Zadanie nr 0

Architektura komputerów, gr. 5, czwartek 13:15  
Wojciech Krzos, Natalia Marszałek

## Przyjęta metodologia tworzenia układów w programie LogiSim

Z racji na konieczność utworzenia 3 różnych układów bramek logicznych zdecydowano, że należy obrać metodologię ich tworzenia. Następujące kroki zostały podjęte:

1. Zbadać podaną funkcję Boolowską za pomocą tablicy prawdy oraz mapy Karnaugh.
2. Użyć oprogramowania LogiSim i zbudować niezoptymalizowany układ bramek logicznych.
3. Jeżeli w za pomocą mapy Karnaugh odkryto możliwość zoptymalizowania funkcji, zbudować zoptymalizowany układ bramek logicznych.
4. Sprawdzić wynik testując układ w LogiSim.

Zdecydowano się na utworzenie zarówno zoptymalizowanego jak i niezoptymalizowanego układu w celu poszukiwania wniosków z ćwiczenia.

## Niezoptymalizowany układ bramek logicznych

Następująca funkcja (Eq. 1) będzie brana pod uwagę w tym referacie:

|  |  |  |
| --- | --- | --- |
|  |  | Eq. 1 |

Do jej analizy, w celu dalszego skomponowania układu w symulatorze oraz optymalizacji, została sporządzona tabela prawdy:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Zmienne | | |  |
|  | a | b | c |  |
| Wartości sygnałów wejściowych | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 |

Figure 1: tablica prawdy funkcji

Na podstawie wzoru funkcji (Eq. 1) oraz tabeli (Figure 1) został sporządzony niezoptymalizowany układ bramek logicznych (Figure 2):

Diagram, schematic

Description automatically generated

Figure 2: Niezoptymalizowany układ dla funkcji

### Minimalizowanie funkcji i optymalizacja układu

Zostało zdecydowane, że zostanie podjęta próba optymalizacji wspomnianej funkcji. Do tego celu, została użyta Mapa Karnaugh, która przyjęła następującą formę:

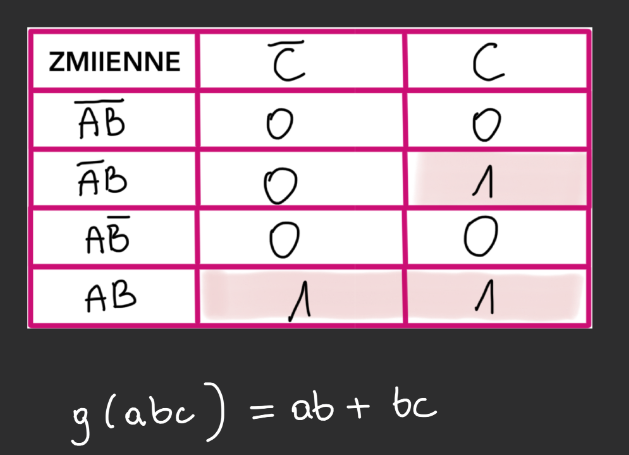


Figure 3: Mapa Karnaugh dla funkcji Eq. 1

Po analizie powyższej mapy (Figure 3) dla funkcji , otrzymujemy zoptymalizowaną funkcję:

|  |  |  |
| --- | --- | --- |
|  |  | Eq. 2 |

Dzięki temu powstał następujący układ bramek logicznych:

Diagram, schematic

Description automatically generated

Figure 4: Zoptymalizowany układ bramek

Dla tego układu również utworzono tablicę prawdy:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Zmienne | | |  |
|  | a | b | c |  |
| Wartości sygnałów wejściowych | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 |

Figure 5: tablica prawdy funkcji

Porównując Figure 1 do Figure 5 upewniamy się, że zachodzi równość w kontekście wartości zwracanych przez obydwie funkcje.

## Konkluzje

Skonstruowanie wymaganych w zadaniu układów nie sprawiło trudności. Dzięki zoptymalizowaniu układu (Figure 4) pozbyliśmy się trzeciej bramki AND z układu niezoptymalizowanego (Figure 2). W większych układach bramek logicznych (np. w skali procesorów), taka optymalizacja jest niezbędna do obniżenia kosztów oraz zwiększenia ich wydajności (Shivakumar, 2015) (Gellert et al, 2019).

Zatem z zadania można wywnioskować, szczególnie z racji na brak kwestii optymalizacji w jego poleceniu, że należy szukać sposobów na uproszczenie każdego układu ze względu na korzyści, jakie niesie ze sobą ten proces.

## Bibliografia

1. Shivakumar SK. 3 - Optimizing Performance of Enterprise Web Application. In: Shivakumar SK, editor. Architecting High Performing, Scalable and Available Enterprise Web Applications.

Boston: Morgan Kaufmann; 2015 [accessed 2023 Mar 3]. p. 101–141. <https://www.sciencedirect.com/science/article/pii/B9780128022580000032>. doi:[10.1016/B978-0-12-802258-0.00003-2](https://doi.org/10.1016/B978-0-12-802258-0.00003-2)

1. Gellert A, Florea A, Fiore U, Zanetti P, Vintan L. Performance and energy optimisation in CPUs through fuzzy knowledge representation. Information Sciences. 2019;476:375–391. doi:[10.1016/j.ins.2018.03.029](https://doi.org/10.1016/j.ins.2018.03.029)