

Laboratorium 3

Sterownik windy

Łukasz Kwinta, Kacper Kozubowski, Ida Ciepiela

maj 2024

Spis treści

1	Cel zadania	3
2	Rozwiązania	3
2.1	Symulator silnika windy	3
2.1.1	Black box	3
2.1.2	Wejścia	4
2.1.3	Wyjścia	4
2.1.4	Realizacja układu	4
3	Testy	6
4	Zastosowania	6
5	Wnioski	6

1 Cel zadania

Proszę zaproponować, zbudować i przetestować układ sterujący windą w przykładowym trzykondygnacyjnym budynku.

Winda posiada:

- wskaźnik ruchu windy
- wskaźnik kierunku ruchu windy
- trzy czujniki otwarcia drzwi, po jednym na każdej kondygnacji
- trzy przyciski przywołania windy, po jednym na każdej kondygnacji
- trzy przyciski wyboru piętra w kabinie windy.

Winda powinna posiadać stale aktualizowany wskaźnik aktualnego piętra.

Rzeczy niedopowiedziane w treści zadania, proszę ustalić, doprecyzować i opisać samodzielnie.

2 Rozwiązania

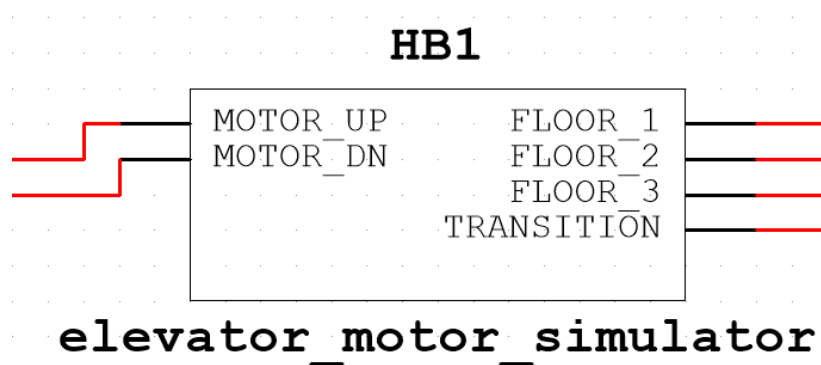
Postanowiliśmy rozbić problem na wiele mniejszych problemów i rozwiązać je osobno aby na końcu połączyć je w jeden działający system. Poniżej zamieszczamy schemat blokowy przedstawiający poszczególne systemy.

2.1 Symulator silnika windy

Aby zasymulować ruch windy wraz z czasem przemieszczania się między piętrami, zdecydowaliśmy się zaimplementować układ bazujący na liczniku i demultiplexerze.

2.1.1 Black box

Poniżej zamieszczamy schemat wyjść i wejść oraz opis logiki układu.



Rysunek 2.1: Black box układu symulującego silnik windy

2.1.2 Wejścia

- MOTOR_UP - sygnał wejściowy nakazujący poruszać się windzie w górę
- MOTOR_DN - sygnał wejściowy nakazujący poruszać się windzie w dół

2.1.3 Wyjścia

- FLOOR_1 - sygnał wyjściowy informujący o tym, że winda znajduje się na 1 piętrze, aktywny w stanie wysokim
- FLOOR_2 - sygnał wyjściowy informujący o tym, że winda znajduje się na 2 piętrze, aktywny w stanie wysokim
- FLOOR_3 - sygnał wyjściowy informujący o tym, że winda znajduje się na 3 piętrze, aktywny w stanie wysokim
- TRANSITION - sygnał wyjściowy informujący o tym że winda jest obecnie w ruchu, aktywny w stanie wysokim

2.1.4 Realizacja układu

Do realizacji układu wykorzystaliśmy 4 bitowy licznik z biblioteki komponentów programu Multisim, układ 74LS193N. Poniżej zamieszczamy tabelkę przedstawiającą działanie układu:

CLR	~LOAD	Up	Down	Mode
H	X	X	X	Reset(Async.)
L	L	X	X	Preset(Async.)
L	H	H	H	No Change
L	H	↑	H	Count Up
L	H	H	↑	Count Down

Tabela 2.1: Źródło: <https://www.multisim.com/help/components/binary-counters/>

Streszczenie

- H - stan wysoki na wejściu
- L - stan niski na wejściu
- X - dowolny stan na wejściu
- ↑ - narastające zbocze sygnału

Stworzyliśmy układ kombinacyjny, który mapuje wyjście zegara, na adres demultiplexera, który z kolei przekazuje jedynekę logiczną na odpowiednie wyjście. Przyjeliśmy, że:

- 0000 - winda jest na 1 piętrze

- 1000 - winda jest na 2 piętrze
- 1111 - winda jest na 3 piętrze
- każdy inny - winda porusza się między piętrami

Jako demultiplexer wykorzystaliśmy układ U7A 4555BD_5V. Jest to demultiplexer 1:4, z 2 bitami adresowymi.

Do wyprowadzenia formuł wykorzystaliśmy skrypt w języku Python, który generuje tabelę prawdy oraz minimalizuje formuły logiczne. Poniżej zamieszczamy kod programu oraz wynik jego działania:

```

1  import logicmin
2  elevator_motor_mux_tt = logicmin.TT(4, 2)
3
4  for i in range(16):
5      permutation = bin(i).removeprefix("0b").rjust(4, '0')
6
7      a1 = '1'
8      b1 = '1'
9
10     if i == 0:
11         a1 = '0'
12         b1 = '0'
13     elif i == 8:
14         a1 = '1'
15         b1 = '0'
16     elif i == 15:
17         a1 = '0'
18         b1 = '1'
19
20     elevator_motor_mux_tt.add(permutation, [a1, b1])
21
22 print("-----elevator_motor_tt")
23 sols = elevator_motor_mux_tt.solve()
24 print(sols.printN(xnames=['QD', 'QC', 'QB', 'QA'], ynames=['1A', '1B']))

```

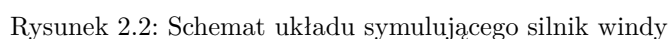
Wynikiem działania skryptu są zminimalizowane formuły logiczne:

```

-----elevator_motor_tt
1B <= QA + QB + QC
1A <= QD'.QA + QC'.QB + QC.QA' + QD.QB'

```

Po zaimplementowaniu układu w programie Multisim, uzyskaliśmy następujący schemat:



3 Testy

4 Zastosowania

5 Wnioski