



POLITECHNIKA OPOLSKA

LABORATORIUM

PRZEDMIOT:	AUTOMATYZACJA NAPĘDU ELEKTRYCZNEGO
------------	---

KIERUNEK STUDIÓW:	<i>Elektrotechnika</i>	ROK STUDIÓW:	<i>1</i>
SEMESTR:	<i>2</i>	ROK AKADEMICKI:	<i>2021/2022</i>

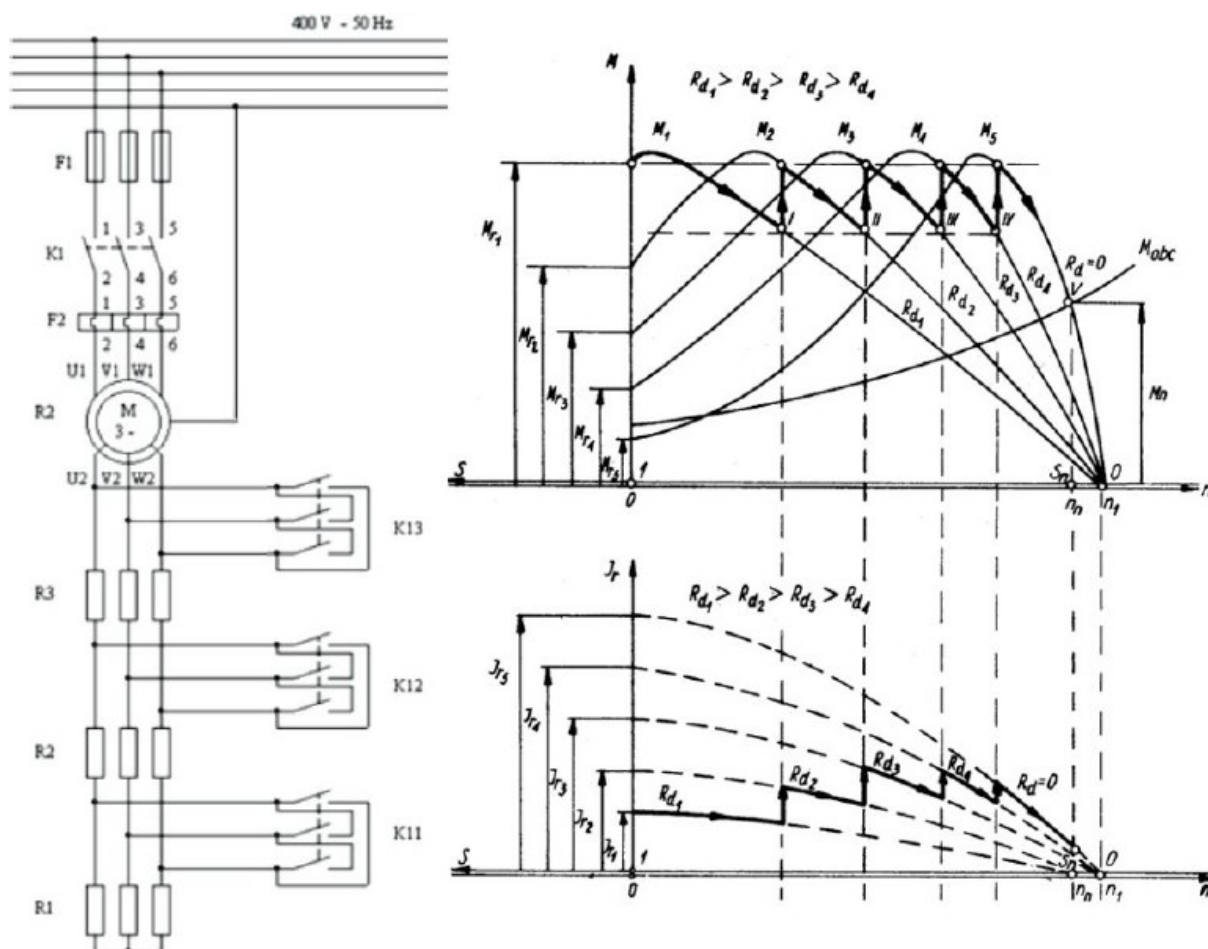
<i>Temat:</i>
Wykonanie układu sterowania rozruchem silnika w ZelioSoft

Termin oddania:	27.01.2022	Nr ćwiczenia:	4
-----------------	------------	---------------	---

<i>Projekt wykonali:</i>			
<i>Nazwisko:</i>		<i>Imię:</i>	<i>Ocena:</i>
1.	HOFFMAN	KACPER	
2.	KOŁODZIEJ	KAMIL	
3.	KORYTKOWSKI	GRACJAN	

1. Opis ćwiczenia

Celem ćwiczenia było wykonanie sterowania rozruchu silnika pierścieniowego przy pomocy tablicy Karnaugh oraz oprogramowania ZelioSoft. W związku z tym możliwe było wykonaniu dwóch typów układu sterowania. Jedną możliwością jest stworzenie systemu czasowego, zgodnie z którym kolejne styki są załączane po ustawionych opóźnieniach czasowych. Druga opcja z kolei polega na pomiarze wartości prądu – kiedy poziom prądu spadnie poniżej minimalnej wartości ustawionej załączony zostanie następny styk. W tym ćwiczeniu wybrano drugą możliwość.



Rys. 1.1: Schemat układu rozruchowego z przebiegami prądu dla kolejnych ustawień styków

2. Tabela Karnaugh

Stworzenie układu spełniającego warunki rozruchu jest możliwe nawet bez specjalnego przygotowania, jednak możliwe jest również zastosowanie tablicy Karnaugh. Tablicą Karnaugh nazywamy specjalną tabelę opisującą stany logiczne układu wraz z odpowiedziami. Tutaj jako wejścia przyjmujemy sygnał Is, flagę M oraz sygnały stanu styków K11s, K12s, K13s. Wartość Is to wynik porównania logicznego, czy wartość mierzonego prądu jest większa od nastawionego minimum. To pozwala na wpisanie to tabeli liczb binarnych stan prądu który sam z siebie jest wartością ciągłą. Flaga M będzie użyta w celu zapamiętania stanu wysokiego. To pozwala na odróżnienie kolejnych rejonów rozruchu. Kiedy Is osiągnie wartość 1, to flaga M zostanie ustawiona na 1. Kiedy zaś stan Is zmieni się na 0, to M pozostanie 1 i pozwoli na uruchomienie kolejnego styku. Sygnały ze styków K11s, K12s oraz K13s są niezbędne w celu odróżnienia, który styk ma zostać załączony. Na początku wykonujemy tabelę z kolejnymi stanami w normalnym procesie rozruchu:

Tabela Początkowa								
Is	M	K11s	K12s	K13s	M	K11	K12	K13
0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0
1	1	0	0	0	1	0	0	0
0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	1	0	0
1	0	1	0	0	1	1	0	0
1	1	1	0	0	1	1	0	0
0	1	1	0	0	0	1	1	0
0	0	1	1	0	0	1	1	0
1	0	1	1	0	1	1	1	0
1	1	1	1	0	1	1	1	0
0	1	1	1	0	0	1	1	1
0	0	1	1	1	0	0	0	1
1	0	0	0	1	1	0	0	1
1	1	0	0	1	1	0	0	1

Rys. 2.1: Tabela początkowa stanów wejściowych i wyjściowych w układzie

Na początku mamy stan wstępny 0, gdzie wszystko jest odłączone. Kiedy po raz pierwszy Is stanie się 1, zapalamy flagę M. Po tym jak Is spadnie na 0, Odczytuje się stan z wyjść. $M = 1$ oraz $K11s = 0$, $K12s = 0$, $K13s = 0$ oznacza, że załączamy styk K11 oraz zerujemy M. Potem prąd wzrasta, $Is = 1$ przez co ustawiamy flagę M. Kiedy $Is = 0$, odczytujemy z wyjść $M = 1$, $K11s = 1$, $K12s = 0$, $K13s = 0$. Decydujemy, że załączy się styk K12 oraz wyzeruje flaga. Ten cykl się powtarza dla K13 z jednym wyjątkiem. Ponieważ na końcu nie ma potrzeby aby wszystko zostało załączone, odłącza się styki K11 i K12. Po wykonaniu tej tablicy początkowej wstawiamy dane go tablic Karnaugh. Polegają one na wstawieniu stanów wyjść do tabeli ze stanami wejść. Wejścia segreguje się zgodnie z kodem Gray'a, czyli kolejne adresy mogą się różnić tylko o jeden bit (np. 00, 01, 11, 10).

Tablice Karnaugh								
M	K11s / K12s / K13s							
Is / M	"000"	"001"	"011"	"010"	"110"	"111"	"101"	"100"
"00"	0	0	x	x	0	0	x	0
"01"	0	0	x	x	0	x	x	0
"11"	1	0	x	x	1	x	x	1
"10"	1	0	x	x	1	x	x	1
K11	K11s / K12s / K13s							
Is / M	"000"	"001"	"011"	"010"	"110"	"111"	"101"	"100"
"00"	0	0	x	x	1	0	x	1
"01"	1	0	x	x	1	x	x	1
"11"	0	0	x	x	1	x	x	1
"10"	0	0	x	x	1	x	x	1
K12	K11s / K12s / K13s							
Is / M	"000"	"001"	"011"	"010"	"110"	"111"	"101"	"100"
"00"	0	0	x	x	1	0	x	0
"01"	0	0	x	x	1	x	x	1
"11"	0	0	x	x	1	x	x	0
"10"	0	0	x	x	1	x	x	0
K13	K11s / K12s / K13s							
Is / M	"000"	"001"	"011"	"010"	"110"	"111"	"101"	"100"
"00"	0	1	x	x	0	1	x	0
"01"	0	1	x	x	1	x	x	0
"11"	0	1	x	x	0	x	x	0
"10"	0	1	x	x	0	x	x	0

Rys. 2.2: Tablice Karnaugh

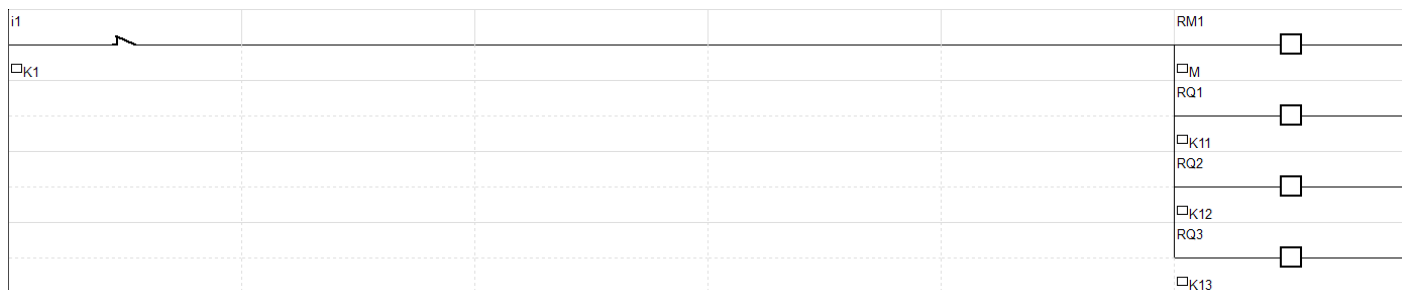
Po wstawieniu wartości znanych należy uzupełnić tabelę o pola które nie występują. Jeśli dany stan jest logicznie niemożliwy, to oznaczamy go jako 'x'. Takie pola mają wartość albo 1 albo 0 zależnie od potrzeby. Po wypełnieniu tablicy możliwe są dwa sposoby na wyznaczenie wzoru na wartość wyjścia. Albo bierzemy pod uwagę tylko jedyńki (dzięki czemu jesteśmy pewni że program zadziała jak należy) albo upraszczamy układ przez uwzględnienie 'x' (co daje łatwiejsze wzory, ale niekoniecznie prawidłowe). Jedyńki zbiera się w prostokątach o wielkościach równych potęgom dwójki (1, 2, 4, 8 itd.). Następnie sprawdzamy które wielkości wejściowe dla danego prostokąta się nie zmieniają i mnożymy je. Dla tabeli M posiadamy dwie możliwości – dokładny prostokąt o 2 wartościach, albo rozszerzony o 8. Dla 2 mamy niezmiennie Is, K11s, K12s oraz odwrócone K13s, a więc wzór na ten prostokąt to $Is * K11s * K12s * \text{NOT } K13s$. Z kolei dla 8 nie zmieniają się Is oraz K11s, a więc wychodzi wzór $Is * K11s$. Wzory prostokątów się dodaje do siebie. Tą metodą uzyskujemy wzory na zmienne wyjściowe które da się dodać do programu.

fM
$Is * K11s * K12s * NOT K13s + Is * NOT K12s * NOT K13s$
$Is * K11s + Is * NOT K12s * NOT K13s = Is * (K11s + NOT K12s * NOT K13s)$
fK11
$K11s * NOT K13s + NOT Is * M * NOT K12s * NOT K13s$
fK12
$K11s * K12s * NOT K13s + NOT Is * M * K11s * NOT K12s * NOT K13s$
$K11s * K12s * NOT K13s + NOT Is * M * K11s = K11s * (K12s * NOT K13 + NOT Is * M)$
fK13
$NOT K11s * NOT K12s * K13s + NOT Is * M * K11s * K12s * NOT K13s + NOT Is * NOT M * K11s * K12s * K13s$
$NOT K11s * K13s + K11s * K13s + NOT Is * M * K11s * K12s = NOT K11s * K13s + K11s * (K13s + NOT Is * M * K12s)$

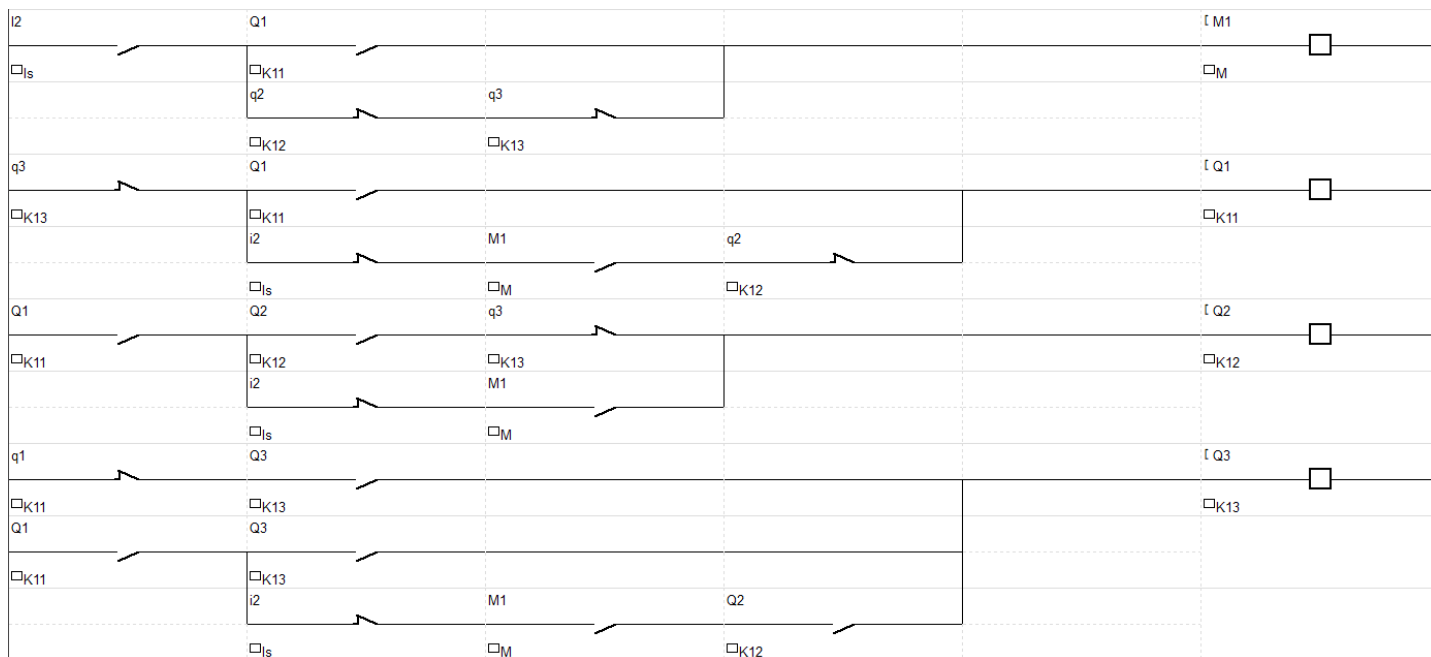
Rys. 2.3: Wzory na wartości wyjściowe ('bezpieczne' wyżej, 'ułatwione' niżej)

3. ZelioSoft

W programie zeliosoft możemy wstawić logiczne elementy reprezentujące nasze wzory. Stosując normalnie otwarte i normalnie zamknięte komponenty odtwarzamy uzyskane powyżej wzory. Na początku należy jednak pamiętać o pominiętym fragmencie ze schematu. Styk K1 to załączenie zasilania układu, i kiedy jest on zerem układ powinien być w stanie zerowym. Uwzględniamy to przy pomocy cewek zerujących wartości wszystkich wyjść.



Rys. 3.1: Część zerująca wartości wyjściowe dla załączenia zasilania



Rys. 3.2: Odtworzenie wzorów logicznych w ZelioSoft

Jak widać, wzory uzyskane zostały odtworzone w ZelioSoft. Mnożeniem logicznym jest operacja AND, reprezentowana połączeniem szeregowym. Z kolei dodawanie logiczne to OR, reprezentowany połączeniem równoległym. Potwierdzenie działania polega na symulacji ZelioSoft. Załączenie zasilania i przełączanie prądu potwierdza, że program zachowuje się dokładnie tak jak zostało to określone w tablicy początkowej.

4. Wnioski

Dzięki wykożystaniu metodyki tablic Karnaugh oraz programu ZelioSoft, utworzony został program zdolny do sterowania rozruchem silnika pierścieniowego. Określenie roli wejść i wyjść pozwoliło stworzyć tabelę stanów logicznych wejść i wyjść. Potem zgodnie z kodem Gray'a stworzono tablice Karnaugh dla wyjść. Wyznaczenie prostokątów z wartościami 1 pozwoliło na obliczenie wzoru logicznego na wyjścia. Te wzory zostały przełożone na język programowania drabinowego. Sprawdzone działanie programu potwierdza, że stworzony układ zachowuje się prawidłowo.