

Technika Cyfrowa.
Ćwiczenie 1.

Maciej Pieta

Piotr Koproń
Rafał Piwowar

Jakub Woś

Marzec 2023

1 Zadanie 1a

1.1 Treść zadania

Bazując wyłącznie na dwuwejściowych bramkach logicznych NAND, proszę od podstaw zaprojektować, zbudować i przetestować układ realizujący funkcję logiczną:

$$Y = \overline{A} \text{ xor } (B + C) \quad (1)$$

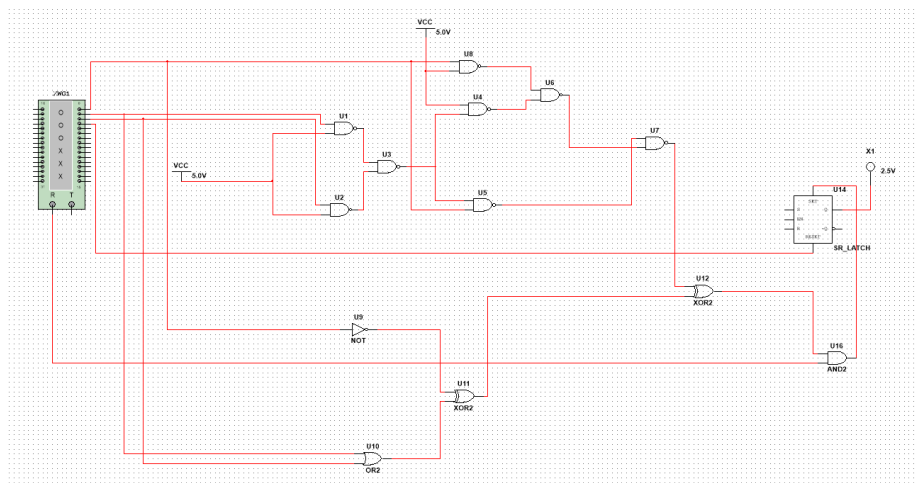
1.2 Rozwiązanie teoretyczne

Dokonujemy następujących przekształceń, korzystając z definicji xor, praw de Morgana oraz prawa podwójnej negacji.

$$\begin{aligned} Y &= \overline{A} \text{ xor } (B + C) \\ &= \overline{A} \cdot \overline{(B + C)} + \overline{\overline{A}} \cdot (B + C) \\ &= \overline{A} \cdot (\overline{B} \cdot \overline{C}) + A \cdot (B + C) \\ &= \overline{A} \cdot \overline{(\overline{B} \cdot \overline{C})} + A \cdot \overline{(\overline{B} \cdot \overline{C})} \\ &= \overline{A} \cdot (\overline{\overline{B} \cdot \overline{C}}) + A \cdot (\overline{\overline{B} \cdot \overline{C}}) \\ |K = \overline{(\overline{B} \cdot \overline{C})}| &= \overline{A} \cdot \overline{K} + A \cdot K \\ &= \overline{\overline{A} \cdot \overline{K} + A \cdot K} \\ &= \overline{\overline{A} \cdot \overline{K}} \cdot \overline{A \cdot K} \\ &= \overline{\overline{A} \cdot (\overline{B} \cdot \overline{C})} \cdot \overline{A \cdot (\overline{B} \cdot \overline{C})} \end{aligned}$$

Otrzymaliśmy równoważny układ bezpośrednio zapisujący się jako zbiór bramek NAND, przy obserwacji, że $x \text{ NAND } 1 = \overline{x}$.

1.3 Implementacja układu 1a w programie Multisim



Rysunek 1: Górna część układu to układ faktyczny, dolna część - układ testujący. Jeżeli w którymkolwiek momencie wykryta zostanie rozbieżność, przerzutnik po prawej "zapamięta" ten fakt.

Moduł XWG1 został zaprogramowany aby sprawdzał wszystkie możliwe kombinacje wartości A,B,C.

1.4 Wnioski

Kompletność NAND Należy zauważyć, że bramka NAND jest wystarczająca do utworzenia pełnego systemu logicznego. Tj, dowolną skończoną funkcję logiczną można fizycznie zaimplementować za pomocą skończonej ilości bramek NAND.

Przykładowe zastosowanie Układ może być wykorzystany do stworzenia systemu oświetlenia w części wspólnej składu w akademiku. (zmodyfikowany wyłącznik schodowy). Sygnał A odpowiadałby stanowi włącznika przy wejściu do składu, sygnały B i C - przy wejściu do pokojów.

2 Zadanie 1b

2.1 Treść zadania

Rozważmy pomieszczenie w którym znajdują się: drzwi wejściowe i dwa okna (wszystko wyposażone w czujniki stanu zamknięcia). Poza tym znajduje się tam: czujnik ruchu, syrena alarmowa (może być reprezentowana wskaźnikiem LED), dwa przyciski: uzbrojenia i rozbrojenia alarmu, dwa wskaźniki LED: alarm uzbrojony i alarm wyłączony, LEDowy czerwony sygnalizator problemu załączenia alarmu.

Alarm można uzbroić dedykowanym przyciskiem tylko wtedy, gdy w pomieszczeniu nie wykryto ruchu, a drzwi i okna są skutecznie zamknięte. Wówczas powinna zaświecić się kontrolka uzbrojenia alarmu. Jeśli warunki te nie są spełnione, zaświeca się czerwony sygnalizator problemu, a alarm pozostaje rozbrojony, co ciągle wówczas sygnalizuje stosowny wskaźnik LED.

Poprawne uzbrojenie alarmu powoduje zgaszenie się wskaźnika rozbrojenia alarmu i sygnalizatora problemu (jeśli jest zaświecony) oraz powoduje zaświecenie się wskaźnika uzbrojenia alarmu.

Alarm uruchamia się, gdy system alarmowy jest uzbrojony i wykryty jest ruch lub sygnalizowane jest otwarcie: drzwi lub któregoś z okien.

W oparciu o dowolne bramki logiczne, przełączniki i wskaźniki LED, proszę zaprojektować, zminimalizować, zbudować i przetestować układ realizujący funkcję opisanego wyżej systemu alarmowego. Rolę czujników mogą tutaj pełnić dowolne (dostępne w Multisimie) źródła sygnału cyfrowego.

2.2 Rozwiązanie teoretyczne

Oznaczenia Przyjmijmy następujące oznaczenia sygnałów:

- $A_1 \dots A_4$ jako sygnały czujników,
- B jako przycisk uzbrojenia,
- C jako przycisk rozbrojenia,
- D jako poprzedni stan odpowiedniego sygnału,
- E jako sygnalizator problemu,
- F jako stan uzbrojenia alarmu.

2.2.1 Łączenie czujników

Zauważmy, że z perspektywy alarmu, dowolne naruszenie czujników jest traktowane identycznie - nie interesuje nas to, czy otwarte są drzwi czy okna, w obu przypadkach można się dostać do pomieszczenia. Możemy więc utworzyć sygnał łączny czujników, dalej oznaczony jako A . Mamy tutaj do czynienia z

następującą tabelą prawdy:

Sygnały czujników $A_1 \dots A_4$	Sygnał A
0000	0
Dowolny inny	1

Z praw logiki można bezpośrednio wyprowadzić $A = A_1 + A_2 + A_3 + A_4$.

2.3 Wyprowadzenie wzorów

2.3.1 Sygnalizator błędu

$E_{\text{err}}[E]:$

$B_1 B_2 \backslash A, U_p$	00	01	11	10
00				
01				
11				1
10				1

$E = A \overline{U_p} B_1$

Rysunek 2: Tabela Karnaugh odpowiadająca sygnałowi błędu.

2.3.2 Stan uzbrojenia

Stan U_n

$B_1 B_2 \backslash A, U_p$	00	01	11	10
00		1	1	
01				
11	1	1	1	
10	1	1	1	

$$U_n = \bar{A}B_1 + \bar{B}_1\bar{B}_2U_p + AU_pB_1$$

Rysunek 3: Tabela Karnaugh odpowiadająca stanowi uzbrojenia.

2.3.3 Syrena alarmowa

Alarm [S]:

$B_1 B_2 \backslash A, U_P$	00	01	11	10
00			1	
01				
11			1	
10			1	

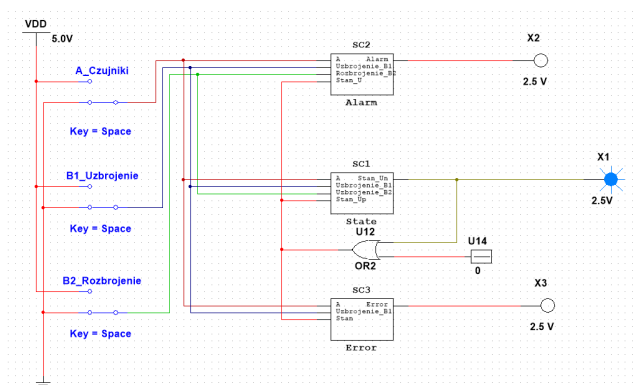
$$S = A \bar{B}_1 \bar{B}_2 U_P + A B_1 U_P$$

Rysunek 4: Tabela Karnaugh odpowiadająca syrenie alarmowej.

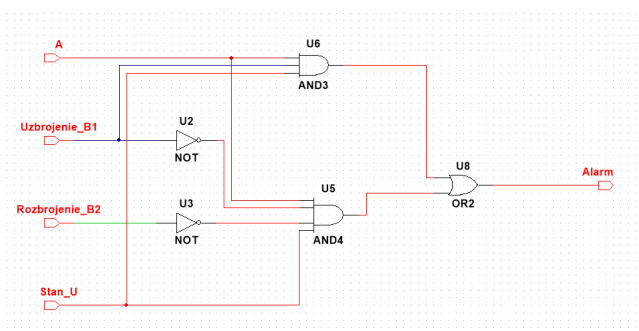
Wejście			Wyjście			
A	B ₁	B ₂	U _p	U	S	E
0	0	0	0	0	0	0
0	0	0	1	1	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	1	0	0
0	1	0	1	1	0	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	0
1	0	0	1	1	1	0
1	0	1	0	0	0	0
1	0	1	1	0	0	0
1	1	0	0	0	0	1
1	1	0	1	1	1	0
1	1	1	0	0	0	1
1	1	1	1	1	1	0

Rysunek 5: Zbiorcza tablica wartości.

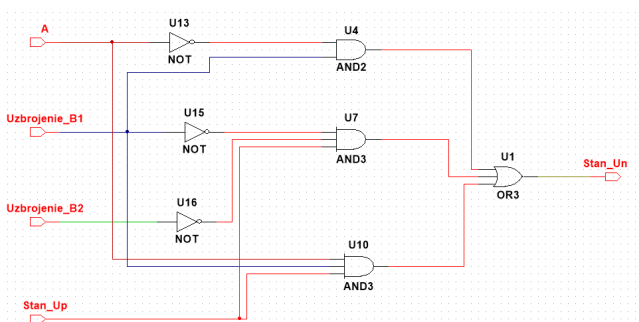
2.4 Implementacja układu 1b w programie Multisim



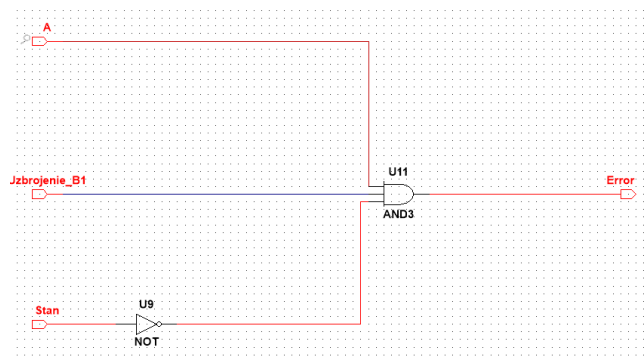
Rysunek 6: Pełny układ, zaimplementowany w programie Multisim. Podprojekty S1-S3 przedstawione poniżej.



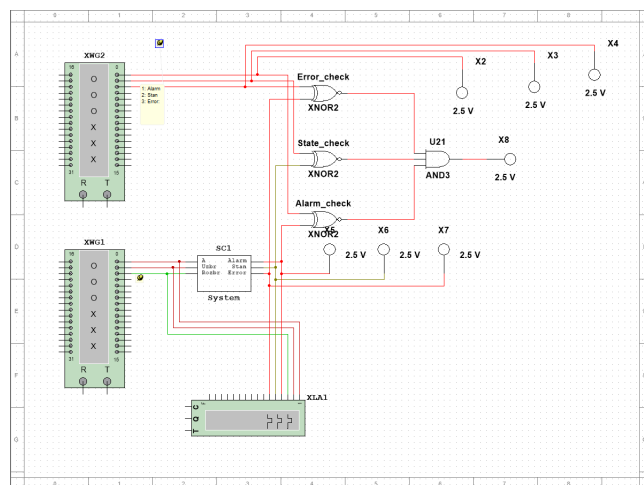
Rysunek 7: Implementacja sygnału uzbrojenia/rozbrojenia alarmu.



Rysunek 8: Implemetacja sygnału syreny alarmowej



Rysunek 9: Implementacja sygnału niepoprawnego uzbrojenia.



Rysunek 10: Układ testujący nasz projekt (SC1). Górne lamki LED generują oczekiwany rezultat, dolne - rzeczywisty. Generator słów tworzy wszystkie możliwe sytuacje.

2.5 Wnioski

Prostota a praktyka Korzystając tylko z podstawowej wiedzy na temat bramek logicznych, jesteśmy w stanie zaprojektować układ stosowalny w życiu codziennym. Co więcej, mamy niemalże gwarancję że układ nasz będzie szybszy niż jakiegokolwiek implementacje oparte o programowanie "klasycznego" procesora.