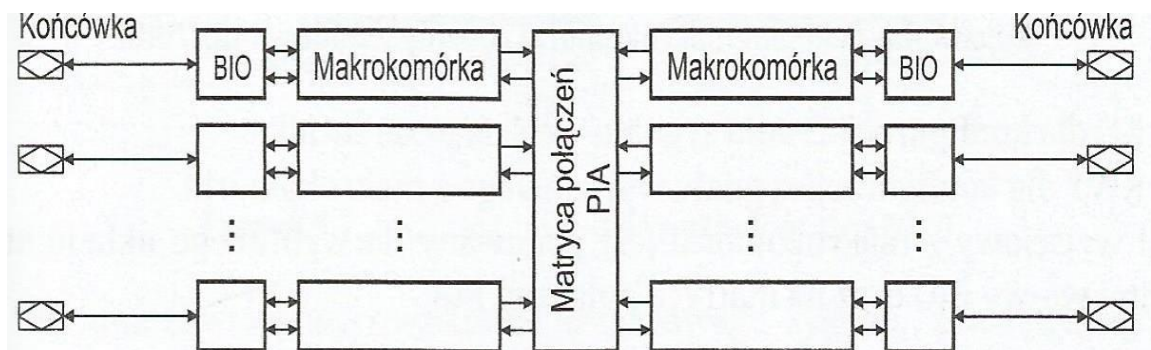
# Przykłady PAL



# Układy CPLD

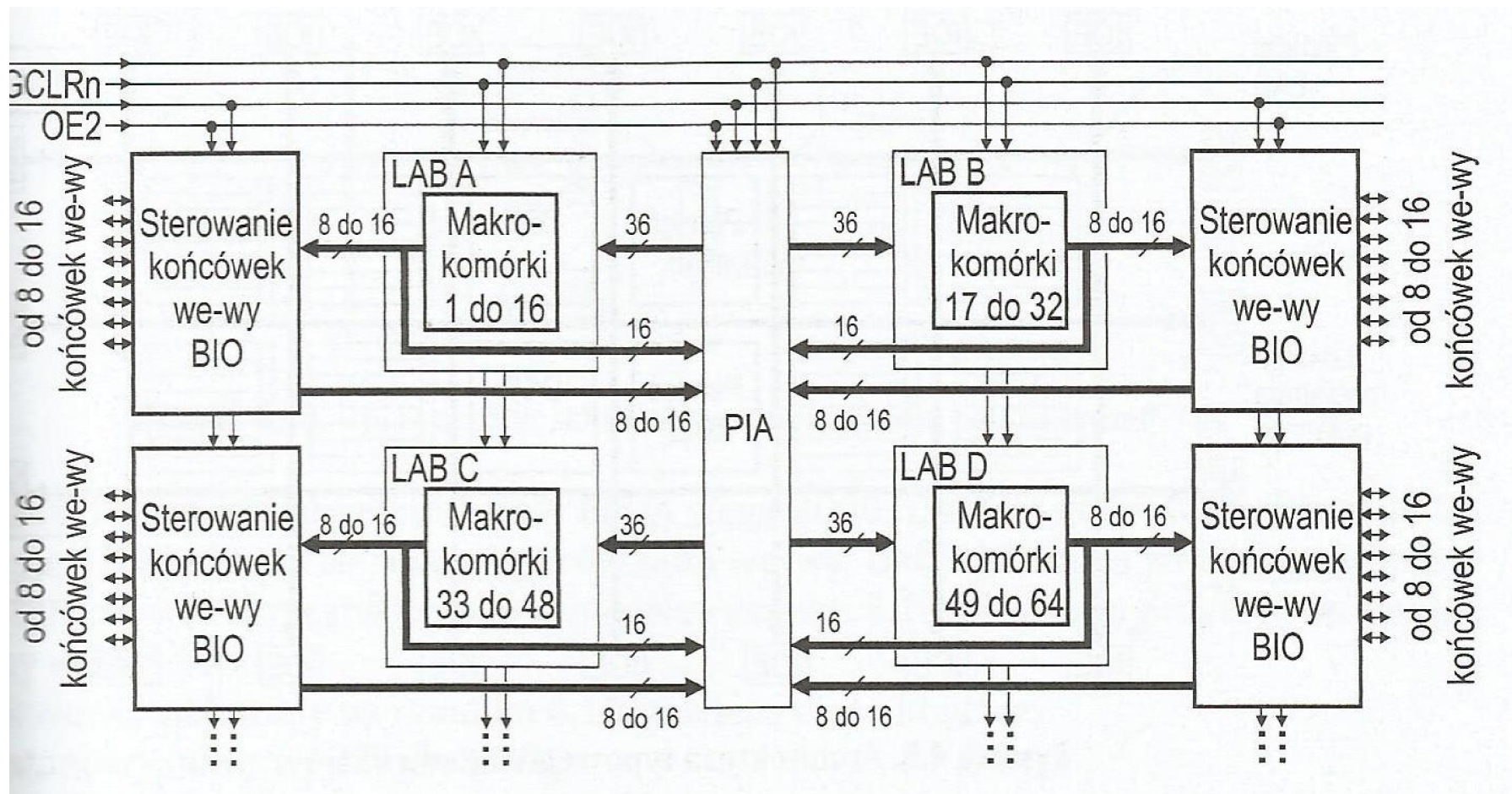
- Matryca połączeń PIA (Programmable Interconnect Array) - dostęp do sygnałów generowanych, wejściowych, wyjściowych
- Bloki wej/wyj BIO
- Makrokomórka
  - programowalna matryca funkcji boolowskich
  - układ konfiguracji źródła sygnału wejścia przerzutnika
  - układ konfiguracji sygnału wyjściowego z makrokomórki
  - układy konfiguracji źródeł sygnałów sterujących clk, clr, enable
- Możliwość zapamiętania konfiguracji układu CPLD w EEPROM (elektrycznie wymazywalnej pamięci ROM) – jeden bit komórki pamięci odzwierciedla stan programowalnego połączenia układu.

# Układ CPLD – struktura





# Układ CPLD – przykład architektury MAX7000

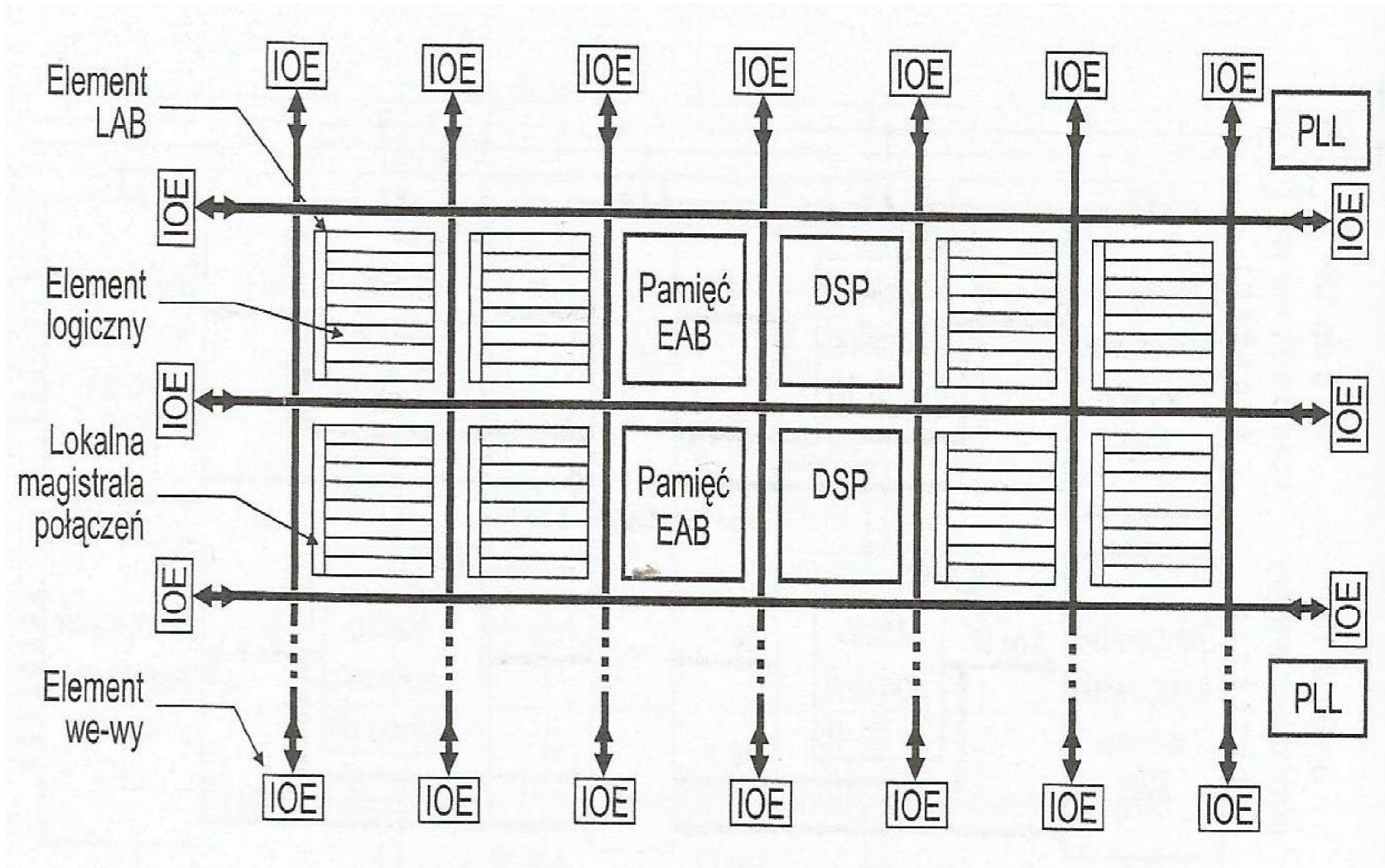


# Układy FPGA

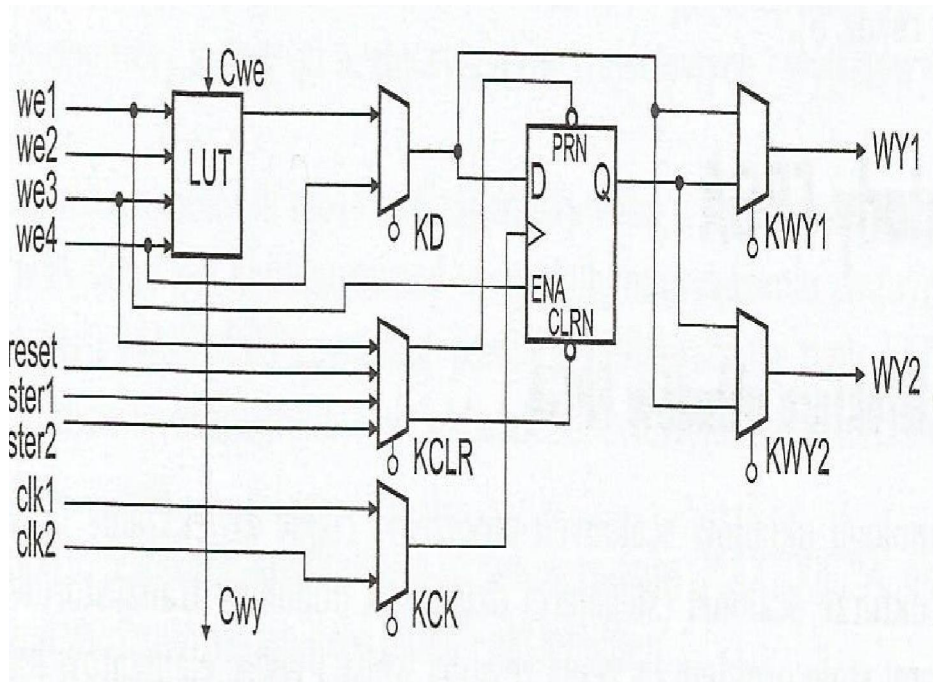
- Wzrost liczby elementów logicznych
- Efektywne rozwiązanie problemu łączenia elementów:
  - Magistrale połączeń międzyblokowych (multitrack interconnection) - wiersze i kolumny konfigurowalnych połączeń (ścieżki dzielone) pomiędzy blokami funkcjonalnymi i elementami wej-wyj IOE
- Bloki funkcjonalne FPGA
  - Matryce LAB (z lokalnymi magistralami) elementów logicznych LE
  - Układy pamięci,
  - Układy DSP (cyfrowego przetwarzania sygnałów) – konfigurowalne układy mnożące,
  - Układy z pętlami fazowymi PLL – generacja synchronizowanych sygnałów zegarowych o żądanych częstotliwościach .



# Architektura FPGA

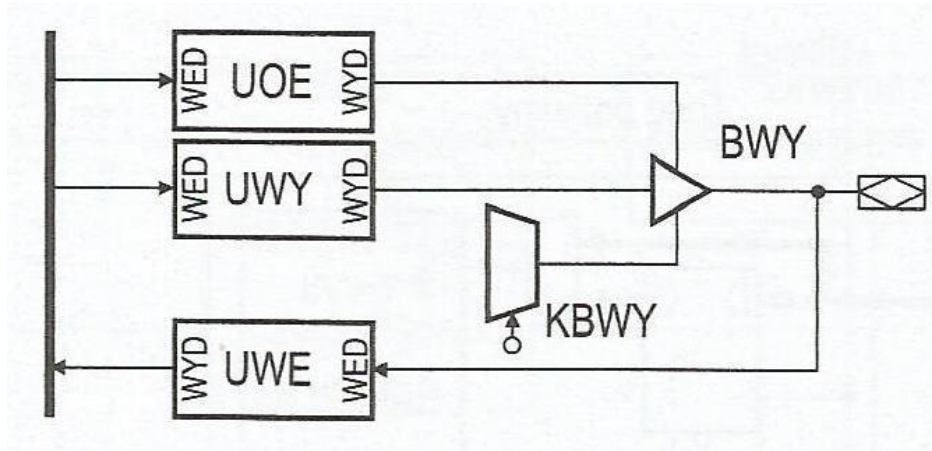


# Struktura LE

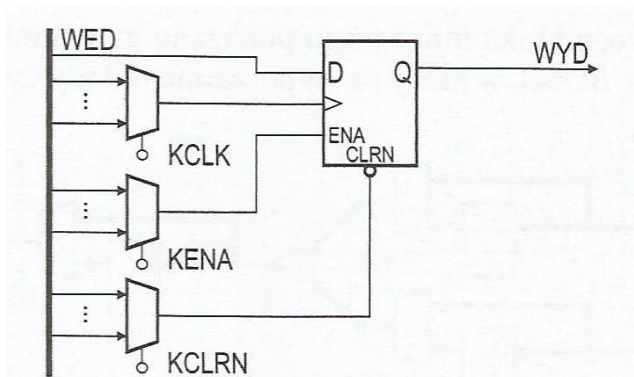


- Rys. 4.9 str 50 pawlowski
- **Pamięć funkcji logicznej LUT** (look up table) programowalna realizuje dowolne funkcje swoich wejść
- Sygnały przeniesienia
- Układy konfiguracji sygnałów sterujących i danych,
- Wyjścia podłączone do magistrali lokalnego LAB i magistrali połączeń międzyblokowych

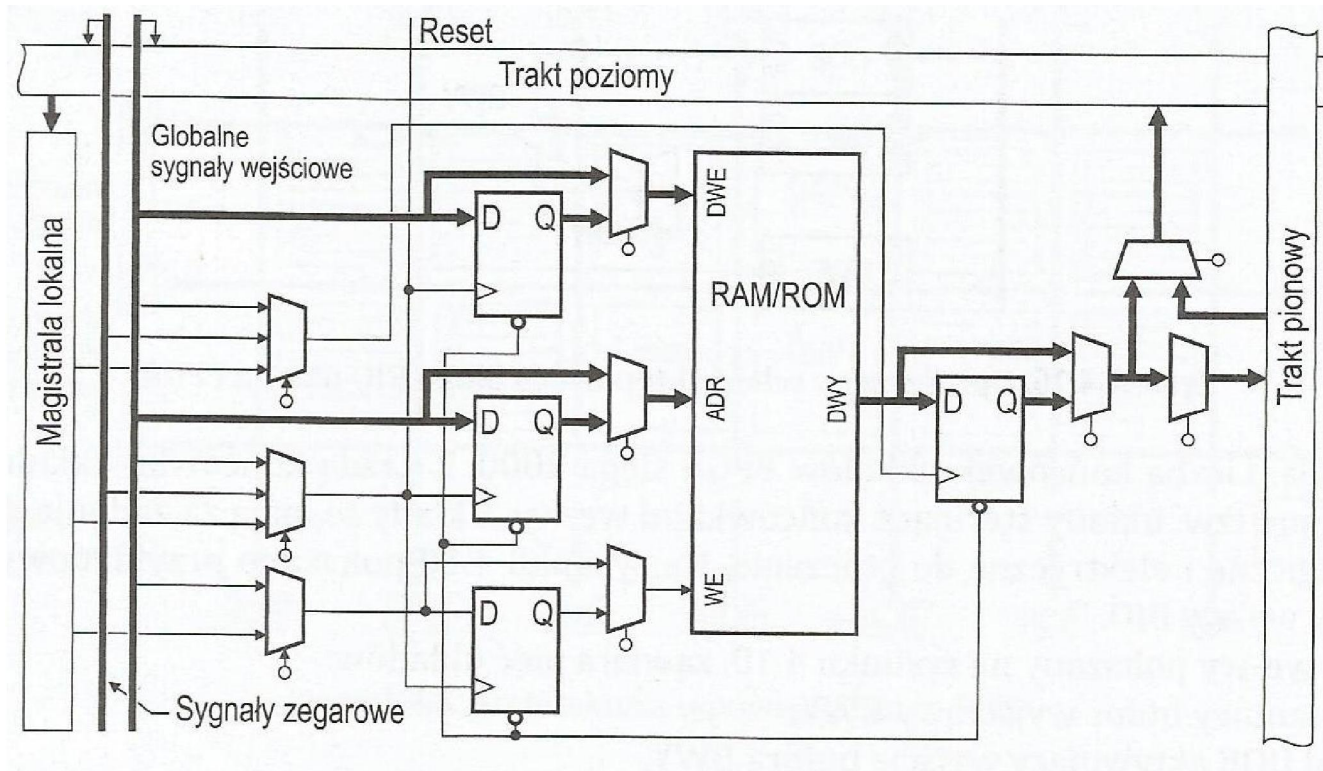
# Struktura bloku wyjściowego



KBWY – układ konfiguracji bufora wyjścia pozwala na dostosowanie bufora do współpracy z układami: TTL, CMOS, o różnych napięciach zasilania itp..

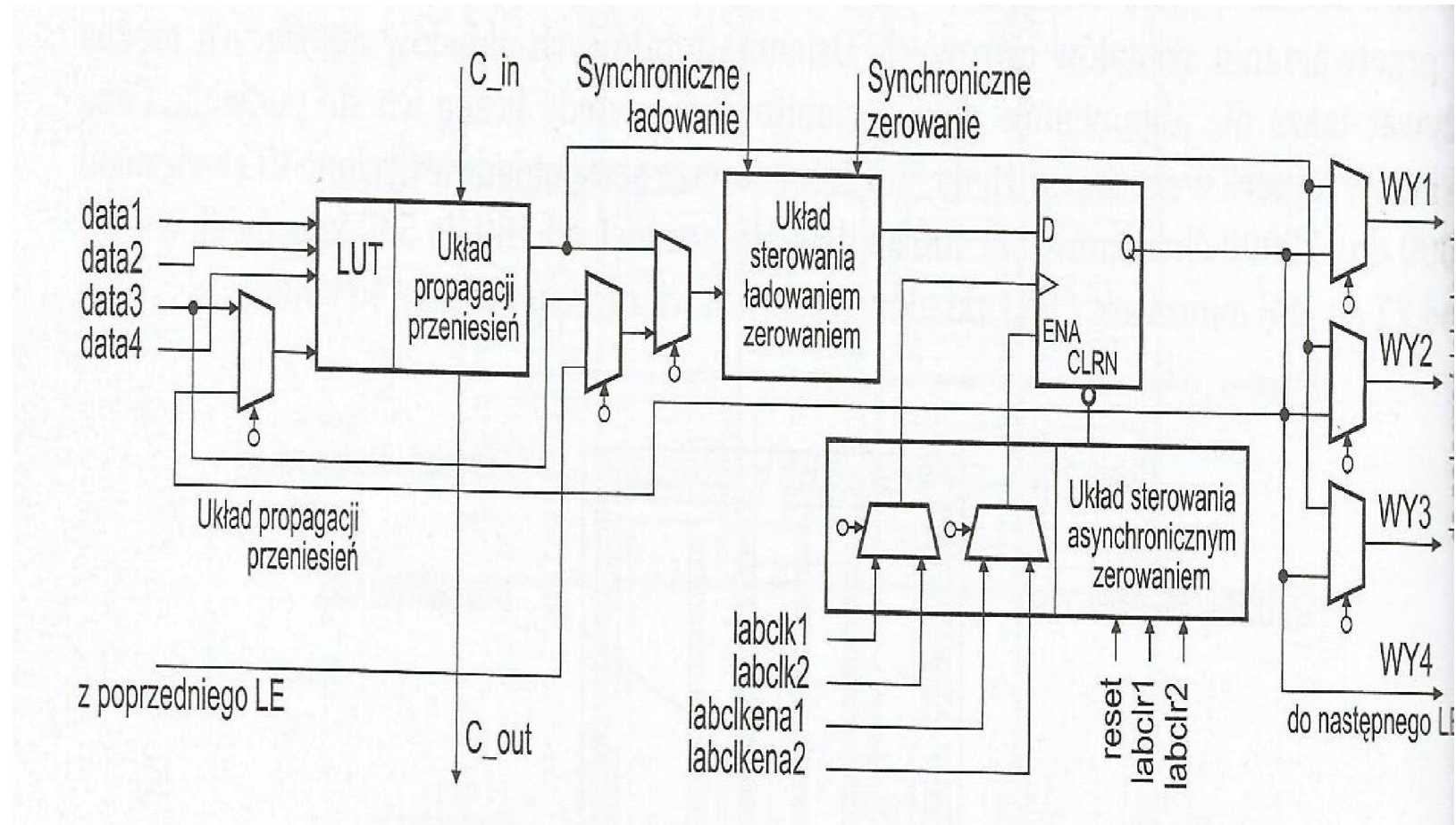


# Blok pamięci wbudowanej w FPGA

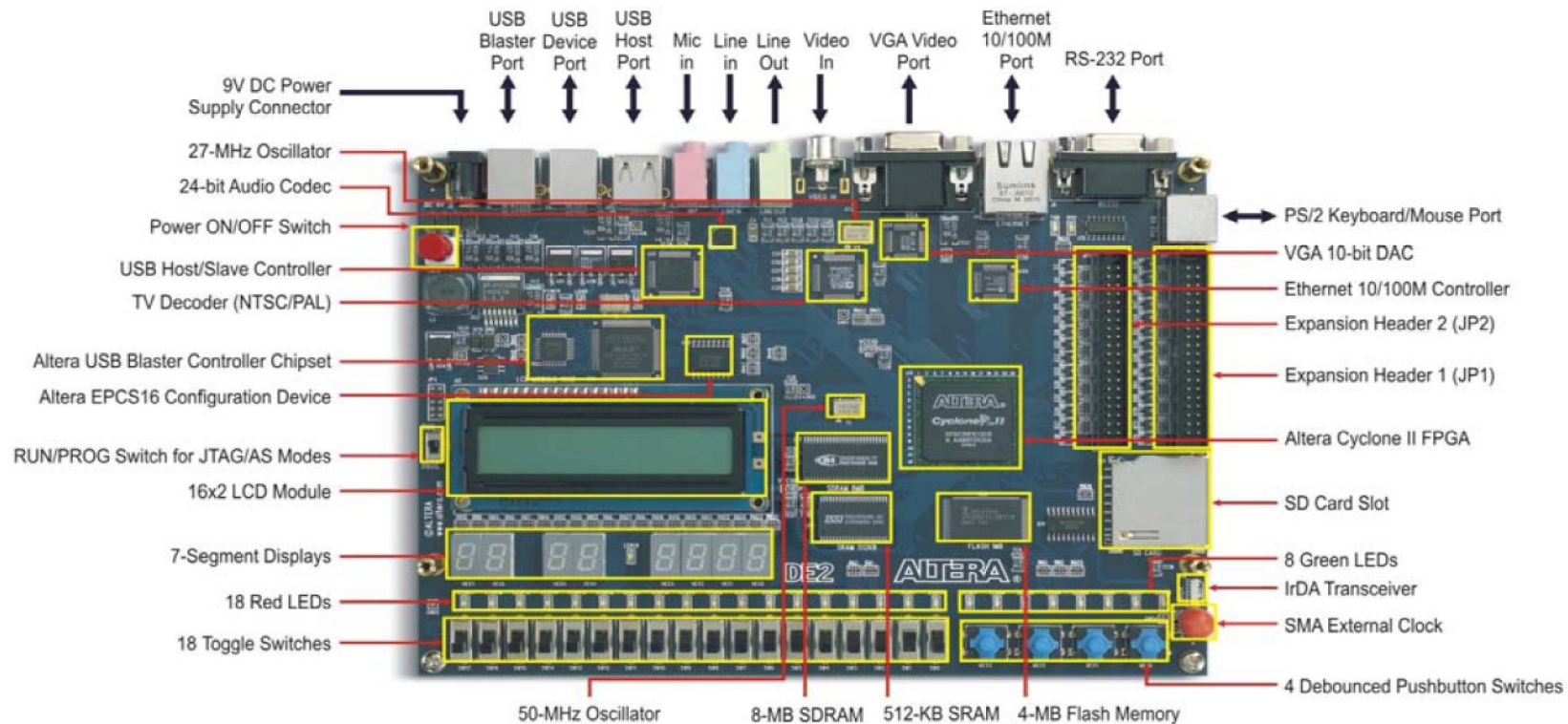




# Schemat LE w Cyclone III



# DE2 Development and Education Board



Korzystamy również z DE2-70 – podobna struktura, ~2x większy FPGA

# Elementy składowe DE2

- **Altera Cyclone® II 2C35 FPGA device**
- Altera Serial Configuration device - EPCS16
- USB Blaster (on board) for programming and user API control; both JTAG and Active Serial
- (AS) programming modes are supported
- 512-Kbyte SRAM
- 8-Mbyte SDRAM
- 4-Mbyte Flash memory (1 Mbyte on some boards)
- SD Card socket
- 4 pushbutton switches
- 18 toggle switches
- 18 red user LEDs
- 9 green user LEDs
- 50-MHz oscillator and 27-MHz oscillator for clock sources
- 24-bit CD-quality audio CODEC with line-in, line-out, and microphone-in jacks
- VGA DAC (10-bit high-speed triple DACs) with VGA-out connector
- TV Decoder (NTSC/PAL) and TV-in connector
- 10/100 Ethernet Controller with a connector
- USB Host/Slave Controller with USB type A and type B connectors
- RS-232 transceiver and 9-pin connector
- PS/2 mouse/keyboard connector
- IrDA transceiver
- Two 40-pin Expansion Headers with diode protection

# Schemat blokowy DE2

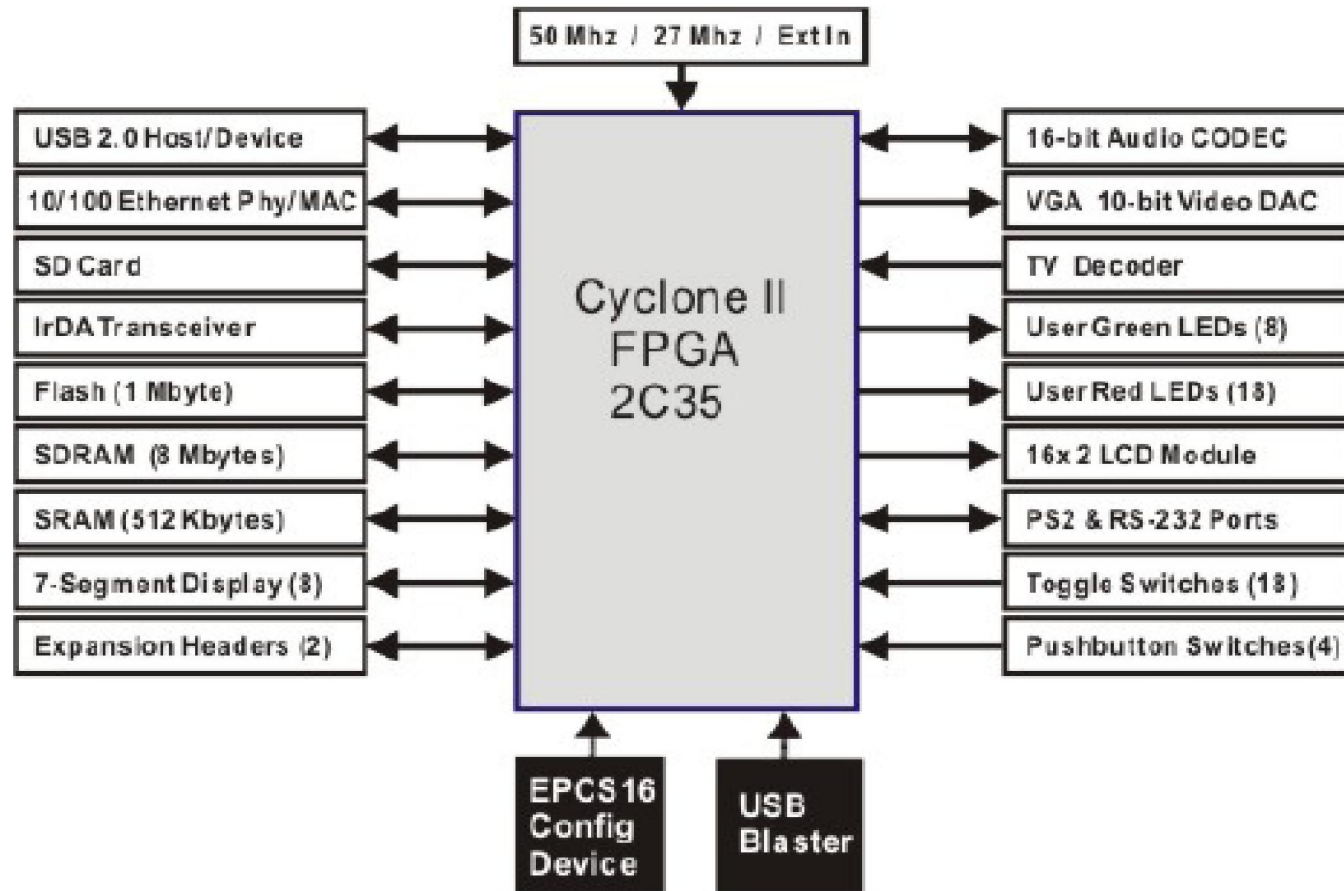


Figure 2.2. Block diagram of the DE2 board.



# Parametry Cyclone II 2C35 FPGA

- 33,216 elementów logicznych/ 70 k LE dla 2C70
- 105 bloków pamięci RAM M4K
- 483,840 liczba bitów RAM
- 35 wbudowanych układów mnożących
- 4 pętle sprzężenia fazowego
- 475 wyprowadzeń użytkownika typu I/O
- obudowa z 672 wyprowadzeniami

# Płyta DE2 i FPGA Cyclone III

- [https://www.cs.put.poznan.pl/rwalkowiak/pliki/DE2\\_UserManual.zip](https://www.cs.put.poznan.pl/rwalkowiak/pliki/DE2_UserManual.zip)
- [https://www.cs.put.poznan.pl/rwalkowiak/pliki/DE2\\_70\\_User\\_manual\\_v108.zip](https://www.cs.put.poznan.pl/rwalkowiak/pliki/DE2_70_User_manual_v108.zip)
- [https://www.cs.put.poznan.pl/rwalkowiak/pliki/cyc2\\_cii51002.zip](https://www.cs.put.poznan.pl/rwalkowiak/pliki/cyc2_cii51002.zip)

# Projektowanie złożonych układów cyfrowych

Etapy:

- Formalizowanie opisu
- Projekt koncepcyjny
- Projekt techniczny
- Uruchomienie i weryfikacja

# Formalizowanie opisu

- Specyfikacja działań układu
- Sposób współpracy z otoczeniem: sygnały danych, sygnały zegarowe, wymagania na sygnały współpracy z urządzeniami i pamięciami.
- Specyfikacja parametrów elektrycznych i czasowych układu
- Specyfikacja parametrów mechanicznych układu (obudowa, interfejsy).

# Projekt koncepcyjny

- Wybór architektury – podział realizowanych funkcji na sprzęt i oprogramowanie, wybór technologii: standardowe układy scalone, układy programowalne (FPGA, CPLD); wybór architektury modułów - możliwe wersje architektury:
  - Standardowe bloki funkcjonalne
  - Struktura mikroprogramowalna
  - System mikroprocesorowy
- Dekompozycja układu - podział na moduły – bloki funkcjonalne:
  - **Metoda projektowania zstępująca top-down** – użycie bloków realizujących skomplikowane funkcje bez określenia sposobu ich realizacji, określenie sposobu współpracy, znając strukturę i zasady współpracy projektuje się poszczególne bloki.
  - **Metoda projektowania wstępująca bottom-up** – zastosowanie bloków o określonej - znanej budowie, dla których należy określić sposób współpracy, aby uzyskać układ realizujący wymagania projektu.

# Projekt koncepcyjny

- Opis połączeń i opis funkcji bloków – metody:
  - Schemat
  - Język HDL (projekt w FPGA)
    - Opis strukturalny – przyjęcie struktury bloków funkcjonalnych, takiej jaka odpowiadałaby realizacji z układów scalonych, projekt jest schematem, którego symbole graficzne układów zamieniono na opis w języku HDL, wymagana jest znajomość opisu budowy układów mogących zrealizować poszczególne funkcje.
    - Opis behawioralny – opis układu za pomocą funkcji arytmetycznych i logicznych, realizacja jest powierzona komputerowym systemom wspomagania projektowania.
    - Można wykorzystać dostępne w układach programowalnych: bloki RAM, bloki mnożące, DSP, PLL dostępne w kodzie HDL jako makrodefinicje.
    - Można wykorzystać specjalne narzędzia do włączenia w projekty zespołów funkcjonalnych: procesory, potrzebne interfejsy.

# Projekt techniczny. Uruchomienie urządzenia i weryfikacja projektu.

- **Projekt techniczny** to szczegółowe zaprojektowanie schematów, programów HDL wszystkich modułów urządzenia i programów sterujących ich działaniem. Projekt techniczny jest weryfikowany na podstawie działania modelu fragmentów urządzenia lub poprzez zastosowanie symulatorów.
- **Wykonanie urządzenia, jego uruchomienie i weryfikacja z założeniami**, która może prowadzić do potrzeby modyfikacji projektu i urządzenia – powrotu w tym iteracyjnej procedurze projektowania do wcześniejszych etapów.

# Projektowanie urządzeń cyfrowych - metodologie

- **Projektowanie klasyczne** – schematy modułów i programy dla użytych procesorów; tworzenie i weryfikacja modelu: budowa i uruchamianie modułów, integracja rzeczywistych modułów z symulatorami niektórych modułów np. wykorzystywanych procesorów.
- **Projektowanie współczesne** - przy użyciu komputerowych systemów wspomagania projektowania – testowanie koncepcji urządzenia bez potrzeby budowy modelu, wykorzystanie kompilatorów języka HDL, symulatorów i analizatorów projektów. Metoda pozwala na realizację modelu w strukturze programowalnej np. przy użyciu płyt modelowych (np. Terasic DE2) podłączanych do komputera w celu programowania i na bieżąco testowania pracy układu, w tym również śledzeniu stanu sygnałów i zawartości rejestrów np. procesora (narzędzia typu SOPC). Umożliwia szybką budowę prototypów – (rapid prototyping).