|  |  |
| --- | --- |
| Laboratorium Teorii Automatów | |
| **Hazardy** | |
| Grupa 4b (wtorek 17.15) | Sonia Wittek, Katarzyna Wątorska, Bartłomiej Mróz |

# Wstęp teoretyczny

Ćwiczenie polegało na zaprojektowaniu i przetestowaniu układu asynchronicznego jakim jest automat obsługujący przejazd kolejowy.

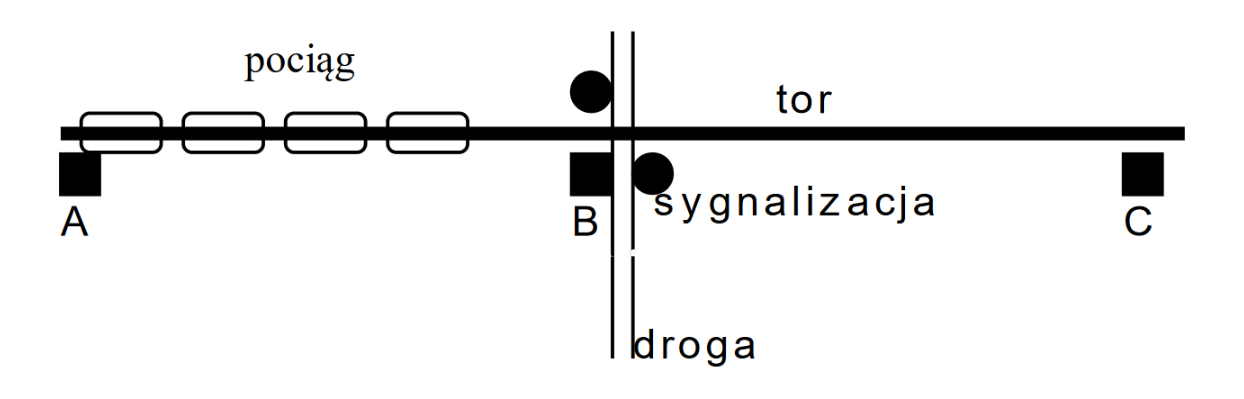
**Założenia zadania:**

1. Pociąg może wjeżdżać na przejazd z dwóch kierunków

2. Czujniki wykrywające obecność pociągu umieszczono w punktach: A, B i C (patrz rysunek)

3. Światła sygnalizacyjne Y mają się zapalić (stan logiczny 1) gdy: (pociąg przesłania A lub C i jedzie w kierunku B) lub (przesłonił A lub C i jedzie w kierunku B) lub (przesłania B).

4. Pociąg nie może manewrować pomiędzy czujnikami A i C (zawracać).



# Przebieg laboratorium

Na samym początku stworzyliśmy tabelę programu dla wejść A, B i C.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Q\ABC | 000 | 001 | 011 | 010 | 110 | 111 | 100 | Y |
| 1 | **1** | 7 | - | - | - | - | 2 | 0 |
| 2 | 3 | - | - | - | 9 | - | **2** | 1 |
| 3 | **3** | - | - | 4 | - | - | - | 1 |
| 4 | 5 | - | 10 | **4** | 9 | - | - | 1 |
| 5 | **5** | 6 | - | - | - | - | 8 | 0 |
| 6 | 1 | **6** | - | - | - | - | - | 0 |
| 7 | 3 | **7** | 10 | - | - | - | - | 1 |
| 8 | 1 | - | - | - | - | - | **8** | 0 |
| 9 | - | - | - | 4 | **9** | 11 | 8 | 1 |
| 10 | - | 6 | **10** | 4 | - | - | - | 1 |
| 11 | - | - | 10 | - | - | **11** | - | 1 |

Ponieważ układ jest symetryczny, zastąpiliśmy sytuacje, gdy pociąg nadjeżdża z lewej lub prawej na jedną – tworząc wejście .

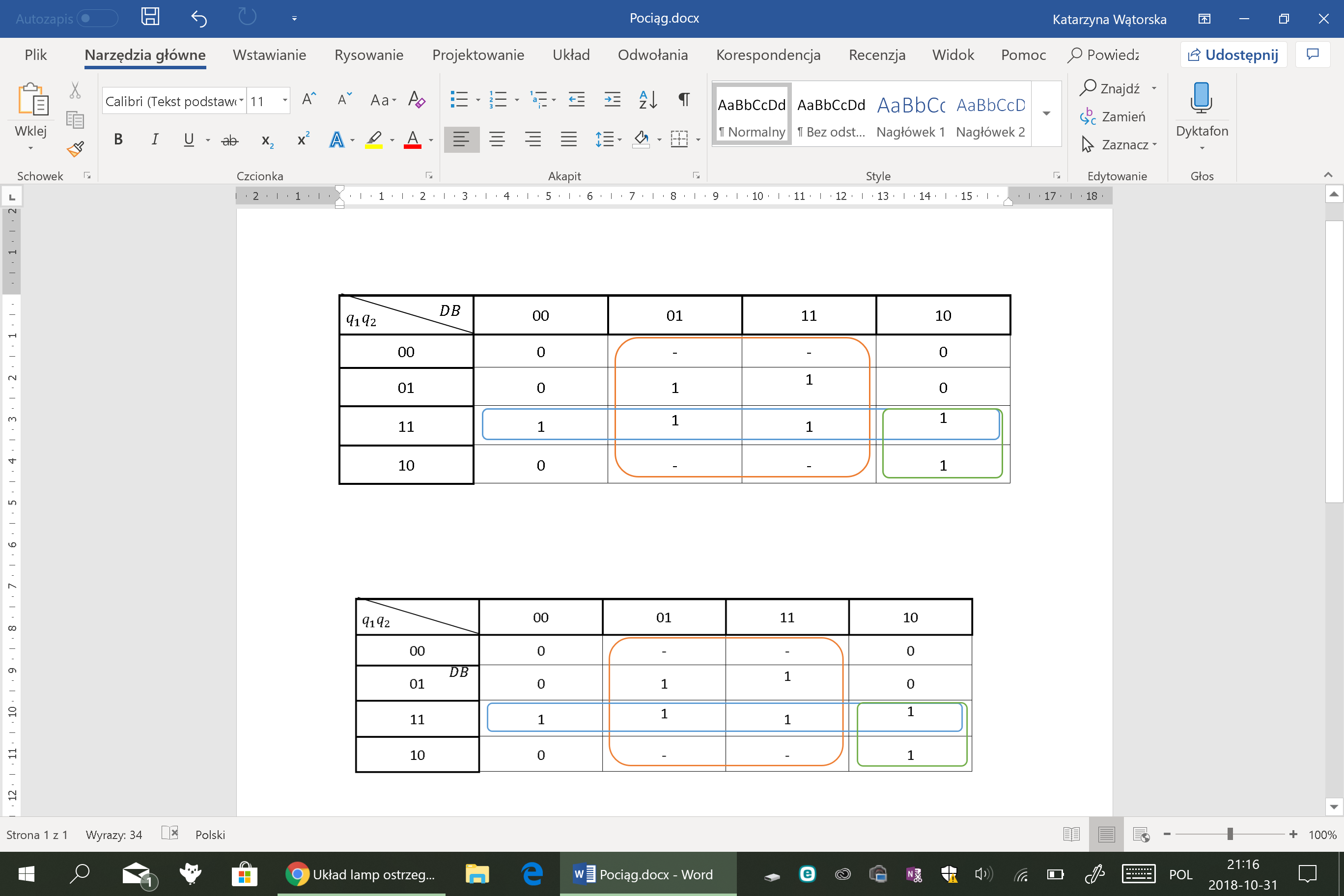
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Q\DB | 00 | 01 | 11 | 10 | Y |
| 1 | **1** | - | - | 2 | 0 |
| 2 | 3 | - | 7 | **2** | 1 |
| 3 | **3** | 4 | - | - | 1 |
| 4 | 5 | **4** | 7 | - | 1 |
| 5 | **5** | - | - | 6 | 0 |
| 6 | 1 | **-** | - | **6** | 0 |
| 7 | - | 4 | **7** | 6 | 1 |

Znajdując stany równoważne (2,3), (4,5) i (6,7) stworzyliśmy na podstawie tej tabeli automat Mealy’ego. Tabela minimalna:

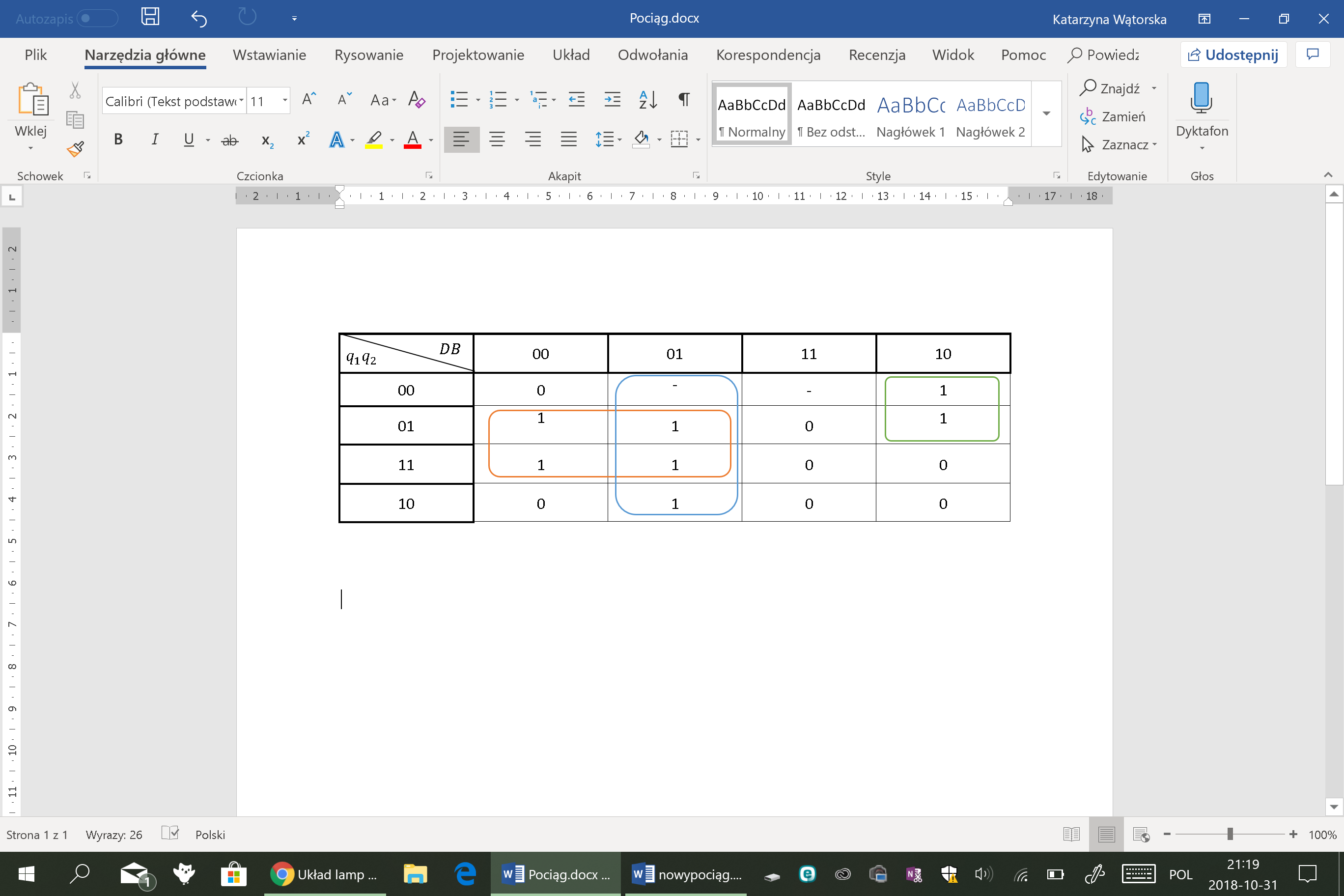
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Q\DB | 00 | 01 | 11 | 10 |
| 1 | **1**, 0 | -,- | -,- | 2 3, 1 |
| 2 3 | **2 3**, 1 | 4 5,1 | 6 7, 1 | **2 3**, 1 |
| 4 5 | **4 5**, 0 | **4 5**, 1 | 6 7, 1 | 6 7, 0 |
| 6 7 | 1, 0 | 4 5, 1 | **6 7**, 1 | **6 7**, 0 |

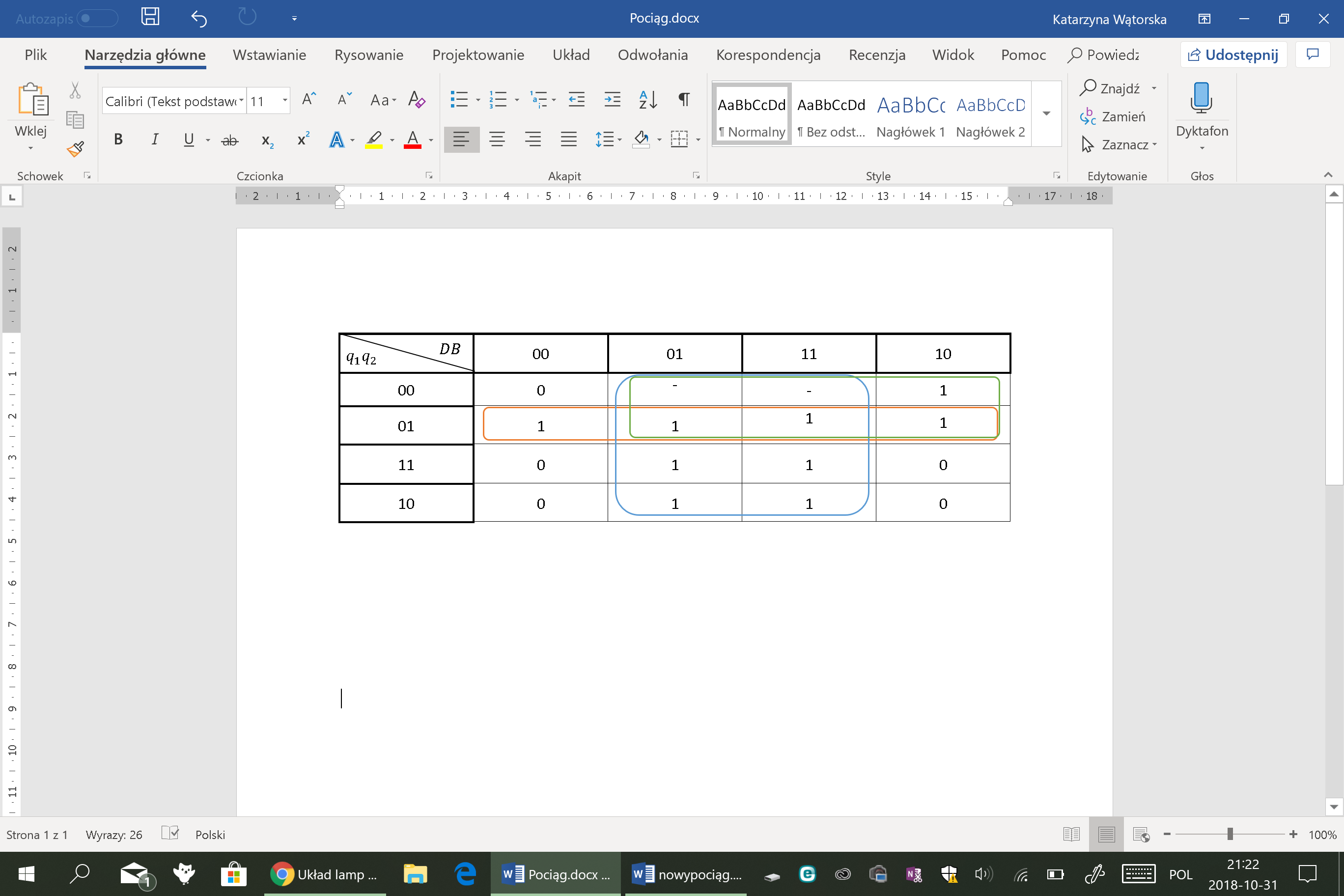
Stany zakodowano w następujący sposób:

* 1 – 00
* 2 3 – 01
* 4 5 – 11
* 6 7 – 10



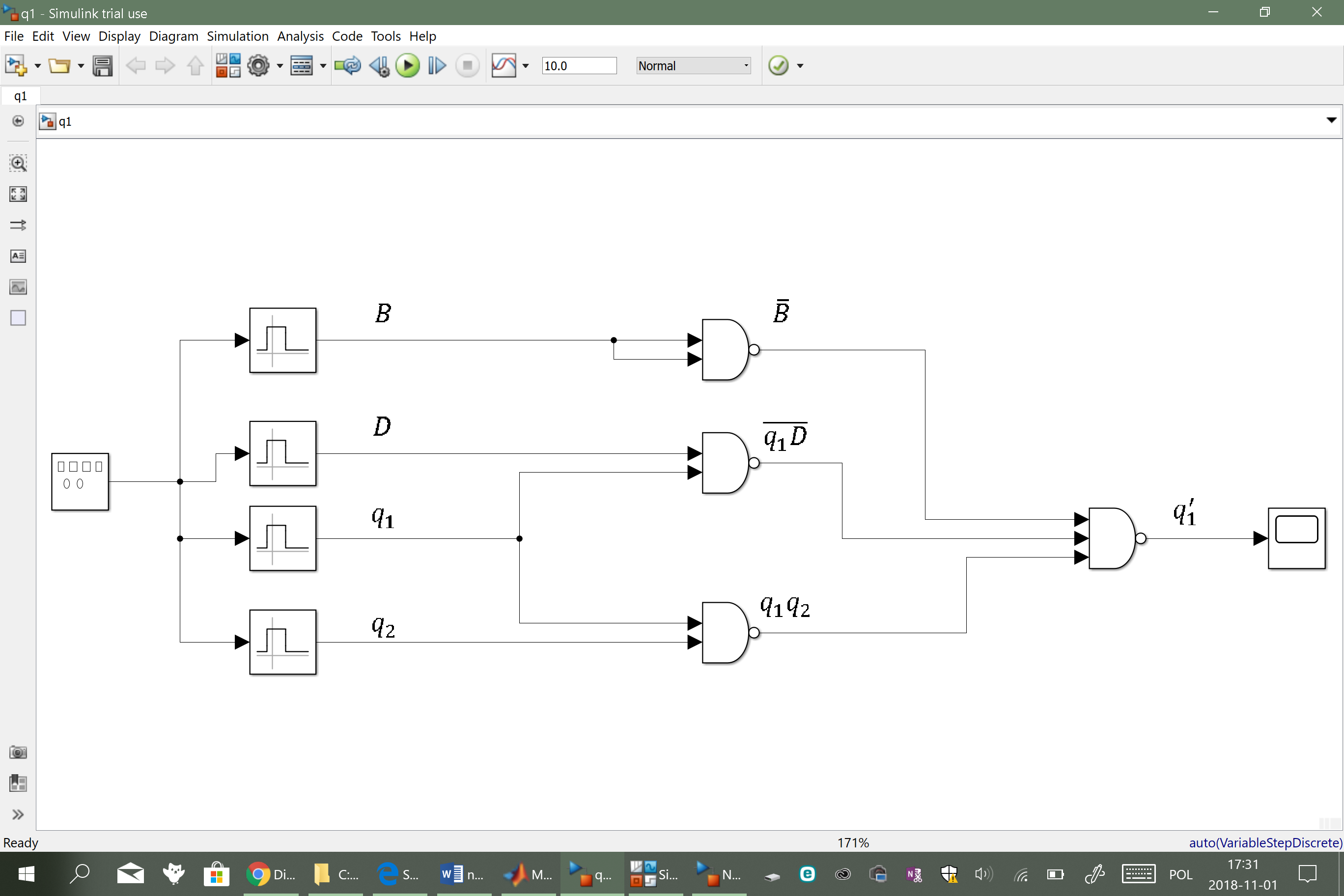
Po wrysowaniu pokryć dla jedynek, otrzymano funkcję:

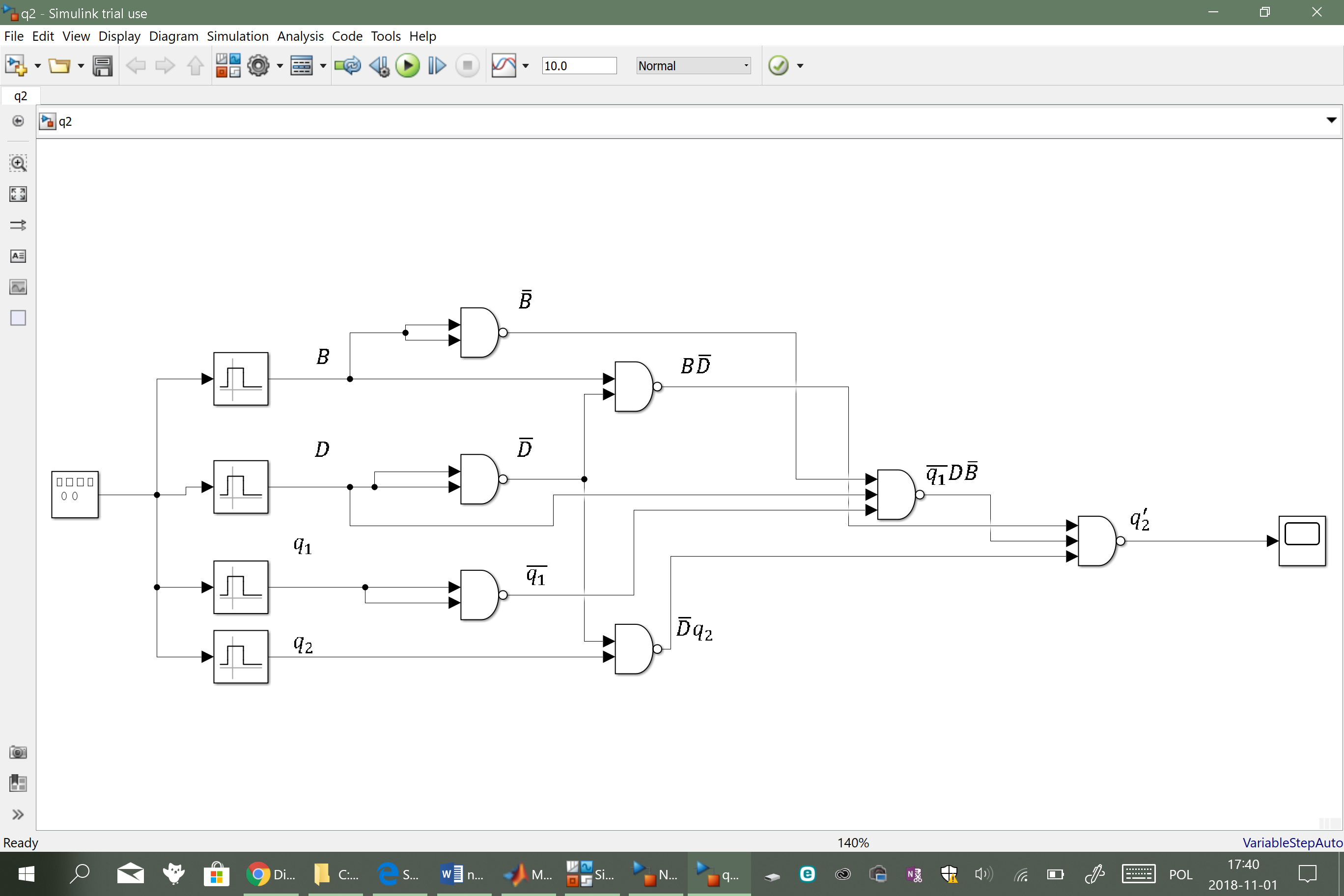


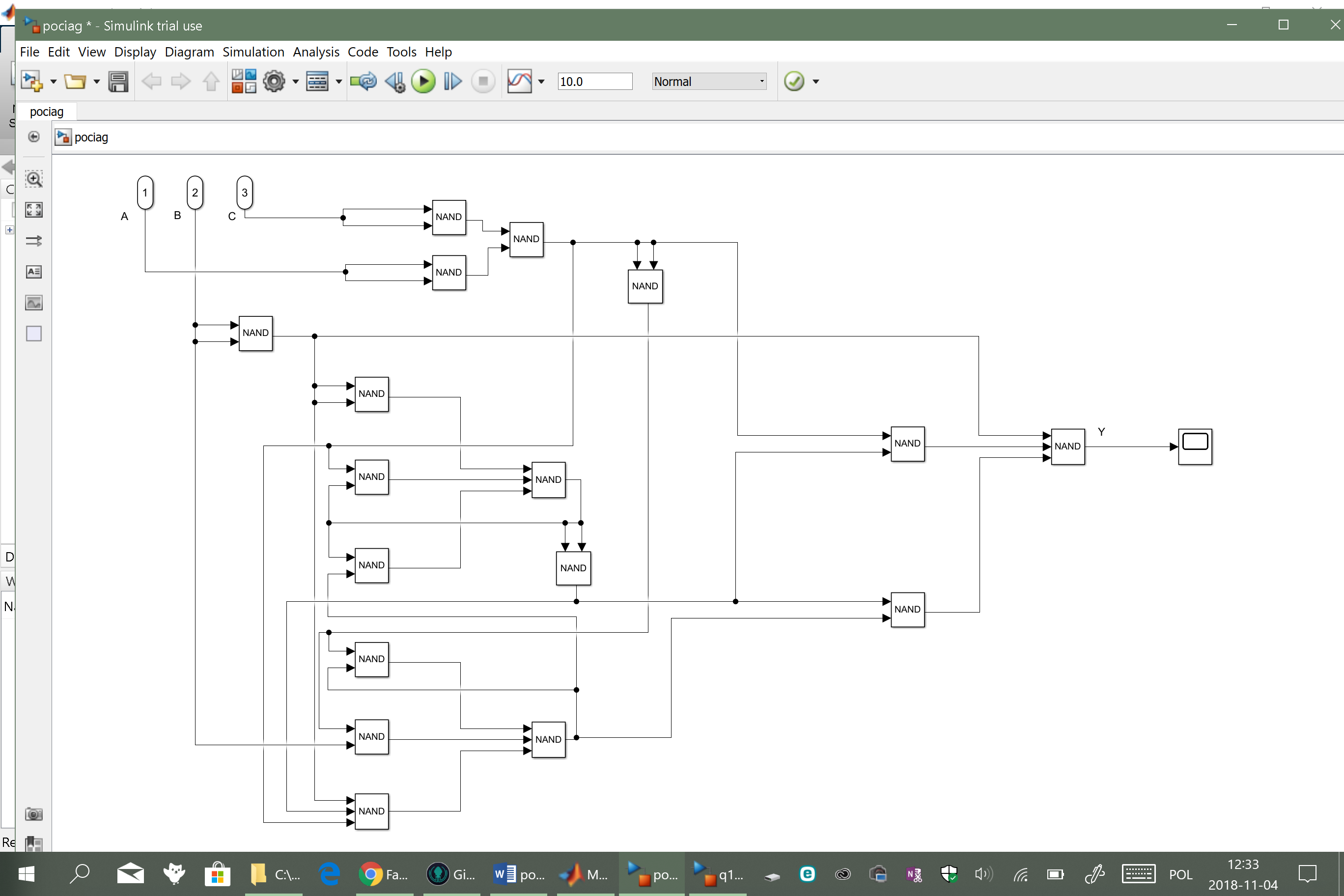


Wykorzystując prawa de Morgana oraz podwójną negację powyższe funkcje logiczne przekształcono do równoważnych postaci możliwych do zrealizowania na bramkach NAND.

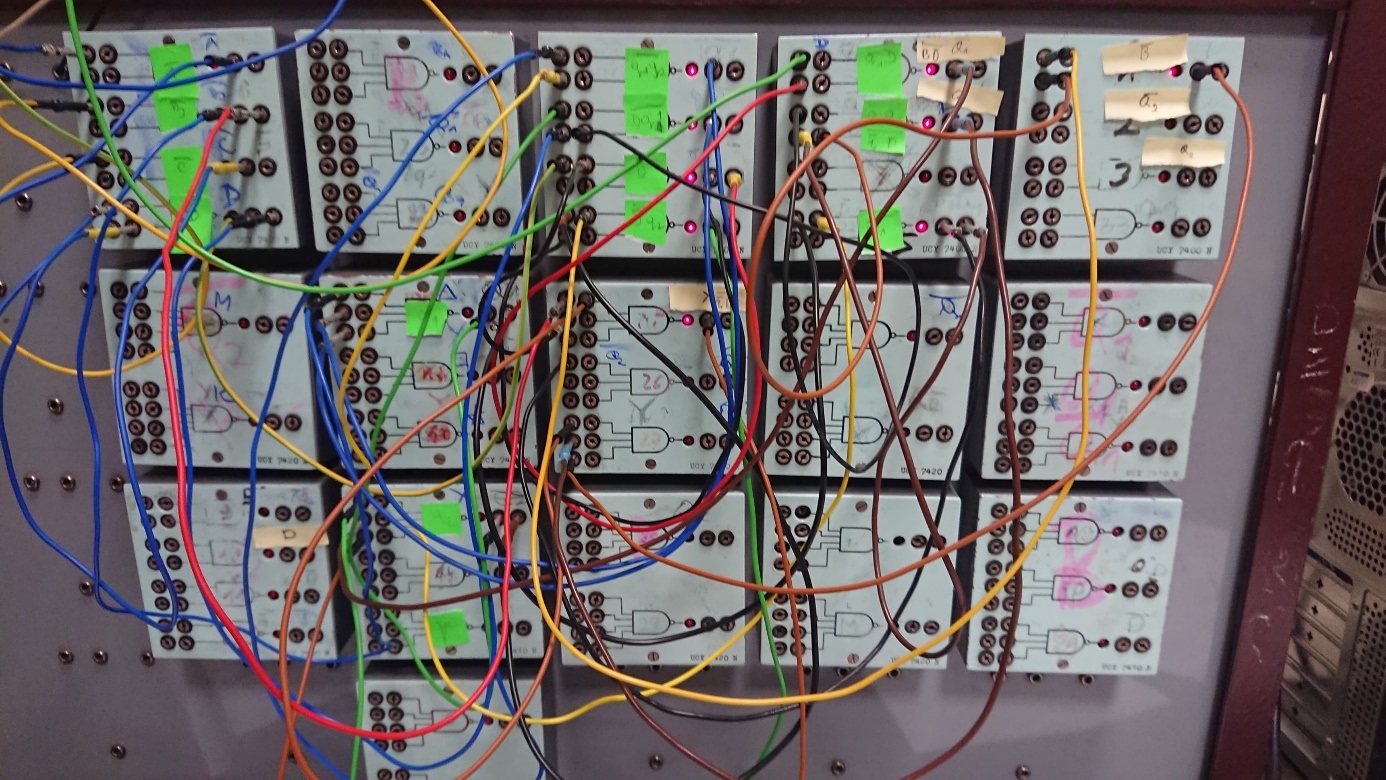
W programie Simulink narysowano schematy dla poszczególnych funkcji.







Końcowy układ:



# Podsumowanie

Po wykonaniu tego laboratorium nauczyliśmy się, w jaki sposób wyglądają układy używane na co dzień i w jaki sposób można je projektować. Dowiedzieliśmy się jak wykonać prosty układ wyłącznie za pomocą bramek NAND oraz poznaliśmy działanie rzeczywistych bramek logicznych.