## **Sprawozdanie z układów logicznych**

**Jan Maciuk - 272664**

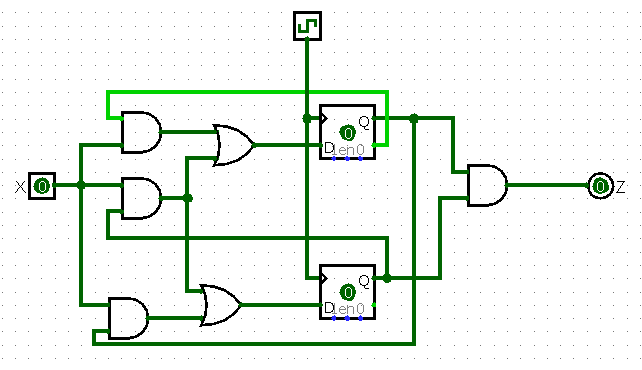
**Michał Dudniczenko - 272677**

**Ćwiczenie 4**

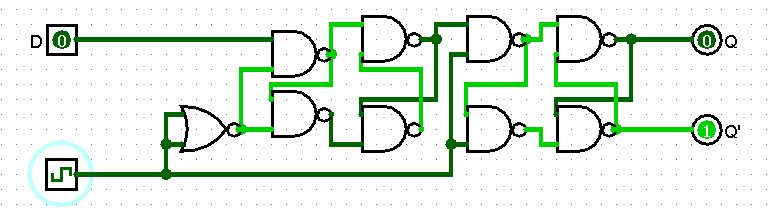
**Grupa 5. Czwartek 13:15-15:00**

**Temat: Analiza układu synchronicznego**

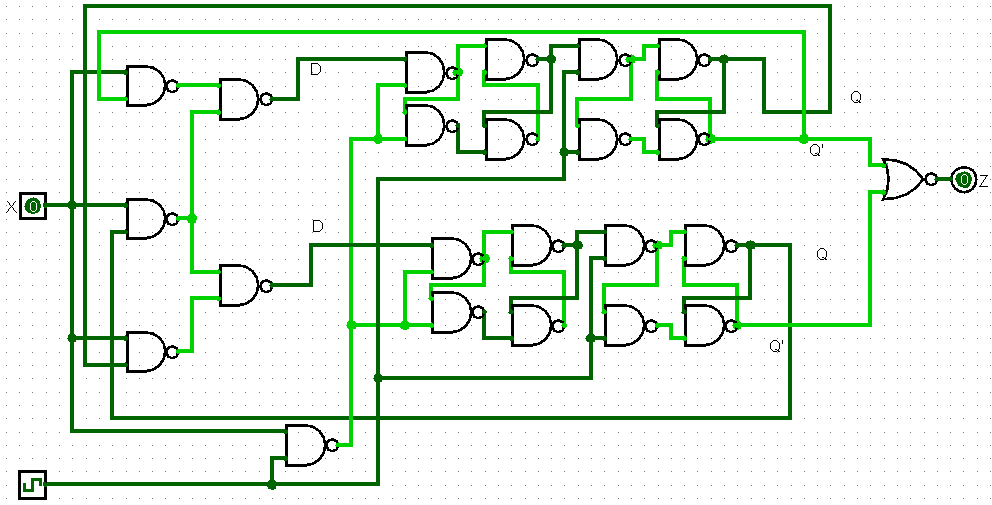
## Ćwiczenia rozpoczęliśmy od realizacji pierwotnego układu zgodnie ze schematem, w rezultacie otrzymaliśmy układ postaci:



Proces przekształcania rozpoczęliśmy od zaprojektowania implementacji przerzutnika D za pomocą bramek NAND. Byliśmy zmuszeni użyć bardziej skomplikowanej konfiguracji master-slave, dzięki czemu uzyskaliśmy przerzutnik, który poprawnie reagował na zbocze narastające zegara, a nie na stan wysoki, jak to było w przypadku prostszej implementacji. Oto nasza implementacja:



Następnie rozpoczęliśmy już właściwy proces modyfikacji pierwotnego układu. Zamieniliśmy gotowe przerzutniki w formie bloków, na naszą implementację oraz zastąpiliśmy obecne w układzie bramki AND i OR odpowiednikami składającymi się z NAND. Układ działał w pełni poprawnie, pozostało jedynie, zgodnie ze specyfikacją pozbyć się wszystkich bramek NOT. Na początku od razu usunęliśmy 4 nadmiarowe bramki NOT, wynikające z połączenia bramek AND z OR, eliminacja bramki NOT na końcu układu możliwa była dzięki użycia zamiast niej bramki NOR, do której doprowadziliśmy zanegowane wyjścia przekaźników w odróżnieniu do wersji pierwotnej. Jedynym zadaniem do wykonania było pozbycie się bramki NOT, potrzebnej do wytworzenia zanegowane sygnału zegarowego, potrzebnego do działania implementacji przerzutników master-slave. Po większych namysłach udało się to uzyskać, dzięki użyciu bramki NAND, do której poprowadziliśmy sygnał zegarowy oraz sygnał wejściowy X. Po testach okazało się, że finalna wersja przekształconego i zminimalizowanego układu działa dokładnie tak jak ta z treści zadania. Oto wynik naszych działań:



Mając gotowy i w pełni funkcjonalny układ w symulatorze, w prosty sposób byliśmy w stanie obserwować zmiany konkretnych elementów układu, dzięki czemu na podstawie naszych obserwacji stworzyliśmy tablicę stanów-wyjść, jak i wypełniliśmy tablicę przejść-wyjść.

Tablica przejść-wyjść Tablica stanów-wyjść

| y1 y2 x | 0 | 1 | Z |  | y1 y2 x | 0 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 0 | 0 0 | 1 0 | 0 |  | A = 00 | A/0 | C/0 |
| 0 1 | 0 0 | 1 1 | 0 |  | B = 01 | A/0 | D/1 |
| 1 0 | 0 0 | 0 1 | 0 |  | C = 10 | A/0 | B/0 |
| 1 1 | 0 0 | 1 1 | 1 |  | D = 11 | A/0 | D/1 |

W analogiczny sposób przebiegła analiza zachowań układu dla podanej sekwencji wejściowej “111100”. Zaobserwowane zachowania zanotowaliśmy w tabeli:

Analiza przejść przy zadanej sekwencji wejściowej:

| czas | t0 | t1 | t2 | t3 | t4 | t5 |
| --- | --- | --- | --- | --- | --- | --- |
| x | 1 | 1 | 1 | 1 | 0 | 0 |
| y1 y2 | 0 0 | 1 0 | 0 1 | 1 1 | 0 0 | 0 0 |
| Z | 0 | 0 | 0 | 1 | 0 | 0 |

Synteza układu do postaci zawierającej tylko bramki NOT, AND oraz przerzutniki typu JK przebiegła bez problemów, głównie dzięki prostocie działania wspomnianych przerzutników. Zastąpiliśmy bramki OR odpowiednikami, oraz z użyciem drobnej modyfikacji z dodatkową ścieżką zanegowanego sygnału, wprowadziliśmy do układu przerzutniki JK. Po testach tu również okazało się, że układ działał dokładnie tak jak układ pierwotny. Oto wynik syntezy:

