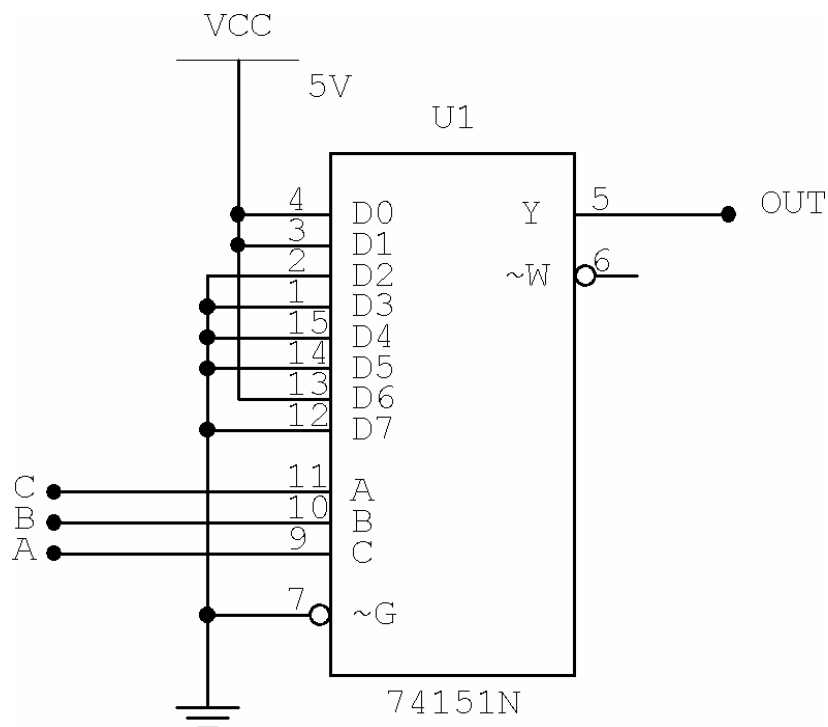


DODATEK I - Realizacja funkcji z użyciem multiplexera

Jednym z zastosowań multiplexerów i demultiplexerów jest możliwość prostej realizacji przy ich pomocy układów kombinacyjnych. Używając go można zrealizować dowolny układ kombinacyjny o ilości wejść równej ilości wejść adresujących i jednym wyjściu. Minimalnie rozbudowując układ można zrealizować także układ o ilości wejść większej o jeden niż ilość wejść adresujących. No ale najpierw wersja (naj)prostsza.

Przeanalizujmy to na przykładzie multiplexera 74151. Multiplexer działa w taki sposób, że przekazuje na swoje wyjście (Y) stan z tego wejścia danych (D0-D7) na które w danym momencie wskazuje stan wejść adresujących CBA. Czyli np. jeżeli CBA = 000, to na wyjściu Y pojawia się stan z wejścia D0, a jeżeli CBA = 001, to na wyjściu Y pojawia się to samo co na wejściu D1 itd. Czyli np. układ pokazany na rysunku:



realizuje funkcję $OUT_{Z_2} = \sum(0,1,6)$, bo 5V (czyli jedynka) jest podłączone do wejść D0, D1 i D6, zatem jedynki pojawią się na wyjściu Y gdy do wejść CBA podłączone zostaną słowa: 000 (dziesiętne 0), 001 (dziesiętne 1), 110 (dziesiętne 6), natomiast podanie na wejścia CBA innych kombinacji będzie skutkowało zerem na wyjściu. Skoro tak, to układ ten realizuje zadaną przykładową funkcję $OUT_{Z_2} = \sum(0,1,6(3,4))$, przy czym stany nieokreślone (3 i 4) zostały przyjęte za zero. Równie dobrze można było na te wejścia podać jedynki, wtedy realizowaną funkcją byłaby funkcja $OUT_{Z_2} = \sum(0,1,3,4,6)$, co z punktu widzenia zadanej funkcji przykładowej nie ma znaczenia.

Należy także pamiętać o wejściu G, jest to wejście opisane jako „gate” lub „strobe”, czyli bramkujące albo strobuujące, mówiąc prościej – zezwalające. Dopóki nie podłączymy tam zera (o tym że wejście jest aktywne zerem mówi nam „falka” przy oznaczeniu, ewentualnie negujące „kółko” na schemacie), układ nie zareaguje na nasze działania, jest to coś w rodzaju „włącznika” dla tego multiplexera.

Korzystając z multiplexera można też zbudować układ kombinacyjny o ilości wejść większej niż liczba wejść adresujących multiplexera. Budowa takiego układu jest niewiele bardziej skomplikowana. Załóżmy że mamy do zbudowania układ opisany tabelą jak niżej:

l.p.	M	N	O	P	OUT	
0	0	0	0	0	0	0
1	0	0	0	1	0	
2	0	0	1	0	1	1
3	0	0	1	1	1	
4	0	1	0	0	0	P
5	0	1	0	1	1	
6	0	1	1	0	1	~P
7	0	1	1	1	0	
8	1	0	0	0	1	
9	1	0	0	1	1	
10	1	0	1	0	1	
11	1	0	1	1	1	
12	1	1	0	0	0	
13	1	1	0	1	X	
14	1	1	1	0	1	
15	1	1	1	1	0	

Jak widać układ ten ma 4 wejścia (M, N, O i P), natomiast my chcemy go zrealizować na multiplexerze 74151, czyli dysponującym jedynie 3 wejściami adresujących. Co w tej sytuacji?

Algorytm postępowania jest następujący:

- ✓ Zaczynamy od podziału tabeli, co dwa wiersze, tak jak pokolorowano wyżej.
- ✓ I patrzymy na pierwszą parę wierszy. Kiedy wejścia MNO są równe 000, wyjście jest równe 0, **niezależnie** od wejścia P.
- ✓ Drugi wiersz. Kiedy MNO=001, OUT=1, **niezależnie** od P.
- ✓ Trzecia para wierszy. MNO=010, OUT jest równe ile? Tym razem nie jest to jednoznaczne, OUT się zmienia, nie możemy powiedzieć że jest równe 0 lub 1 niezależnie od P. Więc może jest **zależne** od P? No właśnie, OUT w tym przypadku jest **takie samo** jak P. No to dalej...
- ✓ Trzecia para wierszy. Znow OUT zależy od P, tyle że jest odwrotnością P, czyli inaczej mówiąc **negacją** P.
- ✓ I tak dalej... do końca tabeli.

Skoro tak, to układ realizujący tę funkcję wygląda tak jak na rysunku poniżej (wyjściem układu jest pin Y (nóżka nr 5)). Wejścia CBA są dla nas w tym momencie wejściami MNO naszego układu, wejście P istnieje „obok”:

