Sprawozdanie 3

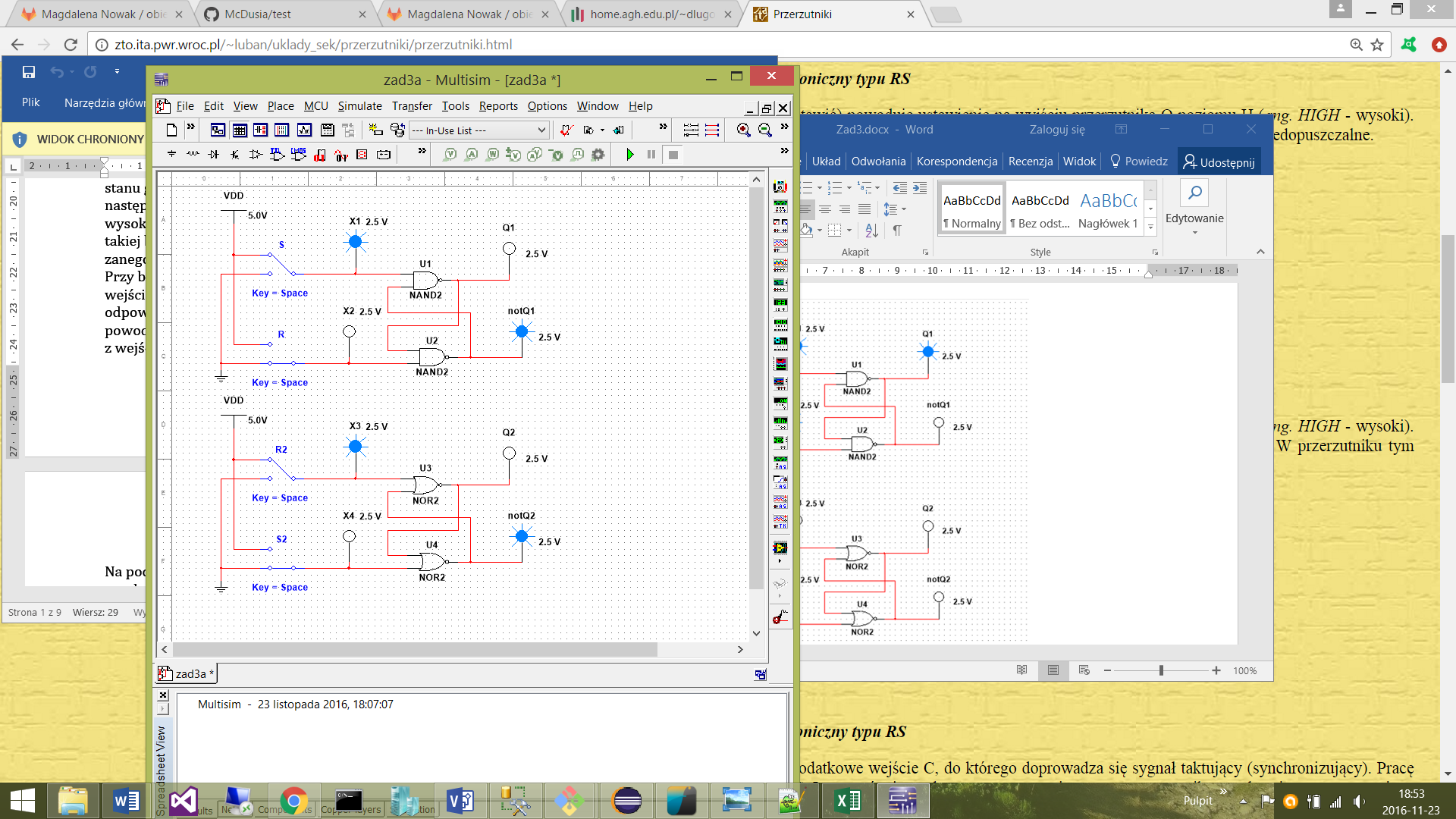
Przerzutniki i rejestry

1. Asynchroniczny przerzutnik „RS”

        1. Przerzutnik asynchroniczny „RS” w oparciu o bramki NAND

        2. Przerzutnik asynchroniczny „RS” w oparciu o bramki NOR

Poniżej zamieszczono schematy obwodów wykonane w programie Multisim.



Na podstawie analizy działania układów zapisano tabele prawdy dla obydwu rodzajów przerzutników.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAND | S | R | Q | ~Q |
|  | 0 | 0 | X | X |
|  | 0 | 1 | 1 | 0 |
|  | 1 | 0 | 0 | 1 |
|  | 1 | 1 | Qn | Qn |

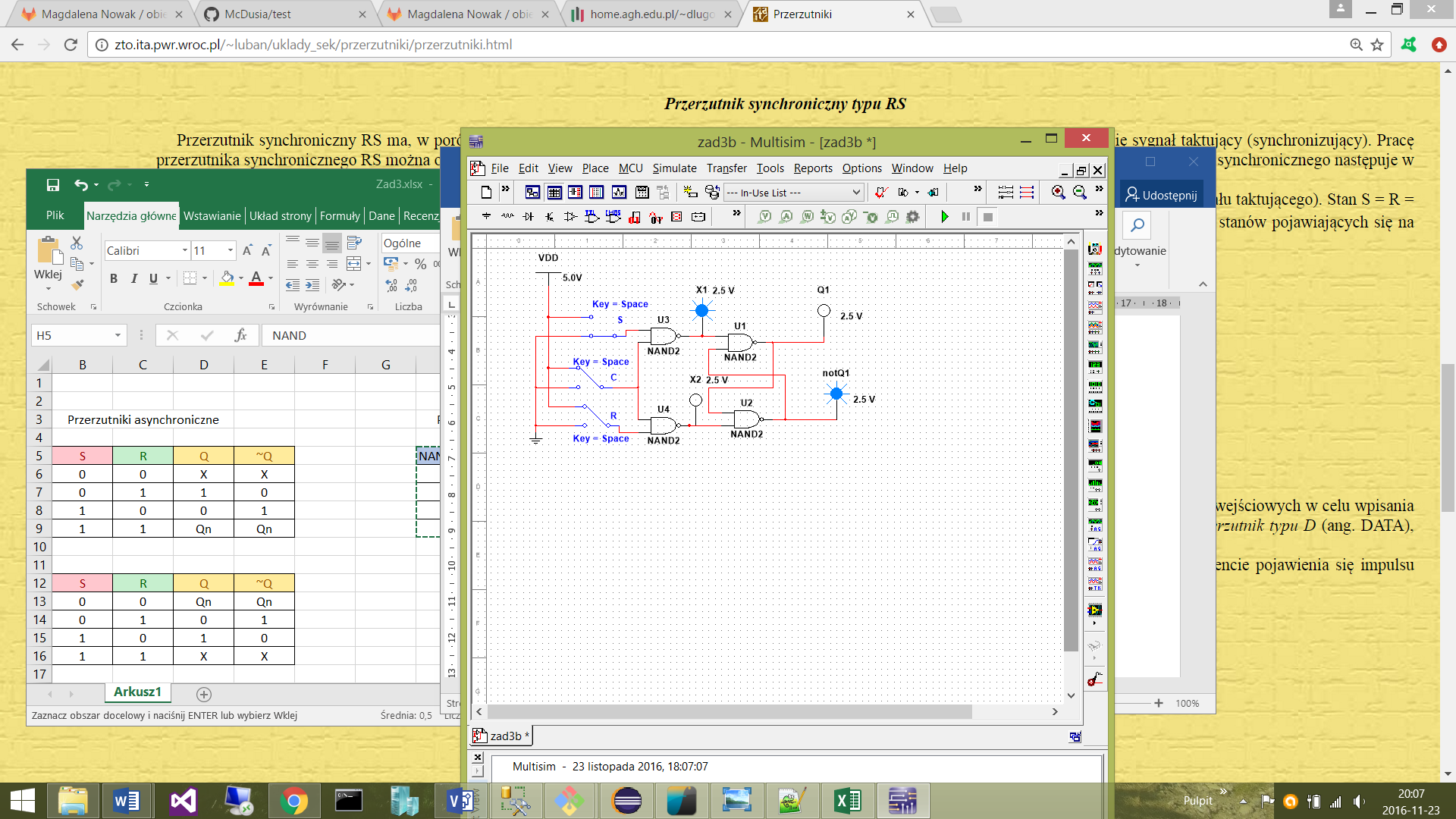
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NOR | S | R | Q | ~Q |
|  | 0 | 0 | Qn | Qn |
|  | 0 | 1 | 0 | 1 |
|  | 1 | 0 | 1 | 0 |
|  | 1 | 1 | X | X |

Dla przerzutnika asynchronicznego zbudowanego z bramek **NAND** doprowadzenie do wejścia S napięcia (poziomu H – High – stan wysoki) przy jednoczesnym doprowadzeniu poziomu niskiego ( L – Low) do wejścia R ustawia na wyjściu sygnał L. Doprowadzenie do wejścia S poziomu L i do wejścia R poziomu H powoduje ustawienie na wyjściu sygnału H. Doprowadzenie napięcia do R i S jednocześnie powoduje, że układ zapamiętuje ustawienie wyjścia, czyli na wyjściu będzie taki sam sygnał, jaki ustawiła poprzednia konfiguracja wejść R i S. Stan niski dla obydwu wejść jest zabroniony, ponieważ wtedy zarówno Q jak i ~Q są w stanie wysokim (nie może być jednocześnie Q prawdą i ~Q prawdą – prawo sprzeczności).

Dla przerzutnika asynchronicznego zbudowanego z bramek **NOR** działanie jest analogiczne, ale dla doprowadzenia stanu wysokiego do wejścia S i stanu niskiego do R na wyjściu ustawi się stan wysoki. Dla odwrotnego ustawienia wejść, stan na wyjściu będzie niski. Jeżeli oba wejścia ustawimy na stan niski, to układ pozostanie w stanie pamiętania. Zaś jednoczesne ustawienie wejść R i S na stan wysoki jest stanem zabronionym (wtedy jednocześnie Q musiałoby być fałszem i ~Q fałszem - sprzeczność).

1. Synchroniczny przerzutnik „RS”, reagujący na stan sygnału zegarowego równy jeden

Przerzutnik synchroniczny „RS” ma oprócz wejść R i S, dodatkowe wejście C, do którego doprowadza się sygnał taktujący, który zapewnia synchroniczne działanie układu. Działanie synchronicznego przerzutnika „RS” jest podobne do działania takiego samego przerzutnika asynchronicznego. Różnica polega na tym, że zmiana stanu przerzutnika następuje w chwilach wyznaczonych przez sygnał taktujący C (Clock). Stan logiczny wyjścia Q w danej chwili zależy od stanu na wejściach R, S i Q w poprzednim takcie zegara. W przedziale czasu między kolejnymi impulsami taktującymi C (gdy wejście C jest w stanie wysokim) przerzutnik nie zmienia stanu, niezależnie od sygnałów na wejściach R i S, czyli pamięta poprzedni stan.   
 Poniżej układ testujący działanie synchronicznego przerzutnika „RS”.



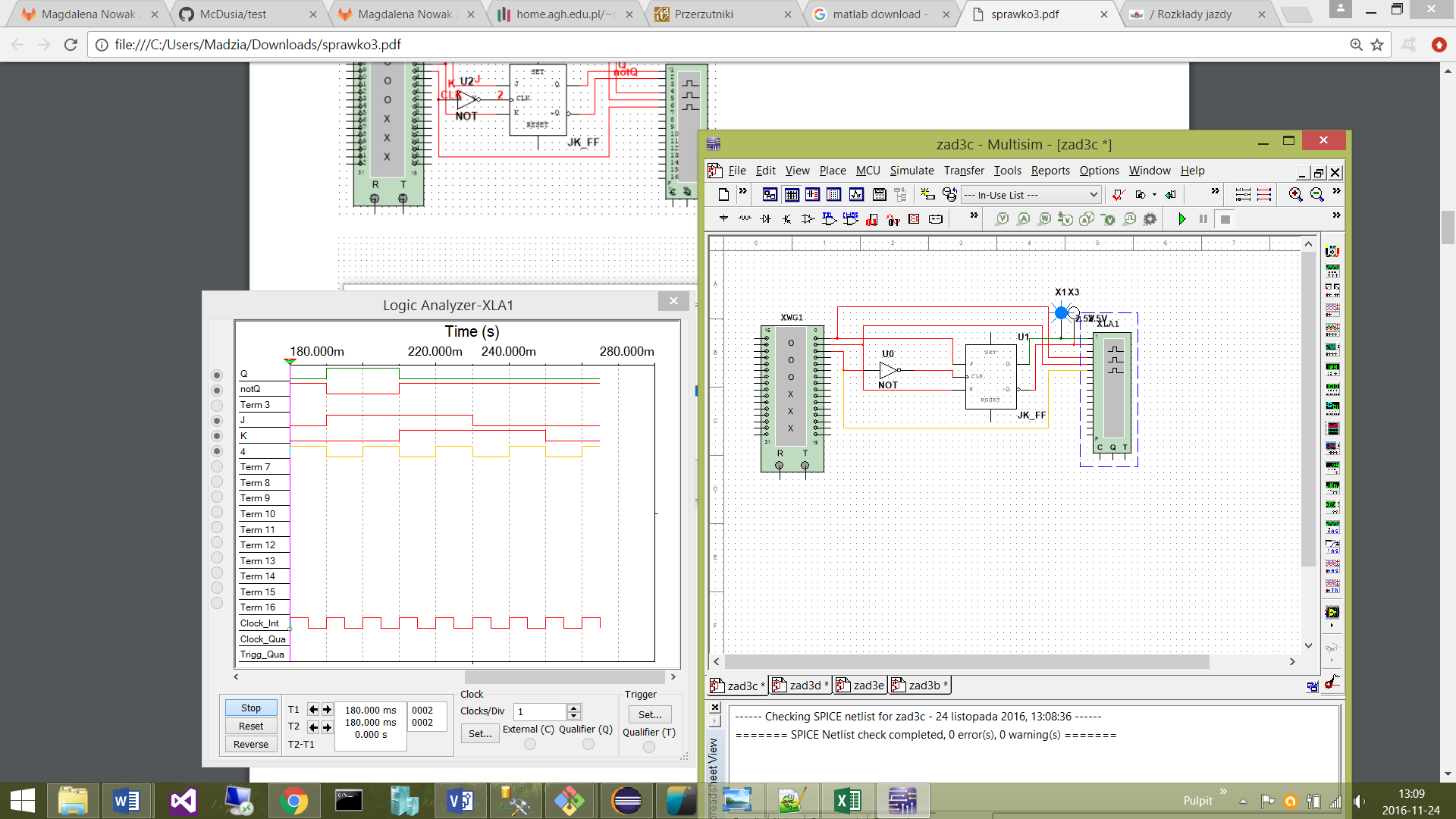
Skoro układ ma reagować na stan sygnału zegarowego równy jeden, musi być zbudowany z bramek NAND. Wtedy stan na wyjściu będzie zależny od sygnału w drugim przewodzie prowadzącym do bramki NAND. Gdyby zastosowano bramkę NOR, to dla sygnału wysokiego dla wejścia C zawsze na wyjściu bramki NOR byłoby 0 (~(prawda or prawda) => fałsz i ~(prawda or fałsz) => fałsz ), czyli układ nie reagowałby na zmiany sygnałów na wejściach R i S.

Poniżej tabela prawdy dla tego rodzaju przerzutnika:

|  |  |  |  |
| --- | --- | --- | --- |
| NAND | S | R | Q (n+1) |
|  | 0 | 0 | Qn |
|  | 0 | 1 | 0 |
|  | 1 | 0 | 1 |
|  | 1 | 1 | X |

1. Synchroniczny przerzutnik „JK”

Przerzutnik „JK” w programie Multisim reaguje na rosnące zbocze sygnału zegarowego. Aby reagował na opadające dodano bramkę NOT przed wejściem Clock.



Poniżej porównanie sygnałów wejścia i wyjścia.



Tabela prawdy dla przerzutnika „JK”:

|  |  |  |
| --- | --- | --- |
| **J** | **K** | **Q(n+1)** |
| 0 | 0 | Qn |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | not(Qn) |

Przerzutnik JK dla stanów niskich zarówno dla J jak i K nie zmienia sygnału wyjściowego, a właściwie „zapamiętuje” go. Dla dwóch stanów wysokich, neguje sygnał wyjściowy, zaś w pozostałych przypadkach działa jak przerzutnik RS zbudowany na bramkach NOR, przy czym J odpowiada S, a K odpowiada wejściu R.

1. Przerzutnik typu „D” bazujący na asynchronicznym przerzutniku „RS”

Poniżej przedstawiono tabelkę, która pokazuje oczekiwane działanie przerzutnika.

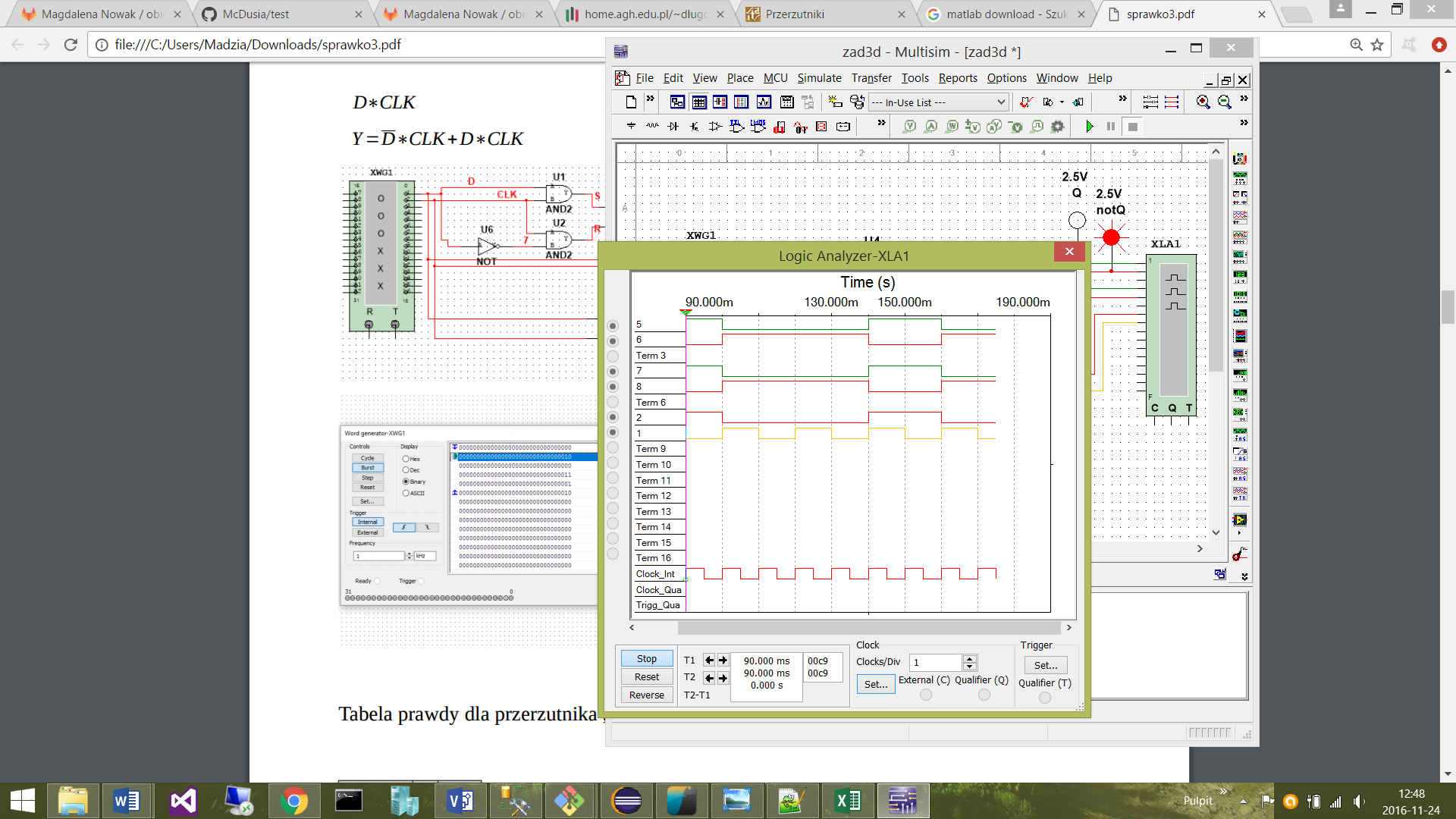
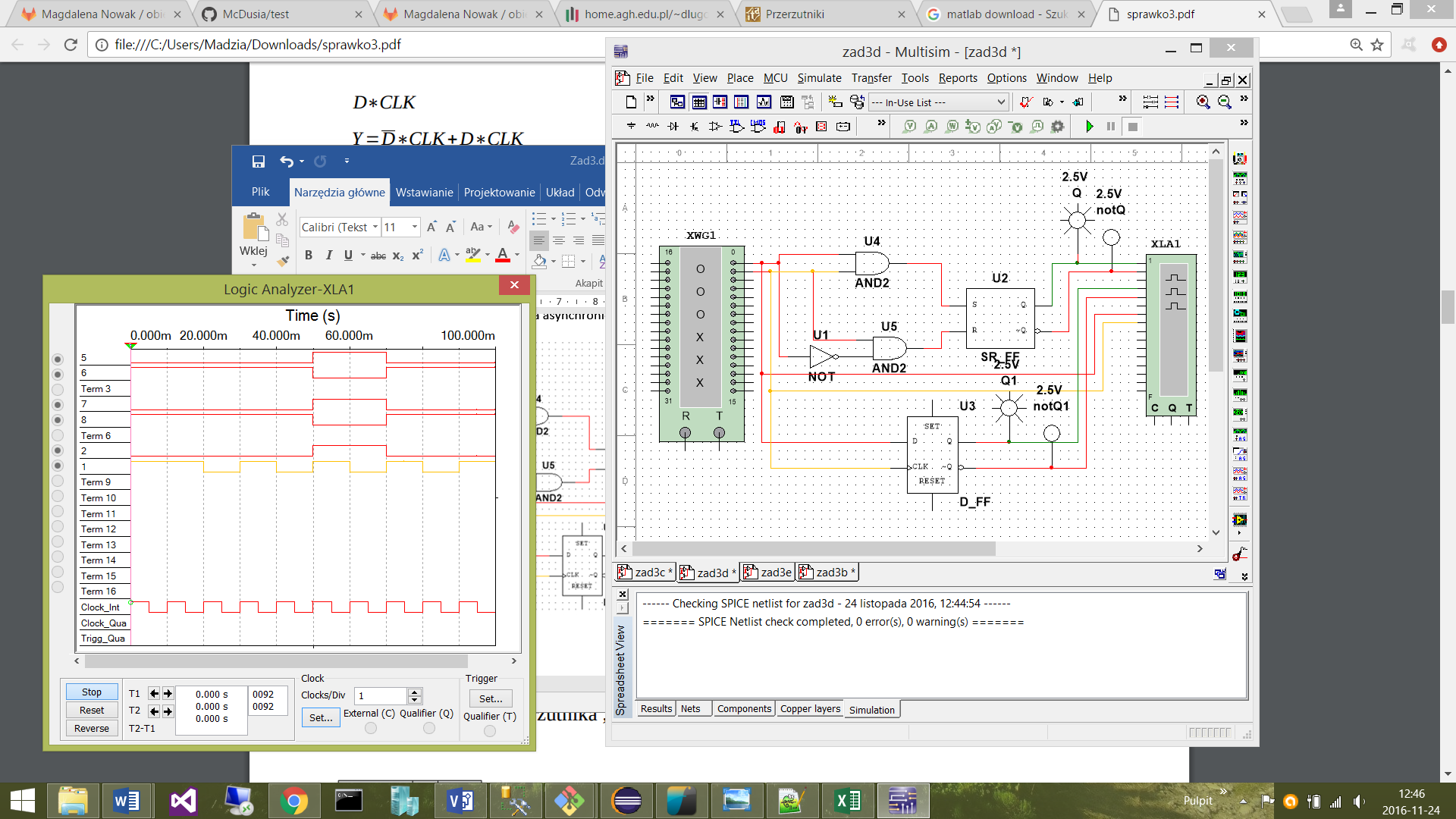
|  |  |  |  |
| --- | --- | --- | --- |
| **D** | **CLK** | **S** | **R** |
| 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |

Tablice Karnough’a.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| dla R |  |  |  | dla S |  |  |
| D\CLK | 0 | 1 |  | D\CLK | 0 | 1 |
| 0 | 0 | 1 |  | 0 | 0 | 0 |
| 1 | 0 | 0 |  | 1 | 0 | 1 |

Ostatecznie funkcja ma postać: Y=D∗CLK+D∗CLK

Układ testujący działanie przerzutnika:

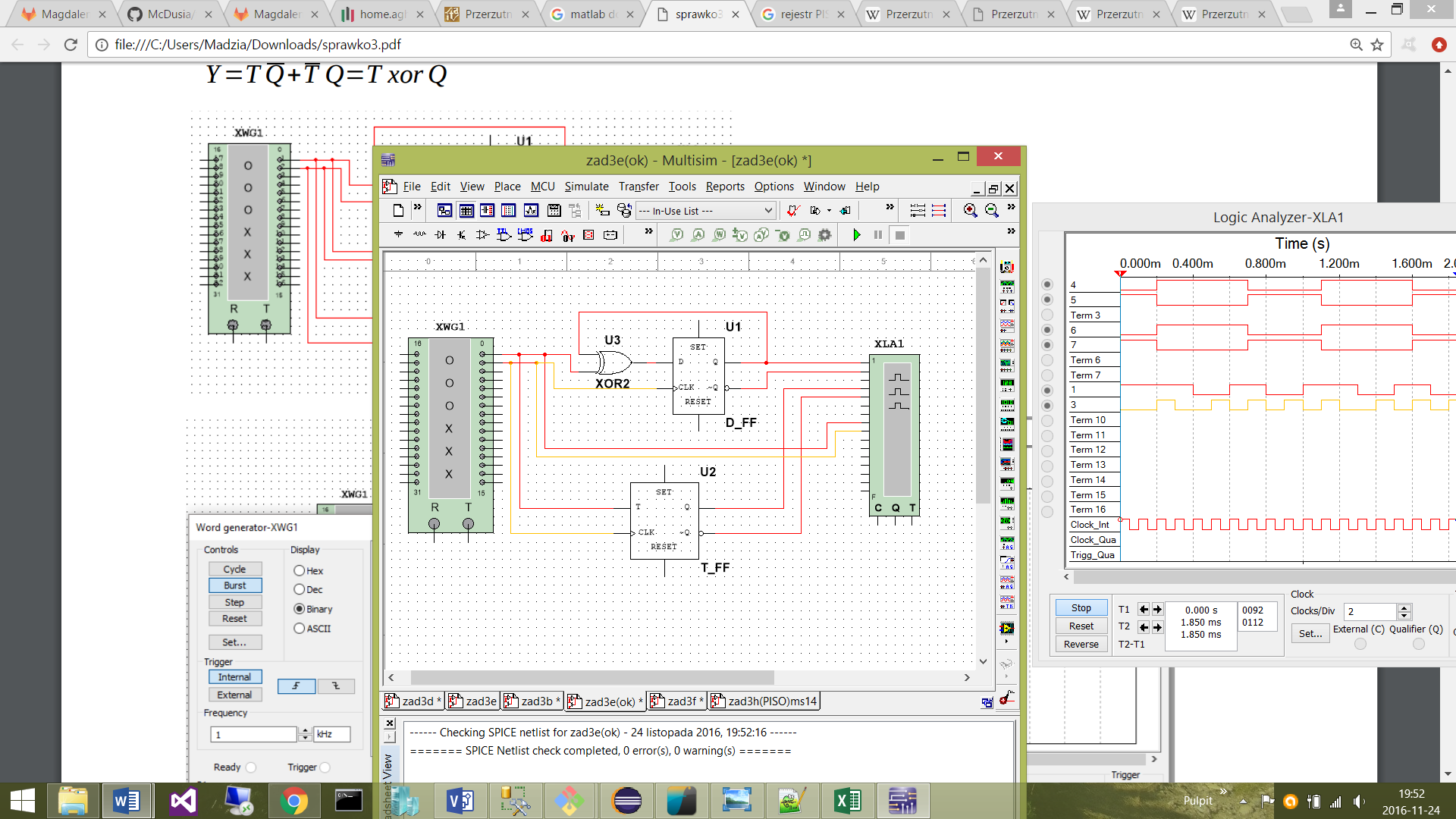


Przerzutnik typu „D” (od angielskiego delay flip-flop) to przerzutnik synchroniczny zwany „opóźniającym”. Przerzutnik ten przepisuje stan wejścia D na wyjście Q, ale dzieję się to tylko wtedy, gdy na zegarze jest sygnał rosnący. W innych przypadkach, przerzutnik nie „reaguje” na sygnał wejściowy, pamięta stan poprzedni.

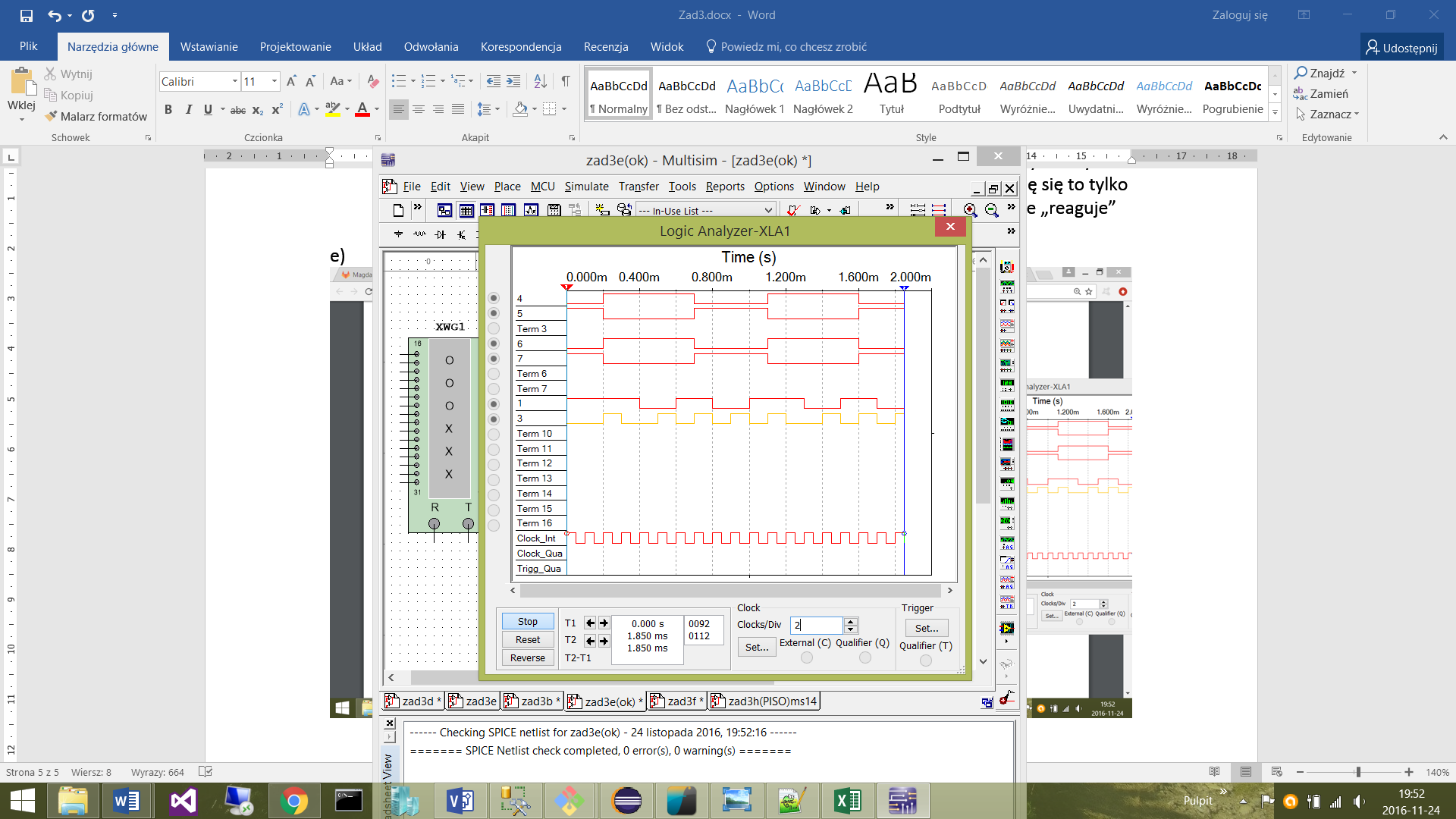
e) Przerzutnik typu „T” bazujący na synchronicznym przerzutniku "D" i bramce XOR.

|  |  |  |
| --- | --- | --- |
| **T** | **Q** | **Q(n+1)** |
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| T\Q | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 1 | 0 |

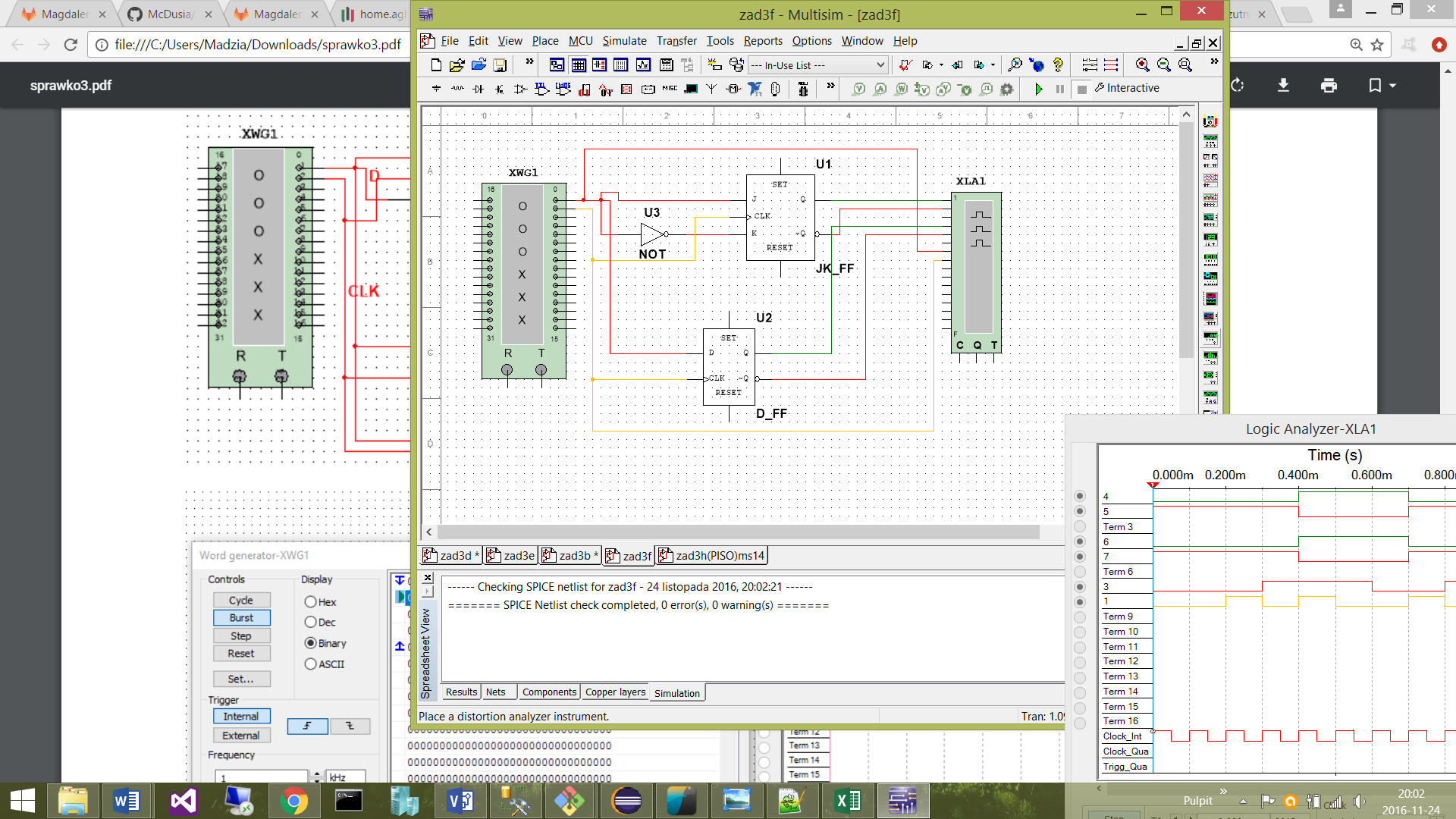


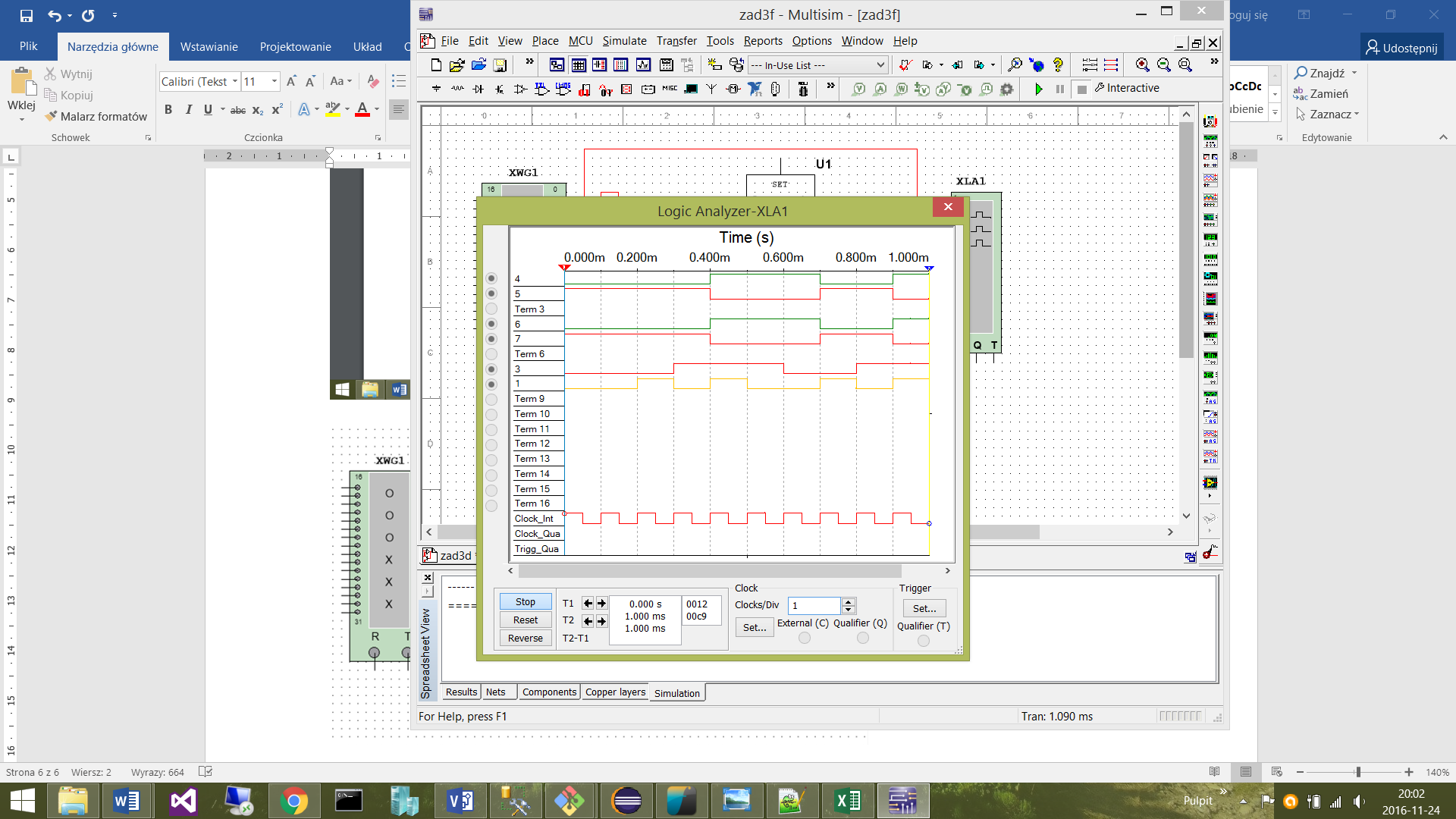
Funkcja wynikowa ma postać: Y=T Q+T Q=T xor Q



1. Przerzutnik typu „D” zbudowany z synchronicznego przerzutnika "JK" i bramki NOT.

|  |  |  |  |
| --- | --- | --- | --- |
| **D** | **J** | **K** | **Q** |
| 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

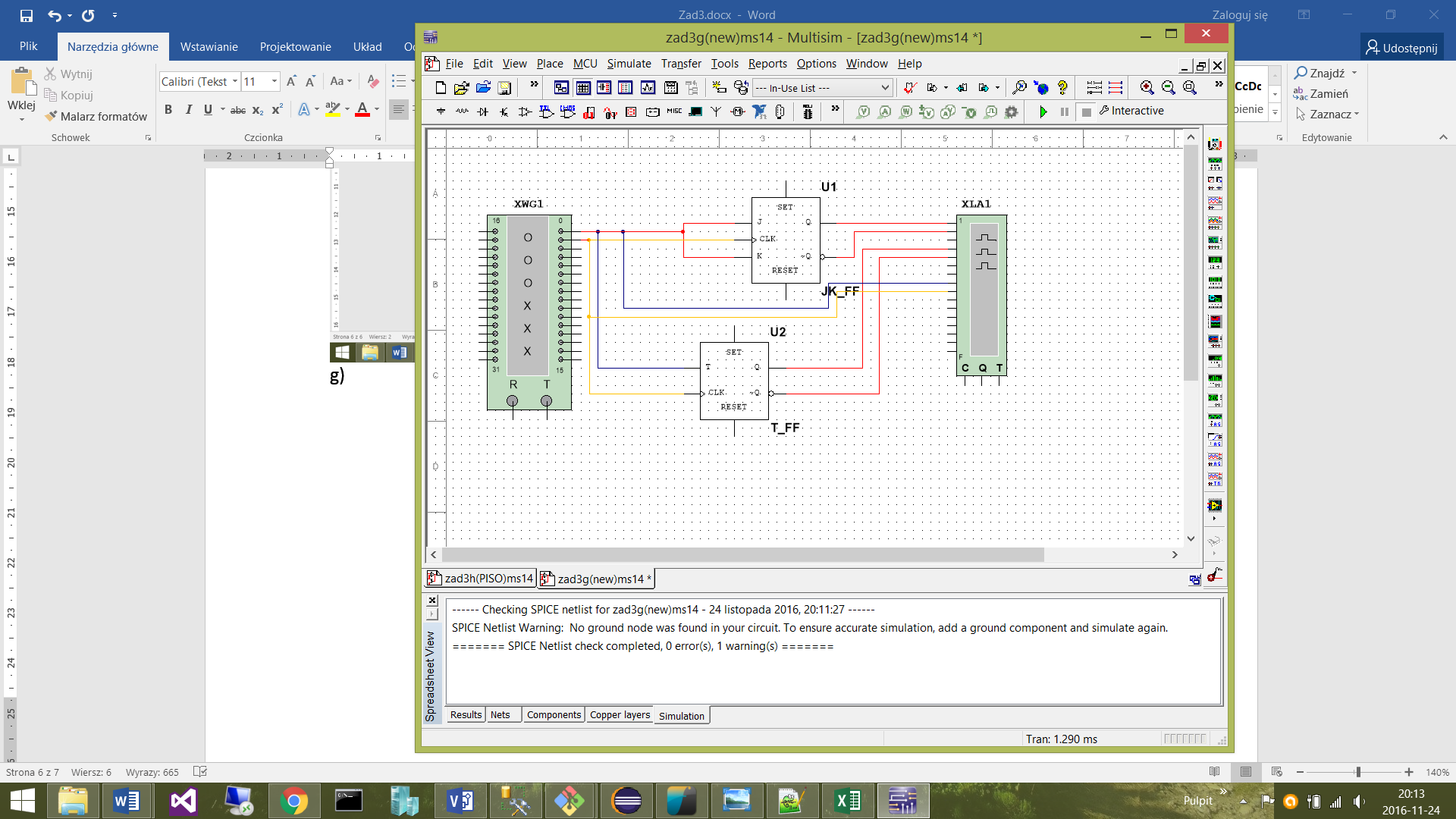


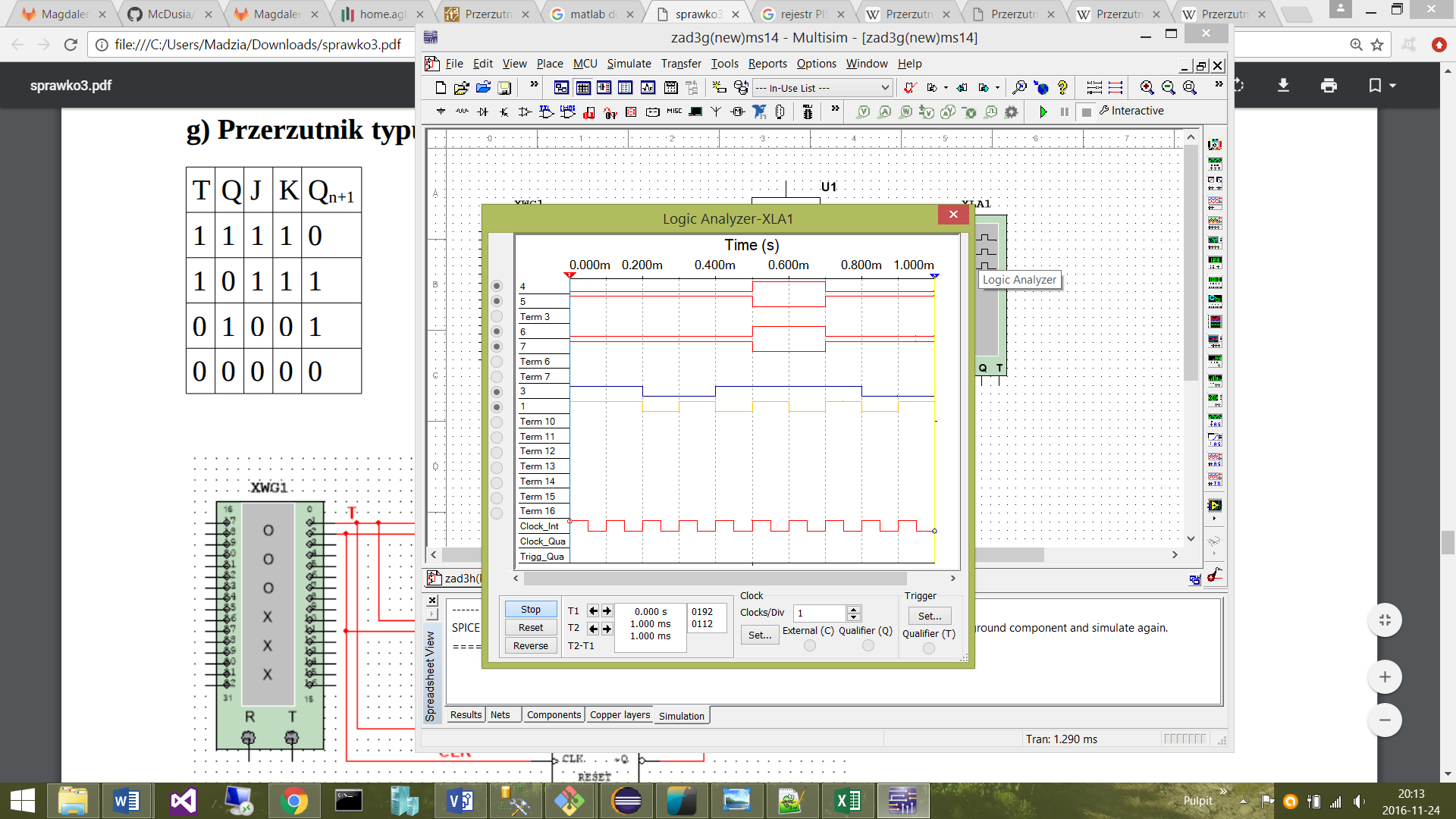


1. Przerzutnik typu „T” bazujący na synchronicznym przerzutniku „JK”.

Przerzutnik T to nic innego jak przerzutnik JK z połączonymi wejściami J i K. Gdy wejście T ma stan niski L, to wyjście zachowuje swoją wartość, gdy natomiast ma ono poziom H, to przerzutnik zmienia stan wyjścia na przeciwny.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **T** | **Q** | **J** | **K** | **Q(n+1)** |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 |



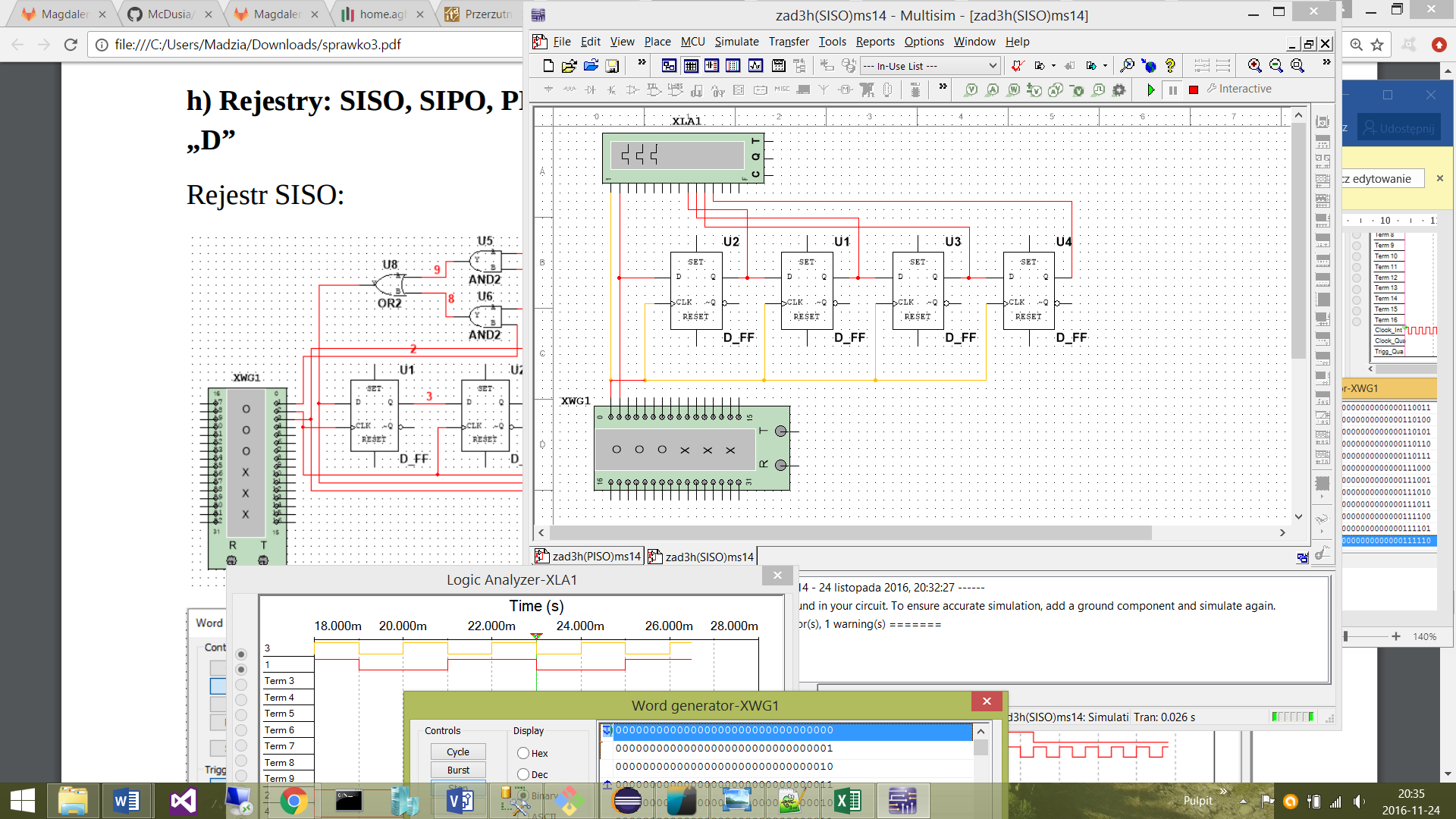


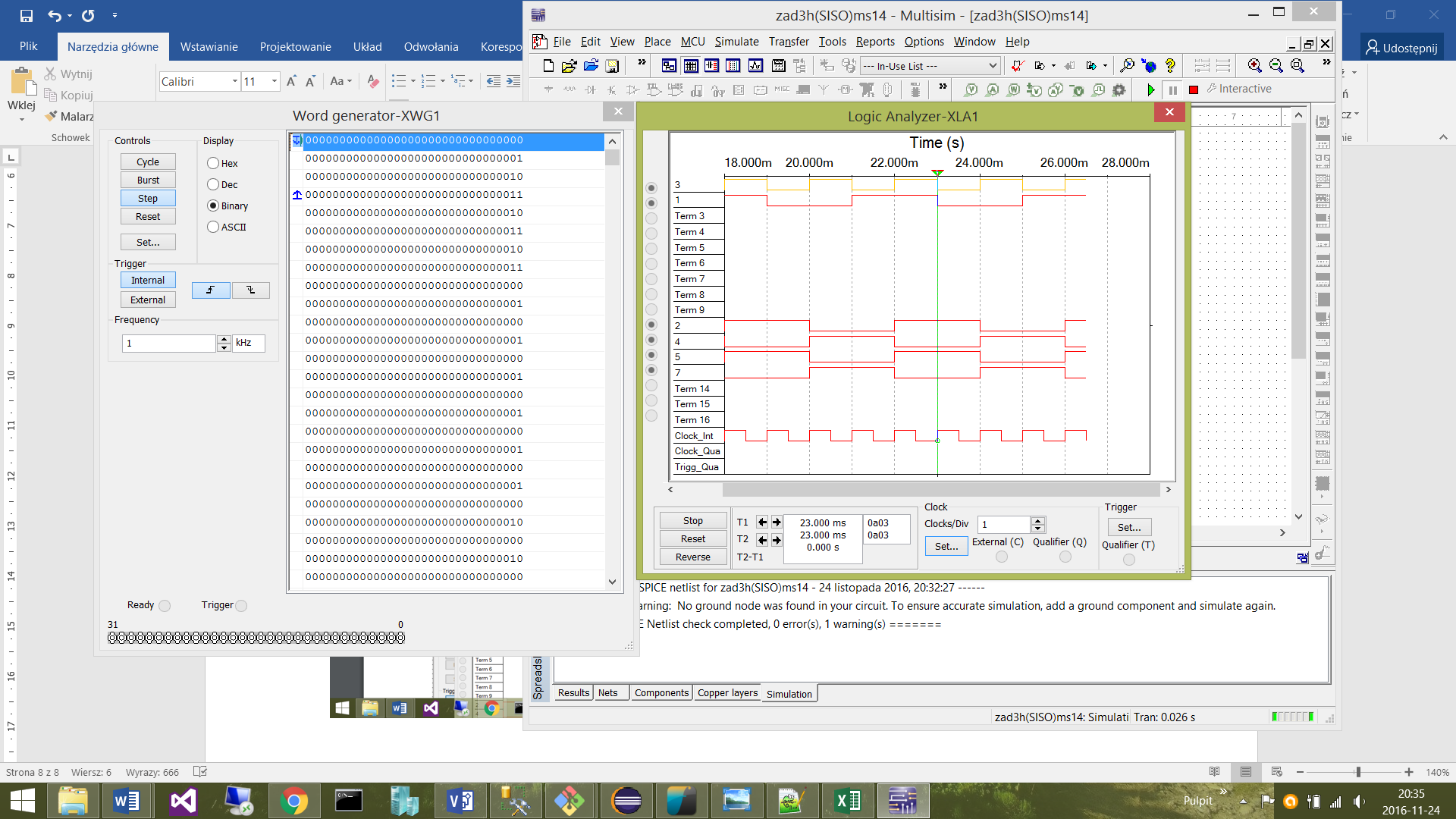
1. Różne rodzaje czterobitowych rejestrów zbudowane na synchronicznych przerzutnikach typu „D” i bramkach logicznych.

Rejestr przesuwający to kaskadowy układ służący do przechowywania informacji.

1. SISO – serial in, serial out – rejestr, który realizuje szeregowe wprowadzanie i szeregowe wyprowadzanie informacji.

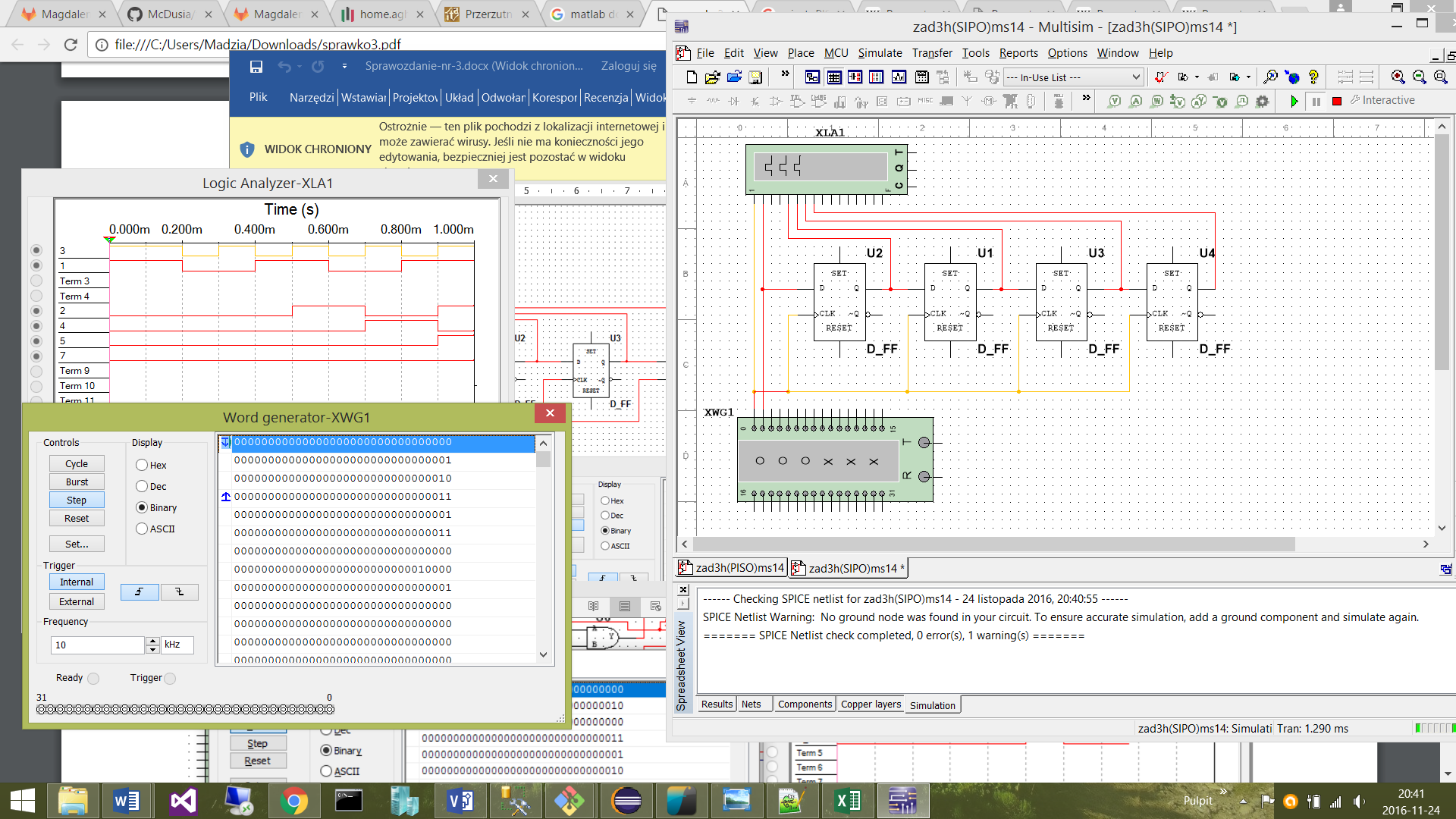
Tak naprawdę poniższy układ powinien mieć wyjście na Logic Analyzer tylko z ostatniego przerzutnika, ale podłączone wszystkie wyjścia, aby łatwiej było zobaczyć, że na wyjściu ostatnim jest przepisywany bit z poprzedniego przerzutnika, a na poprzedni (3) przerzutnik jest przepisywany z 2 przerzutnika, na drugi z pierwszego, a na pierwszy jest przepisywany sygnał wejściowy z Word Generatora.

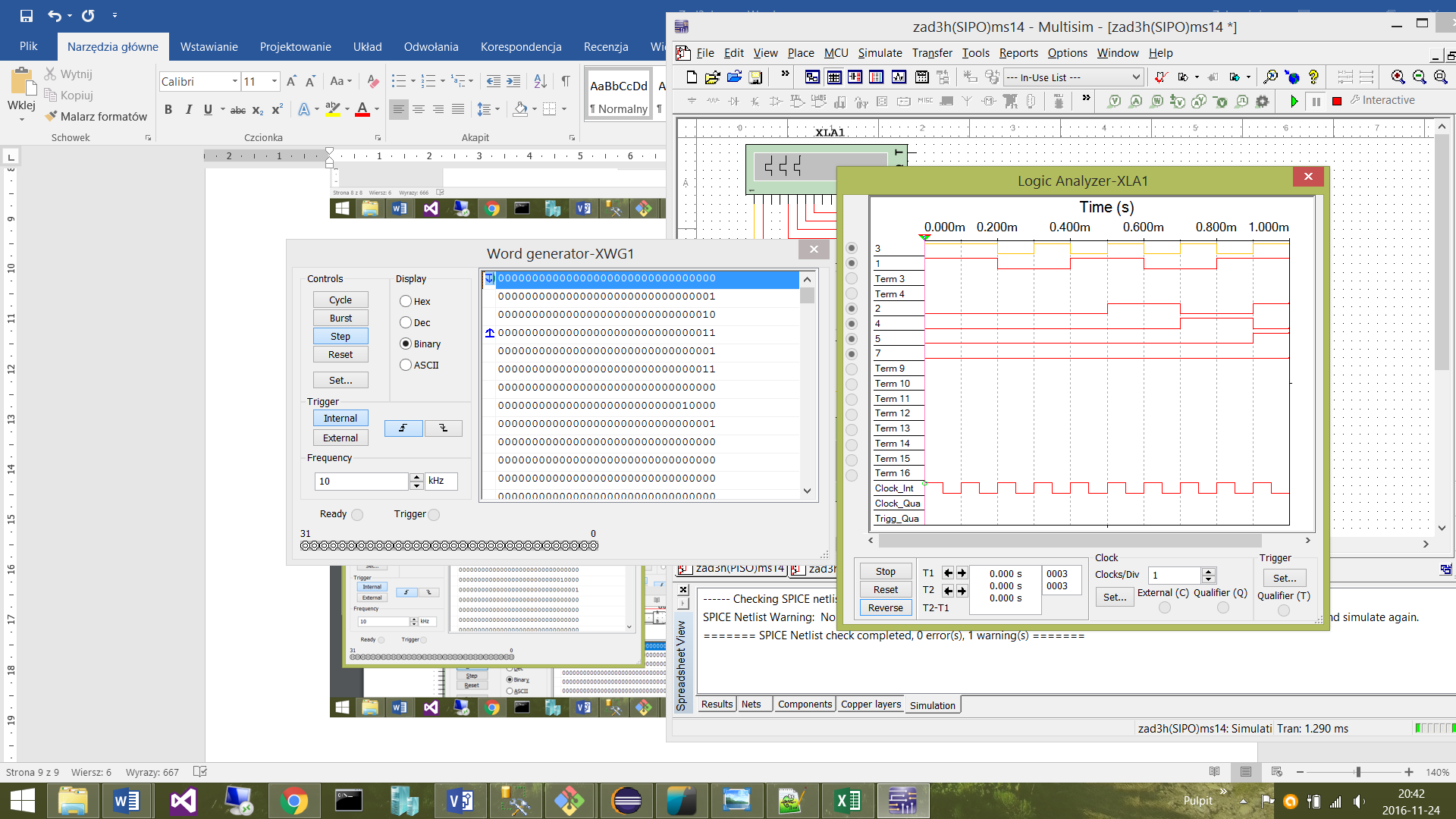
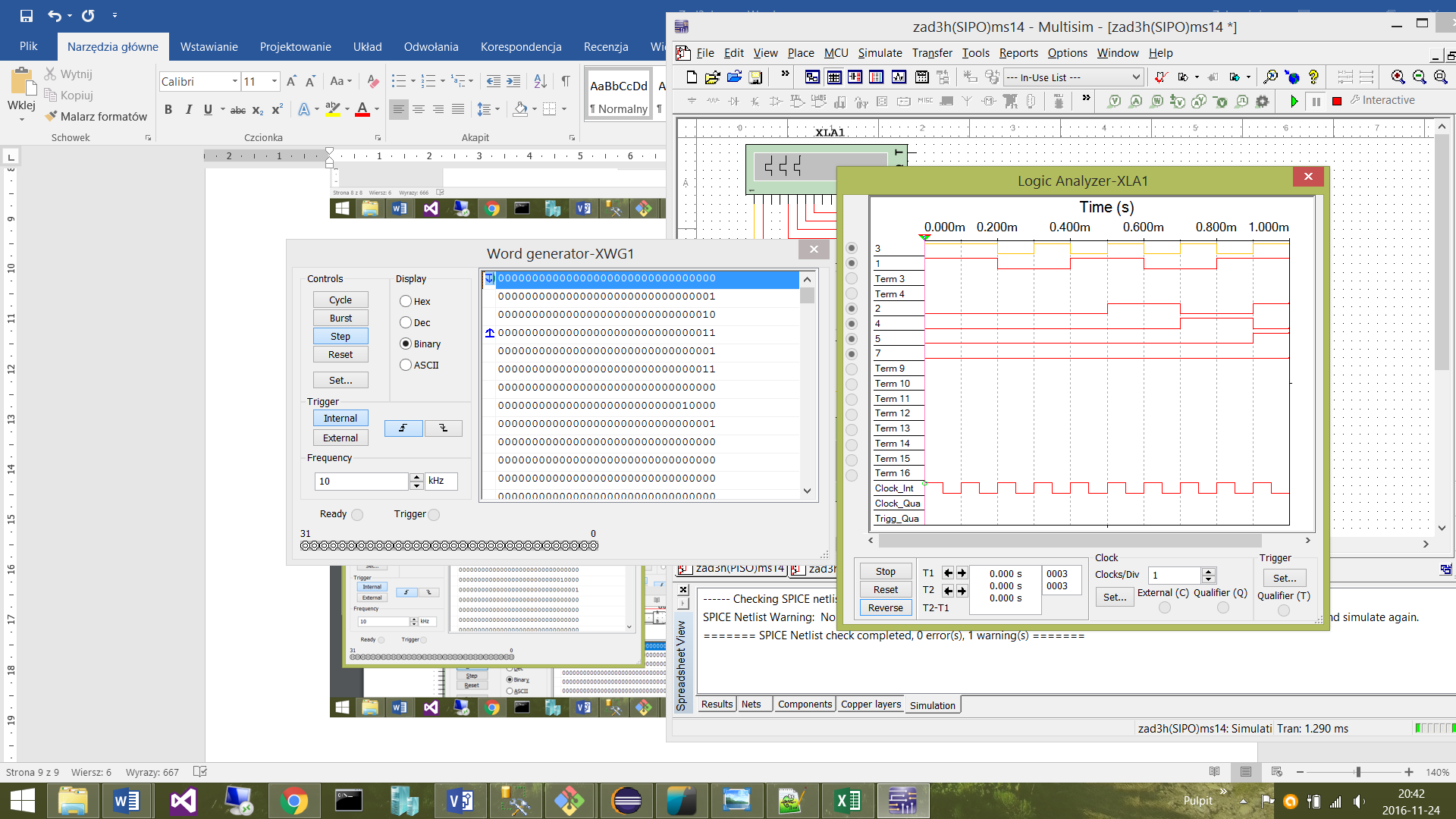




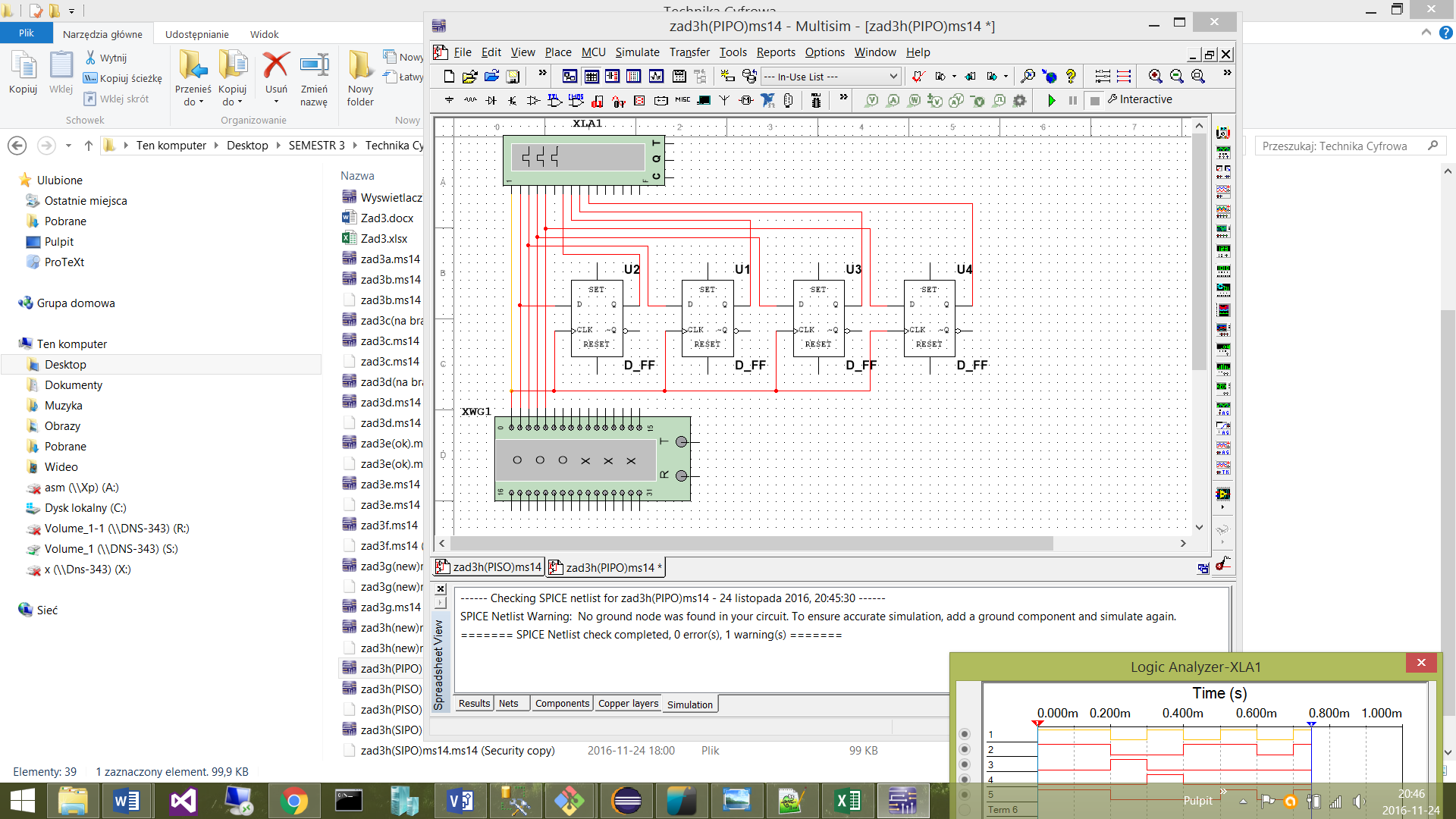
1. SIPO – serial in, parallel out, rejestr realizujący szeregowe wpisywanie i równoległe wypisywanie informacji.

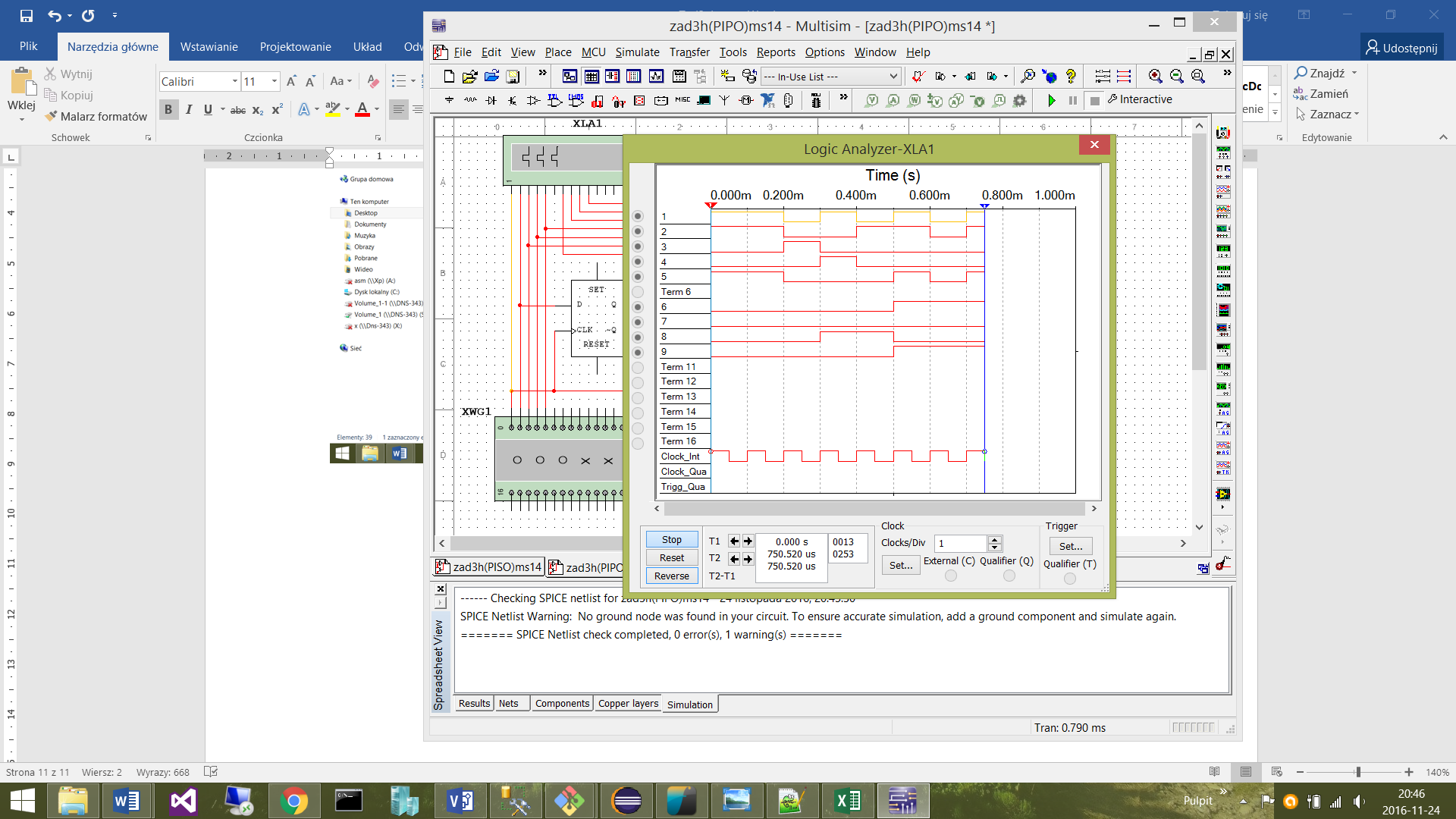
W tym przypadku wyprowadzenie na Logic Analyzer wyjść wszystkich przerzutników ma zasadnicze znaczenie, ponieważ z zasady na wyjściu układu powinniśmy otrzymać sygnał ze wszystkich przerzutników jednocześnie (równolegle ze wszystkich).



1. PIPO – parallel in, parallel out, rejestr realizujący równoległe wczytywanie informacji i równoległe jej odczytywanie. Dopóki nie zostanie podany rosnący sygnał zegarowy, informacja będzie zapamiętana wewnątrz przerzutników.





1. PISO – parallel in, serial out.

Ten rejestr ma specjalny bit sterujący (zaznaczony kolorem granatowym). Podanie na niego sygnału niskiego powoduje, ze układ wpisze (write) na wszystkie przerzutniki jednocześnie (równolegle) sygnał wejściowy. Podanie sygnału wysokiego będzie oznaczało, że układ zacznie przesuwać informację (shift) tzn. z 4 przerzutnika przekaże na wyjście rejestru, z 3 do 4 itd. Należy przy tym pamiętać, że jakakolwiek zmiana odbywa się tylko wtedy, gdy sygnał Clock jest narastający, we wszystkich innych przypadkach nic się nie zmieni.

