



AKADEMIA GÓRNICZO HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE

Technika cyfrowa

WINDA

DULEWICZ ANTONI
SMYDA TOMASZ

7 CZERWCA 2024

Spis treści

1	Treść ćwiczenia	2
2	Opis rozwiązania	2
3	Schemat automatu	3
4	Tabela prawdy	4

1 Treść ćwiczenia

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.

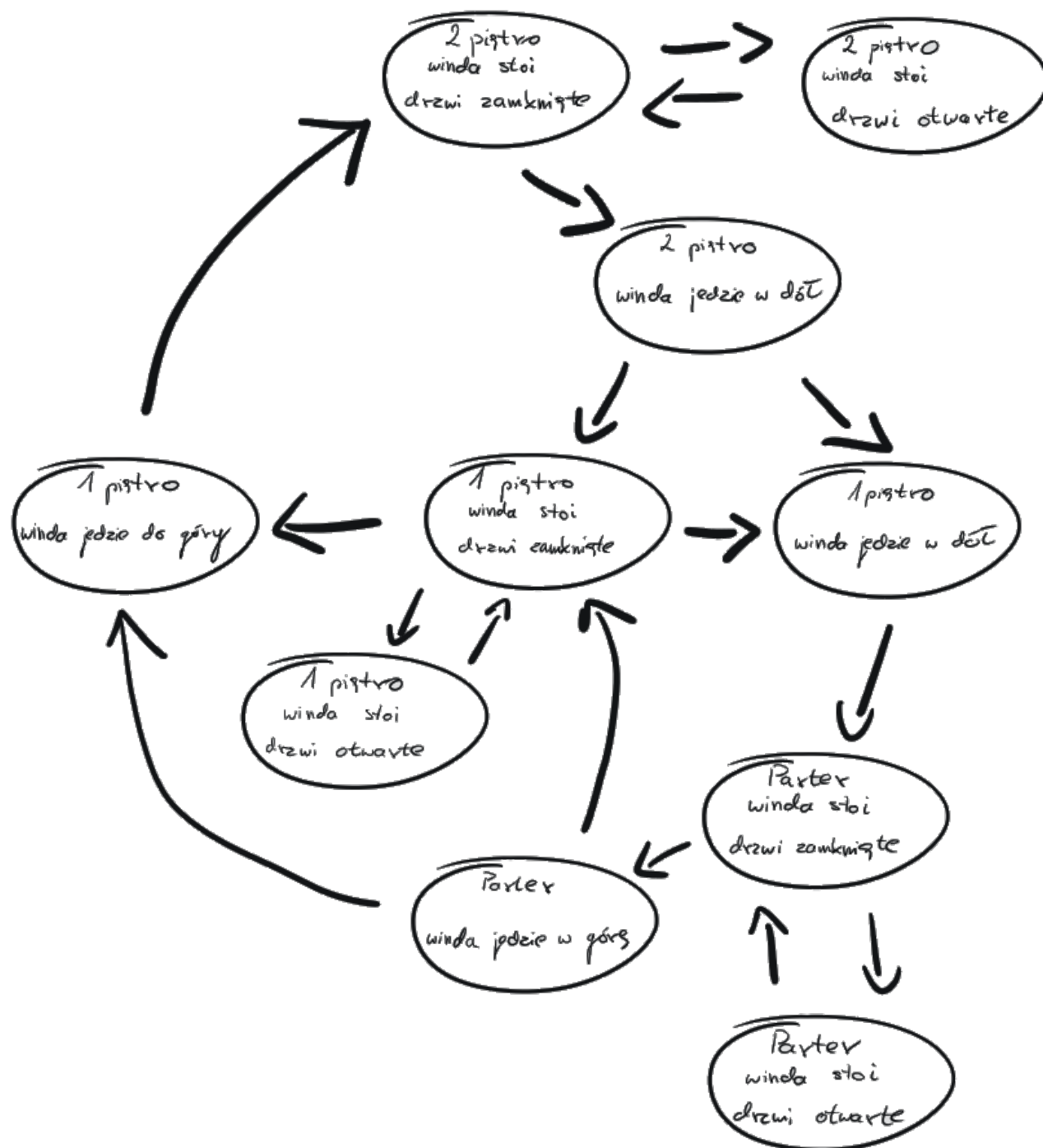
2 Opis rozwiązania

W celu zrealizowania zadania, postanowiliśmy zbudować automat, którego stanem manipulujemy za pomocą przycisków: trzech w kabinie windy i po jednym na każdej kondygnacji. Każdy stan tego automatu składa się z 7 bitów, kolejno:

- F_1F_2 - dwa bity wskaźnika piętra. Kolejne wartości oznaczają:
 - 00 - piętro 0
 - 01 - piętro 1
 - 10 - piętro 2
 - 11 - stan nieokreślony
- D_1D_2 - dwa bity wskaźnika kierunku windy i otwarcia drzwi kabiny. Kolejne wartości oznaczają:
 - 00 - winda stoi i drzwi są zamknięte
 - 01 - winda jedzie do góry
 - 10 - winda jedzie w dół
 - 11 - winda stoi i drzwi są otwarte
- B_0, B_1, B_2 - przyciski windy. Wartość każdego to:
 - 0 - przycisk niewciśnięty
 - 1 - przycisk wciśnięty

W celu skonstruowania układu w programie Multisim, stworzymy tabelę prawdy dla kolejnych stanów automatu oraz zminimalizujemy funkcje logiczne korzystając z algorytmu napisanego w języku Python. Następnie, aby upewnić się, że nasze przekształcenia są poprawne napiszemy algorytm testujący.

3 Schemat automatu



Rysunek 1: Schemat przejść ze stanu aktualnego do stanu następnego

4 Tabela prawdy

Stan aktualny							Stan następny						
F_1	F_2	D_1	D_2	B_0	B_1	B_2	F'_1	F'_2	D'_1	D'_2	B'_0	B'_1	B'_2
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	1	0	0	1
0	0	0	0	0	1	0	0	0	0	1	0	1	0
0	0	0	0	0	1	1	0	0	0	1	0	1	1
0	0	0	0	1	0	0	0	0	1	1	0	0	0
0	0	0	0	1	0	1	0	0	1	1	0	0	1
0	0	0	0	1	1	0	0	0	1	1	0	1	0
0	0	0	0	1	1	1	0	0	1	1	0	1	1
0	0	0	1	0	0	0	-	-	-	-	-	-	-
0	0	0	1	0	0	1	0	1	0	1	0	0	1
0	0	0	1	0	1	0	0	1	0	0	0	1	0
0	0	0	1	0	1	1	0	1	0	0	0	1	1
0	0	0	1	1	0	0	-	-	-	-	-	-	-
0	0	0	1	1	0	1	0	1	0	1	1	0	1
0	0	0	1	1	1	0	0	1	0	0	1	1	0
0	0	0	1	1	1	1	0	1	0	0	1	1	1
0	0	1	0	0	0	0	-	-	-	-	-	-	-
0	0	1	0	0	0	1	-	-	-	-	-	-	-
0	0	1	0	0	1	0	-	-	-	-	-	-	-
0	0	1	0	0	1	1	-	-	-	-	-	-	-
0	0	1	0	1	0	0	-	-	-	-	-	-	-
0	0	1	0	1	0	1	-	-	-	-	-	-	-
0	0	1	0	1	1	0	-	-	-	-	-	-	-
0	0	1	0	1	1	1	-	-	-	-	-	-	-
0	0	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	0	0	0	0	0	0	1
0	0	1	1	0	1	0	0	0	0	0	0	1	0
0	0	1	1	0	1	1	0	0	0	0	0	1	1
0	0	1	1	1	0	0	0	0	1	1	0	0	0
0	0	1	1	1	0	1	0	0	1	1	0	0	1
0	0	1	1	1	1	0	0	0	1	1	0	1	0
0	0	1	1	1	1	1	0	0	1	1	0	1	1
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	0	0	0	1	0	1	0	1	0	0	1
0	1	0	0	0	1	0	0	1	1	1	0	0	0
0	1	0	0	0	1	1	0	1	1	1	0	0	1
0	1	0	0	1	0	0	0	1	1	0	1	0	0
0	1	0	0	1	0	1	0	1	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	1	1	0	0
0	1	0	0	1	1	1	0	1	1	1	1	0	1
0	1	0	1	0	0	0	-	-	-	-	-	-	-
0	1	0	1	0	0	1	1	0	0	0	0	0	1

Stan aktualny							Stan następny						
F_1	F_2	D_1	D_2	B_0	B_1	B_2	F'_1	F'_2	D'_1	D'_2	B'_0	B'_1	B'_2
0	1	0	1	0	1	0	-	-	-	-	-	-	-
0	1	0	1	0	1	1	1	0	0	0	0	1	1
0	1	0	1	1	0	0	-	-	-	-	-	-	-
0	1	0	1	1	0	1	1	0	0	0	1	0	1
0	1	0	1	1	1	0	-	-	-	-	-	-	-
0	1	0	1	1	1	1	1	0	0	0	1	1	1
0	1	1	0	0	0	0	-	-	-	-	-	-	-
0	1	1	0	0	0	1	-	-	-	-	-	-	-
0	1	1	0	0	1	0	-	-	-	-	-	-	-
0	1	1	0	0	1	1	-	-	-	-	-	-	-
0	1	1	0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	1	0	1	0	0	0	0	1	0	1
0	1	1	0	1	1	0	0	0	0	0	1	1	0
0	1	1	0	1	1	1	0	0	0	0	1	1	1
0	1	1	1	0	0	0	0	1	0	0	0	0	0
0	1	1	1	0	0	1	0	1	0	0	0	0	1
0	1	1	1	0	1	0	0	1	1	1	0	0	0
0	1	1	1	0	1	1	0	1	1	1	0	0	1
0	1	1	1	1	0	0	0	1	0	0	1	0	0
0	1	1	1	1	0	1	0	1	0	0	1	0	1
0	1	1	1	1	1	0	0	1	1	1	1	0	0
0	1	1	1	1	1	1	0	1	1	1	1	0	1
1	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	1	1	0	1	1	0	0	0
1	0	0	0	0	1	0	1	0	1	0	0	1	0
1	0	0	0	0	1	1	1	0	1	1	0	1	0
1	0	0	0	1	0	0	1	0	1	0	1	0	0
1	0	0	0	1	0	1	1	0	1	1	1	0	0
1	0	0	0	1	1	0	1	0	1	0	1	1	0
1	0	0	0	1	1	1	1	0	1	1	1	1	0
1	0	0	1	0	0	0	-	-	-	-	-	-	-
1	0	0	1	0	0	1	-	-	-	-	-	-	-
1	0	0	1	0	1	0	-	-	-	-	-	-	-
1	0	0	1	0	1	1	-	-	-	-	-	-	-
1	0	0	1	1	0	0	-	-	-	-	-	-	-
1	0	0	1	1	0	1	-	-	-	-	-	-	-
1	0	0	1	1	1	0	-	-	-	-	-	-	-
1	0	0	1	1	1	1	-	-	-	-	-	-	-
1	0	1	0	0	0	0	-	-	-	-	-	-	-
1	0	1	0	0	0	1	-	-	-	-	-	-	-
1	0	1	0	0	1	0	0	1	0	0	0	1	0
1	0	1	0	0	1	1	0	1	0	0	0	1	1
1	0	1	0	1	0	0	0	1	1	0	1	0	0
1	0	1	0	1	0	1	0	1	1	0	1	0	1

Stan aktualny							Stan następny						
F_1	F_2	D_1	D_2	B_0	B_1	B_2	F'_1	F'_2	D'_1	D'_2	B'_0	B'_1	B'_2
1	0	1	0	1	1	0	0	1	0	0	1	1	0
1	0	1	0	1	1	1	0	1	0	0	1	1	1
1	0	1	1	0	0	0	1	0	0	0	0	0	0
1	0	1	1	0	0	1	1	0	1	1	0	0	0
1	0	1	1	0	1	0	1	0	0	0	0	1	0
1	0	1	1	0	1	1	1	0	1	1	0	1	0
1	0	1	1	1	0	0	1	0	0	0	1	0	0
1	0	1	1	1	0	1	1	0	1	1	1	0	0
1	0	1	1	1	1	0	1	0	0	0	1	1	0
1	0	1	1	1	1	1	1	0	1	1	1	1	0
1	1	0	0	0	0	0	-	-	-	-	-	-	-
1	1	0	0	0	0	1	-	-	-	-	-	-	-
1	1	0	0	0	1	0	-	-	-	-	-	-	-
1	1	0	0	0	1	1	-	-	-	-	-	-	-
1	1	0	0	1	0	0	-	-	-	-	-	-	-
1	1	0	0	1	0	1	-	-	-	-	-	-	-
1	1	0	0	1	1	0	-	-	-	-	-	-	-
1	1	0	0	1	1	1	-	-	-	-	-	-	-
1	1	0	1	0	0	0	-	-	-	-	-	-	-
1	1	0	1	0	0	1	-	-	-	-	-	-	-
1	1	0	1	0	1	0	-	-	-	-	-	-	-
1	1	0	1	0	1	1	-	-	-	-	-	-	-
1	1	0	1	1	0	0	-	-	-	-	-	-	-
1	1	0	1	1	0	1	-	-	-	-	-	-	-
1	1	0	1	1	1	0	-	-	-	-	-	-	-
1	1	0	1	1	1	1	-	-	-	-	-	-	-
1	1	1	0	0	0	0	-	-	-	-	-	-	-
1	1	1	0	0	0	1	-	-	-	-	-	-	-
1	1	1	0	0	1	0	-	-	-	-	-	-	-
1	1	1	0	0	1	1	-	-	-	-	-	-	-
1	1	1	0	1	0	0	-	-	-	-	-	-	-
1	1	1	0	1	0	1	-	-	-	-	-	-	-
1	1	1	0	1	1	0	-	-	-	-	-	-	-
1	1	1	0	1	1	1	-	-	-	-	-	-	-
1	1	1	1	0	0	0	-	-	-	-	-	-	-
1	1	1	1	0	0	1	-	-	-	-	-	-	-
1	1	1	1	0	1	0	-	-	-	-	-	-	-
1	1	1	1	0	1	1	-	-	-	-	-	-	-
1	1	1	1	1	0	0	-	-	-	-	-	-	-
1	1	1	1	1	0	1	-	-	-	-	-	-	-
1	1	1	1	1	1	0	-	-	-	-	-	-	-
1	1	1	1	1	1	1	-	-	-	-	-	-	-

5 Minimalizacja funkcji logicznych

Do utworzenia i zminimalizowania funkcji logicznych użyliśmy biblioteki `logicmin` w Pythonie. Wykorzystuje ona m.in. metodę Karunaugha do uproszczenia funkcji do ich minimalnych form.