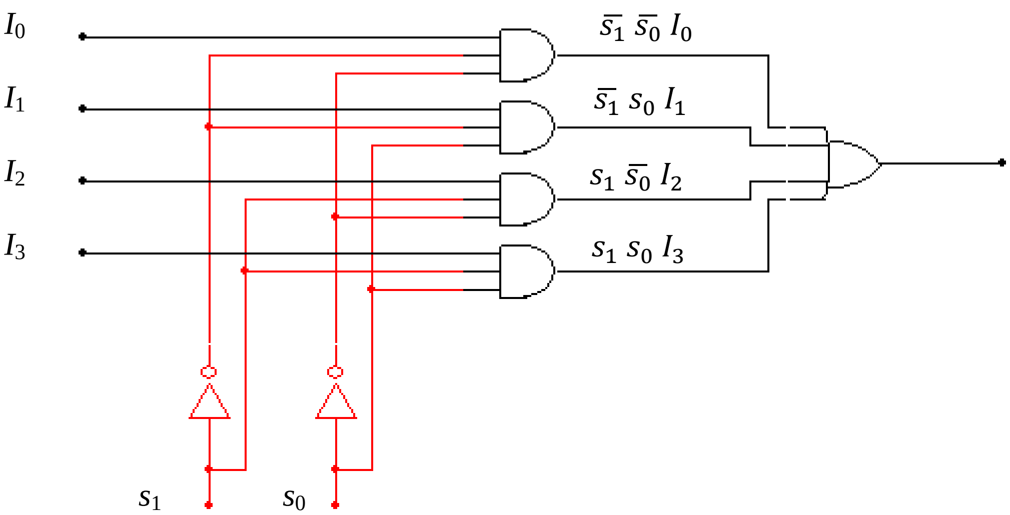
TECHNIKA CYFROWA
Laboratorium 2014/15
ćw. 2
Multipleksery i demultipleksery

Funkcjonalność **multipleksera** prezentuje poniższy rysunek. Multiplekser umożliwia wybór jednego z wejść informacyjnych ($I_0 ... I_3$), z którego sygnał jest przesyłany na wyjście. Numer wejścia jest określony przez sygnały na wejściach adresowych ($s_0 ... s_1$).



Podstawą dla realizacji funkcji logicznych z użyciem multipleksera jest **twierdzenie Shannona o dekompozycji**.

Dla dekompozycji wobec jednej zmiennej:

$$g(x_1x_2x_3\ldots x_n) = x_1 g(1x_2\ldots x_n) + \overline{x_1} g(0x_2\ldots x_n)$$ **- dla postaci dysjunkcyjnej**

$$g(x_1x_2x_3\ldots x_n) = (x_1 + g(0x_2\ldots x_n))(\overline{x_1} + g(1x_2\ldots x_n))$$ **- dla postaci koniunkcyjnej**

Dla dekompozycji wobec dwóch zmiennych odpowiednio:

$$g(x_1x_2x_3\ldots x_n) = x_1x_2 g(11x_3\ldots x_n) + x_1\overline{x_2} g(10x_3\ldots x_n) + \overline{x_1}x_2 g(01x_3\ldots x_n) + \overline{x_1}\,\overline{x_2} g(00x_3\ldots x_n)$$

$$g(x_1x_2x_3\ldots x_n) = (x_1 + x_2 + g(00x_3\ldots x_n)) + (x_1 + \overline{x_2} + g(01x_3\ldots x_n)) + (\overline{x_1} + x_2 + g(10x_3\ldots x_n)) + (\overline{x_1} + \overline{x_2} + g(11x_3\ldots x_n))$$

<u>Przykład 1</u>

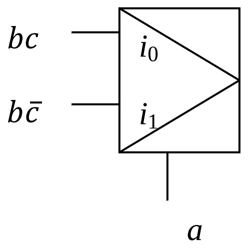
Niech $f(a,b,c) = \bar{a}bc + ab\bar{c}$. Dokonajmy dekompozycji funkcji względem zmiennej *a*:

$$f(a,b,c) = \bar{a}\underbrace{f(0,b,c)}_{I_0} + a\underbrace{f(1,b,c)}_{I_1}$$

Wyznaczamy funkcje resztkowe:

$$I_0 = \bar{0}\cdot bc + 0\cdot b\bar{c} = 1\cdot bc = bc$$
$$I_1 = \bar{1}\cdot bc + 1\cdot b\bar{c} = 1\cdot b\bar{c} = b\bar{c}$$

Funkcję I_0 przyporządkowujemy do wejścia informacyjnego i_0, a I_1 – odpowiednio do wejścia i_1:

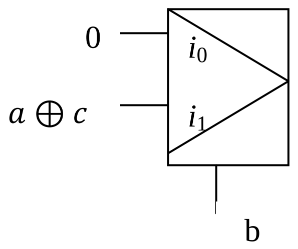


Jeżeli jednak dekompozycja miałaby być dokonana względem zmiennej b, to otrzymuje się odpowiednio:

$$f(a,b,c) = \bar{b}\underbrace{f(a,0,c)}_{I_0} + b\underbrace{f(a,1,c)}_{I_1}$$

i wobec tego:

$$I_0 = 0 \cdot \bar{a}c + 0 \cdot a\bar{c} = 0$$
$$I_1 = 1 \cdot \bar{a}c + 1 \cdot a\bar{c} = \bar{a}c + a\bar{c} = a \oplus c$$



<u>Przykład 2</u>

Rozważmy realizację funkcji logicznej $f\,(a,b,c,d)=\sum m(0,1,2,3,5,6,9,10) + d(12,14)$ z użyciem multipleksera o dwóch wejściach adresowych. Procedura ustalenia funkcji resztkowych wymaga specyficznego podejścia, jeżeli wartości funkcji dla niektórych argumentów są *don't care*, gdyż stosując dekompozycję Shannona wobec minimalnej postaci dysjunkcyjnej, traci się informację o położeniu wartości *don't care* i w efekcie postać funkcji resztkowych może nie być minimalna.

Najprościej jest ustalić funkcje resztkowe korzystając z fragmentów siatki Karnaugha. W przypadku dekompozycji względem zmiennych, których wartości sąsiadują ze sobą w wierszach lub kolumnach oryginalnej siatki Karnaugha. Przykładowo, zdekomponujmy funkcję *f* względem zmiennych (*a*, *b*). Oryginalna siatka Karnaugha pokazana jest na rysunku poniżej. Rozpatrując niezależnie od siebie poszczególne wiersze, niejako zakładamy konkretną wartość zmiennych (*a*, *b*), a wartości w danym wierszu zależą już wyłącznie od pozostałych zmiennych (*c*, *d*) – są to wartości funkcji resztkowych, które należy w następnym kroku zminimalizować.

ab \ cd	00	01	11	10
00	1	1	1	1
01	0	1	0	1
11	x	0	0	x
10	0	1	0	1

Ustalenie zminimalizowanej postaci funkcji resztkowych nie nastręcza trudności. Prosta analiza wartości funkcji resztkowych dla poszczególnych wartości (*c*, *d*) pozwala stwierdzić, że :

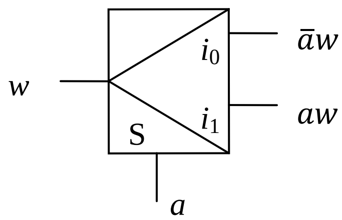
- funkcja resztkowa I_0 przyjmuje wartość 1 dla dowolnych wartości (*c*, *d*), zatem $I_0 = 1$,
- funkcja resztkowa I_1 przyjmuje wartośc 1 dla (*cd*) = (01) i (*cd*) = (10), więc $I_1 = c \oplus d$,
- funkcja resztkowa I_2 jest identyczna jak I_1,
- funkcja resztkowa I_3 przyjmuje wartość 0 dla (*cd*) = (01) i (*cd*) = (11), a poza tym wartości są *don't care*, zatem najprostszym rozwiązaniem jest $I_3 = 0$.

W sytuacji, gdy żądana jest dekompozycja funkcji względem zmiennych, których wartości nie sąsiadują ze sobą w wierszach/kolumnach siatki Karnaugha, należy przygotować zmodyfikowaną siatkę Karnaugha. Pamiętamy o tym, że poszczególne pola nowej siatki odpowiadają teraz innym indeksom *i* identyfikującym stany wejściowe. Jeżeli *a* jest zmienną zapisywaną na najbardziej znaczącej pozycji, $i = 2^3a + 2^2b + 2^1c + 2^0d$. W poniższej tabeli pokazano indeksy dla poszczególnych pól siatki Karnaugha, jeżeli wiersze odpowiadają poszczególnym parom wartości zmiennych

ad \ bc	00	01	11	10
00	0	2	6	4
01	1	3	7	5
11	9	11	15	13
10	8	10	14	12

* * *

Demultiplekser działa odwrotnie w stosunku do multipleksera: sygnał z pojedynczego wejścia informacyjnego jest przesyłany na wyjście, którego numer jest określony poprzez „adres” podany na wejścia sterujące.



Niech funkcja wejściowa będzie oznaczona w. Można zauważyć, że na wyjściu i_0 realizowana jest funkcja $\bar{a}w$, a na wyjściu i_1 – funkcja aw.

W szczególności, jeśli w = 1, na poszczególnych wyjściach demultipleksera realizowane są wszystkie mintermy dla tylu zmiennych logicznych, ile jest wejść adresowych demultipleksera.

Zadanie 1. (1p) Zrealizuj wykorzystując MUX o dwóch wejściach sterujących funkcję $f(a,b,c,d) = \sum m(0,2,3,5,6,12) + d(9,14)$.

Zaliczenie zadania:
- na kartce: wyznaczenie funkcji resztkowych
- w symulatorze: demonstracja działającego układu

Zadanie 2. (1p) Powtórz zadanie 1. wykorzystując multiplekser o trzech wejściach sterujących. Zmienne, wobec których ma być wykonana dekompozycja, poda prowadzący.

Zaliczenie zadania:
- na kartce: wyznaczenie funkcji resztkowych
- w symulatorze: demonstracja działającego układu

Zadanie 3. (0,5 p) Zrealizuj funkcję z zadania 1. wykorzystując tak jak wówczas jeden multiplekser o dwóch wejściach sterujących. Na jego wejścia sterujące podaj te same zmienne co w zad. 1. Jednak zamiast bramek i negatorów do realizacji funkcji resztkowych (podawanych na wejścia informacyjne) można wykorzystać wyłącznie multipleksery o jednym wejściu sterującym.

Zaliczenie zadania: należy wykonać symulację układu w Multisimie.

Zadanie 4. (0,5 p) Wypełnij poniższą tabelę jedynkami w polach odpowiadających literom, które występują w Twoim imieniu i nazwisku. Zrealizuj funkcję o takiej tablicy prawdy korzystając z demultipleksera o czterech wejściach adresowych. Na jego wyjściach wolno użyć bramek logicznych jednego typu: NAND lub AND.

a	ą	ł	e	ć	b	d	g	f	ę	c	h	i	l	k	j
n	m	t	o	ó	ń	ż	s	ś	p	r	u	y	w	ź	z
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**

Zaliczenie zadania: pokazanie działającego układu w MultiSim.