

Jan Chyczyński  
Błażej Nowicki  
Bartłomiej Słupik  
Przemysław Węglik

# Technika cyfrowa Sprawozdanie 1

5 marca 2022

# 1. Zadanie 1a

## 1.1. Wstęp

Zadanie polega na zaprojektowaniu układu realizującego funkcję logiczną:

$$Y = \overline{A}C + B(A + B)$$

Układ można zrealizować wyłącznie dzięki użyciu bramek NAND ponieważ bramka NAND jest systemem funkcjonalnie pełnym tzn. korzystając wyłącznie z niej można przedstawić dowolną funkcję boolowską.

## 1.2. Rozwiązanie teoretyczne

Pierwszym krokiem jest przekształcenie wyrażenie do zawierającego wyłącznie operacje NAND.

$$\begin{aligned} Y &= \overline{A}C + B(A + B) \\ &= \overline{\overline{\overline{A}C} \overline{B(A + B)}} \end{aligned}$$

## 1.3. Symulacja w programie Multisim

Program został napisany w języku Python w wersji 3.8. Do stworzenia aplikacji graficznej użyto biblioteki pygame<sup>1</sup>.

## 1.4. Wnioski

Celem implementacji struktur jest stworzenie optymalnego algorytmu odpowiadającego na zapytanie:

---

<sup>1</sup> <https://www.pygame.org>

## 2. Zadanie 1b

Wykorzystamy funkcję działającą w następujący sposób:

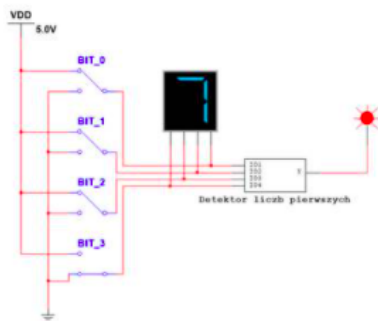
1. porównanie obszaru zapytania z aktualnie rozważanym wierzchołkiem,
  - a) jeśli obszar zawiera rozważany punkt, dodaj go do zbioru wynikowego,
2. wywołaj funkcję rekurencyjnie dla każdego podobszaru przecinającego się z obszarem zapytania.

W ten sposób w każdym kroku zmniejszamy obszar poszukiwań.

Złożoność:

- $O(\log n)$  dla pojedynczego punktu,
- $O(m \log n)$  dla zadanego obszaru, gdzie  $m$  to ilość punktów w tym obszarze.

Na Rys. 2.1 przedstawiono animację znajdującą pojedynczy punkt na płaszczyźnie (sytuacja ta jest zatem tożsama ze wstawianiem punktu). Jak widzimy podziały nie zawsze są równe. Wynika to z niezbalansowania drzewa.



Rysunek 2.1: Wyszukiwanie pojedynczego punktu w drzewie czwórkowym