Jan Chyczyński
Błażej Nowicki
Bartłomiej Słupik
Przemysław Węglik

Technika cyfrowa
Sprawozdanie 1

7 marca 2022

1. Zadanie 1a

Zadanie polega na zaprojektowaniu układu realizującego funkcję logiczną:

$$Y = \overline{A}C + B(A + B)$$

1.1. Rozwiązanie teoretyczne

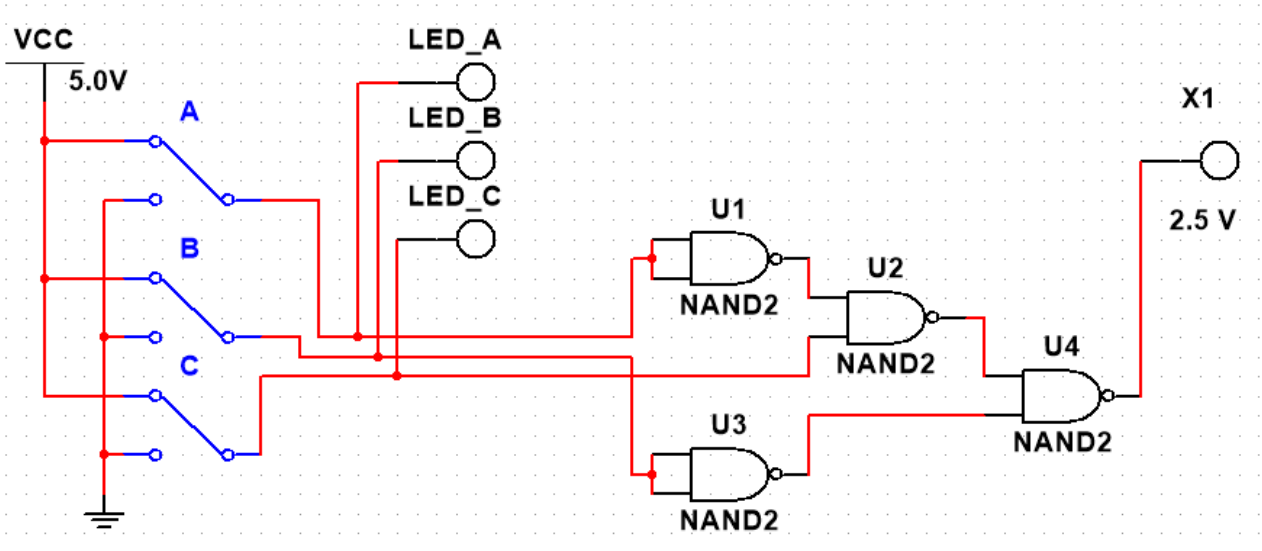
Układ można zrealizować wyłącznie dzięki użyciu bramek NAND ponieważ bramka NAND jest systemem funkcjonalnie pełnym tzn. korzystając wyłącznie z niej można przedstawić dowolną funkcję boolowską.

Pierwszym krokiem jest przekształcenie wyrażenie do zawierającego wyłączenie operacje NAND.

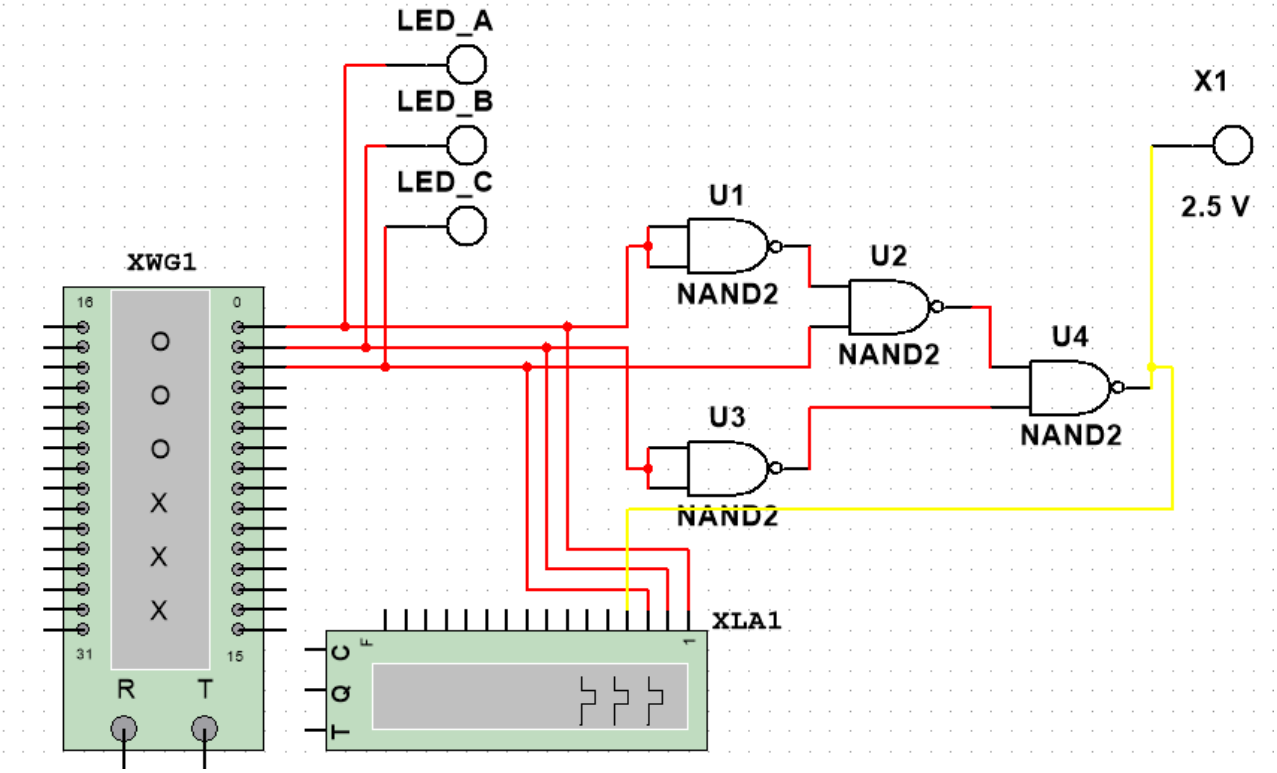
$$\begin{aligned} Y &= \overline{A}C + B(A + B) \\ &= \overline{A}C + B \\ &= \overline{\overline{\overline{A}C}\,\overline{B}} \end{aligned}$$

1.2. Symulacja w programie Multisim

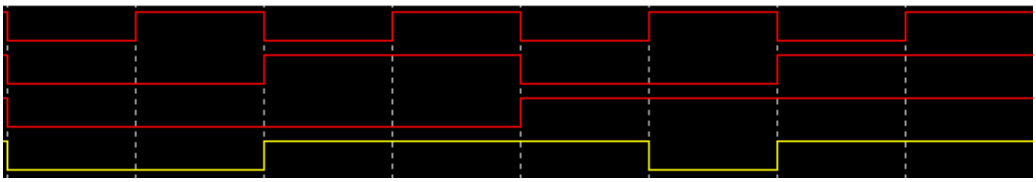
Układ zbudowano w dwóch wersjach: pierwszej, z ręcznymi przełącznikami pozwalającymi na zmianę stanu A, B i C oraz drugiej, z generatorem słów bitowych i analizatorem stanów logicznych.



Rysunek 1.1: Układ z ręcznymi przełącznikami



Rysunek 1.2: Układ z generatorem i analizatorem



Rysunek 1.3: Przebieg czasowy sygnałów. Od góry: A, B, C, Y

1.3. Wnioski

1. Dzięki prawom logiki możemy uprosić skomplikowane wyrażenia w taki sposób, aby używały mniejszej ilości operacji logicznych.
2. Przedstawienie funkcji logicznej tylko za pomocą bramek NAND ma praktyczne zastosowanie, ponieważ komercyjnie dostępne chipy często zawierają kilka bramek tego samego rodzaju (np. 4xNAND, 4xOR itp.). Dzięki uproszczeniu układu do bramek NAND, możemy go skonstruować w rzeczywistością używając tylko jednego chipu 4xNAND zamiast trzech różnych: 4xNOT, 4xOR i 4xAND.
3. Podana funkcja logiczna po drobnym przekształceniu

$$Y = \overline{A}C + B(A + B) = \overline{A}C + B$$

przedstawia równanie charakterystyczne przerzutnika asynchronicznego typu RS. Po podstawieniach

$$\begin{aligned} A &= R \\ B &= S \\ C &= Q_{n-1} \\ Y &= Q_n \end{aligned}$$

otrzymujemy

$$Q_n = S + \overline{R}Q_{n-1}$$

2. Zadanie 1b

Celem zadania jest zaprojektowanie, zbudowanie i przetestowanie układu detekcji liczby pierwszej w binarnym słowie czterobitowym.

2.1. Rozwiązanie teoretyczne

W binarnym czterobitowym słowie można zapisać liczby od 0 do 15. Liczby pierwsze w tym zakresie to 2,3,5,7,11,13. Przy oznaczeniach wartości bitów A,B,C,D w kolejności od najbardziej znaczącego funkcja logiczna ma postać

$$f(A,B,C,D) = \sum(2,3,5,7,11,13)$$

W celu wyznaczenie najprostszej postaci wpisano ją w tabeli Karnough

AB/CD	00	01	10	11
00	0	0	1	1
01	0	1	0	1
10	0	0	0	1
11	0	1	0	0

Tabela 2.1: Tabela Karnough dla badanej funkcji

Z tabeli można odczytać postać funkcji logicznej

$$f(A,B,C,D) = \widetilde{A}\widetilde{B}C + \widetilde{A}B\widetilde{C}D + AB\widetilde{C}D + A\widetilde{B}CD + \widetilde{A}BCD$$

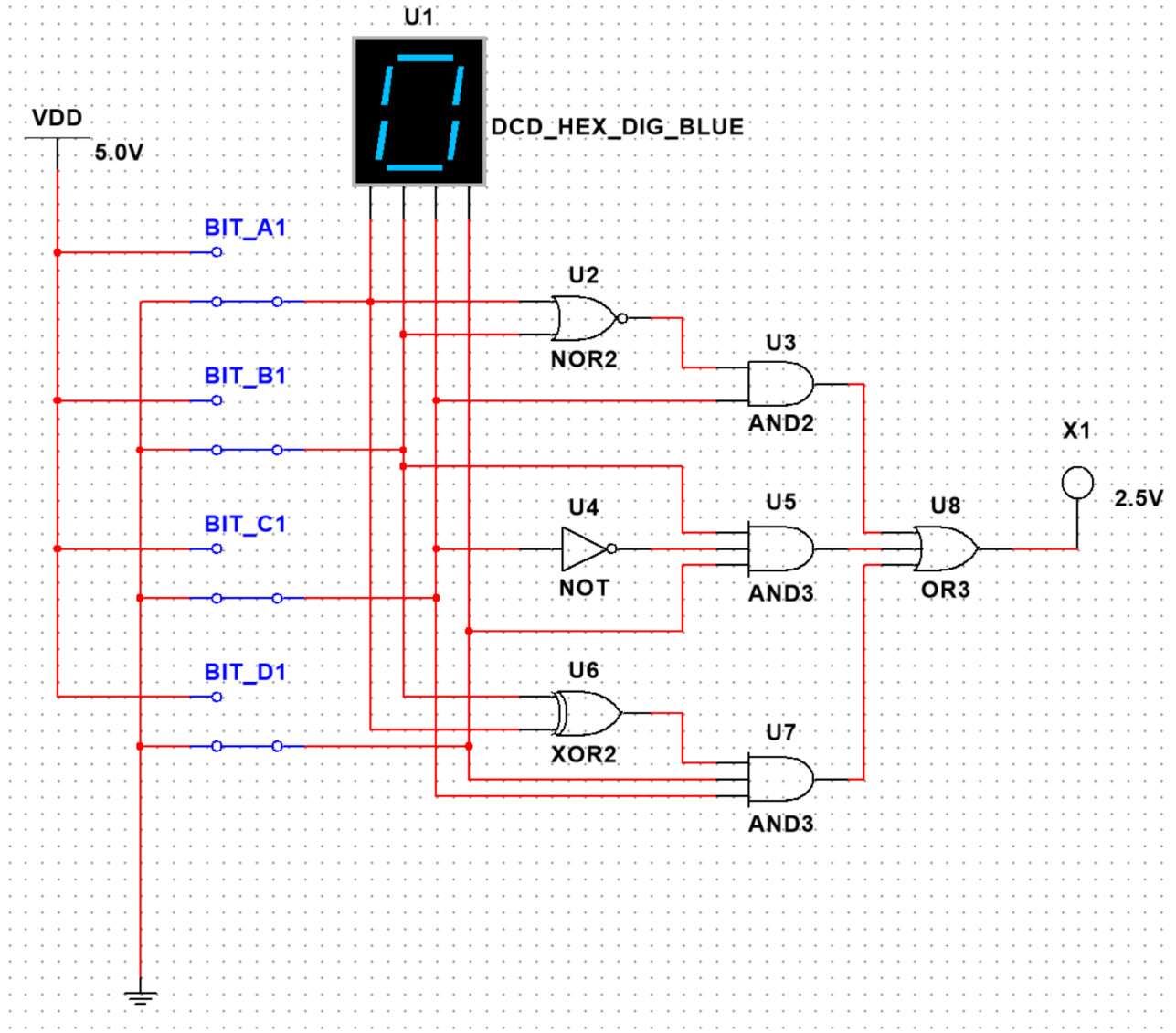
Stosując prawa przekształceń równoważnych

$$f(A,B,C,D) = \widetilde{A}\widetilde{B}C + (\widetilde{A}+A)B\widetilde{C}D + A\widetilde{B}CD + \widetilde{A}BCD$$
$$f(A,B,C,D) = \widetilde{A}\widetilde{B}C + B\widetilde{C}D + (A\overline{B} + \overline{A}B)CD$$

2.2. Budowa i symulacja układu w programie mulitsim

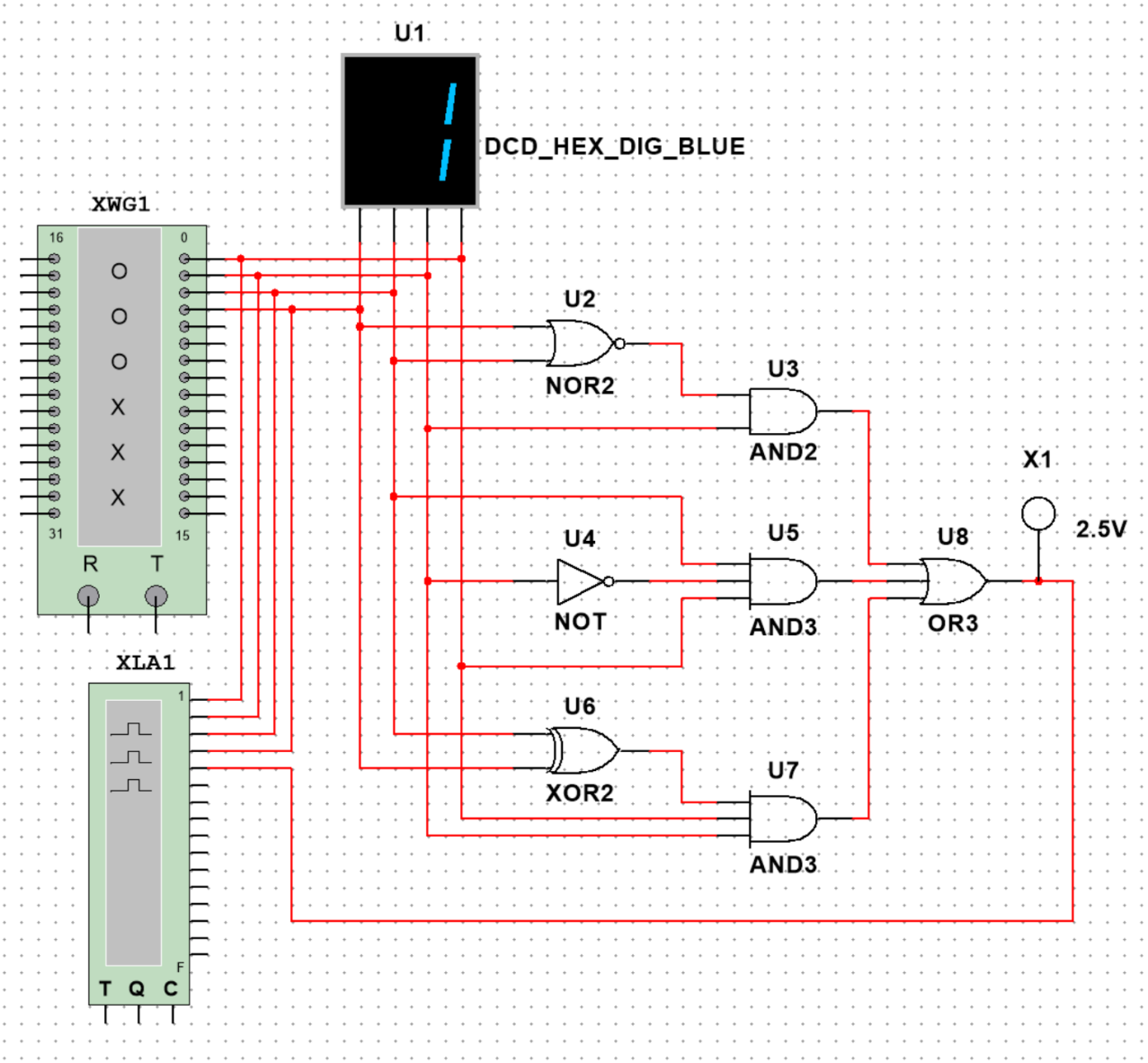
W celu konstrukcji układu należy przełożyć wzór funkcji na bramki logiczne. Funkcja jest alternatywą trzech składników. Pierwszy z nich $\widetilde{A}\widetilde{B}C$ można reprezentować przez bramkę NOR2 i AND2. Wyrażenie $B\widetilde{C}D$ można zastąpić bramką NOT i AND3 a $(A\overline{B}+\overline{A}B)CD$ jako połączenie XOR2 i AND3. Trzy wyrażenia połączone zostały bramką logiczną OR3. Słowa bitowe reprezentujące testowane liczby można wprowadzać przez cztery przełączniki. Gdy wprowadzone zostanie słowo reprezentujące liczbę pierwszą zapala się dioda.

Opisany schemat w programie multisim.

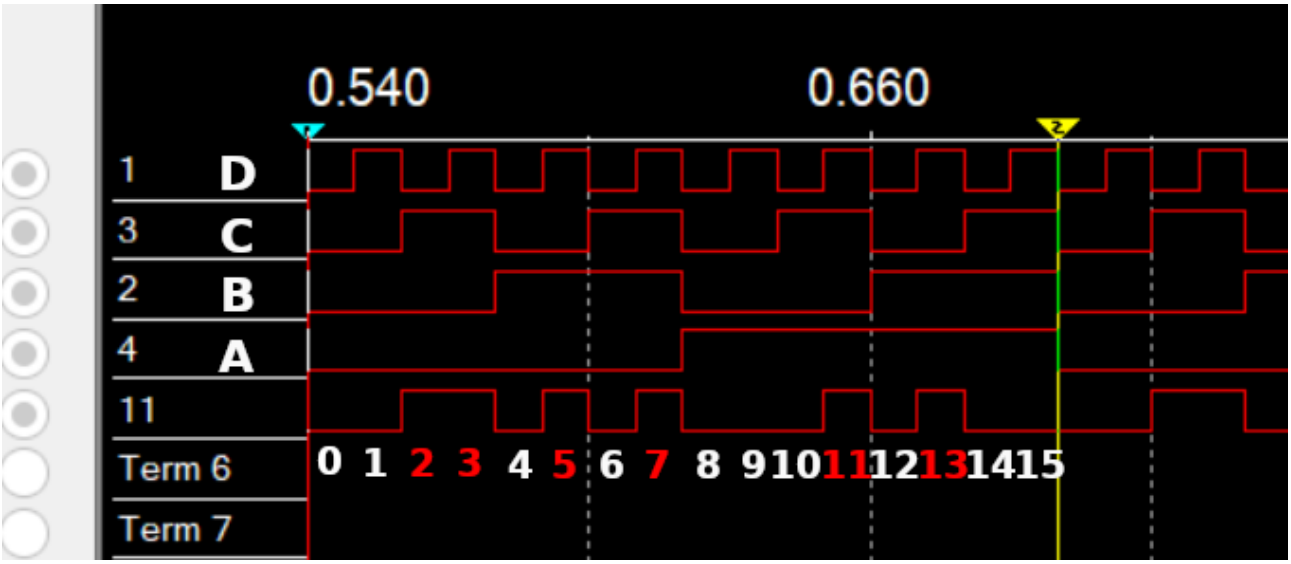


Rysunek 2.1: Układ z ręcznymi przełącznikami

W celu kompleksowego przetestowania układu przyciski zamieniono na generator słów bitowych XWG1 oraz podłączono analizator stanów logicznych XLA1. Odpowiednio dobierając częstotliwości próbkowania i generowanych sygnałów możemy zweryfikować poprawność układu.



Rysunek 2.2: Układ testowy



Rysunek 2.3: Wskazania analizatora

Ze wskazań analizatora można stwierdzić że dla każdego możliwego słowa bitowego otrzymano poprawną wartość. Układ wygenerował stan wysoki dla słów odpowiadającym 2,3,5,7,11,13.

2.3. Wnioski

1. Układ jest prosty do zaimplementowania ponieważ polega na bezpośrednim mapowaniu danych wejściowych do oczekiwanego wyjścia ale przez to nie jest to rozwiązanie które można by skalować dla dłuższych słów bitowych.
2. Sprzętowe generowanie liczb pierwszych może wspomóc oprogramowanie wykonujące testy pierwszości poprzez szybkie sprawdzenie małych liczb pierwszych.
3. W celu przeskalowania układu do praktycznych zastosowań można użyć układów sprzętowego tablicowania wartości

2.4. Podsumowanie

1. Funkcja logiczna została przekształcona do postaci pozwalającej na użycie jak najmniejszej ilości bramek.
2. Układ pozwala identyfikować słowa binarne reprezentujące liczby pierwsze.
3. Przetestowano poprawność układu stosując analizator stanów logicznych i generator słów losowych.