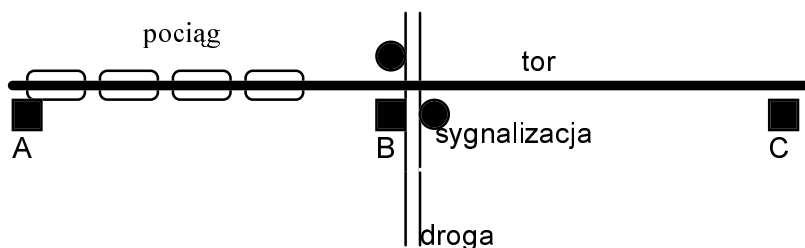

LABORATORIUM TEORII AUTOMATÓW

PRZEJAZD KOLEJOWY

Układ lamp ostrzegawczych na niestrzeżonym przejeździe kolejowym

Założenia zadania:

1. Pociąg może wjeżdżać na przejazd z dwóch kierunków
2. Czujniki wykrywające obecność pociągu umieszczono w punktach A,B,C (patrz rysunek)
3. Światła sygnalizacyjne Y mają się zapalić (stan logiczny 1) gdy: (pociąg przesłania A **lub** C i jedzie w kierunku B) **lub** (przesłonił A **lub** C i jedzie w kierunku B) **lub** (przesłania B).
4. Pociąg nie może manewrować pomiędzy czujnikami A i C (zawracać).



A,B,C - czujniki obecności pociągu

Rysunek 1 Przejazd kolejowy z sygnalizacją

Tworzymy pierwotną siatkę programu.

ABC	000	001	011	010	110	111	100	Y
1	1	6	-	-	-	-	4	0
2	2	-	-	8	-	-	-	1
3	3	7	-	-	-	-	5	0
4	2	-	-	-	9	-	4	1
5	1	-	-	-	-	-	5	0
6	2	6	10	-	-	11	-	1
7	1	7	-	-	-	-	-	0
8	3	-	10	8	9	-	-	1
9	-	-	-	8	9	11	5	1
10	-	7	10	8	-	-	-	1
11	-	-	10	-	9	11	-	1

Tabela pierwotna

Każdy stan Q (numerowany od 1 do 11) wyróżnia nową sytuację związaną z warunkami zadania, stany pierwotne mogą posiadać te same wartości zmiennych wejściowych i wyjściowych. na przykład: stany **2** i **3** mają identyczne wartości: A=0 B=0 C=0 Y=0 tymczasem **2** odpowiada sytuacji - pociąg jest pomiędzy AB lub BC i zmierza do B; **3** odpowiada sytuacji - pociąg jest pomiędzy AB lub BC i zmierza w kierunku przeciwnym do B;

Sytuacja $ABC=101$ nie istnieje. Oznaczała by ona: pociąg przesłania A, przesłania C i nie przesłania B - czyli rozerwał się skład pociągu.

Tabela pierwotna odpowiada zawsze automatowi Moore'a.

Korzystając z symetrii utworzymy prostszą tabelę pierwotną o dwóch wejściach: **d=avc** oraz **b**

Q \ DB	00	01	11	10	Y
1	1	-	-	2	0
2	3	-	7	2	1
3	3	4	-	-	1
4	5	4	7	-	1
5	5	-	-	6	0
6	1	-	-	6	0
7	-	4	7	6	1

Pociąg nie może manewrować (zawracać) pomiędzy czujnikami A i C w związku z powyższym nie można przejść ze stanu 2 do 1, ze stanu 3 do 2, ze stanu 4 do 3, ze stanu 5 do 4, ze stanu 6 do 5. Podobnie nie jest możliwa zmiana równoczesna na dwóch bitach wejściowych, a więc co drugie pole od stanu stabilnego w danym wierszu jest wypełnione kreską.

Z tabeli pierwotnej tworzymy tabelę minimalną dla automatu Mealy'ego.

Można połączyć wiersze: 2-3 oraz 4-5-7 (wersja I) lub wiersze 2-3 4-5 6-7 (wersja II). Otrzymuje się dwa automaty Mealy'ego o liczbie stanów 4. Jest to minimalna liczba stanów.

Automat Moore'a ma o jeden stan więcej, ponieważ nie można łączyć wierszy o różnych wyjściach.

Wersja I

Q \ DB	00	01	11	10
1	1, 0	-, -	-, -	23, 1
23	23, 1	457, 1	457, 1	23, 1
457	457, 0	457, 1	457, 1	6, 0
6	1, 0	-, -	-, -	6, 0

Wersja II

Q \ DB	00	01	11	10
1	1, 0	-, -	-, -	23, 1
23	23, 1	45, 1	67, 1	23, 1
45	45, 0	45, 1	67, 1	67, 0
67	1, 0	45, 1	67, 1	67, 0

Wersja I

Koduje się stany	stan 1	kod	00
	stan 23	kod	01
	stan 457	kod	11
	stan 6	kod	10

Wersja II

Koduje się stany	stan 1	kod	00
	stan 23	kod	01
	stan 45	kod	11
	stan 67	kod	10

Automat Wersja I

q1,q2 \ D,B	00	01	11	10
00	00, 0	-, -	-, -	01, 1
01	01, 1	11, 1	11, 1	01, 1
11	11, 0	11, 1	11, 1	10, 0
10	00, 0	-, -	-, -	10, 0

q1'q2', Y

Automat Wersja II

q1,q2 \ D,B	00	01	11	10
00	00, 0	-, -	-, -	01, 1
01	01, 1	11, 1	10, 1	01, 1
11	11, 0	11, 1	10, 1	10, 0
10	00, 0	11, 1	10, 1	10, 0

q1'q2', Y

Jak widać automaty obu wersji różnią się tam gdzie to zaznaczono wytłuszczając druk.

Tabelę przejść/wyjsc automatu należy rozpisać na trzy odrębne tabele dla:

q1', q2' i Y

q1,q2 \ D,B	00	01	11	10
00	0	-	-	0
01	0	1	1	0
11	1	1	1	1
10	0	-	-	1

q1'

Z tabeli widać, że: $q_1' = B \vee q_1q_2 \vee Dq_1$

q1,q2 \ D,B	00	01	11	10
00	0	-	-	1
01	1	1	1 werI 0 werrII	1
11	1	1	1 werI 0 werrII	0
10	0	1	1 werI 0 werII	0

q2'

Z tabeli widać, że: $q_2' = B \vee D(\text{nie}(q_1)) \vee (\text{nie}(D))q_2$ dla wersji I

$q_2' = B(\text{nie}(D)) \vee D(\text{nie}(q_1))(\text{nie}(B)) \vee (\text{nie}(D))q_2$ dla wersji II

$q_1, q_2 \setminus D, B$	00	01	11	10
00	0	-	-	1
01	1	1	1	1
11	0	1	1	0
10	0	1	1	0

Y

$$Y = B \vee (\text{nie}(q_1))q_2 \vee D(\text{nie}(q_1))$$

A zatem można przystąpić do realizacji automatu. Jeżeli realizacja ma miejsce przy użyciu bramek NAND to należy przekształcić funkcje logiczne do odpowiedniej postaci. jeżeli realizacja odbywa się przy użyciu sterownika logicznego to należy zamodelować równania logiki.