

# UKŁADY KOMBINACYJNE I SEKWENCYJNE

*dr inż. Grzegorz Góra*

*D3-4.02*

*ggora@agh.edu.pl*

*<http://home.agh.edu.pl/~ggora/>*

Katedra Robotyki i Mechatroniki

Wydział Inżynierii Mechanicznej i Robotyki

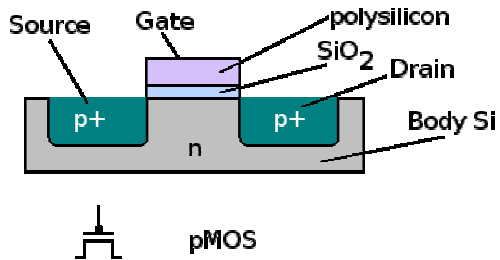
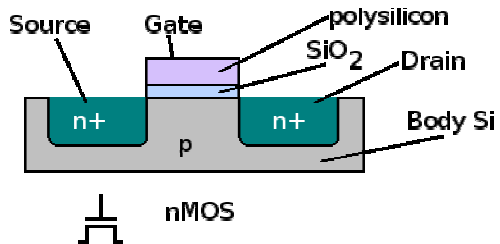
Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie



# UKŁADY KOMBINACYJNE I SEKWENCYJNE

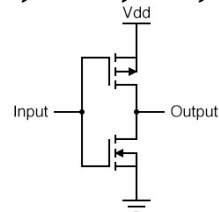
## PROJEKTOWANIE UKŁADÓW KOMBINACYJNYCH I SEKWENCYJNYCH

### TRANZYSTORY (np. N-MOS i P-MOS)

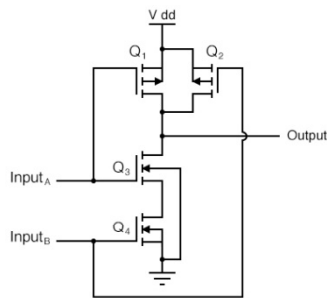


### BRAMKI I PRZERZUTNIKI (np. NOT, AND, OR, itd.)

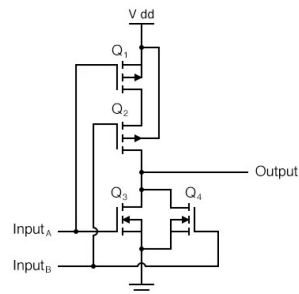
NOT



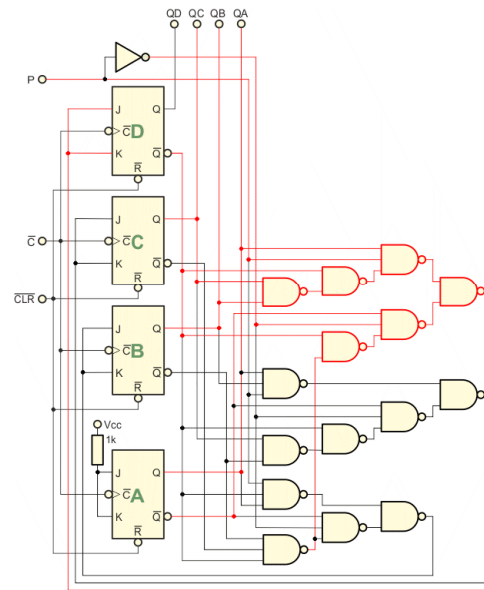
NAND



NOR

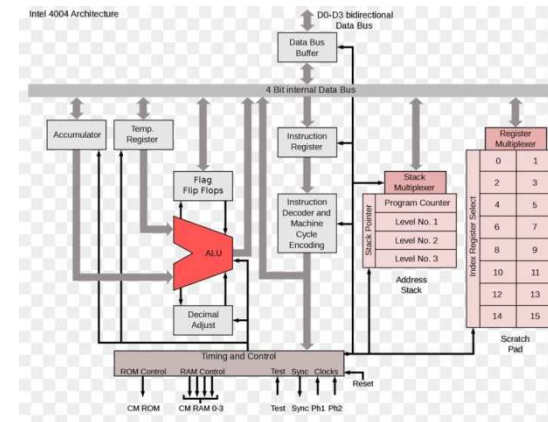


### UKŁADY KOMBINACYJNE I SEKWENCYJNE (np. liczniki, sumatory, multipleksery, transkodery, automaty itd.)



Licznik mod 5

### ZŁOŻONE UKŁADY CYFROWE (np. ALU, liczniki rozkazów, kontrolery pamięci, itd.)



# UKŁADY KOMBINACYJNE I SEKWENCYJNE

Po co uczyć się projektowania układów kombinacyjnych i sekwencyjnych?

## UKŁADY FPGA, CPLD oraz SoC



Znajomość projektowania oraz implementacji układów kombinacyjnych i sekwencyjnych jest **podstawową wiedzą** dotyczącą implementacji algorytmów w układach reprogramowalnych typu **FPGA/CPLD** oraz hybrydowych **SoC (FPGA + procesor)**.

## ELEKTRONIKA



- Proste operacje na sygnałach przy wykorzystaniu elementów 74HC (np. zmiana A/B na Step/Dir, proste układy zabezpieczeń).



- Znajomość budowy układów peryferyjnych mikrokontrolera (programowanie niskopoziomowe mikrokontrolerów).
- Czytanie i rysowanie schematów dotyczących architektury i działania układów cyfrowych (licznik, multiplexer, komparator, dekodery, sumatory, itd.).

## POZOSTAŁE



- Znajomość działania układów cyfrowych, w szczególności procesora/mikroprocesora.
- Działanie wyświetlacza 7-segmentowego, modulacja PWM, sterowanie silnikiem DC, sterowanie silnikiem krokowym.
- Rozwijanie umiejętności: logicznego myślenia, synteza informacji za pomocą aparatu matematycznego.

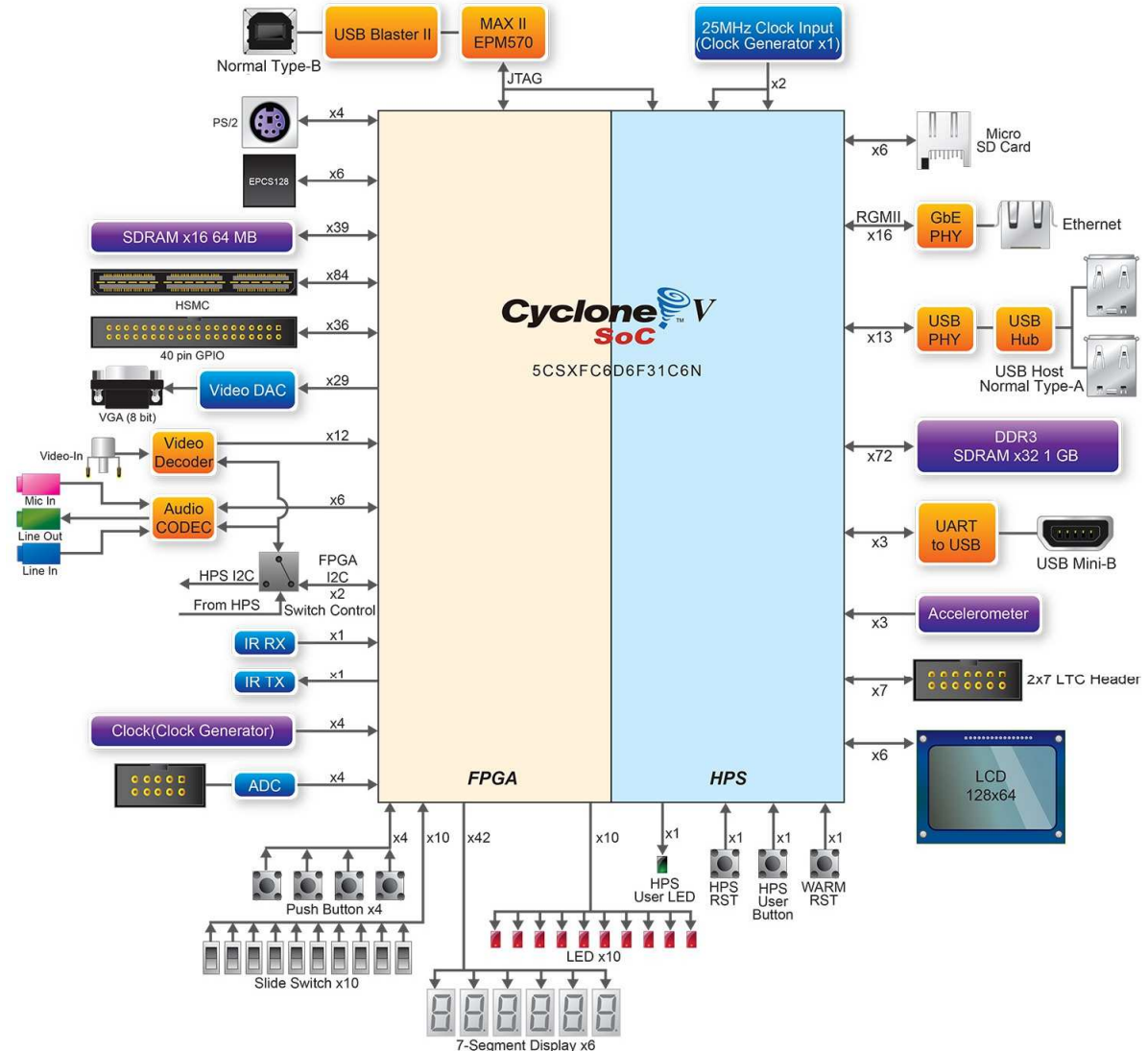
# UKŁADY KOMBINACYJNE I SEKWENCYJNE

Po co uczyć się projektowania układów kombinacyjnych i sekwencyjnych?

## UKŁADY FPGA oraz SoC



Znajomość projektowania oraz implementacji układów kombinacyjnych i sekwencyjnych jest **podstawową wiedzą** dotyczącą implementacji algorytmów w układach reprogramowalnych typu **FPGA/CPLD** oraz hybrydowych **SoC (FPGA + procesor)**.



# UKŁADY KOMBINACYJNE I SEKWENCYJNE

Po co uczyć się projektowania układów kombinacyjnych i sekwencyjnych?

## ELEKTRONIKA

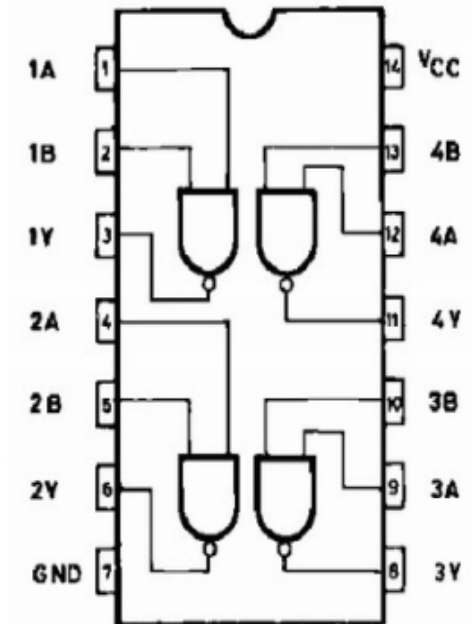


- Proste operacje na sygnałach przy wykorzystaniu elementów 74HC (np. zmiana A/B na Step/Dir, proste układy zabezpieczeń).



- Znajomość budowy układów peryferyjnych mikrokontrolera (programowanie niskopoziomowe mikrokontrolerów).
- Czytanie i rysowanie schematów dotyczących architektury i działania układów cyfrowych (licznik, multiplekser, komparator, dekodery, sumator, itd.).

Part No.	Description
74HC00	74HC00 Quad 2-input NAND Gate
74HC02	74HC02 Quad 2-input NOR Gate
74HC04	74HC04 Hex Inverter
74HC08	74HC08 Quad 2-input AND Gate
74HC10	74HC10 Triple 3-input NAND Gate
74HC11	74HC11 Triple 3-input AND Gate
74HC14	74HC14 Hex Inverter Schmitt Trigger
74HC20	74HC20 Dual 4-input NAND Gate
74HC30	74HC30 8-input NAND Gate
74HC32	74HC32 Quad 2-input OR Gate
74HC74	74HC74 Dual D Flip-Flop
74HC75	74HC75 Quad BiStable Transparent Latch
74HC85	74HC85 4-bit Magnitude Comparator
74HC86	74HC86 Quad EXCLUSIVE-OR Gate
74HC107	74HC107 Dual JK Flip-Flop with Reset
74HC123	74HC123 Dual Mono Multivibrator
74HC125	74HC125 Quad Bus Buffer Tri-State
74HC126	74HC126 Quad Buffer/Line Driver, 3-State
74HC132	74HC132 Quad 2-input NAND Schmitt Trigger



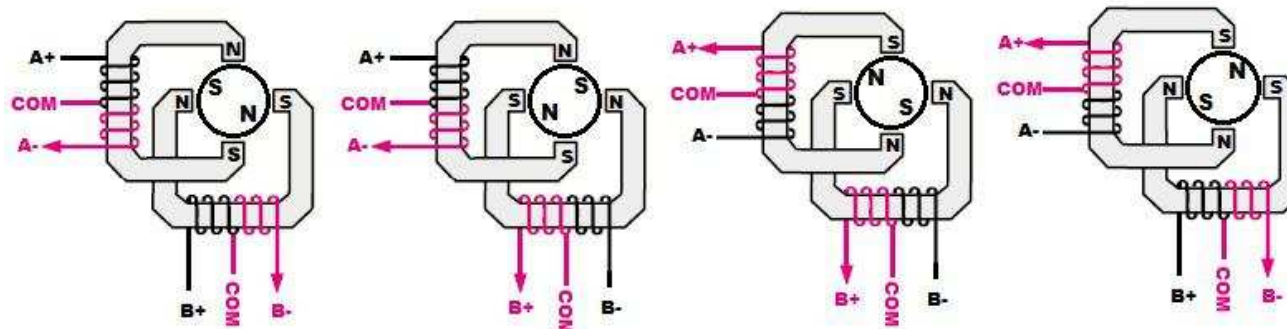
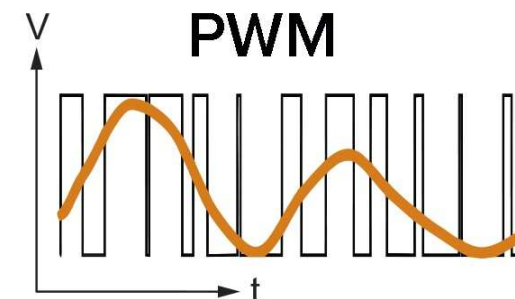
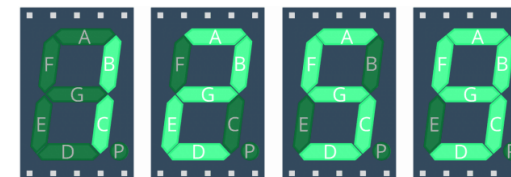
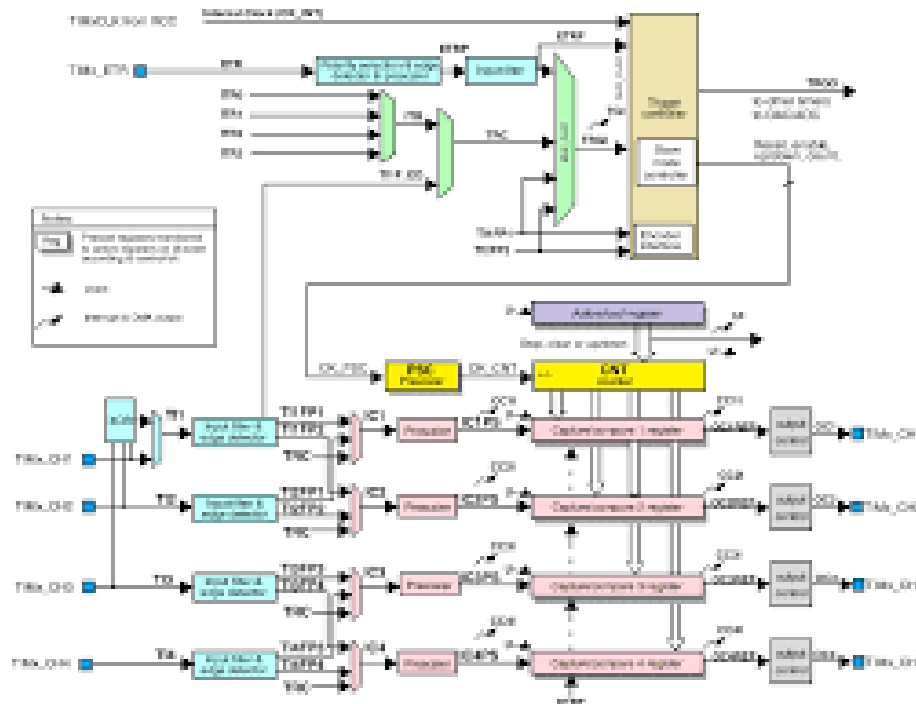
# UKŁADY KOMBINACYJNE I SEKWENCYJNE

Po co uczyć się projektowania układów kombinacyjnych i sekwencyjnych?

## POZOSTAŁE



- Znajomość działania układów cyfrowych, w szczególności układów peryferyjnych procesora / mikroprocesora.
- Działanie wyświetlacza 7-segmentowego, modulacji PWM, sterowanie silnikiem DC, sterowanie silnikiem krokowym.
- Rozwijanie umiejętności: logicznego myślenia, synteza informacji za pomocą aparatu matematycznego.



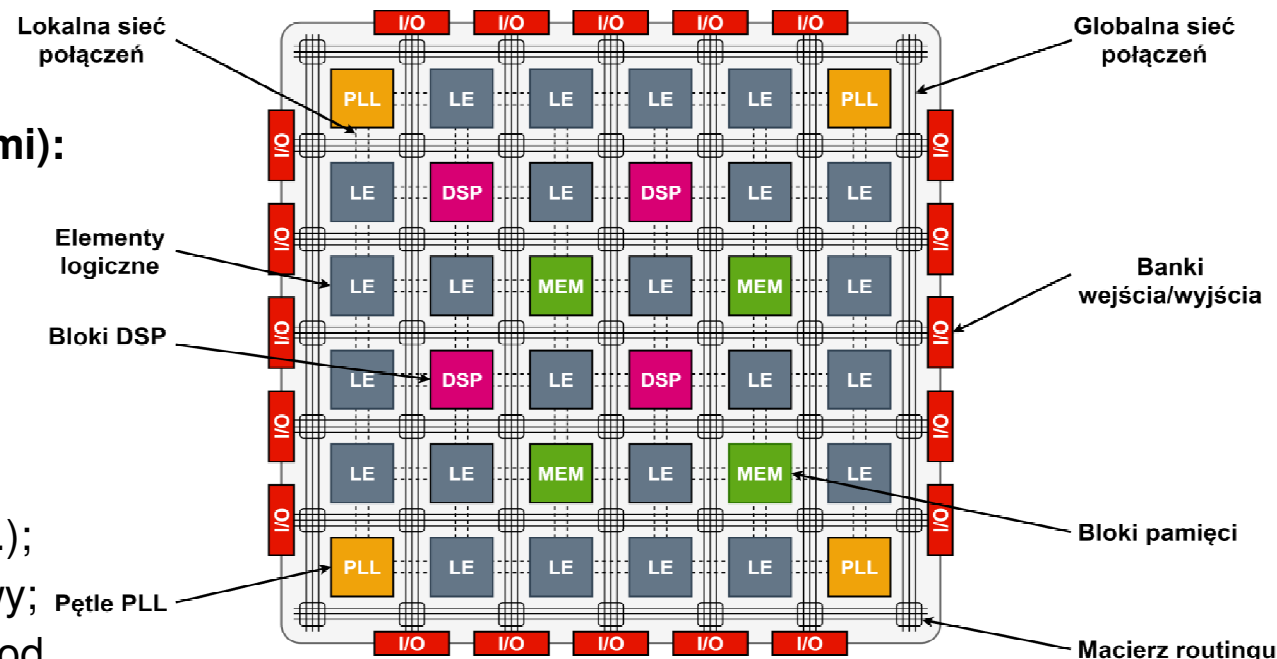
# UKŁADY FPGA (ang. Field Programmable Gate Arrays)

## Budowa układu FPGA:

- **komórki LEs** (ang. *Logic Elements*);
- **sieć połączeń** – zapewnia komunikację lokalną z sąsiednimi LEs oraz komunikację w obrębie całego układu FPGA (tzw. połączenia globalne o różnym zasięgu);
- **elementy dodatkowe** (np. pamięć, mnożarki, pętle PLL, ADCs, itd.);
- **inne** (układ zasilania, porty wejścia/wyjścia).

## Różnice pomiędzy układami FPGA (rodzinami):

- ilość elementów logicznych (ilość LEs);
- poziom złożoności LEs;
- szybkość układu (maksymalna częstotliwość pracy, wyrażona jako czas propagacji sygnału przez LEs od 1-9 ns);
- ilość i rodzaj wbudowanych elementów dodatkowych (pamięć, mnożarki, pętle PLL, itd.);
- ilość portów wejścia/wyjścia – rodzaj obudowy;
- wersja (commercial / industrial - zależna np. od zakresu temperatury pracy).



## **FPGA vs MCU: Podobieństwa i różnice**

### **MCU**



### **FPGA**



### **Różnice**

- Stała architektura narzucona przez producenta
- Język programowania wykorzystywany jako sekwencyjnie wykonywane polecenia
- Wykonywanie wielu zadań realizowane poprzez przełączanie się pomiędzy nimi
- Porty dedykowane, niewielka możliwość zmiany funkcjonalności : np. PWM, I2C, SPI
- Wbudowane przetworniki ADC i DAC

- Zmienna architektura konfigurowana przez użytkownika
- Język programowania wykorzystywany do wewnętrznej konfigurowania układu
- Zadania wykonywane jednocześnie i równolegle
- Porty ogólnego przeznaczenia dowolnie konfigurowane
- Najczęściej bez wbudowanych przetworników ADC i DAC

### **Podobieństwa**

- Układ cyfrowy w formie układu scalonego
- Przeznaczony do realizacja zaimplementowanego algorytmu
- Programowany z wykorzystaniem dedykowanego języka

## **Firmy wykorzystujące układy FPGA/SoC**

Firmy wykorzystujące technologie/umiejętności dotyczące FPGA/CPLD/SoC lub VHDL/Verilog działające w Krakowie i okolicach:



ARROW ELECTRONICS, INC.



**WOODWARD**



**Honeywell**



**HITACHI**



**cādence**

**NOKIA**

# PRZYGOTOWANIE DO ZAJĘĆ

***W instrukcji każdego zadania wymienione są wymagania (zakres wiedzy i umiejętności) konieczne do prawidłowego wykonania ćwiczenia.***

**Mechatroniczne Systemy Wykonawcze Sensoryczne i Sterujące**

*Lab: Transkoder do wyświetlacza 7-segmentowego*



## ***Transkoder do wyświetlacza 7-segmentowego***

Wymagania do realizacji ćwiczenia (zakres zagadnień i umiejętności koniecznych do wykonania zadania):

- reprezentacja liczb w systemie dziesiętnym, binarnym i szesnastkowym;
- zakładanie projektu w programie Quartus Prime;
- tworzenie modułu sprzętowego (symbolu) w programie Quartus Prime na podstawie pliku schematu (\*.bdf);
- tworzenie modułu sprzętowego (symbolu) w programie Quartus Prime na podstawie pliku z kodem źródłowym (np. \*.vhd);
- umiejętność upraszczania wyrażeń logicznych metodą tablic Karnaugh;
- umiejętność implementacji schematu z bramek logicznych na podstawie równania algebraicznego;

## PLAN ZAJĘĆ

### 1. Układy kombinacyjne I

**Pro:** Ćwiczenia tablicowe z tablic Karnaugh, rozdanie zadań na sprawozdanie (czas realizacji 2 tygodnie);

**Lab:** Wprowadzenie do tematu projektowania układów kombinacyjnych i sekwencyjnych; projekt bazowy w programie Quartus Prime (założenie projektu bazowego z bramką AND, zaimplementowanie układu kombinacyjnego w formie schematu, zaimportowanie pliku VHDL z migającą diodą i jego uruchomienie);

### 2. Układy kombinacyjne II

**Lab:** Wprowadzenie do implementacji układów kombinacyjnych w FPGA:

implementacja układu kombinacyjnego do sterowania wyświetlaczem 7-segmentowym.

**Pro:** Implementacja układu kombinacyjnego multiplexera,  
implementacja sterownika świateł drogowych.

### 3. Układy sekwencyjne I

**Lab:** Implementacja licznika – podstawy;

implementacja indywidualnych zadań dotyczących liczników.

**Pro:** Implementacja regulatora obrotów silnika DC (modulator PWM).  
Implementacja sterownika silnika krokowego.

### 4. Układy sekwencyjne II

**Lab:** Implementacja automatu – projekt podstawowy;

implementacja indywidualnych zadań dotyczących automatów.

## CZĘŚĆ PROJEKTOWA ZAJĘĆ

- **Zadania projektowe:**
  - multiplexer;
  - sterownik świateł drogowych;
  - regulator obrotów silnika DC (modulator PWM);
  - sterownik silnika krokowego.
- **Ocena projektu:**
  - działanie układu „na sprzęcie”;
  - **odręczne** notatki;
  - odpowiedź na pytania dotyczące zadania (np. Jak działa multiplexer? Co to jest PWM? Jak zmienić prędkość/kierunek obrotów silnika DC/krokowego?); sugerowane pytania będą w opisach zadań.
- **Czas realizacji:**

projekt można oddać na dowolnych zajęciach poświęconych implementacji układów kombinacyjnych i sekwencyjnych.

## **PLAN ZAJĘĆ - PUNKTACJA**

- 1. Układy kombinacyjne I (2 pkt)**
  - sprawozdanie z tablic Karnaugh 2 pkt
  
- 2. Układy kombinacyjne II (8 pkt)**
  - implementacja w układzie FPGA układu komb. do wyświetlacza 7-segmentowego 4 pkt
  - implementacja w układzie FPGA multipleksera 2 pkt
  - implementacja w układzie FPGA sterownika świateł drogowych 2 pkt
  
- 3. Układy sekwencyjne I (6 pkt)**
  - implementacja w układzie FPGA przykładowego licznika 1 pkt
  - projekt oraz implementacja w układzie FPGA licznika z dodatkowym wejściem dir (zmiana kierunku liczenia – góra/dół), zadanie indywidualne dla każdego 1 pkt
  - projekt oraz implementacja regulatora obrotów silnika DC (modulacja PWM) 2 pkt
  - projekt oraz implementacja sterownika silnika krokowego 2 pkt
  
- 4. Układy sekwencyjne II (4 pkt)**
  - implementacja w układzie FPGA przykładowego automatu 2 pkt
  - projekt oraz implementacja w układzie FPGA automatu zadanie indywidualne dla każdego zespołu 2 pkt