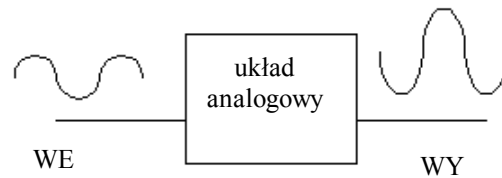


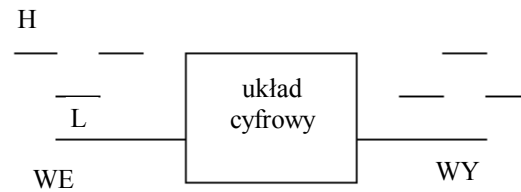
# WYKŁAD 7

## CYFROWE UKŁADY SCALONE

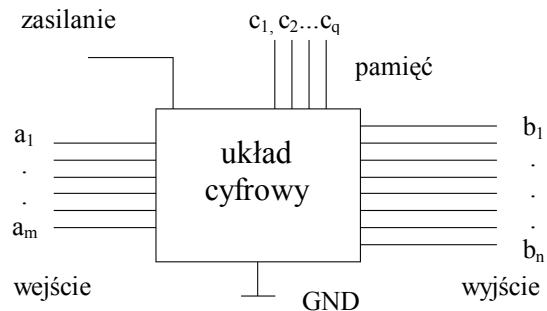
Układy analogowe są przystosowane do przetwarzania napięć (lub prądów), których wartości zawierają się w pewnym przedziale wartości.



Układy cyfrowe służą do przetwarzania sygnałów o dwóch wielkościach napięć (ewentualnie prądów): wysokiej (H-high) i niskiej (L-low).



Na ogół układ cyfrowy posiada  $n$  wejść,  $m$  wyjść i  $q$  stanów pamięciowych. Każdy z wektorów  $a$ ,  $b$ , czy  $c$  nazywamy **słowem logicznym**. Każdy element słowa logicznego nazywamy **bitem**. Słowo ośmiobitowe nazywane jest **bajtem**. Stany na wyjściu zależą od aktualnej sytuacji na wejściu. Stany pamięciowe zależą zarówno od aktualnej konfiguracji na wejściu jak i od słów, jakie istniały tam w poprzednich chwilach czasu.

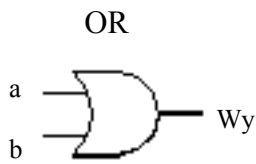


Pracę układów cyfrowych opisuje się za pomocą dwuwartościowej algebry Boole'a, zwanej logiką matematyczną. W tym celu poziomom H i L układu cyfrowego przyporządkowuje się wartości logiczne - np. odpowiednio „1” (prawda) i „0” (fałsz) (tzw. logika dodatnia) lub odwrotnie „0” i „1” (logika ujemna). Układy cyfrowe są więc układami wykonującymi pewne funkcje logiczne.

Podstawowe twierdzenie logiczne :

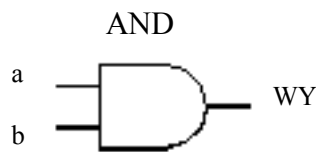
**Każdą funkcję logiczną można złożyć z kombinacji trzech podstawowych działań logicznych : sumy (alternatywy- lub - OR), iloczynu (koniunkcji - i - AND) oraz negacji (inwersji - nie - NOT).**

Urządzenia elektroniczne realizujące te funkcje nazywamy **bramkami** odpowiednio **OR**, **AND** i **NOT**. Są one dostarczane w wyspecjalizowanych układach cyfrowych.



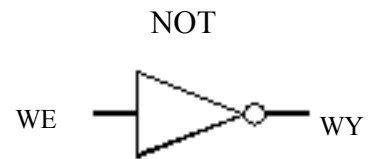
a	b	Wy
1	1	1
1	0	1
0	1	1
0	0	0

$$Wy = a + b$$



a	b	Wy
1	1	1
1	0	0
0	1	0
0	0	0

$$Wy = a * b$$



WE	WY
1	0
0	1

$$WY = \overline{WE}$$

Powszechnie wykorzystuje się **prawa de Morgana** :

$$\overline{a + b} = \overline{a} * \overline{b}$$

$$\overline{a * b} = \overline{a} + \overline{b}$$

Najbardziej uniwersalnymi bramkami są bramki NAND (NOT-AND) i NOR (NOT-OR).

NAND



a	b	Wy
1	1	0
1	0	1
0	1	1
0	0	1

NOR



a	b	Wy
1	1	0
1	0	0
0	1	0
0	0	1

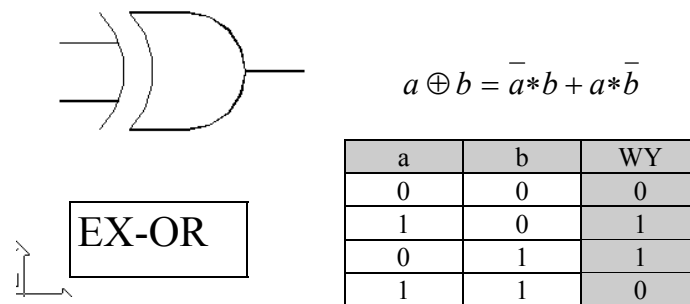
Każą funkcję logiczną można utworzyć z pewnej kombinacji **tylko** bramek NAND lub **tylko** bramek NOR.

Dowód - w trakcie wykładu.

Zmiana funkcji logicznej danej bramki w przypadku zmiany rodzaju logiki

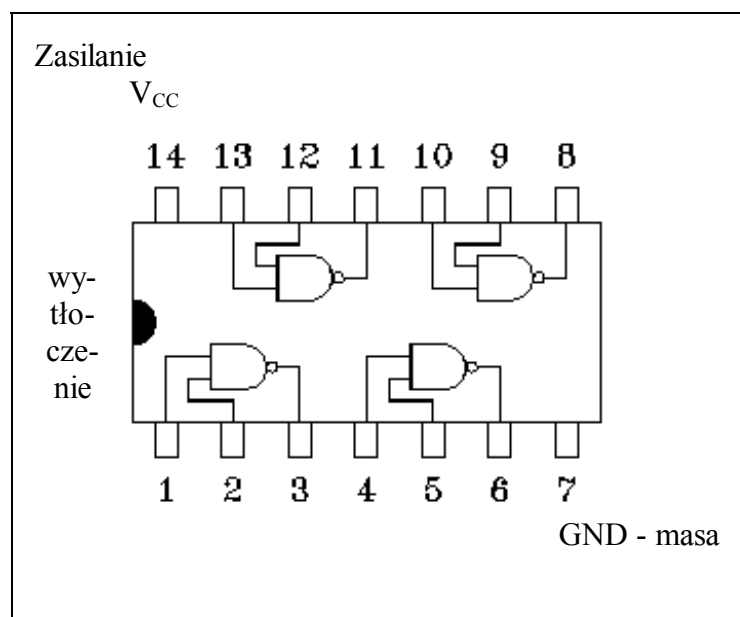
LOGIKA	
dodatnia	ujemna
AND	OR
OR	AND
NAND	NOR
NOR	NAND

Jedną z bardziej użytecznych funkcji logicznych jest Exclusive OR.



Z bramek cyfrowych (bramek logicznych) można łatwo budować rozmaite użyteczne układy elektroniczne. Grupy bramek cyfrowych tworzą tzw. rodziny. Najbardziej rozpowszechniona jest rodzina bramek TTL (Transistor - Transistor Logic), a w niej seria 74. Na przykład, w układzie scalonym typu 74xx00 znajdują się cztery bramki NAND (xx oznacza rodzaj bramki: S-szybka, LS-szybka małej mocy itd):

Po zasileniu układu scalonego (między końcówkami oznaczonymi przez  $V_{CC}$  i GND) założony schemat realizuje się poprzez proste łączenie wejść i wyjść bramek. Poziomy logiczne określone są przez wartość napięcia odpowiednio między wejściem (lub wyjściem) a GND.



Inne układy z tej serii zawierają inne funkcje logiczne, np. 7402 - cztery bramki NOR, 7440 - 8-wejściową bramkę NAND itd. - patrz - instrukcja do ćwiczenia „**Cyfrowe układy scalone**”.

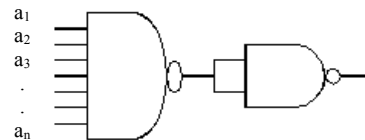
Uprawiając elektronikę z układami TTL serii 74 należy wiedzieć, że :

- układy zasilane napięciem  $5 \pm 0.25$  V;
- układy pracują w logice dodatniej;
- napięcie odpowiadające logicznemu zeru zawiera się między 0 a 0.4 V z dopuszczalnym marginesem błędów 0.4 V;
- napięcie odpowiadające logicznej jedynce wynosi 3.3 V lecz nie mniej niż 2.4 V z marginesem błędów 0.4 V;
- wejście bramki niepodłączone do niczego znajduje się w stanie logicznym „1”;
- wyjścia bramek nie wolno łączyć równolegle!!! Może to spowodować ich uszkodzenie;
- średni czas propagacji sygnału przez bramkę wynosi od 1 do 30 ns (typowo - około 10 ns);
- średnie zużycie mocy przez bramkę wynosi około 10 mW;

Użyteczne schematy:

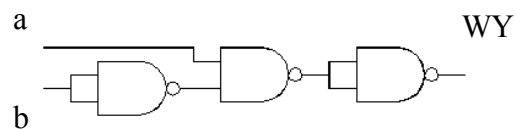
- Wielowejściowa funkcja AND.

Wartość logiczna „1” pojawia się na wyjściu wtedy i tylko wtedy, gdy stan logiczny wszystkich wejść wynosi „1”. Przez fizyków bywa nazywany układem koincydencyjnym.



- Układ antykoincydencyjny

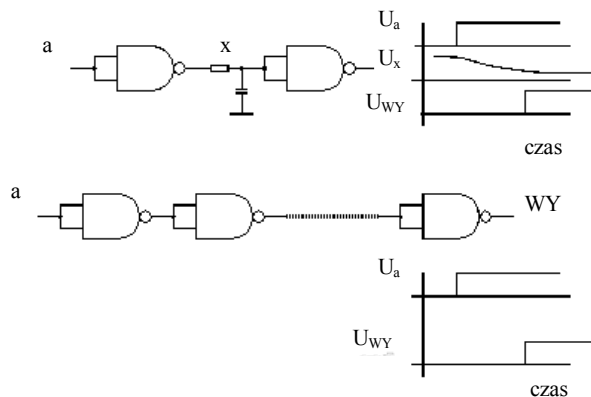
$$f = a * \bar{b}$$



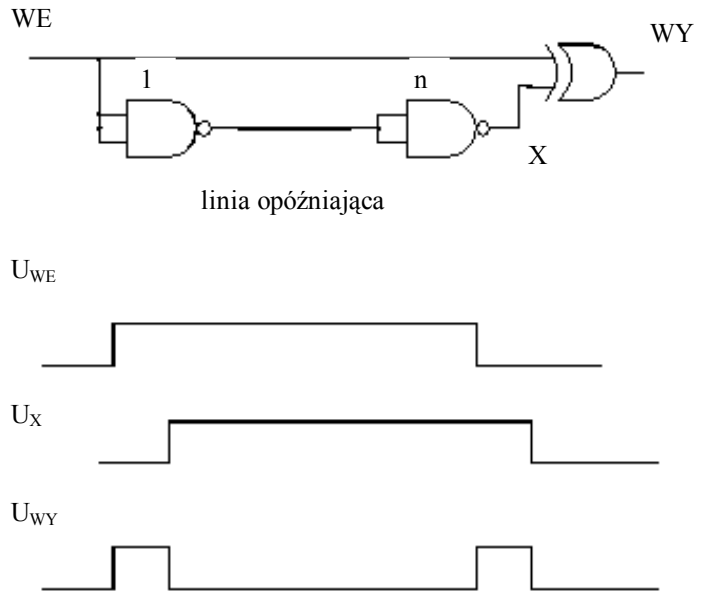
- Układ opóźniający.

W pierwszym przypadku - opóźnienie jest proporcjonalne do stałej czasowej RC.

W drugim - do liczby bramek o czasu propagacji sygnału przez nie.

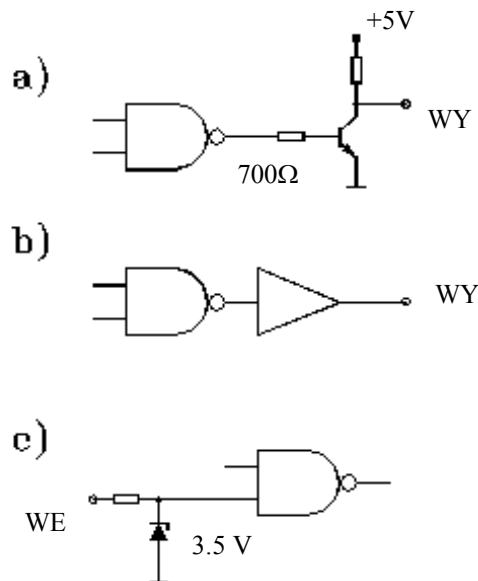


- Cyfrowy układ różniczkujący - wytwarzający sygnały w momentach rozpoczęcia i zakończenia pewnego sygnału. W przypadku, gdy liczba bramek ( $n$ ) w linii opóźniającej jest nieparzysta, sygnał wyjściowy ma odwróconą polaryzację.



Bramki nie powinny być nadmiernie obciążane. Każdy układ cyfrowy ma określoną **obciążalność**, czyli liczbę mówiącą ile wejść cyfrowych może być podłączonych do danego wyjścia.

W przypadku, gdy układ cyfrowy ma sterować innym układem należy posłużyć się wzmacniaczem np. tranzystorowym (a) lub **driverem** (b) - wzmacniaczem znajdującym się w rodzinie cyfrowych układów scalonych zwiększającym obciążalność wyjścia bramki.



Gdy do układu cyfrowego wprowadza się sygnał sterujący z zewnątrz, należy zadbać o zachowanie standardowych napięć i polaryzacji.

Na rysunku c pokazano przykład rozwiązania za pomocą diody Zenera, która nie dopuszcza do przekroczenia na wejściu bramki napięcia 3.5 V, jak również do pojawienia się napięć o odwróconej polaryzacji i napięciu większym niż -0.7 V.

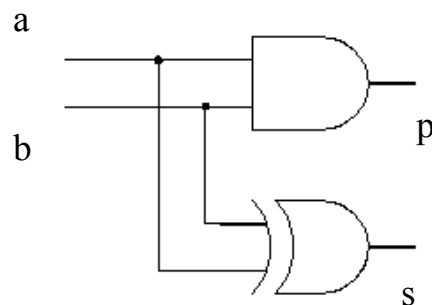
## Układy arytmetyczne.

Każde słowo logiczne może być interpretowane jako pewna liczba zapisana w danym **kodzie binarnym**. Na przykład słowo (1011) w kodzie naturalnym jest liczbą 11 :  $1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3$ . Za pomocą cyfrowych układów elektronicznych można konstruować układy dokonujące operacji arytmetycznych na takich liczbach. Ich podstawą są **półsumatory** - układy dodające dwie liczby jednobitowe a i b. W wyniku sumowania powstaje liczba dwubitowa której elementami są suma s i przeniesienie p :

a	b	s	p
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

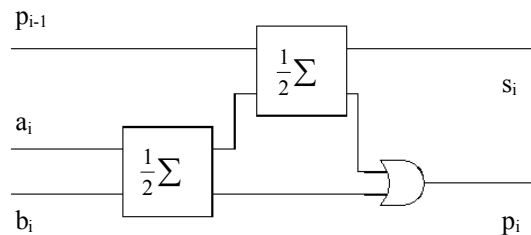
s - funkcja EXOR,

p - funkcja AND.

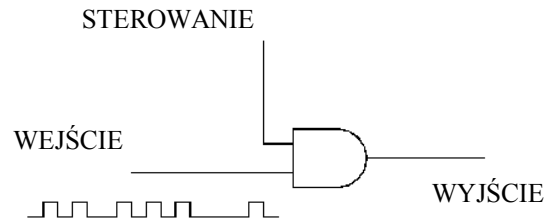


**Sumator jednobitowy**, który może pracować przy sumowaniu na i-tej pozycji poza danymi  $a_i$  i  $b_i$  przyjmuje także przeniesienie z pozycji poprzedniej  $p_{i-1}$  ; generuje sumę  $s_i$  i przeniesienie na pozycję następną  $p_i$ :

$a_i$	$b_i$	$p_{i-1}$	$s_i$	$p_i$
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
0	0	1	1	0
1	1	0	0	1
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

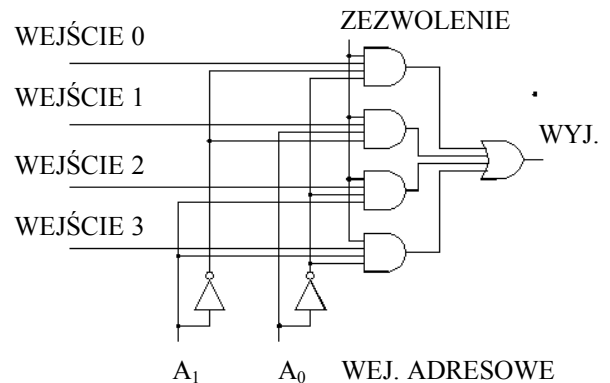


Bramka AND może być wykorzystana do sterowania przepływem informacji. Ciąg impulsów podany na wejście układu dostanie się do wyjścia wtedy i tylko wtedy gdy na wejściu sterującym pojawi się stan logiczny „1”.



Urządzeniami przeznaczonymi do kontroli przepływu informacji są **multipleksery** i **demultipleksery**.

W przedstawionym obok przykładzie multipleksera informacja podawana jest na czterokanałowe wejście. Do wyjścia dostanie się tylko informacja z kanału, którego adres zostanie wywołany przez podanie na wejście adresowe dwubitowego adresu wejścia informacyjnego.



W demultiplekszerze informacja z wejścia jest kierowana do tego wyjścia, którego adres został wywołany przez podanie na wejście adresowe numery wyjścia informacyjnego.

Działanie obu urządzeń jest możliwe dopiero wtedy, gdy wejście „zezwoleń” znajduje się w stanie logicznym „1”.

