Programowalna matryca logiczna (PLD) C-4
Logika kombinacyjna

Celem ćwiczenia jest zaprojektowanie układów kombinacyjnych oraz sprawdzenie ich działania w praktyce za pomocą matrycy logicznej. Matryca zasilana jest napięciem pojedynczym +5V względem masy.

Realizując funkcje logiczne należy pamiętać iż matryca umożliwia bezpośrednią implementacje funkcji logicznych w postaci sumy iloczynów np. $Y=A\overline{B}D + \overline{C}A$, gdzie symbol $A\overline{B}D$ oznacza iloczyn logiczny sygnałów A, nie B (negacja B) oraz D, czyli trójwejściową bramkę AND. Symbol nadkreślenia (kreski nad symbolem) oznacza jego negację, natomiast znak + to suma logiczna (bramka OR). Takimi zasadami zapisu funkcji logicznych należy się kierować również w sprawozdaniu. Do realizacji zadań w tym ćwiczeniu będzie wymagana znajomość: algebry Boole'a, praw De Morgana oraz optymalizacji funkcji logicznych z wykorzystaniem tablic Karnaugh'a. Do realizacji logiki sekwencyjnej wymagana jest również wiedza na temat przerzutnika typu D.

W sprawozdaniu dla każdego z punktów ćwiczenia należy przedstawić:
- funkcje logiczne będące wynikiem danego zadania,
- tabelę stanów układu kombinacyjnego lub tabelę stanów następnych (logika sekwencyjna)
- tabele Karnaugh'a **wraz z zaznaczonymi grupami** jeśli w danym punkcie ćwiczenia używano tej metody optymalizacji (punkty 2 - 4 ćwiczenia)
- zdjęcie matrycy pokazujące zaimplementowane funkcje logiczne,

1. Realizacja bramek logicznych

Korzystając z 7 wyjść matrycy (w trybie kombinacyjnym) zaprojektować w postaci sumy iloczynów następujące bramki logiczne: NOT, AND, NAND, OR, NOR, XOR, XNOR (każda bramka na kolejnym wyjściu matrycy O_1, O_2, itd.). Wszystkie bramki logiczne powinny być dwu-wejściowe i zależeć od sygnałów wejściowych A i B (wyjątkiem jest inwerter (NOT), który będzie zależał tylko od A). Bramki zmontować na matrycy logicznej, zweryfikować ich poprawność (czy dla 4 kombinacji sygnałów wejściowych A i B wyjście bramki przyjmuje wartości zgodne z przewidywaniami teoretycznymi).

2. Pomiar czasu propagacji

Dla wybranej bramki logicznej, podając na jej wejście przebieg prostokątny (niski poziom (ang. **L**ow level)=0V, wysoki poziom (ang. **H**igh level)=5V), zmierzyć czasy przełączenia (zmiana stanu na wyjściu) ze stanu niskiego do wysokiego T_{LH} oraz wysokiego do niskiego T_{HL}.

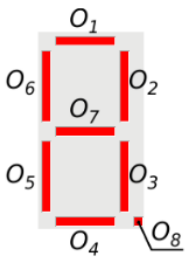
3. Dekoder kodu binarnego na wyświetlacz 7-segmentowy

Zbudować układ kombinacyjny dekodujący liczby w kodzie binarnym na wyświetlacz 7-segmentowy prezentujący liczby w kodzie dziesiętnym. Należy użyć wejść A – D jako 4 bitów liczby binarnej (A jest najmłodszym bitem) oraz wyjść O_1 − O_8 jako sygnałów sterujących poszczególnymi segmentami wyświetlacza (wszystkie wyjścia powinny zostać skonfigurowane jako wyjścia kombinacyjne). Zaprojektowany dekoder powinien prezentować liczby dziesiętne od 0 do 9, odpowiednio dla binarnych 0000 – 1001 **oraz spełniać wymagania szczególne** na podstawie poniższej tabeli w zależności od numeru wersji ćwiczenia podanej przez prowadzącego:

Wersja ćwiczenia	*Szczegóły dekodera*
1	dla wartości na wejściu 10-15 (binarnie 1010-1111) wyświetlacz prezentuje resztę z dzielenia przez 10, zaświecając dodatkowo kropkę na wyświetlaczu, która sygnalizuje ten fakt (np. wartość 1 zaświeca 1, 11 zaświeca 1. itd.)
2	dla wartości na wejściu 10-15 (binarnie 1010-1111) wyświetlacz pokazuje tylko "-" symbolizując przekroczenie zakresu. Dla wartości wejściowych poniżej 10 kropka na wyświetlaczu świeci się dla liczb nieparzystych.
3	dla wartości na wejściu 10-15 (binarnie 1010-1111) wyświetlacz pokazuje wartość zakodowaną szesnastkowo: 10→A, 11→b, 12→C, itd. Kropka na wyświetlaczu powinna stale być wyłączona/nieużywana.
4	dla wartości na wejściu 10-15 (binarnie 1010-1111) wyświetlacz pokazuje tylko "o" - przekroczenie zakresu. Dla wartości wejściowych poniżej 10 kropka na wyświetlaczu świeci się dla liczb parzystych (zero jest parzyste).
5	dla wartości na wejściu 10-15 (binarnie 1010-1111) wyświetlacz prezentuje liczbę równą (18 - wartość wejściowa), w efekcie pokazując cyfry na wyświetlaczu (0, 1 ... 9, 8, 7, 6, 5, 4, 3) - najpierw po kolei, a potem w odwróconej kolejności. Kropka na wyświetlaczu powinna stale być wyłączona/nieużywana.

Zadanie należy rozpocząć od zaprojektowania tabeli stanów (obowiązkowo do zamieszczenia w sprawozdaniu), gdzie dla każdego z 16 (2^4) stanów 4-bitowego słowa wejściowego DCBA przypisana zostanie jednoznaczna wartość wszystkich 8 wyjść dekodera, związanych z odpowiednimi segmentami wyświetlacza. Przykład tabeli stanów pokazujący jak zakodować kilka podstawowych liczb/symboli przedstawiono poniżej:

D	C	B	A		O1	O2	O3	O4	O5	O6	O7	O8	Symbol
0	0	0	0		1	1	1	1	1	1	0	0	0
0	0	0	1		0	1	1	0	0	0	0	0	1
0	0	1	0		1	1	0	1	1	0	1	0	2
1	1	1	0										
1	1	1	1		1	0	0	0	1	1	1	0	F



Podstawowy wygląd symboli na wyświetlaczu 7-segmentowym pokazuje rysunek poniżej:

W dalszej części dokonujemy minimalizacji funkcji logicznych dla każdego z 8 wyjść O_1 – O_8 wykorzystując metodę tablic Karnaugh'a. Kolejność sygnałów wejściowych w tabelach proszę przyjąć według następującej zasady. Indeksowanie wierszy według najstarszych bitów w tabeli stanów (tutaj D, potem C), natomiast kolumny według dwóch młodszych (tutaj B, potem A). Przykład poniżej:

DC \ BA	00	01	11	10
00				
01				
11				
10				