

# **PROGRAMOWALNE UKŁADY ELEKTRONICZNE**

**Prowadzący:**

**Dr inż. Michał Krystkowiak**

**[Michal.Krystkowiak@put.poznan.pl](mailto:Michal.Krystkowiak@put.poznan.pl)**

# **Orcad jako narzędzie symulacyjne wspomagające projektowanie cyfrowych układów elektronicznych**

## **(przypomnienie - część I)**

### Plan wykładu:

**1. Wstęp**

**2. Prezentacja modułu SCHEMATICS**

- zapoznanie się z edytorem graficznym,
- omówienie bibliotek wybranych elementów,
- sposób realizacji modeli symulacyjnych

**3. Deklaracja parametrów analizy TRANSIENT**

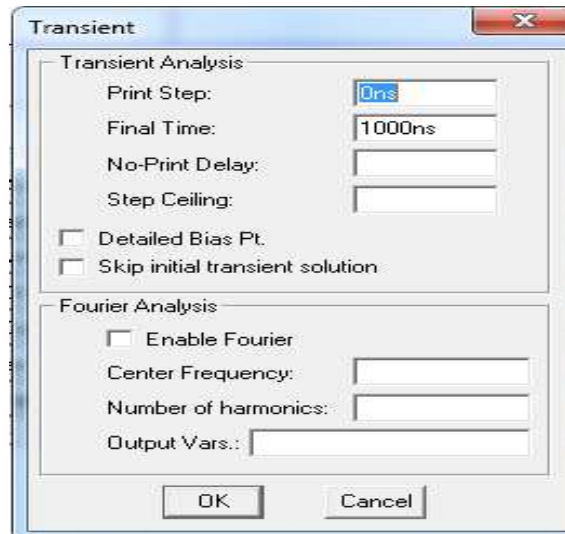
**4. Analiza FOURIERA**

**5. Charakterystyka modułu PROBE**

**6. Realizacja i badania prostego modelu symulacyjnego**

**Uwaga:** wymienione zagadnienia zostaną zaprezentowane z wykorzystaniem aplikacji ORCAD FAMILY RELEASE.

# Parametry analizy czasowej TRANSIENT



W czasie wykonywania analizy czasowej (**TRANSIENT**) obliczana jest odpowiedź układu w przedziale od  $t = 0s$  do podanej wartości końcowej (**Final Time**). Warunki początkowe obliczane są w czasie wykonanej wcześniej analizy stałoprądowej (dla założenia, że przed chwilą  $t = 0s$  panował w układzie stan ustalony). Istnieje możliwość pominięcia warunków początkowych poprzez uaktywnienie opcji **Skip initial transient solution**.

W celu umożliwienia wykonania analizy czasowej konieczna jest deklaracja następujących parametrów:

- **Print Step** (wartość kroku, z jakim ma być wykonywana analiza),
- **Final Time** (wartość końcowa przedziału czasowego).

Proces próbkowania w trakcie trwania analizy czasowej nie jest równomierny (duża dynamika zmian sygnałów w czasie wiąże się z dużą częstotliwością próbkowania). **Step Ceiling** jest parametrem pozwalającym na określenie granicznej wartości, dla której powinny zostać wykonane obliczenia.

# Architektura programowalnych układów logicznych PLD

## Plan wykładu:

**1. Klasyfikacja programowalnych układów logicznych**

**2. Omówienie architektury układów PLD**

- układy SPLD,
- układy CPLD,
- układy FPGA

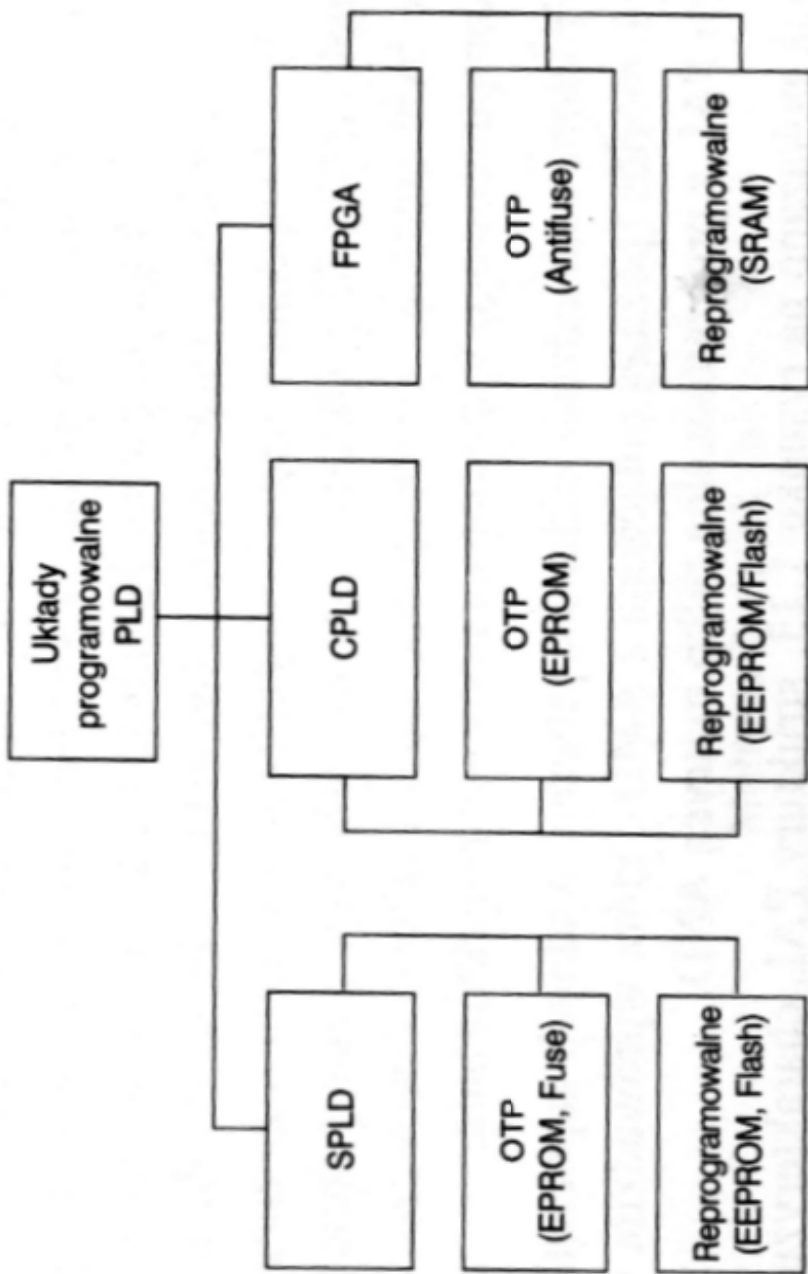
**3. Prezentacja systemu uruchomieniowego bazującego na układzie CPLD z rodziny MAXII firmy ALTERA**

- ogólna charakterystyka systemu uruchomieniowego,
- omówienie poszczególnych komponentów,
- prezentacja architektury systemu

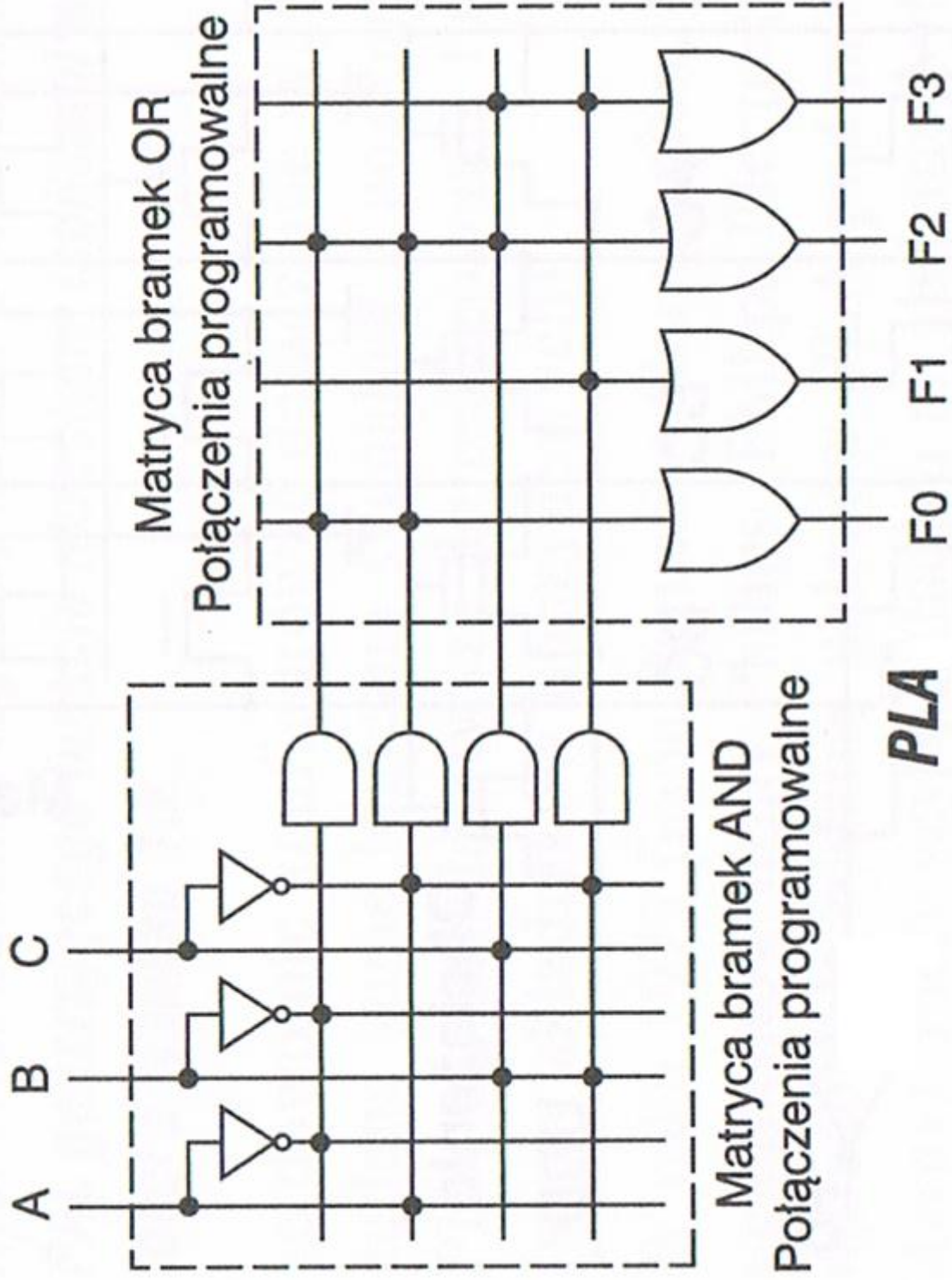
**4. Omówienie organizacji pamięci FLASH układu EPM1270F256C5**

- obszar pamięci UFM,
- obszar pamięci CFM

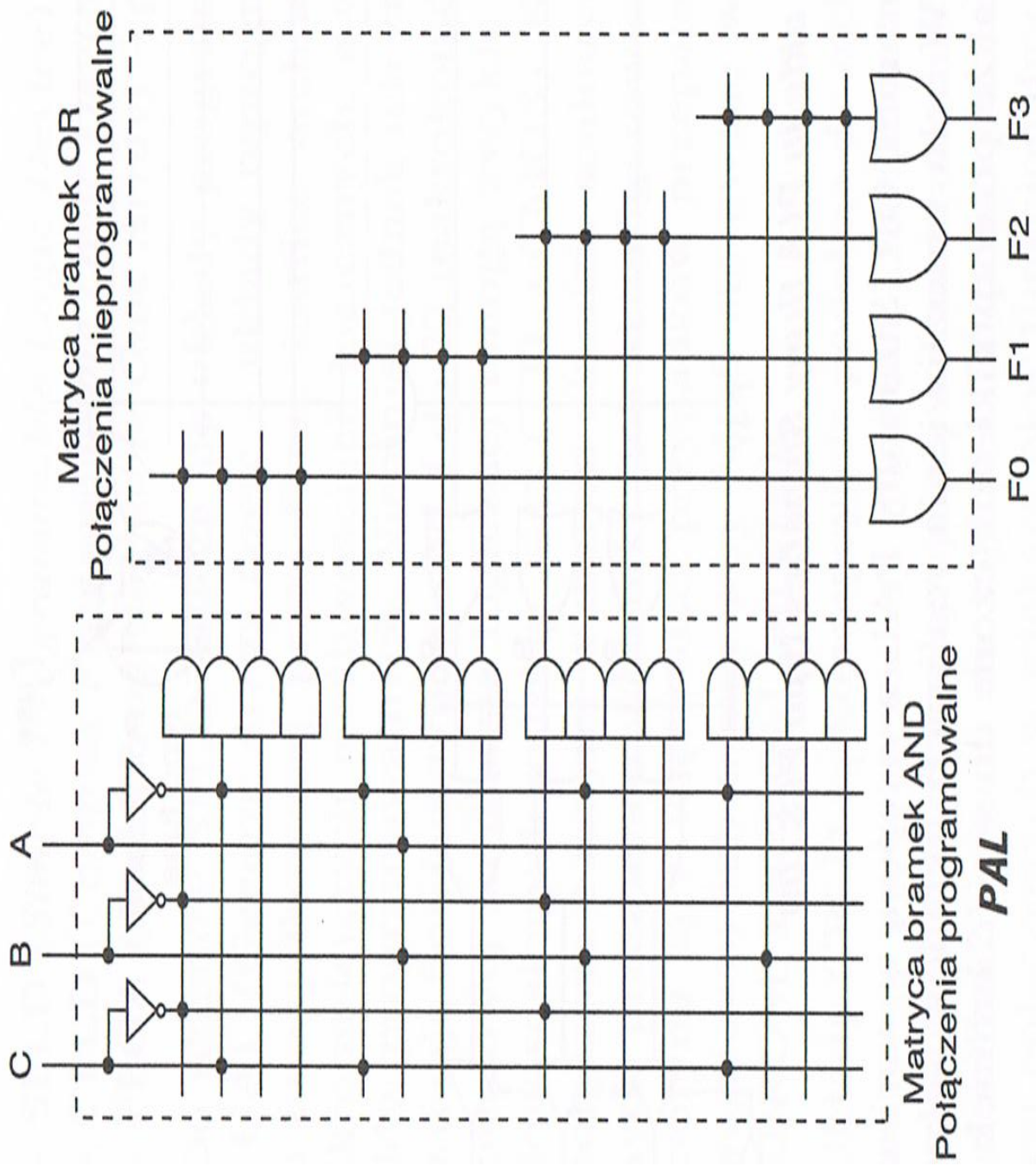
**Uwaga:** wybrane zagadnienia zostaną zaprezentowane z wykorzystaniem dokumentacji technicznej systemu uruchomieniowego firmy Altera



***Klasyfikacja układów programowalnych***

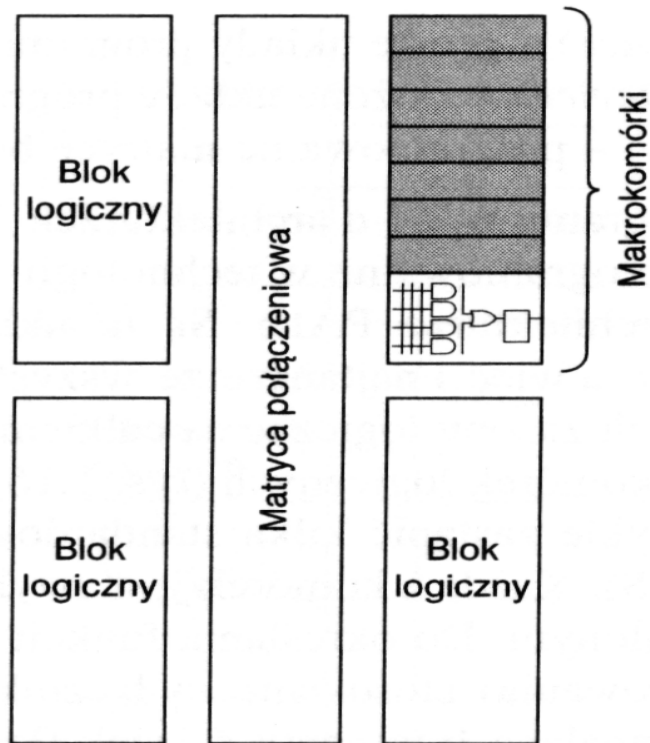


**PLA**



Matryca bramek AND  
Potężenia programowalne  
**PAL**

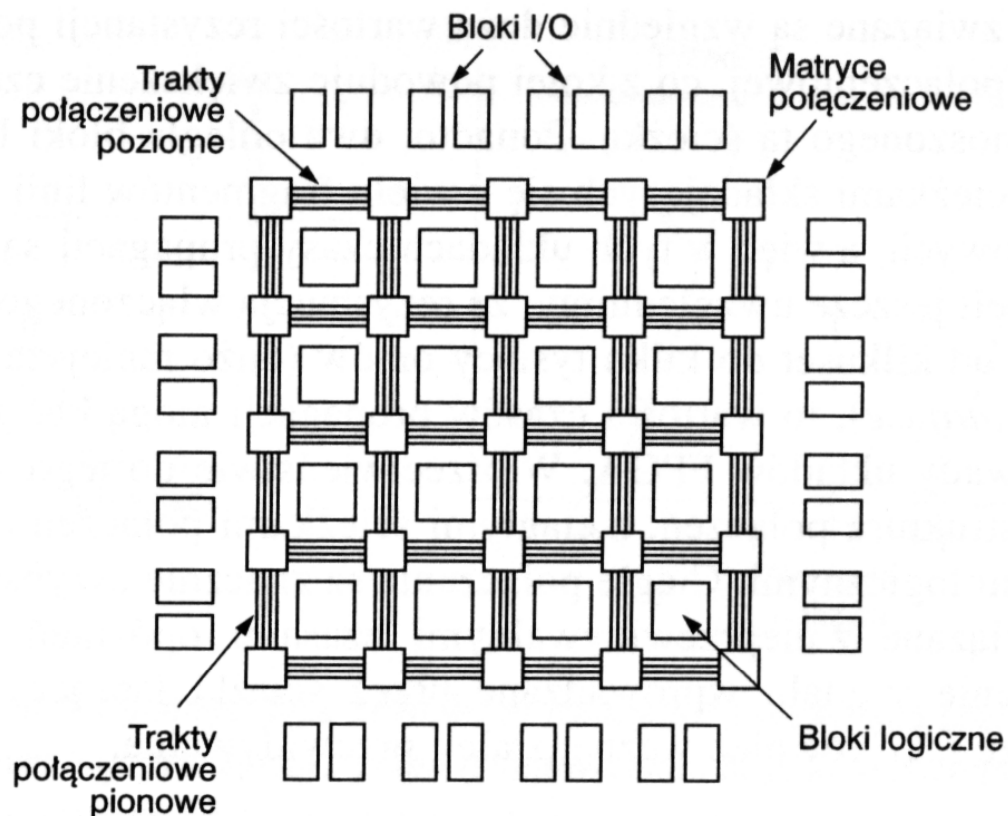
Matryca bramek OR  
Potężenia nieprogramowalne



**Rys. 1.17. Szkic typowej architektury układów CPLD**

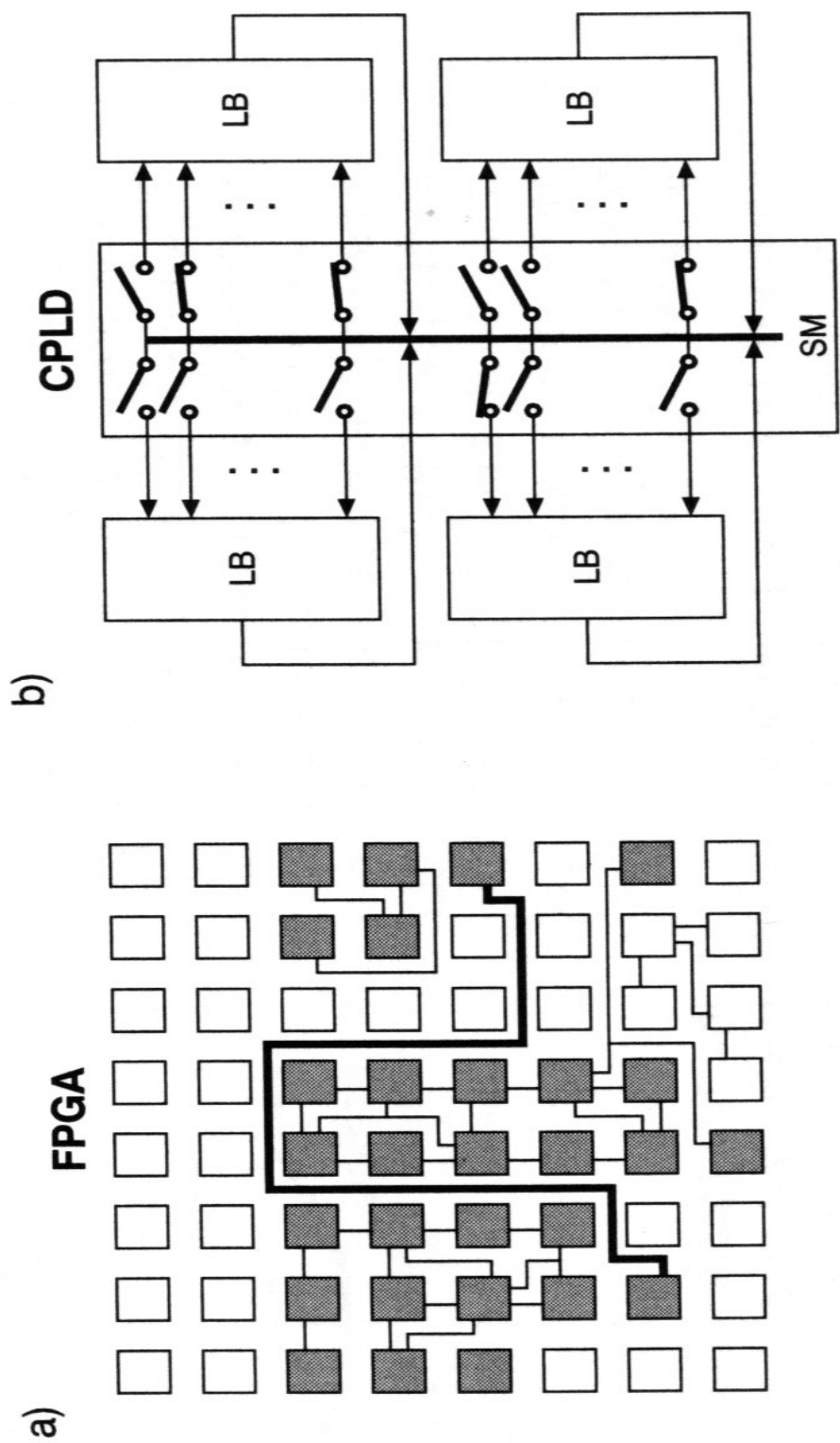
Układy CPLD są koncepcyjnie podobne do układów SPLD, lecz są bardziej złożone: mają większe zasoby logiczne i możliwości funkcjonalne. Ich architektura ma strukturę hierarchiczną opartą na makrokomórkach logicznych, których zawierają od kilkudziesięciu do kilkuset. Typowo od czterech do szesnastu makrokomórek jest połączonych w większy blok funkcjonalny (blok logiczny). Makrokomórki tworzące blok funkcjonalny są zwykle w pełni połączone. Jeżeli układ zawiera wiele bloków funkcjonalnych, wówczas muszą one być łączone między sobą. Do tego właśnie celu służy matryca połączeniowa kluczy.





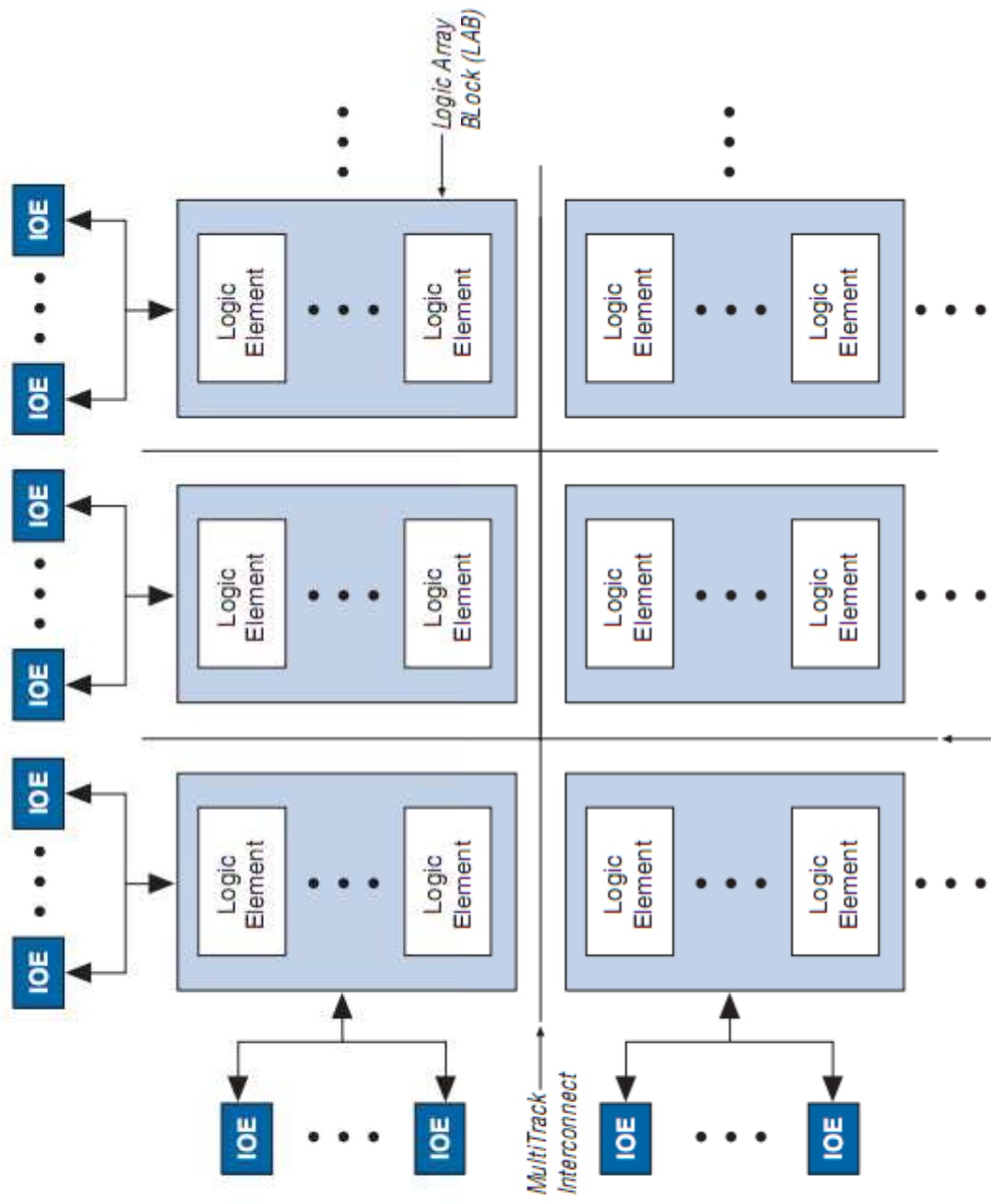
**Rys. 1.18. Szkic typowej architektury układów FPGA**

Architektura układów FPGA jest zróżnicowana, różniąc się od architektury układów CPLD. Na ogół układy FPGA zawierają rozmieszczone matrycowo bloki logiczne. Poszczególne bloki logiczne są łączone ze sobą za pośrednictwem linii traktów połączeniowych oraz programowalnych matryc kluczy połączeniowych umieszczonych w miejscu krzyżowania się traktów poziomych i pionowych. Na obrzeżu bloków logicznych znajdują się programowalne bloki I/O.



**Rys. 1.19. Ilustracja połączenia dwóch bloków logicznych w układach: a) FPGA, b) CPLD**

**Figure 2-1. MAX II Device Block Diagram**



**Figure 1. MAX II Development Board Top View**

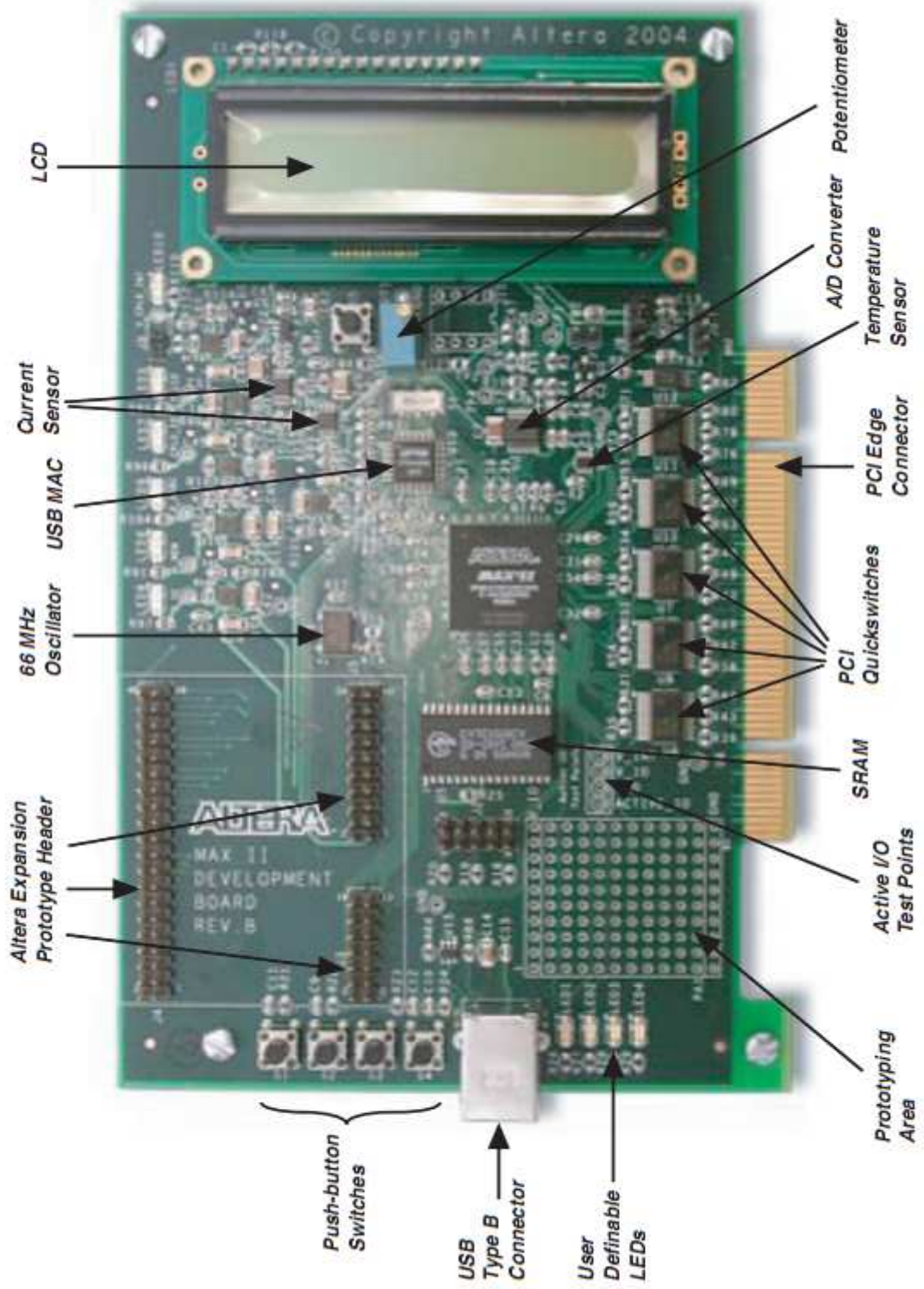
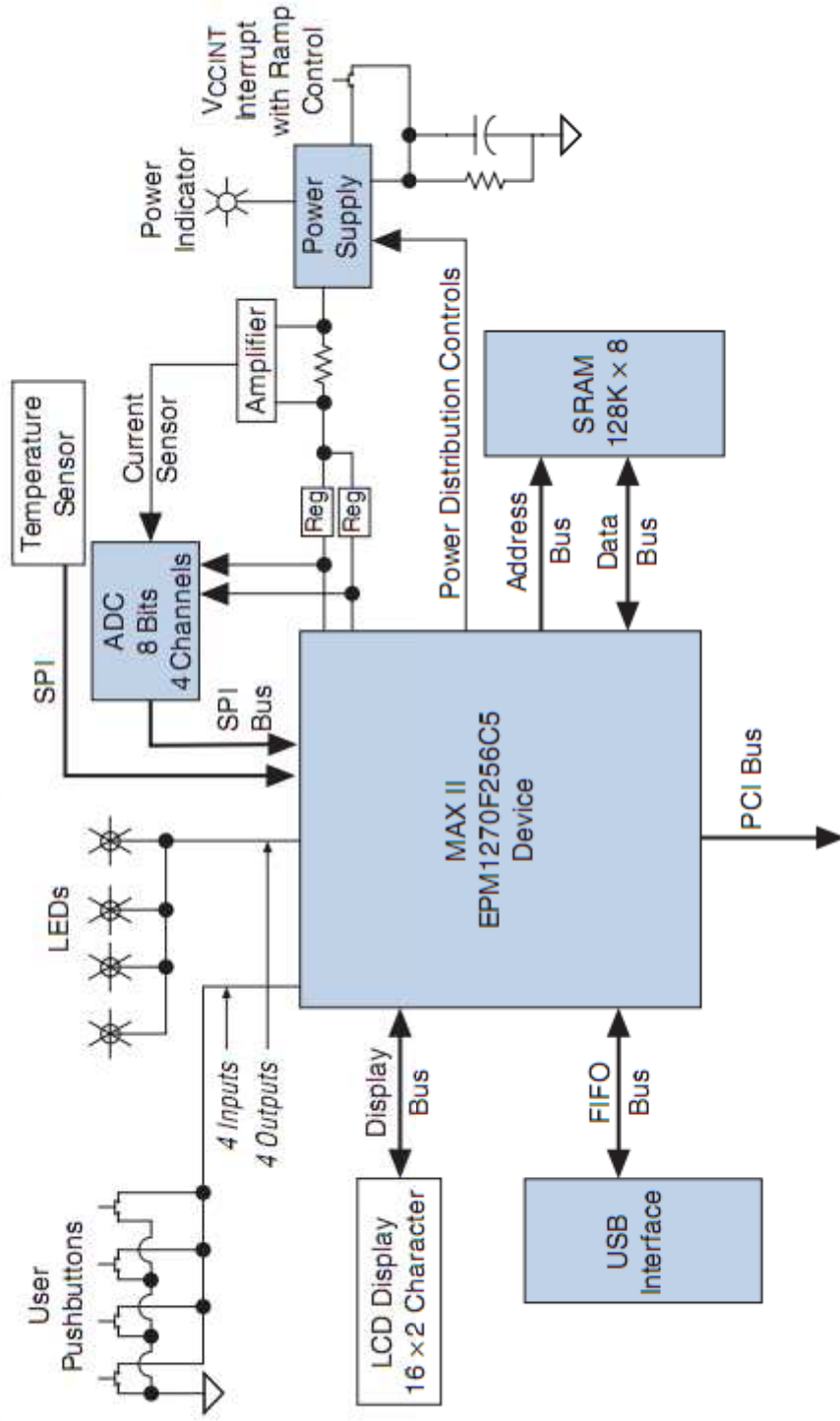


Figure 3. MAX II Development Board Block Diagram



**Figure 2-6. MAX III LE**

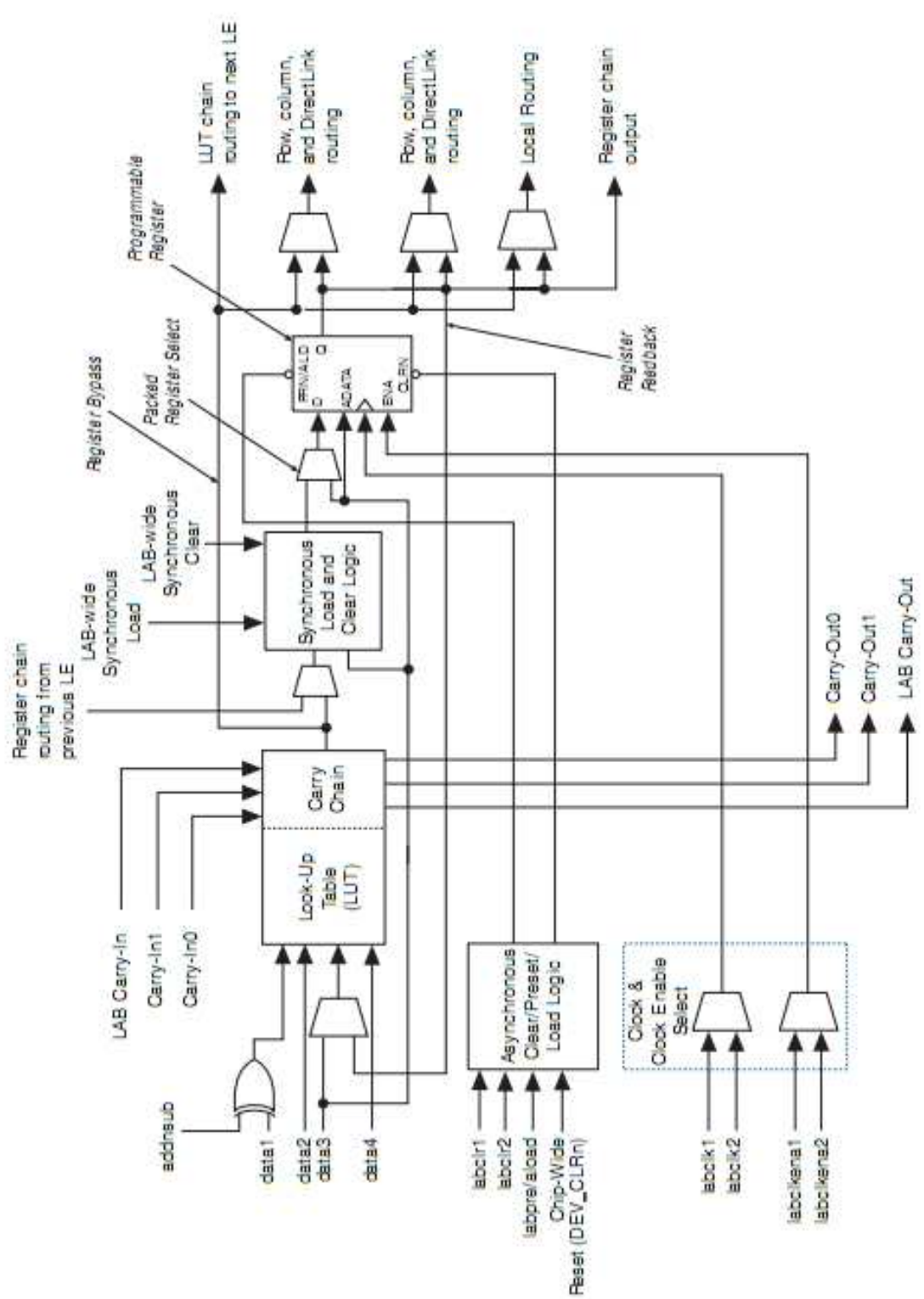
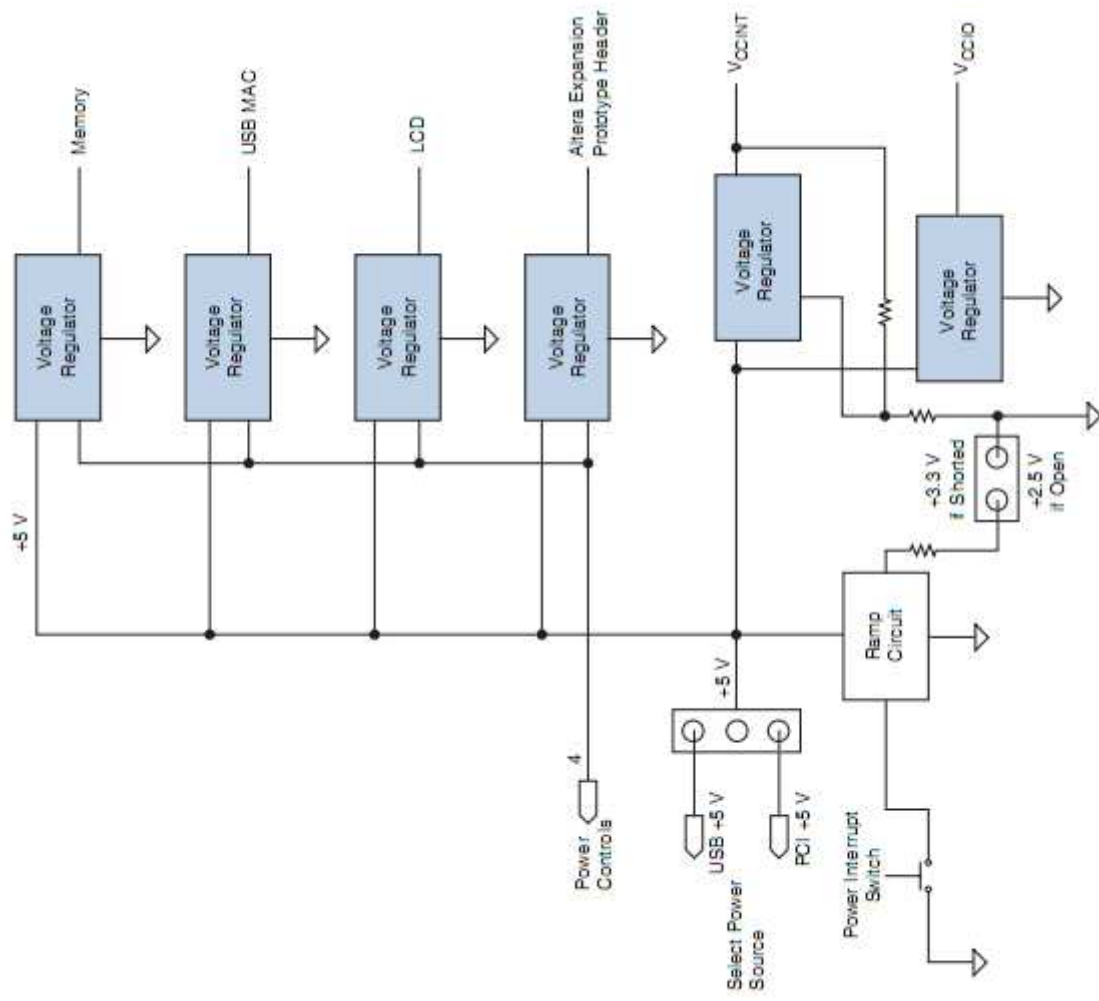
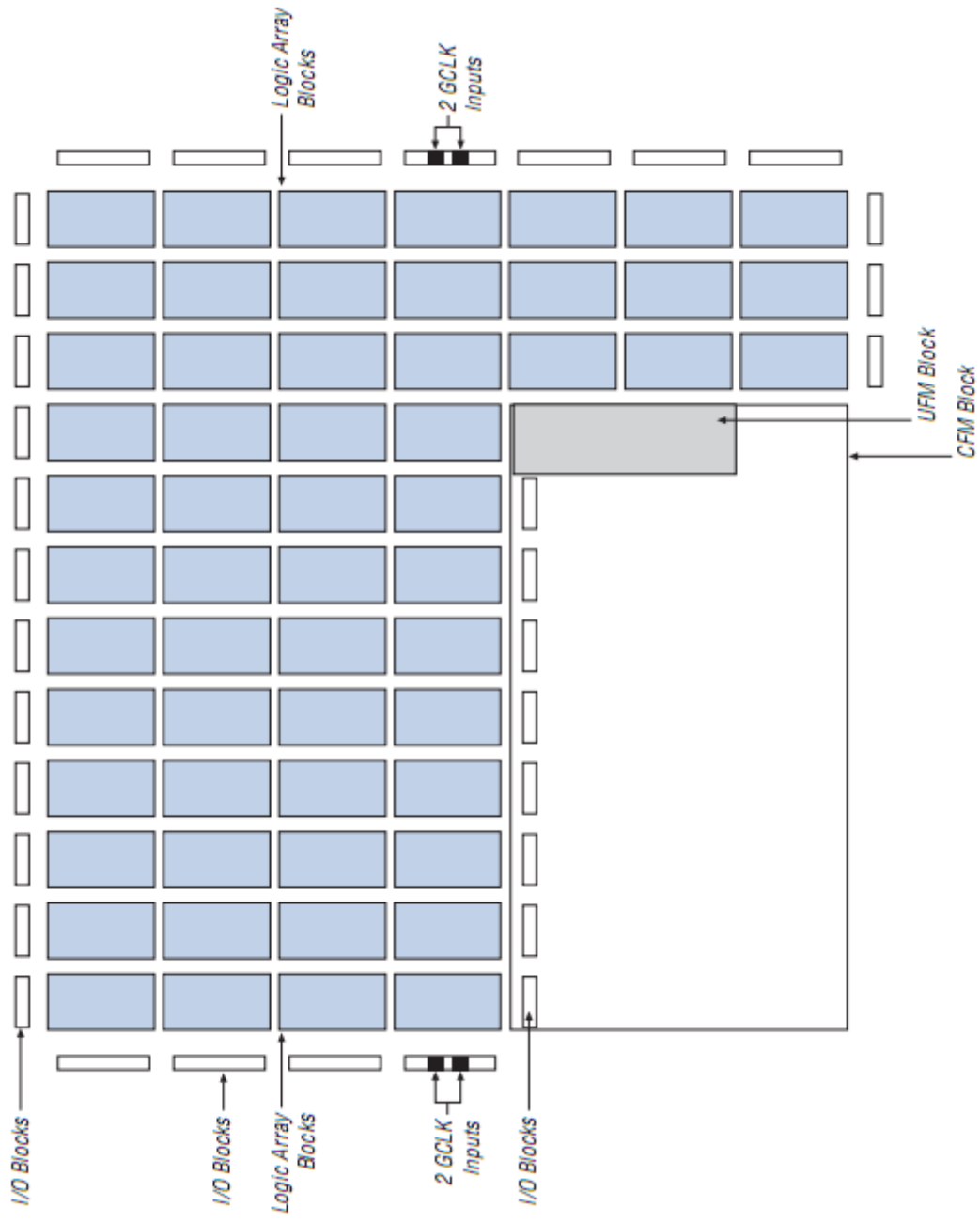


Figure 4. Power Supply Distribution Block Diagram



**Figure 2-2. MAX II Device Floorplan** *Note (1)*





**Table 9. Push-Button Switch Pin-Outs**

Signal Name	Description	MAX II Pin
S1	Reset, DEV_CLRn	M9
S2	User defined	R14
S3	User defined	T15
S4	User defined	R16
S5	Power interrupt	None

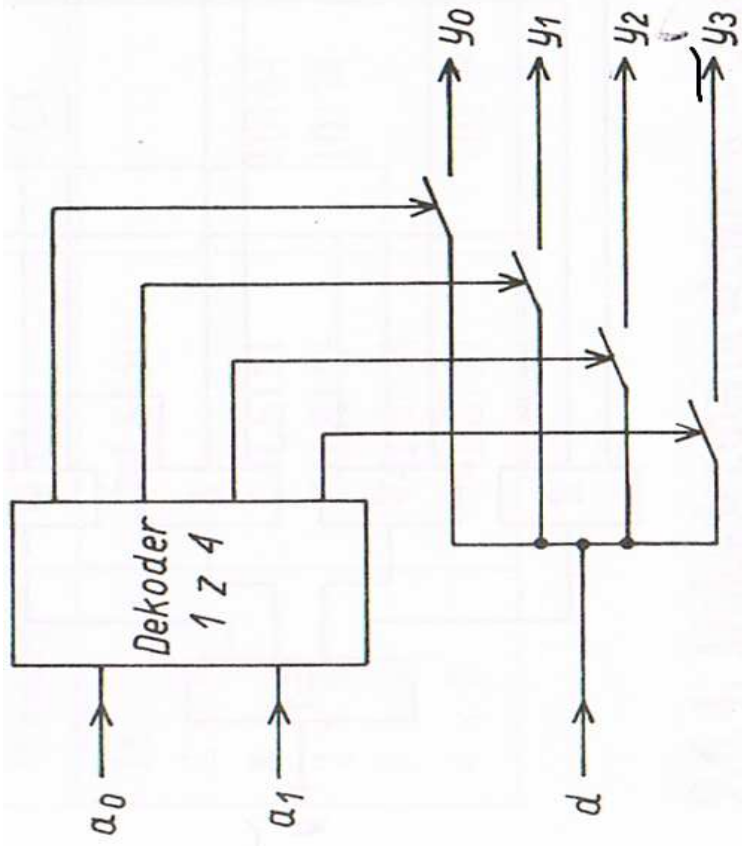
**Table 11. LED Pin-Outs**

Signal Name	Description	MAX II Pin
LED1	User-defined	T13
LED2	User-defined	R13
LED3	User-defined	P13
LED4	User-defined	T12
LED5	LCD power indicator	N/A
LED6	SRAM power indicator	N/A
LED7	V <sub>CCIO</sub> power indicator	N/A
LED8	Altera Expansion Prototype Header power indicator	N/A
LED9	USB power indicator	N/A
LED10	V <sub>CCINT</sub> power indicator	N/A

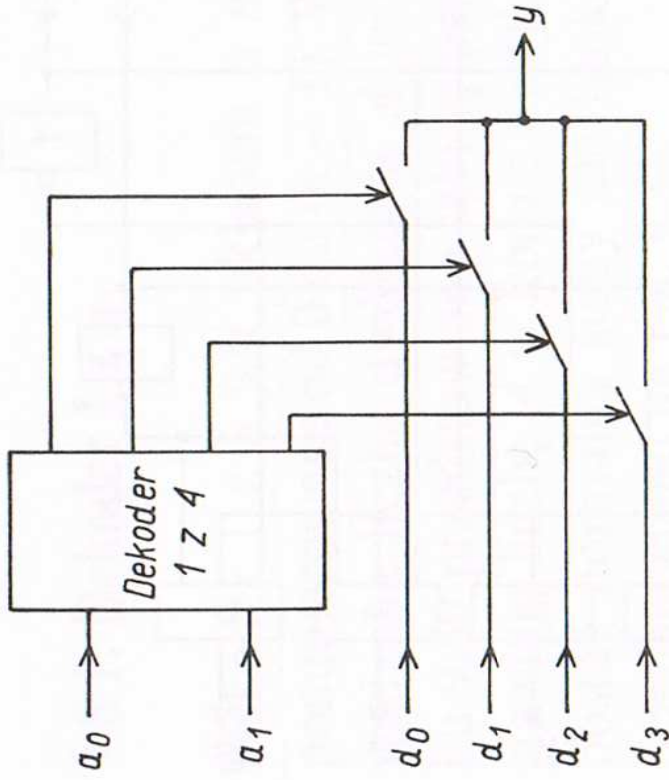
# Wybrane zagadnienia dotyczące tematyki elektroniki cyfrowej

## Plan wykładu:

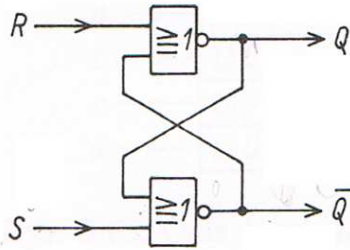
1. Pojęcia demultipleksera oraz multipleksera
2. Układy sekwencyjne: przerzutniki proste i złożone
3. Realizacja układów licznikowych:
  - liczniki asynchroniczne,
  - liczniki synchroniczne
4. Zasada działania rejestrów przesuwanych
5. Realizacja i badania prostego licznika w środowisku ORCAD
6. Środowisko programistyczne QUARTUS



Zasada działania demultiplexera



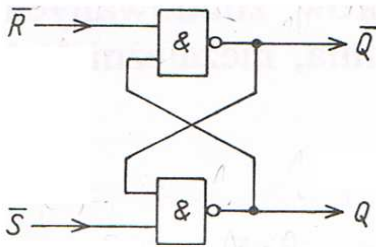
Zasada działania multiplexera



Przerzutnik RS zbudowany z bramek NOR

S	R	Q	$\bar{Q}$
0	0	$Q_{-1}$	$\bar{Q}_{-1}$
0	1	0	1
1	0	1	0
1	1	(0)	(0)

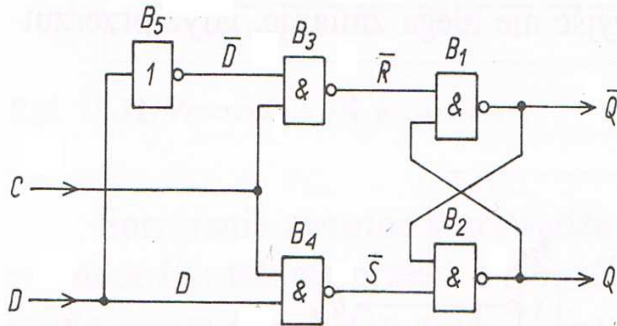
Tablica przejść przerzutnika RS



Przerzutnik RS zbudowany z bramek NAND

$\bar{S}$	$\bar{R}$	Q	$\bar{Q}$
0	0	(1)	(1)
0	1	1	0
1	0	0	1
1	1	$Q_{-1}$	$\bar{Q}_{-1}$

Tablica przejść przerzutnika RS z bramek NAND

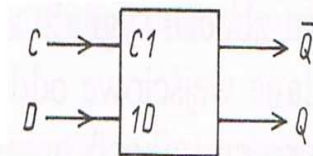


Przerzutnik D typu „zatrask”

C	D	Q
0	0	$Q_{-1}$
0	1	$Q_{-1}$
1	0	0
1	1	1

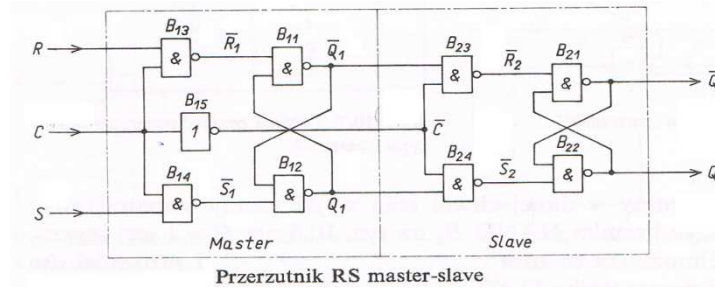
Tablica przejść przerzutnika D typu „zatrask”

## Przerzutniki proste



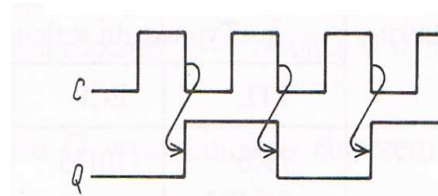
Symbol graficzny przerzutnika D typu „zatrask”

## Przerzutniki złożone

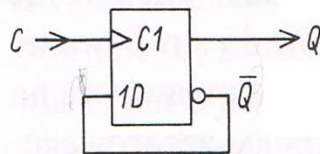


J	K	Q
0	0	$Q_{-1}$ (niezmienione)
0	1	0
1	0	1
1	1	$\bar{Q}_{-1}$ (zanegowane)

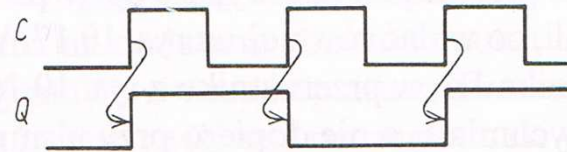
Stan wyjścia przerzutnika JK master-slave po cyklu zegara (010)



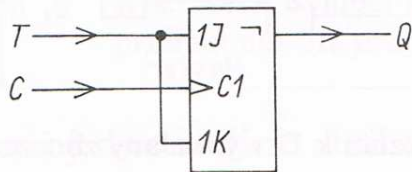
Przerzutnik JK master-slave jako dzielnik częstotliwości ( $J = K = 1$ )



Przerzutnik D wyzwalany zboczem jako dzielnik częstotliwości

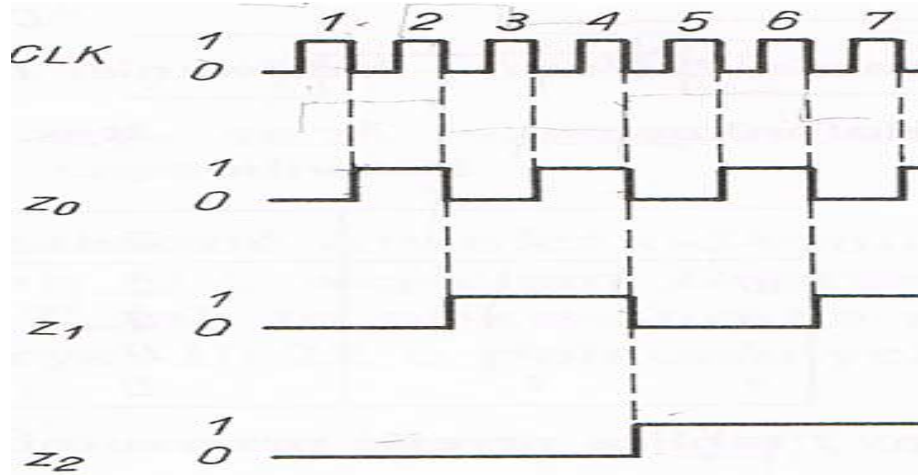


Przebiegi czasowe w dzielniku częstotliwości



$$Q = \begin{cases} Q_{-1} & \text{dla } T = 0 \\ \bar{Q}_{-1} & \text{dla } T = 1 \end{cases}$$

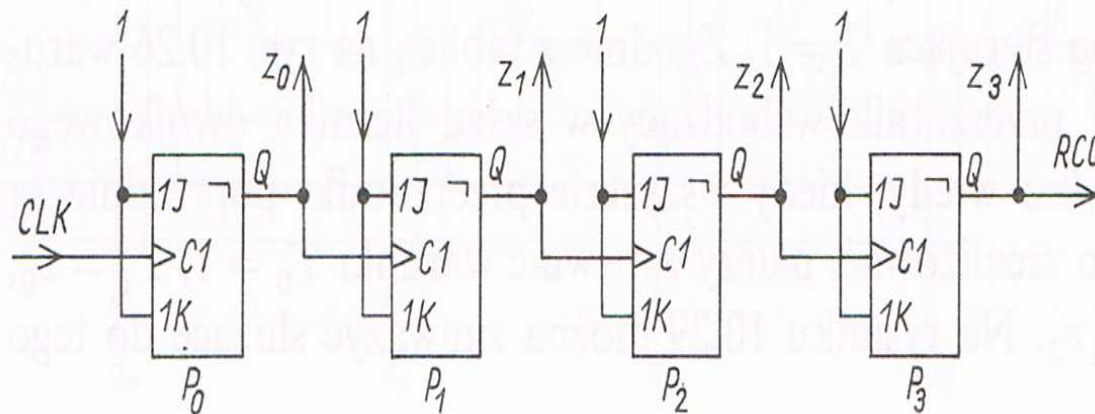
Symbol przerzutnika T



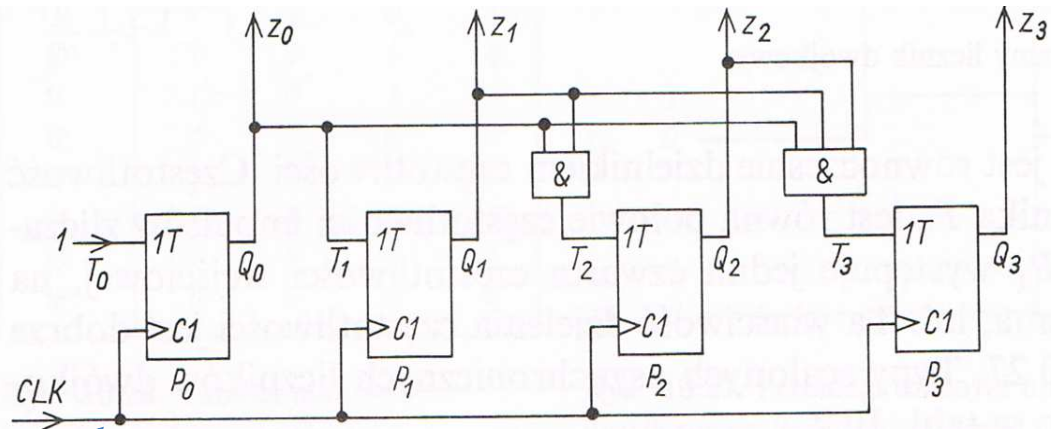
Przebiegi na wyjściach licznika dwójkowego

Z	$Z_2$	$Z_1$	$Z_0$
	$2^2$	$2^1$	$2^0$
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

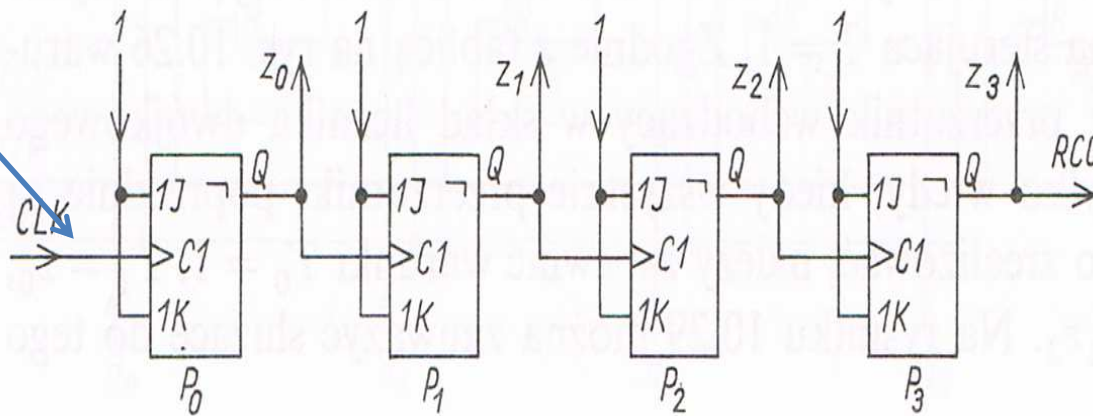
Tablica stanów licznika dwójkowego



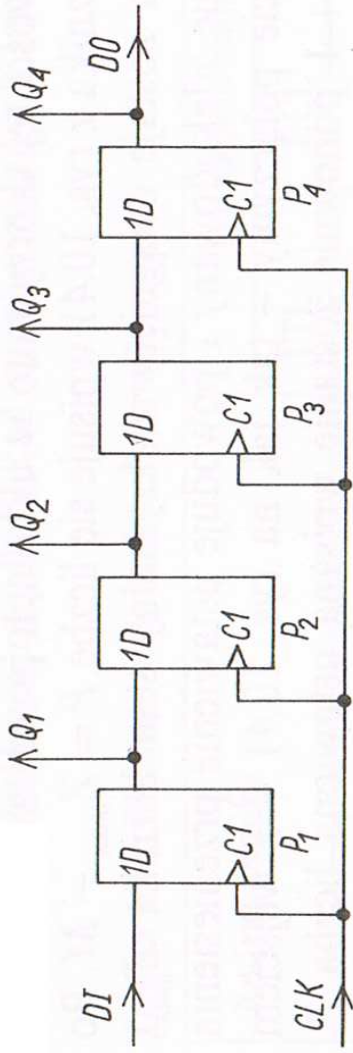
Licznik asynchroniczny



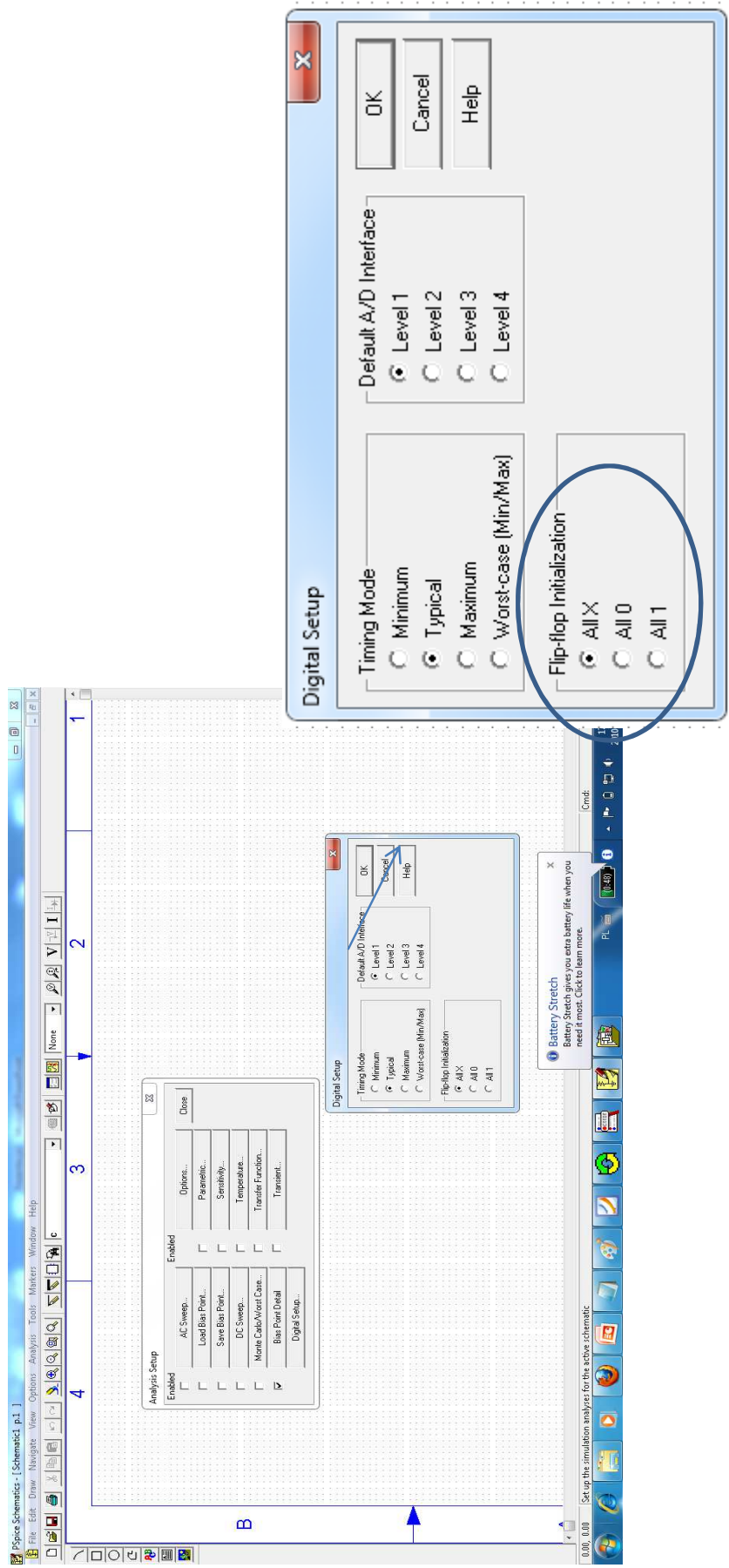
Licznik synchroniczny



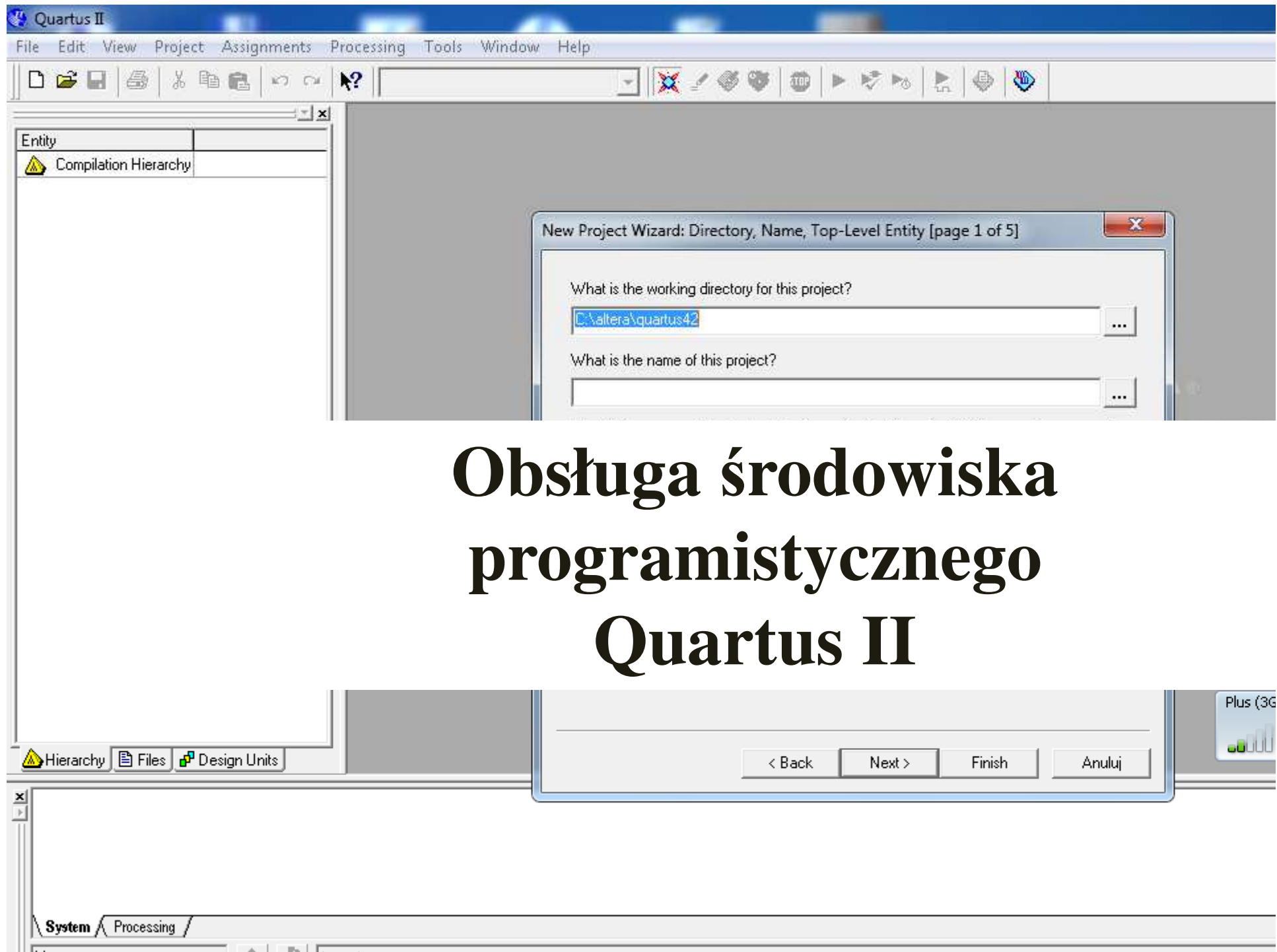
Licznik asynchroniczny



Najprostsza realizacja 4-bitowego rejestru przesuwającego







# Obsługa środowiska programistycznego Quartus II

# **Modelowanie układów cyfrowych w środowisku Orcad oraz wybrane parametry układów PLD**

## **Plan wykładu:**

### **1. Modelowanie układów sekwencyjnych w środowisku Orcad**

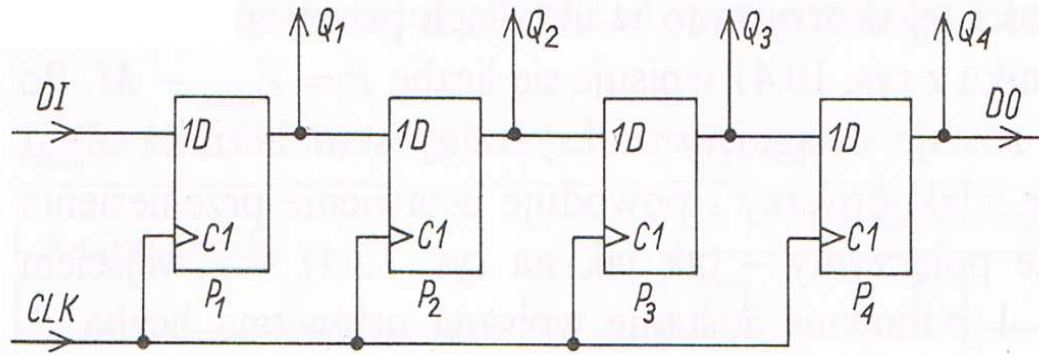
- parametry analizy Transient,**
- dodatkowe parametry dla symulacji cyfrowych,**
- realizacja prostego modelu symulacyjnego układu sekwencyjnego**

### **2. Wybrane parametry dynamiczne układów PLD**

### **3. Straty mocy w układach PLD**

### **4. Zasilanie układów PLD**

# Modelowanie układów sekwencyjnych w środowisku Orcad – przykładowa aplikacja



Najprostsza realizacja 4-bitowego rejestru przesuwającego

Analysis Setup

Enabled	AC Sweep...	Enabled	Options...	Close
<input type="checkbox"/>	Load Bias Point...	<input type="checkbox"/>	Parametric...	
<input type="checkbox"/>	Save Bias Point...	<input type="checkbox"/>	Sensitivity...	
<input type="checkbox"/>	DC Sweep...	<input type="checkbox"/>	Temperature...	
<input type="checkbox"/>	Monte Carlo/Worst Case...	<input type="checkbox"/>	Transfer Function...	
<input checked="" type="checkbox"/>	Bias Point Detail	<input type="checkbox"/>	Transient...	
<input type="checkbox"/>	Digital Setup...			

Digital Setup

Timing Mode

- Minimum
- Typical
- Maximum
- Worst-case (Min/Max)

Default A/D Interface

- Level 1
- Level 2
- Level 3
- Level 4

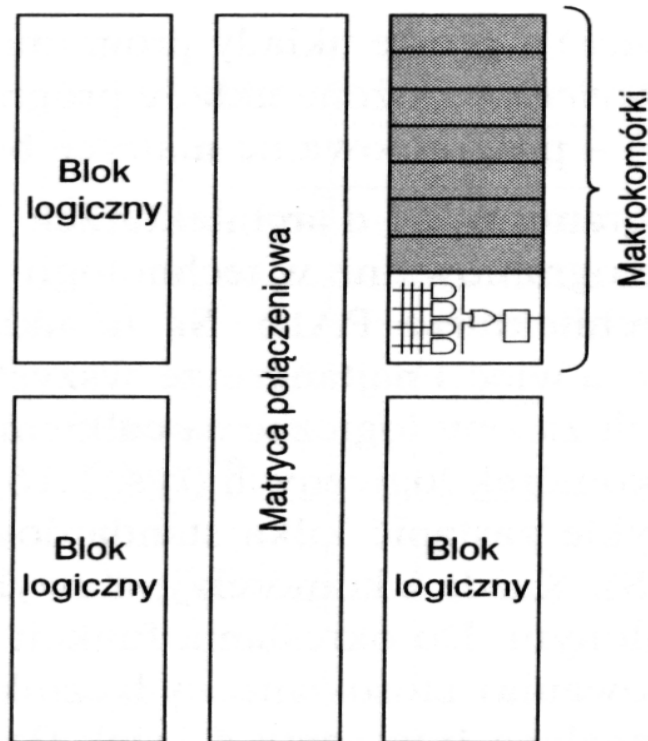
Flip-flop Initialization

- All X
- All 0
- All 1

Battery Stretch

Battery Stretch gives you extra battery life when you need it most. Click to learn more.

# Pojęcie makromórkki - przypomnienie

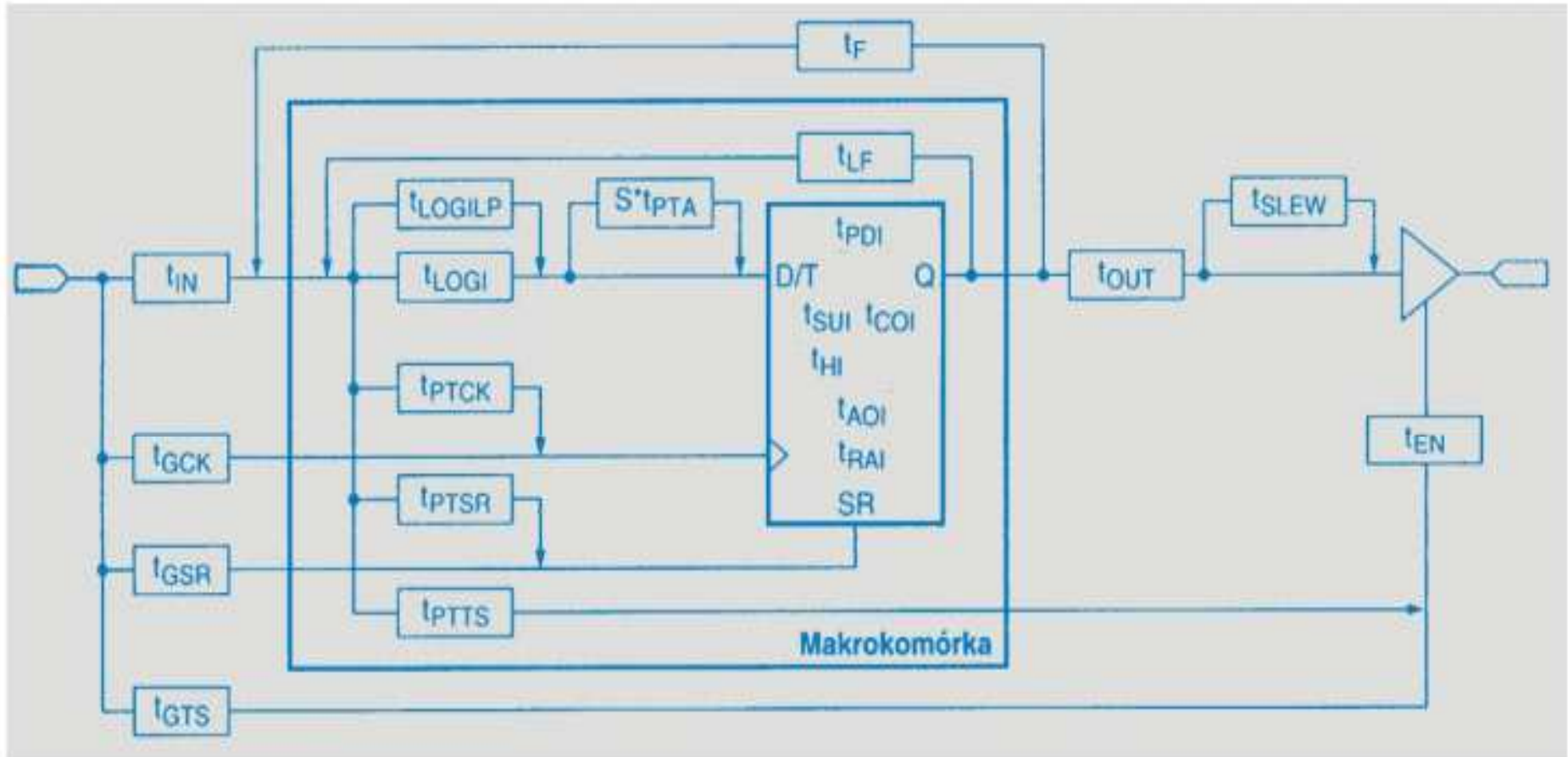


**Rys. 1.17. Szkic typowej architektury układów CPLD**

Układy CPLD są koncepcyjnie podobne do układów SPLD, lecz są bardziej złożone: mają większe zasoby logiczne i możliwości funkcjonalne. Ich architektura ma strukturę hierarchiczną opartą na makromórkkach logicznych, których zawierają od kilkudziesięciu do kilkuset. **Typowo od czterech do szesnastu makromórek jest połączonych w większy blok funkcjonalny (blok logiczny).**

**Makromórkki tworzące blok funkcjonalny są zwykle w pełni połączone.** Jeżeli układ zawiera wiele bloków funkcjonalnych, wówczas muszą one być łączone między sobą. Do tego właśnie celu służy matryca połączeniowa kluczy.

# Parametry dynamiczne układów PLD



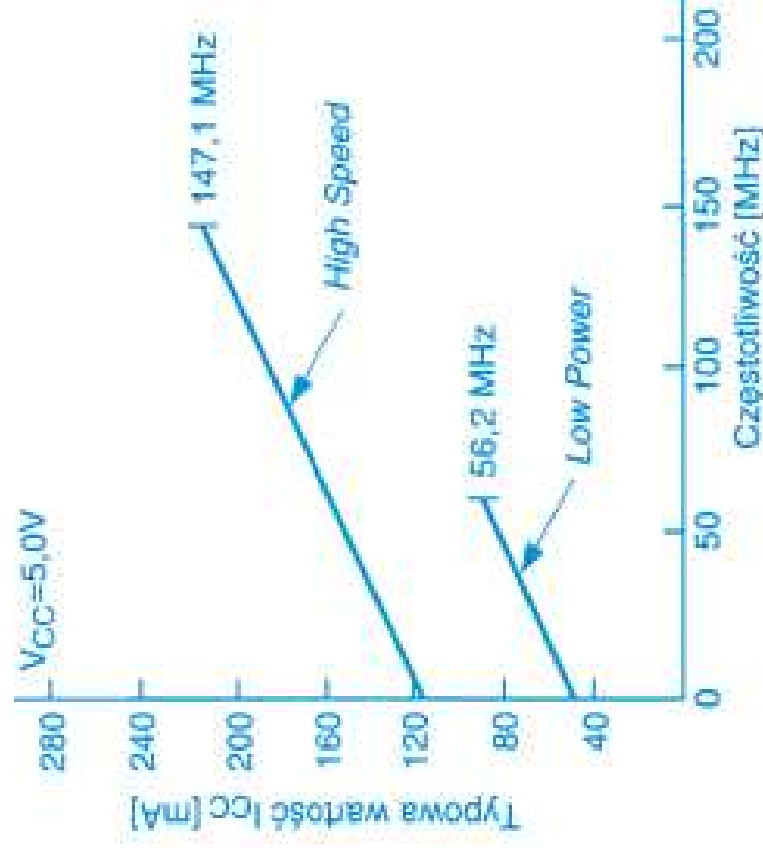
Model dynamiczny rodziny XC9500

## Parametry modelu dynamicznego (XC9500)

Symbol	Opis
$t_{IN}$	czas opóźnienia bufora wejściowego
$t_{GCK}$	czas opóźnienia bufora globalnego sygnału zegarowego rejestrów
$t_{GSR}$	czas opóźnienia bufora globalnego sygnału zerującego rejestry
$t_{GTS}$	czas opóźnienia bufora globalnego sygnału zezwolenia dla buforów wyjściowych
$t_{OUT}$	czas opóźnienia bufora wyjściowego
$t_{EN}$	czas opóźnienia w buforze trójstanowym od sygnału enable/disable do ustalenia się odpowiednich stanów na jego wyjściu
$t_{PTCK}$	czas opóźnienia „lokalnego” sygnału zegarowego (wytworzanego w macyry termów, w makromórcce) do wejścia zegarowego rejestru
$t_{PTSR}$	czas opóźnienia „lokalnego” sygnału zerującego/ustawiającego (wytworzanego w macyry termów) do wejścia SR rejestru
$t_{PTTS}$	czas opóźnienia „lokalnego” sygnału zezwolenia (wytworzanego w macyry termów) do wejścia bufora trójstanowego
$t_{PDI}$	czas opóźnienia dla układu kombinacyjnego omijającego rejestr wewnętrzny
$t_{SUI}$	czas ustalania rejestru wewnętrznego
$t_{H}$	czas przeciągania rejestru wewnętrznego
$t_{COI}$	czas opóźnienia ustalenia się danych na wyjściu rejestru względem sygnału zegarowego
$t_{AOI}$	czas opóźnienia asynchronicznego wyzerowania/ustawienia rejestru względem sygnału zerującego/ustawiającego
$t_{RAI}$	czas wyprzedzenia asynchronicznego zerowania/ustawiania rejestru przed narastającym zboczem sygnału zegarowego
$t_{LOGI}$	czas opóźnienia układów kombinacyjnych makromórci
$t_{LOGILP}$	czas opóźnienia układów kombinacyjnych makromórci dla trybu poboru małej mocy
$t_F$	czas opóźnienia sygnału zwrotnego w szybkiej macyry połączeniowej
$t_{LF}$	czas opóźnienia lokalnego sygnału sprzężenia zwrotnego w makromórcce (bloku funkcjonalnym)
$t_{PTA}$	czas opóźnienia alokatora termów umożliwiającego zwiększenie ich liczby
$t_{SLEW}$	czas opóźnienia sygnału wnoszony przez układ ogranicznika szybkości narastania impulsów (slew rate)

## Pobór mocy

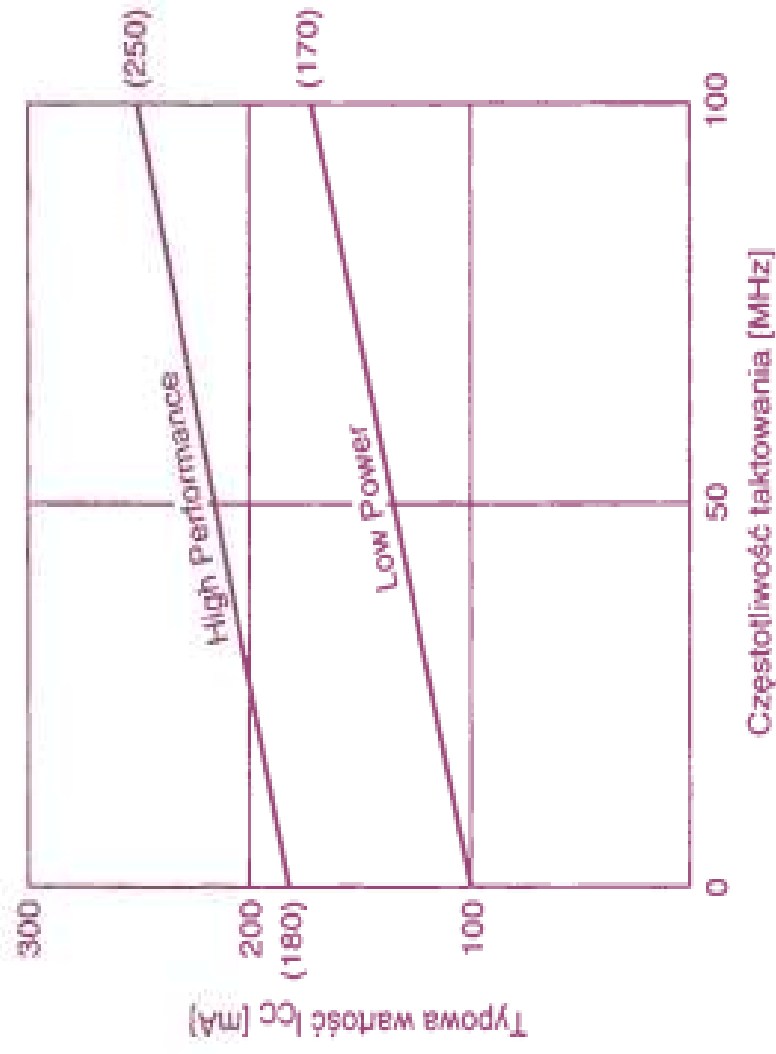
### Straty mocy w układach logicznych CPLD (MAX7000)



Zależność: prąd zasilania w funkcji częstotliwości taktowania wybranego układu CPLD

## Pobór mocy

### Straty mocy w układach logicznych CPLD (XC9500)

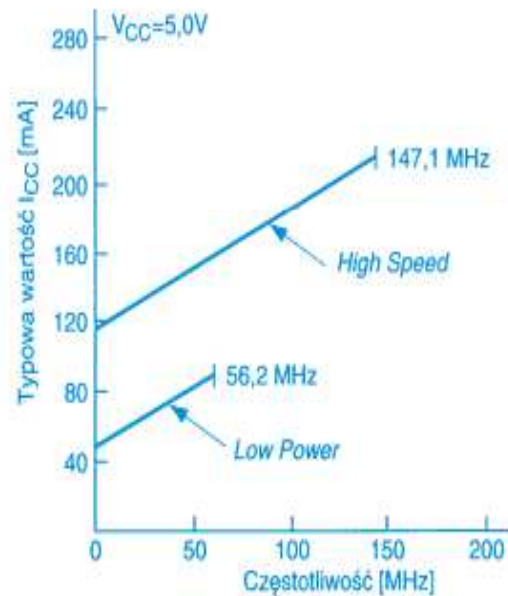




# Porównanie pobory mocy dla układów CPLD z rodziny: MAX7000 i XC9500

## Pobór mocy

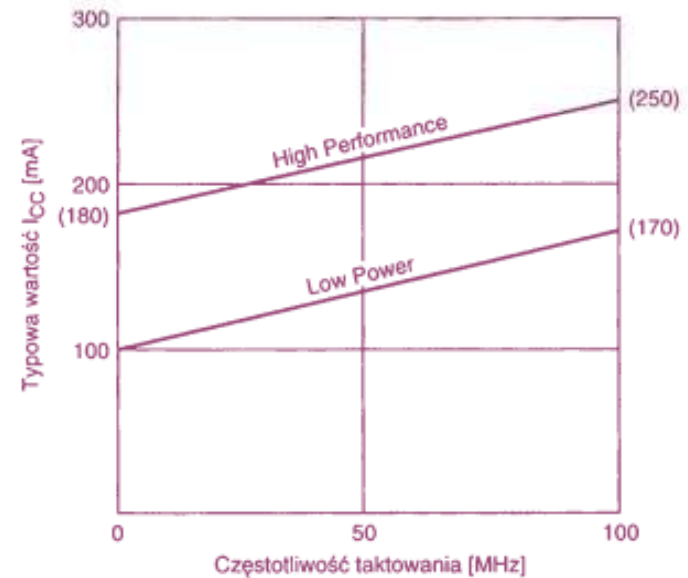
Straty mocy w układach logicznych CPLD (MAX7000)



Zależność: prąd zasilania w funkcji częstotliwości taktowania wybranego układu CPLD

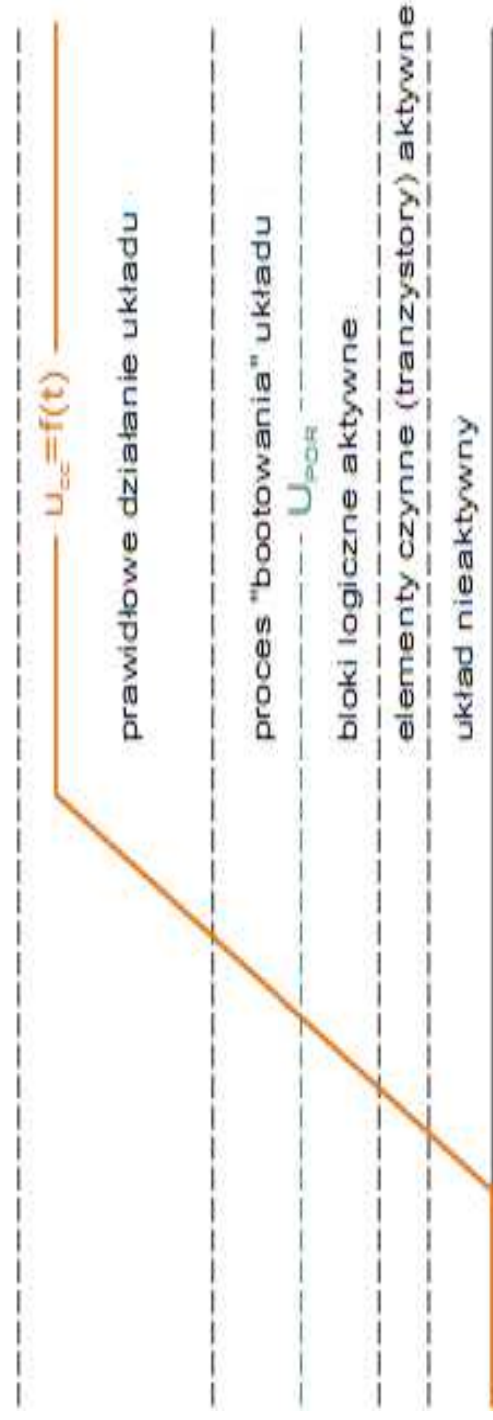
## Pobór mocy

Straty mocy w układach logicznych CPLD (XC9500)



## Zasilanie układów programowalnych

Aktywacja układów PLD - po złączeniu napięcia zasilania



**Proces aktywacji układów CPLD trwa od kilkadziesiąt do kilkuset  $\mu s$**