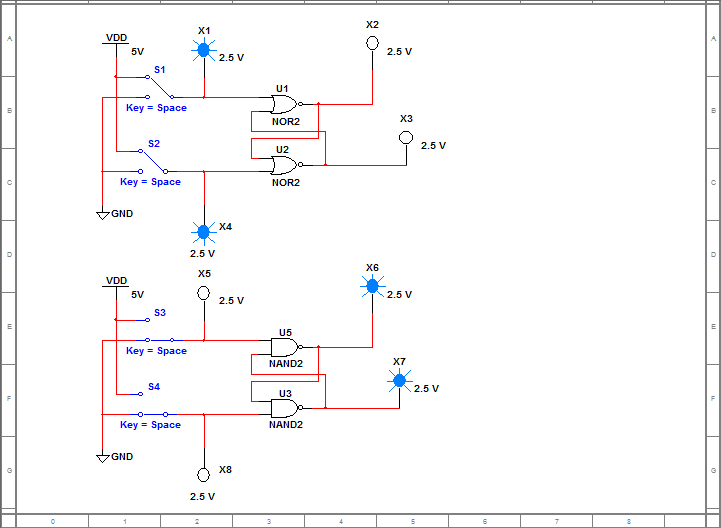
Sprawozdanie nr 3

Daniel Kłódka

Zadanie 1.

Zrealizować asynchroniczny przerzutnik "RS” najpierw w oparciu o bramki NAND a następnie w oparciu o bramki NOR. Po czym przetestować obydwie wersje przerzutnika RS w programie Multisim. Sporządzić tabele prawdy, opisać różnice i ewentualnie inne wnioski.

Realizacja przerzutnika RS na bramkach NAND oraz NOR jest identyczna jeśli chodzi o schemat połączeń lecz zachowanie jakie uzyskamy tworząc przerzutnik z bramek NOR a z bramek NAND różni się znacząco. Schemat połączeń jak i symulacja działania znajduje się na poniższym GIF’ie.

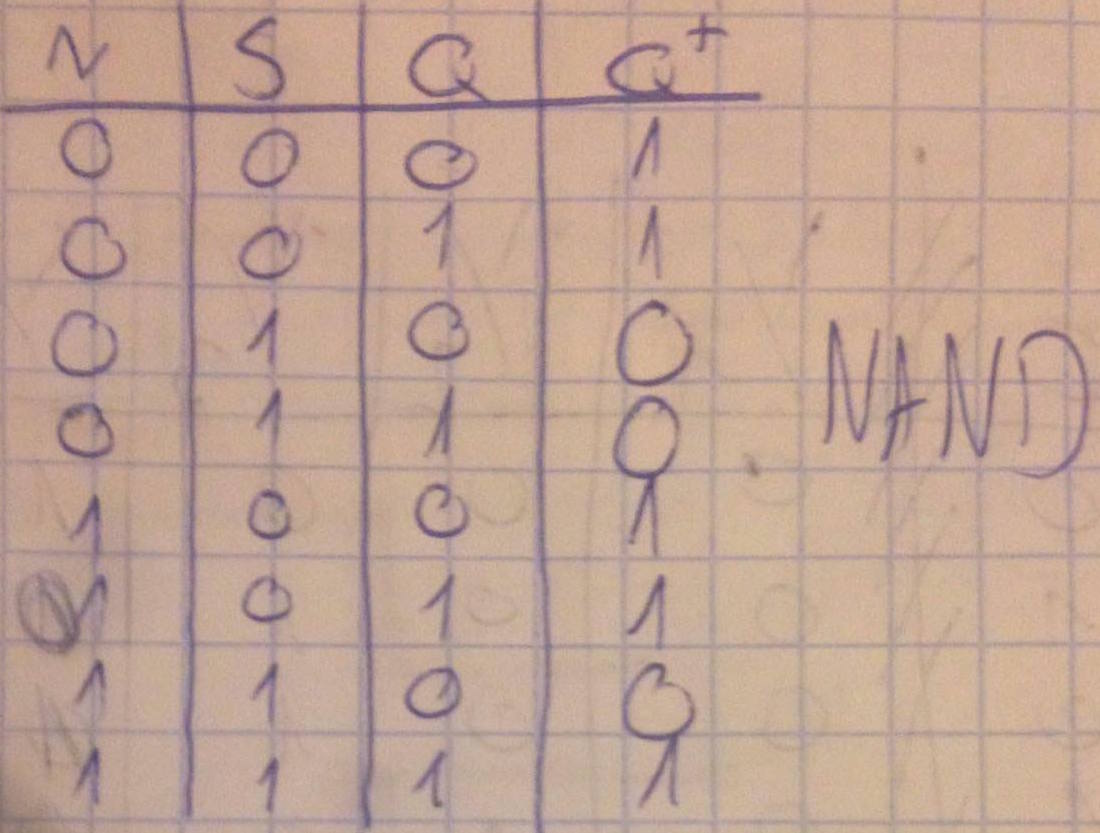


Analiza układu pozwala nam stwierdzić, że układ w zależności od tego na jakiego rodzaju bramkach został zbudowany, reaguje inaczej na sygnały wejściowe.

Przerzutnik stworzony na bramkach NOR posiada tryb pamiętania ostatniego stanu gdy na wejściu R i S panuje stan niski. Resetowanie lub Setowanie następuje gdy na odpowiednie wejście podamy stan wysoki. Podanie dwóch wysokich sygnałow na wejście R i S nie powinno mieć miejsca ponieważ dla takiej bramki występuje wtedy stan zabroniony i wyjście Q jak i jego zanegowany odpowiednik są wtedy w stanie niskim.

Przy bramce NAND sytuacja jest odwrotna. Stan zabroniony występuje gdy na wejścia R i S podamy stan niski, wtedy na wyjściu Q i jego zanegowanym odpowiedniku panuje stan wysoki. Podanie na wejście S lub R stanu niskiego powoduje odpowiednio setowanie lub resetowanie przekaźnika a gdy na każdym z wejść mamy stan wysoki to przekaźnik jest w stanie pamiętania.

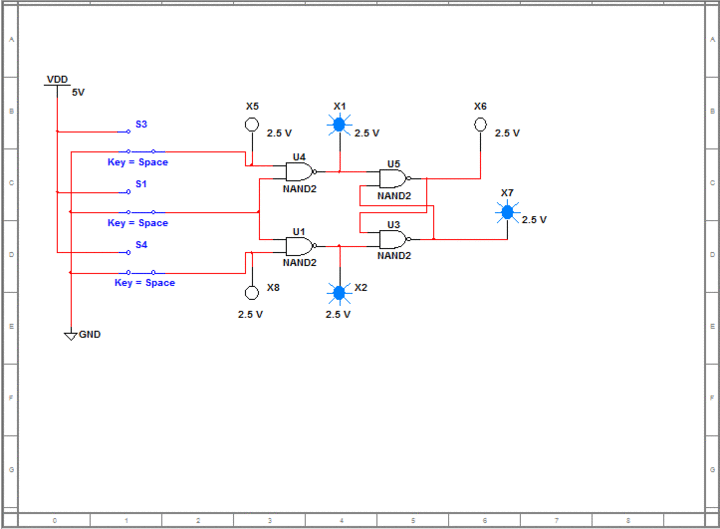
Na podstawie tych informacji możemy stworzyć tabele prawd tych przekaźników.



Zadanie 2.

W oparciu o dowolne bramki logiczne zbudować i przetestować synchroniczny przerzutnik "RS", reagujący na stan sygnału zegarowego równy jeden.

Aby uzyskać synchroniczny przerzutnik RS wykorzystamy schemat układu utworzony w poprzednim zadaniu (Przerzutnik RS na bramkach NAND). Zauważmy, że synchornizacja ma następować tylko gdy stan sygnały zegarowego równy jest jeden. Widać tutaj od razu, iż należy zastosować bramkę AND dla każdego z sygnałów wejściowych z sygnałem zegarowym. Lecz zwykła bramka AND przy braku sygnału zegarowego na wyjściu będzie dawała nam stan niski co wprowadzi nasz przerzutnik w stan zabroniony, a chcielbyśmy aby brak sygnału zegarowego wprowadził przerzutnik w stan pamiętania. Rozwiązanie jest proste, należy dołożyć nie bramkę AND a bramkę NAND co pozwoli nam uzyskać żądane zachowanie przerzutnika. GIF ze schematem połączeń jak i przetestowaniem układu znajduje się poniżej.



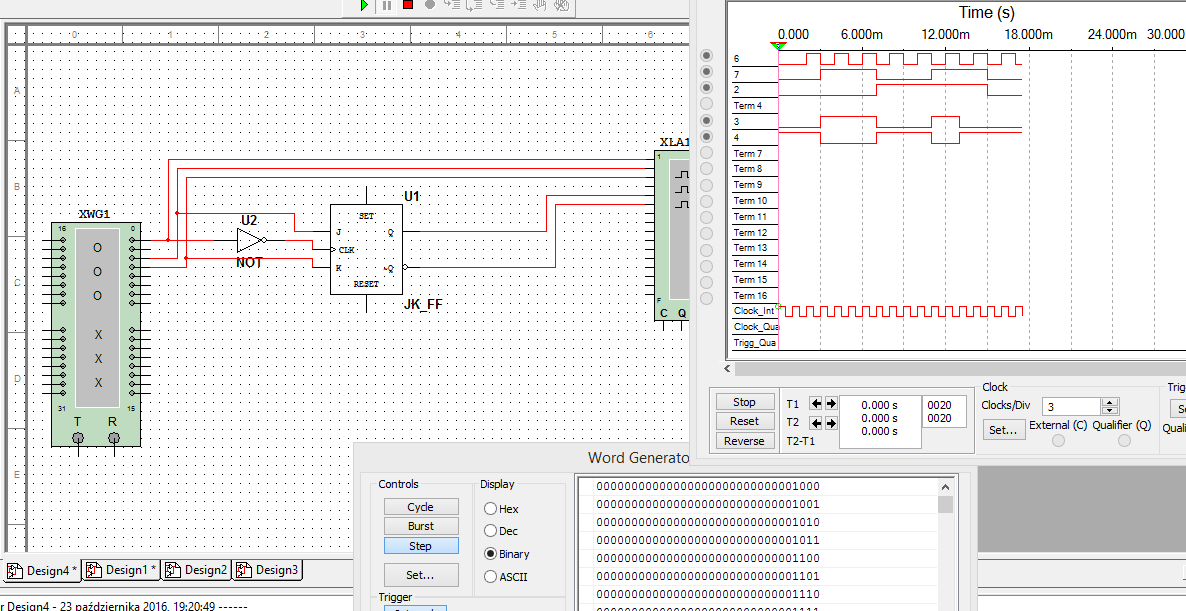
Widać, iż układ działa prawidłowo zgodnie z oczekiwaniami.

Zadanie 3.

W programie Multisim przetestować i sporządzić tablicę prawdy dla synchronicznego przerzutnika "JK" reagującego na opadające zbocze sygnału zegarowego.

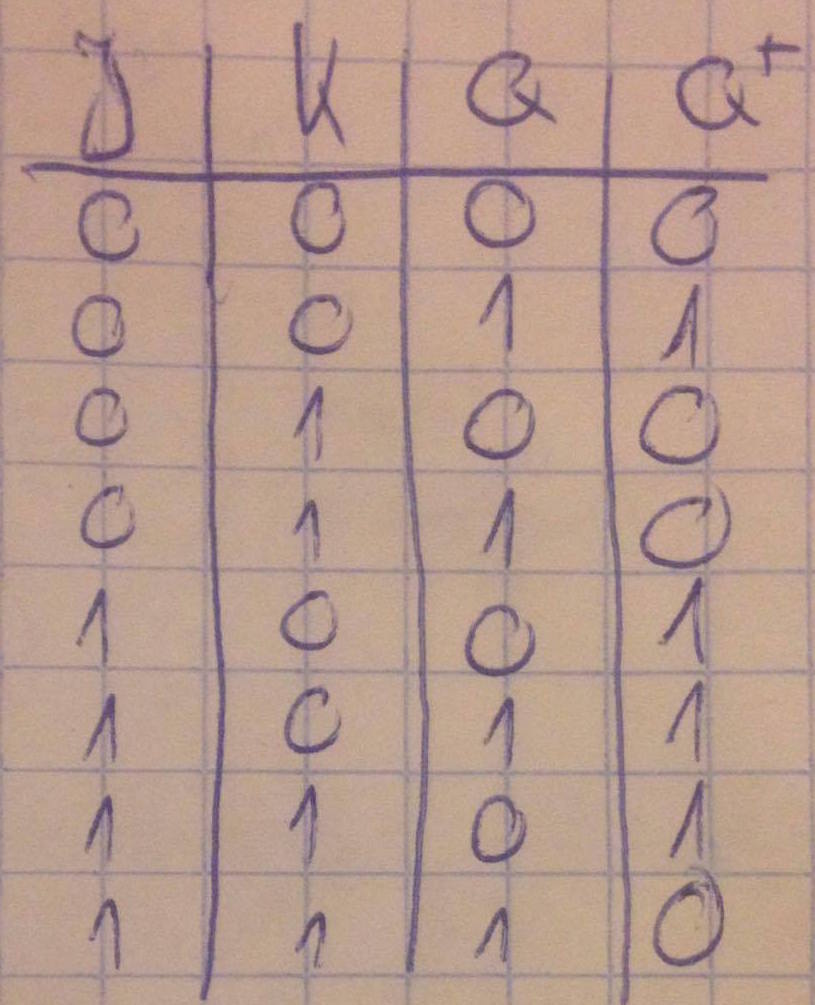
Z powodu braku przerzutnika JK reagującego na opadające zbocze sugnału zegarowego zastosowałem bramkę NOT na wejściu zegarowym.

Schemat połączeń przerzutnika z przyrządami pomiarowymi znajduje się na poniższym rysunku.



Na podstawie otrzymanych wykresów z analizatora logicznego zauważamy, iż przerzutnik JK działa prawie identycznie jak przerzutnik RS za wyjątkiem tego, że w momencie gdy przerzutnik RS został by ustawiony w stan zabroniony I na wyjściu stale utrzymywał by jedną stałą wartość to przerzutnik JK za każdym razem gdy zostaje synchronizowany, zmienia swój stan na przeciwnym.

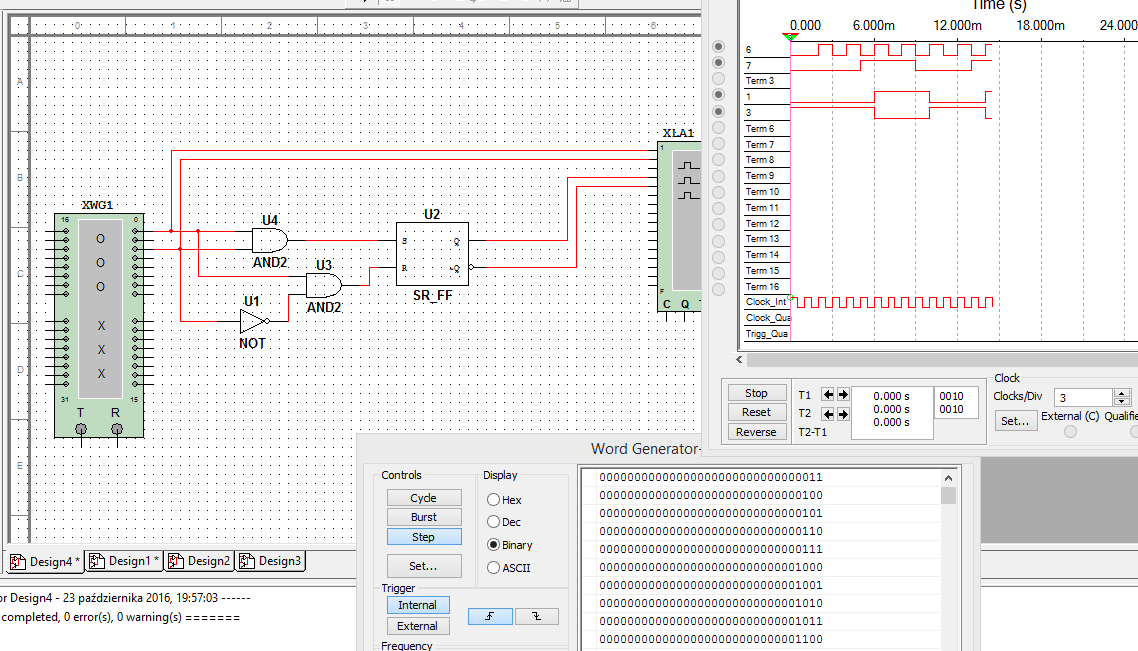
Zgodnie z otrzymanym wynikiem możemy stworzyć tabele prawdy dla tego przerzutnika.



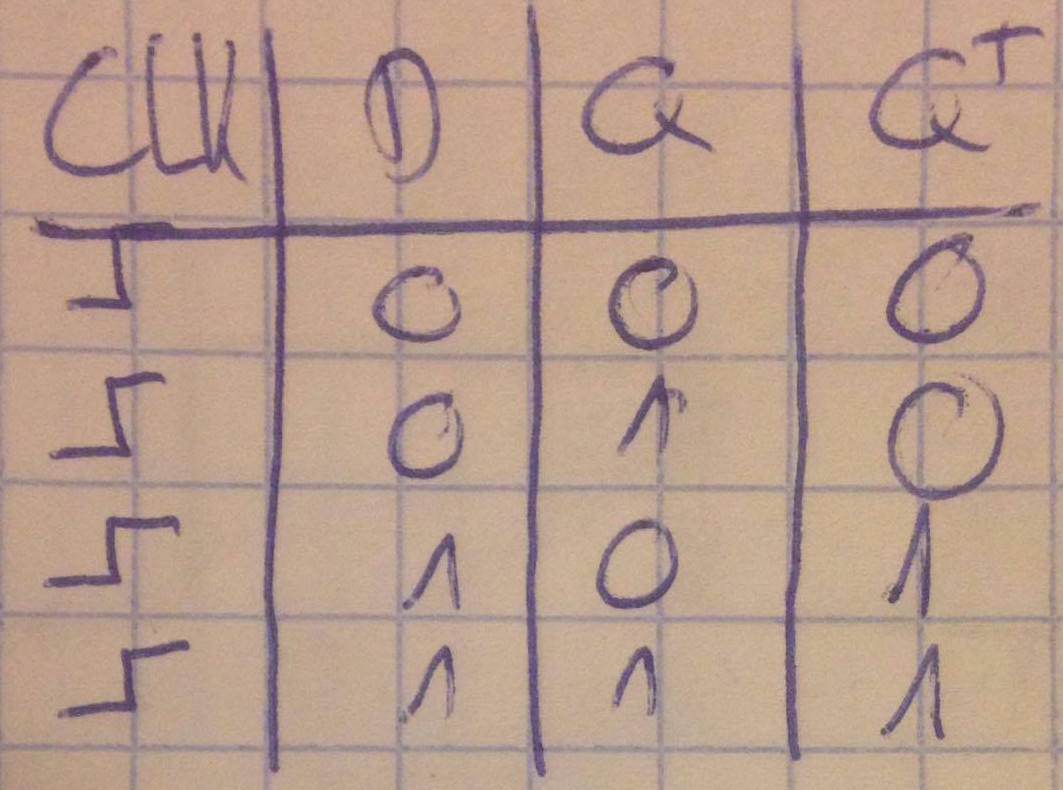
Zadanie 4.

Bazując na asynchronicznym przerzutniku RS i ewentualnie bramkach logicznych, zbudować i przetestować przerzutnik typu "D".

Wiedząc to, iż przerzutnik D jest synchronizowany sygnałem zegarowym możemy odrazu zauważyć, iż wejście S będzie musiało być iloczynem sygnału zegarowego i wejścia D. Wejście przerzutnika R także będzie musiało być iloczynem sygnału zegarowego i w tym przypadku zanegowanego wejścia D ponieważ chcemy resetować układ gdy wejście D jest w stanie niskim.



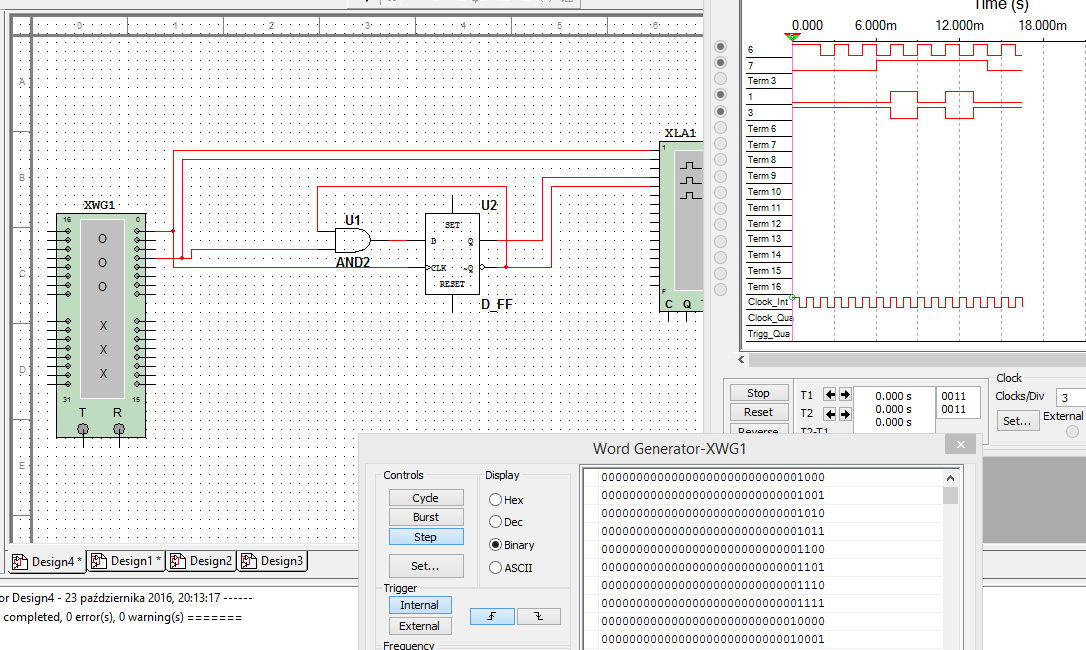
Na podstawie uzyskanych wyników możemy stworzyć tabele prawdy przerzutnika D. W tym przypadku przerzutnik nie jest synchroniowany zboczem narastającym a sygnałem wysokim lecz sporządzona tabelka będzie odpowiadała rzeczywistemu przerzutnikowi.

Brak sygnału synchronizującego wprowadza przerzutnik w stan pamiętania.

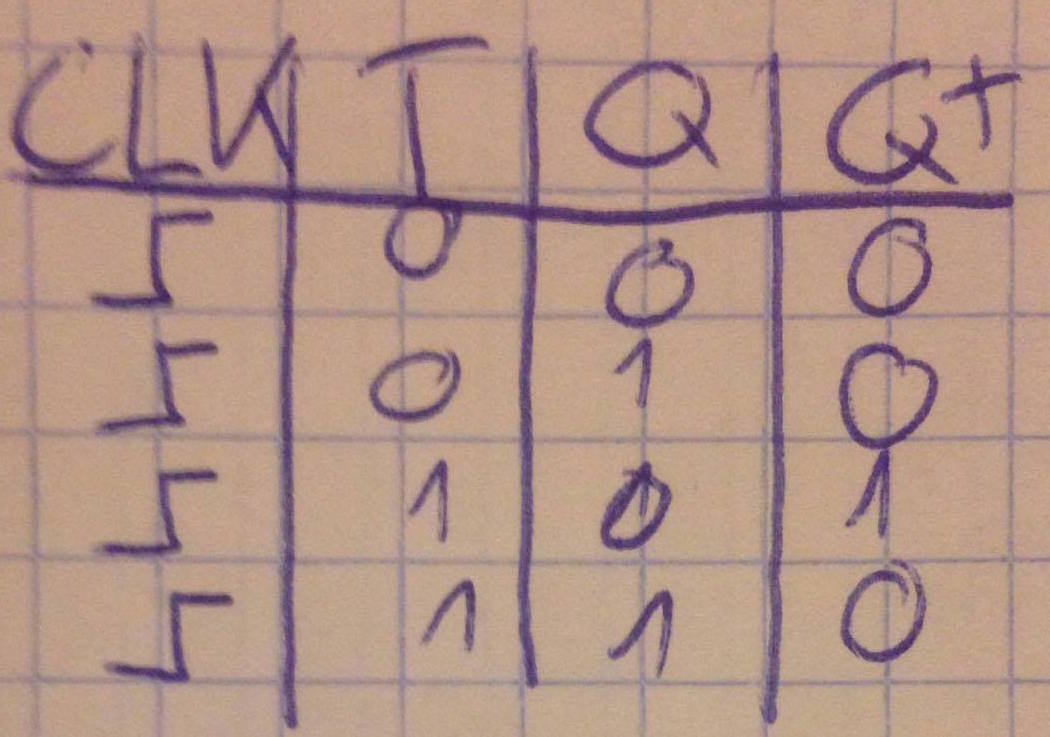
Zadanie 5.

Bazując na synchronicznym przerzutniku "D" i ewentualnie bramkach logicznych, zbudować i przetestować przerzutnik typu "T".

Wiedząc, że przekaźnik typu T działa w sposób taki, iż gdy na wejściu T mamy stan wysoki oraz na wejściu zegarowym mamy sygnał synchronizujący stan wyjścia zmienia się na przeciwny. Analogicznie do przerzutnika JK dla dwóch stanów wysokich na wejściu. Gdy mamy na wejściu T stan niski lub mamy dowolny przy braku sygnału synchronizującego to układ jest w stanie pamiętania. Widzimy tutaj, iż sygnał wejściowy T będzie musiał być doprowadzony do wejścia D lecz nie bezpośrednio. Lecz aby zabezpieczyć się przed ciągłym stamen wysokim na wyjściu wystarczy sygnał wejściowy T połączyć bramką AND z zanegowanym wyjściem Q. W ten sposób otrzymamy żądane zachowanie przekaźnika. Schemat połączeń jak I wykresy analizatora logicznego znajdują się na poniższym obrazku.



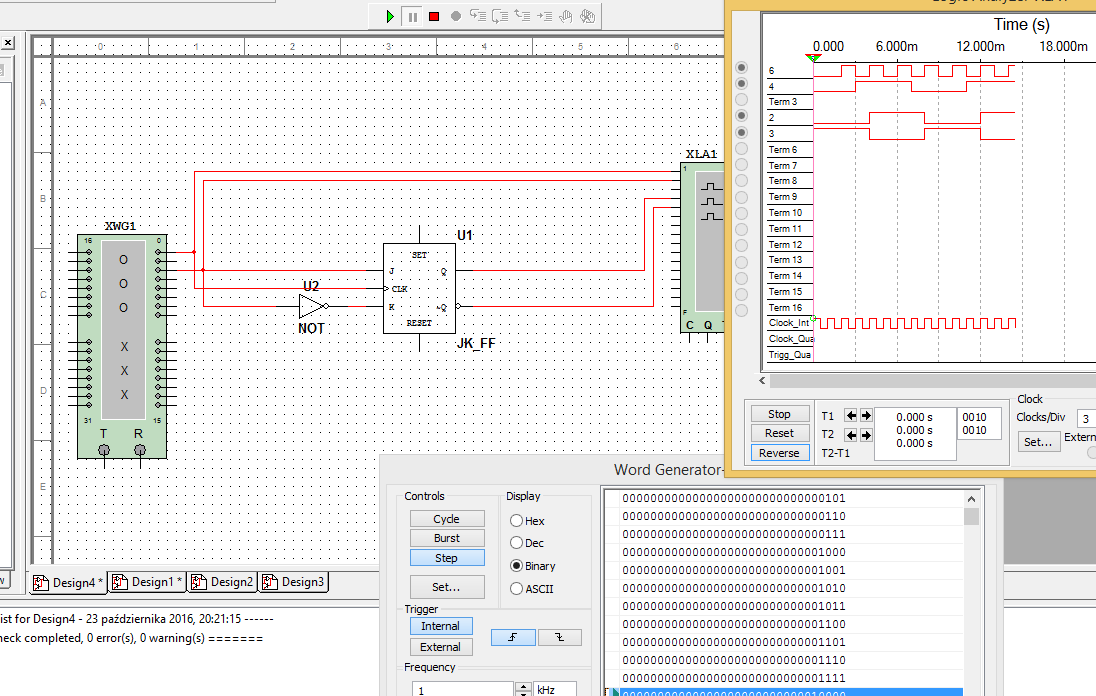
Na podstawie otrzymanych wyników możemy stworzyć tabele prawdy dla takiego przekaźnika.

Brak sygnału synchronizującego powoduję wejście układu w stan pamiętania.

Zadanie 6.

Bazując na synchronicznym przerzutniku "JK" i ewentualnie bramkach logicznych, zbudować i przetestować przerzutnik typu "D".

Zadanie to jest dość proste, wystarczy zauważyć że dla zera na wejściu D chcemy mieć stan niski więc jest on równoznaczny ze stanem niskim na wejściu J i stanem wysokim na wejściu R. Stan wysoki na wejściu D powinien być równoznaczny ze stanem wysokim na wejściu J i stanem niskim na wejściu R. Widzimy tutaj wprost, iż sygnał wejściowy D trzeba zanegować przed podaniem na wejście R. Schemat połączeń na rysunku poniżej.

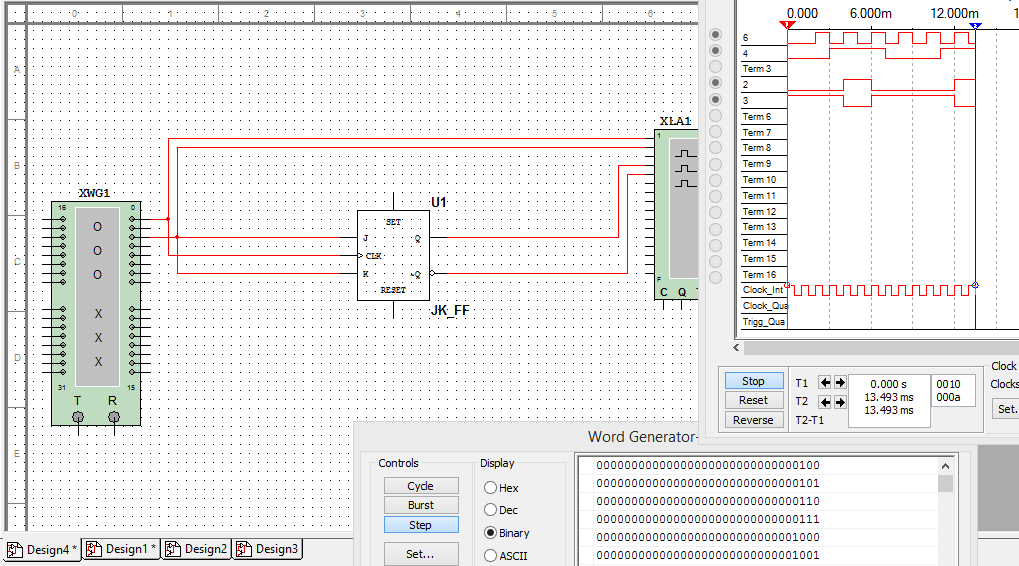


Na podstawie otrzymanych wyników możemy stworzyć tablice prawdy. (Taka sama jak w zadaniu 4)

Zadanie 7.

Bazując na synchronicznym przerzutniku "JK" i ewentualnie bramkach logicznych, zbudować i przetestować przerzutnik typu "T".

Zadanie jest proste jeżeli pamiętamy jakie wyciągneliśmy wnioski w zadaniu 5. Widzimy iż sygnał wejściowy T należy doprowadzić do wejść J i K bezpośrednio. Schemat poniżej.

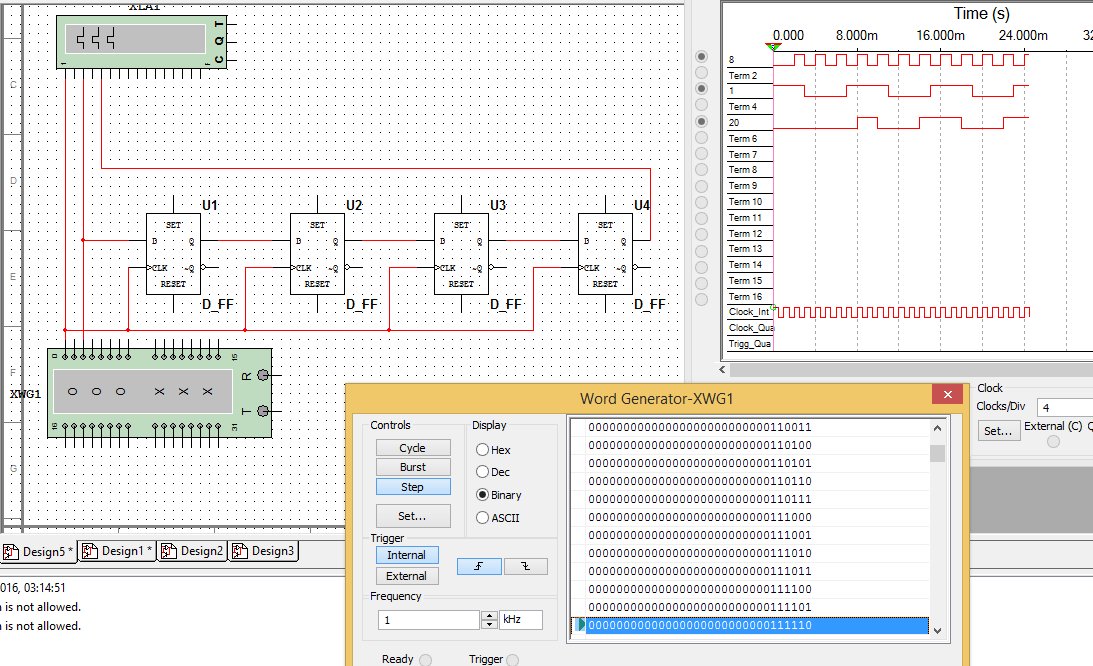


Na podstawie otrzymanych wyników możemy stworzyć tablice prawdy. (Taka sama jak w zadaniu 5)

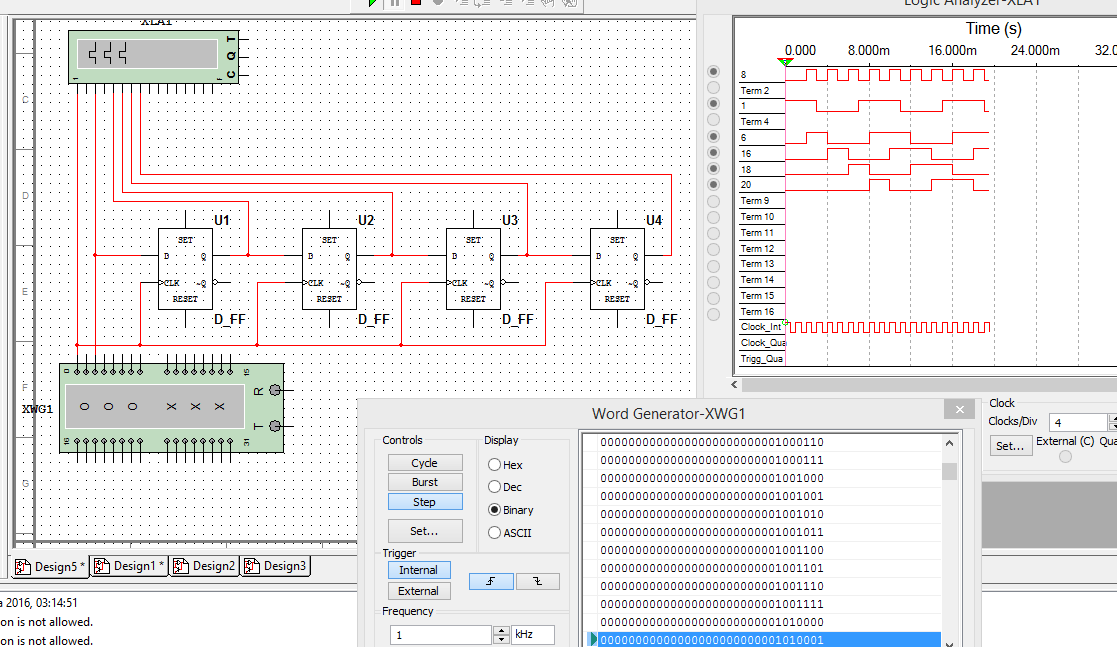
Zadanie 8.

Bazując na synchronicznych przerzutnikach "D" i ewentualnie bramkach logicznych, zaproponować schematy i przetestować czterobitowe rejestry: SISO, SIPO, PIPO i PISO.

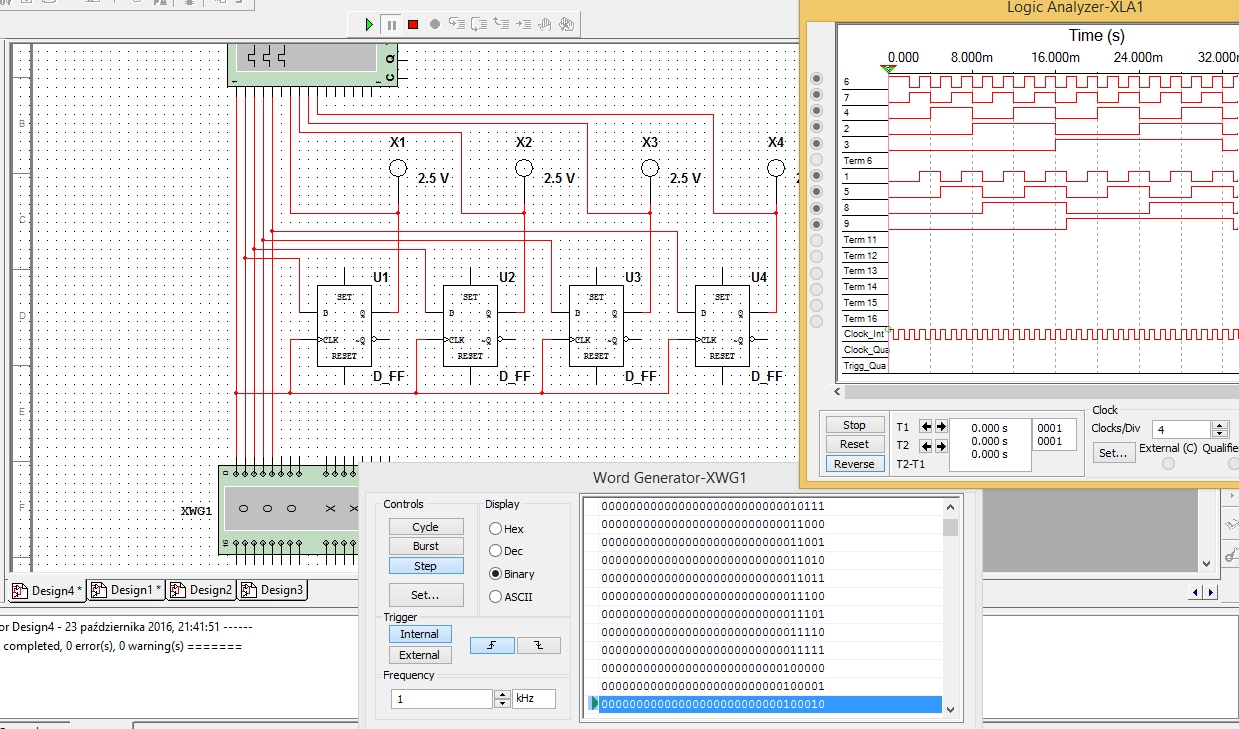
Rejestr SISO jest prostym układem połączenia szeregowego przerzutników D w ten sposób, że każde wyjście jest połączone z wejściem kolejnego za wyjątkiem ostatniego przerzutnika którego wyjście jest miejscem sczytywania danych. Układ działa w taki sposób, iż podając sekwencję bitów na wejściu pierwszego przerzutnika otrzymamy na wyjściu tą samą sekwencję jeśli nastąpi pełen cykl zapisu I odczytu.



Rejestr SIPO jeśli chodzi o zapis danych działa identycznie jak poprzedni rejestr SISO lecz odczyt odbywa się w inny sposób. Odczyt odbywa się równolegle, nie mamy tutaj jednego wyjścia jak poprzednio. Schemat połączeń jest identyczny jak poprzednio, różni się tym iż do każdego wyjścia przerzutnika dodatkowo dokładamy węzeł a sygnał z tego węzła jest dla nas wyjściem z którego będziemy odczytywać dane, w ten sposób otrzymujemy 4 wyjścia. Odczyt odbywa się po pełnym cyklu zapisu do rejestru.



Rejestr PIPO jest najszybszym rejestrem ze wszystkim ponieważ odczyt I zapis odbywa się praktycznie w tym samym czasie. Każde wejście przekaźnika jest indywidualnie podłączone do odpowiadającego mu sygnału wejściowego a każde wyjście jest bezpośrednio wyjściem z którego sczytujemy dane.



Rejestr PISO, w przypadku tego rejestru zapis odbywa się natychmiastowo lecz aby odczytać dane musimy przejść cały cykl odczytu. Dodatkowo wymaga on zabezpieczenia przed tym aby nie nadpisać danych których właśnie czytamy. Dlatego na układzie poniżej znajduje się kombinacja bramek logicznych która w zależności od sygnału wejściowego wprowadza układ w stan odczytu lub zapisu.

