$$p_1' = \bar{a}b + \bar{a}p_1 + bp_1 = (\bar{a}+b)(\bar{a}+p_1)(b+p_1)$$
$$p_2' = abp_2 + \bar{a}\bar{b}p_2 = p_2(a+\bar{b})(\bar{a}+b)$$

Poszukując wspólnych członów obu funkcji można napisać

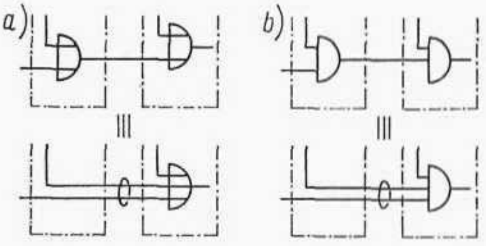
$$p_1' = (\bar{a}+b)(p_1+\bar{a}b) = (\bar{a}+b)(p_1+\overline{a+\bar{b}})$$

Schemat układu przedstawiony jest na rys. 3-20b. Dla pierwszego segmentu jest $p_1 = 0$. $p_2 = 1$ („poprzednie odcinki równe"), więc

$$p_1' = \bar{a}b = \overline{a+\bar{b}}$$
$$p_2' = ab + \bar{a}\bar{b} = (a+\bar{b})(\bar{a}+b)$$

Przeniesienia z ostatniego segmentu są wyjściami układu. Można zauważyć, że y_2 jest iloczynem członów typu $(a+\bar{b})(\bar{a}+b)$, więc realizację iteracyjną tego sygnału można zastąpić działaniem równoległym, a wówczas — przy np. $n = 5$ — zamiast pięciu elementów I 3-wejściowych potrzebny będzie jeden element 10-wejściowy.

Takie rozwiązanie dosyć często znajduje zastosowanie w układach i — ogólnie — polega na zastąpieniu jednego przewodu wiązką innych. Przykłady tworzenia wiązek przedstawia rys. 3-21.



Rys. 3-21. Przykłady stosowania wiązek

3.3. UKŁADY Z ELEMENTÓW NOR ALBO NAND

3.3.1. TRANSFORMACJA UKŁADÓW

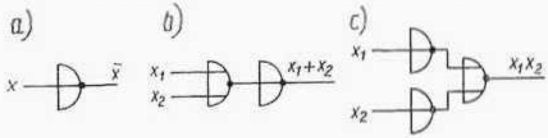
Podane wyżej zasady budowania układów minimalnych z elementów I, LUB, NIE mają bardzo ograniczony zakres użyteczności w praktyce, gdyż prawie wszystkie systemy elementów logicznych wykorzystują elementy NOR albo NAND, co najwyżej uzupełniając je pomocniczymi elementami I albo LUB. Znajomość metod syntezy układów z elementami I, LUB, NIE jest jednak niezbędna do pełnego poznania zasad budowy układów z elementów NOR i NAND, a najprostsza metoda projektowania tych ostatnich polega po prostu na przekształcaniu schematów z elementami I, LUB, NIE. Metoda ta nosi nazwę *transformacji* układów i wykorzystuje podstawowe schematy realizacji sumy, iloczynu i negacji za pomocą elementów NOR albo NAND.

NOR realizuje funkcję

$$x_1 \downarrow x_2 = \overline{x_1 + x_2} = \bar{x}_1 \cdot \bar{x}_2$$

za pomocą której można łatwo uzyskać negację:

$$x_1 \downarrow x_1 = \bar{x}_1 \quad \text{albo} \quad x_1 \downarrow 0 = \bar{x}_1$$



Rys. 3-22. Realizacja negacji, sumy i iloczynu z elementów NOR

Ponieważ sygnał 0 może być zwykle utożsamiany z brakiem sygnału, negatorem będzie też NOR o jednym wejściu (rys. 3-22). Negując wyjściowy sygnał NOR'a uzyskuje się sumę logiczną (rys. 3-22b):

$$\overline{x_1 \downarrow x_2} = \overline{\overline{x_1 + x_2}} = x_1 + x_2$$

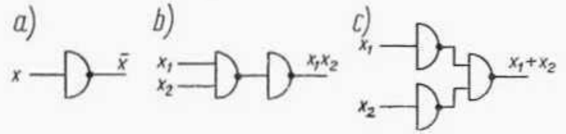
natomiast negując sygnały wejściowe — uzyskuje się iloczyn logiczny (rys. 3-22c):

$$\bar{x}_1 \downarrow \bar{x}_2 = \bar{\bar{x}}_1 \cdot \bar{\bar{x}}_2 = x_1 \cdot x_2$$

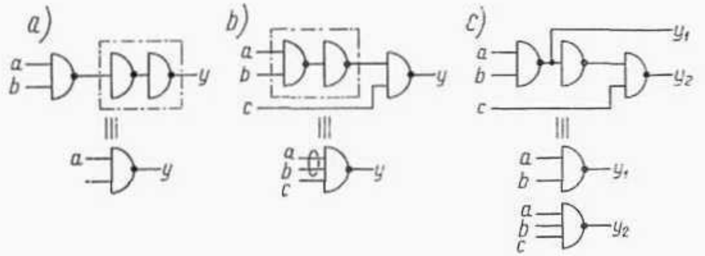
W podobny sposób można otrzymać sumę, iloczyn i negację z elementów NAND (rys. 3-23). Oprócz tych schematów elementarnych, przy transformacji wykorzystuje się jeszcze dosyć oczywiste właściwości przedstawione na rys. 3-24, słuszne zarówno dla elementu NOR jak i dla NAND.

Przykładem zastosowania tych przekształceń niech będzie funkcja równoważności

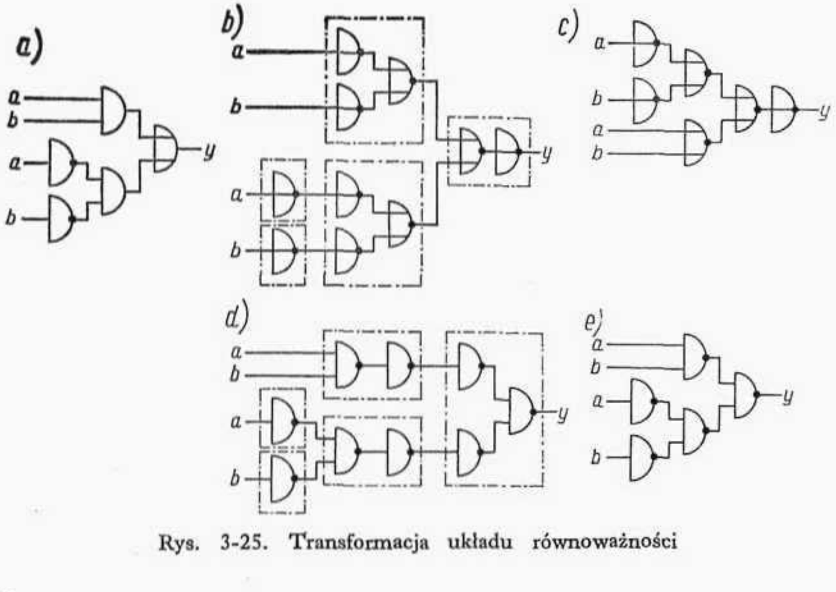
$$y = ab + \bar{a}\bar{b}$$



Rys. 3-23. Realizacja negacji, iloczynu i sumy z elementów NAND



Rys. 3-24. Przekształcanie układów z elementami NAND i NOR



Rys. 3-25. Transformacja układu równoważności

Schemat z rys. 3-25a można przekształcić jak na rys. 3-25b za pomocą rys. 3-22, a stąd — na podstawie rys. 3-24 — uzyskuje się rozwiązanie jak na rys. 3-25c. Podobnie — korzystając ze schematu z rys. 3-23 — uzyskuje się rys. 3-25d i e. Taka transformacja schematów jest pracochłonna i dlatego zwykle, już na podstawie formuły logicznej, rysuje się uproszczony schemat logiczny, korzystając z prostych zasad, wynikających z rys. 3-26. Przedstawiono tu wyniki przekształceń z rys. 3-25 oraz podobne dla funkcji równoważności zapisanej w postaci

$$y = (a+\bar{b})(\bar{a}+b)$$

Zaznaczone poziomy schematów z elementami I, LUB, NIE ułatwiają rysowanie odpowiedników z elementami NOR albo NAND.

Zasady ogólne transformacji można zapisać w sposób następujący:

1. Jeśli transformowana jest formuła typu „suma iloczynów" to:

— poziom I buduje się z dwóch elementów NOR albo jednego elementu NAND;

— poziom II zawiera tyle elementów NOR albo NAND, ile elementarnych iloczynów występuje w formule;

— poziom III w przypadku elementów NOR wytwarza negacje zmiennych, które w formule nie były zanegowane, a w przypadku elementów NAND — negacje tych zmiennych, które w formule były zanegowane.

2. Jeśli transformowana jest formuła typu „iloczyn sum", to:

— poziom I buduje się z dwóch elementów NAND albo jednego elementu NOR;

— poziom II zawiera tyle elementów NAND albo NOR, ile elementarnych sum występuje w formule;

— poziom III w przypadku elementów NAND wytwarza negacje zmiennych, które w formule nie były zanegowane, a w przypadku elementów NOR — negacje tych zmiennych, które w formule były zanegowane.

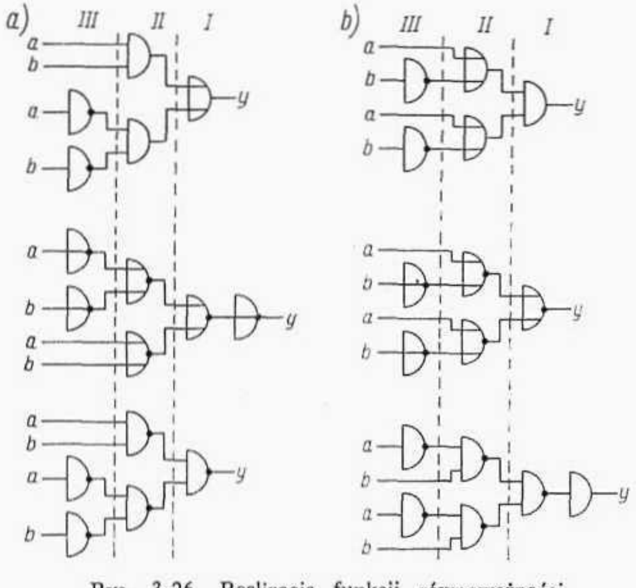
Stosowanie tych zasad umożliwia bezpośrednie rysowanie schematów na podstawie wyrażenia logicznego o postaci normalnej. Takie same rezultaty można uzyskać, przekształcając zapis funkcji w taki sposób, by wyrugować operacje LUB (jeśli układ ma być budowany z elementów

NAND) albo operacje I (w przypadku elementów NOR). Wykorzystuje się w tym celu prawa de Morgana; na przykład

$$y = ab+\bar{a}\bar{b} = \overline{\overline{ab+\bar{a}\bar{b}}} = \overline{\overline{\overline{(\bar{a}+\bar{b})}+\overline{(a+b)}}} = \overline{\overline{(ab)}\,\overline{(\bar{a}\bar{b})}}$$

$$y = (a+\bar{b})(\bar{a}+b) = \overline{\overline{(a+\bar{b})(\bar{a}+b)}} = \overline{\overline{(a+\bar{b})}+\overline{(\bar{a}+b)}} = \overline{\overline{\overline{(\bar{a}b)}\,\overline{(a\bar{b})}}}$$

Każda kreska negacji w tych wyrażeniach odpowiada jednemu elementowi z rys. 3-26.



Rys. 3-26. Realizacja funkcji równoważności

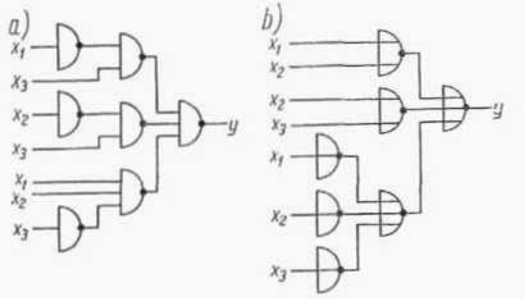
Liczba elementów potrzebnych do realizacji funkcji silnie zależy od liczby argumentów zanegowanych, ale na podstawie podanych zasad można stwierdzić, że przy jednakowych pozostałych warunkach układ typu „suma iloczynów" jest prostszy z elementami NAND, a układ typu „iloczyn sum" — z elementami NOR. Wynika to ze sposobów realizacji ostatniego stopnia układu (poziom I).

Na przykład funkcję

$$y = \bar{x}_1x_3+\bar{x}_2x_3+x_1x_2\bar{x}_3 = (x_1+x_2)(x_2+x_3)(\bar{x}_1+\bar{x}_2+\bar{x}_3)$$

8 NOR, 7 NAND   7 NOR, 8 NAND

można zrealizować za pomocą podanej liczby elementów (gdy negacje zmiennych nie są dane), więc odpowiednie układy będą miały postać jak na rys. 3-27.



Rys. 3-27. Realizacja funkcji za pomocą elementów NAND i NOR

Gdy negacje argumentów są do dyspozycji, najprostsze układy uzyskuje się zazwyczaj z postaci normalnych funkcji (tzn. sum iloczynów lub iloczynów sum). Gdy jednak negacje argumentów nie są dane, istotne uproszczenie układu można osiągnąć przez faktoryzację i inne metody przekształcania funkcji, np.

$$\underset{6\ \mathrm{NOR}}{(a+\bar{b})(a+\bar{d})c} = \underset{4\ \mathrm{NOR}}{(a+\bar{b}\,\bar{d})c}$$

$$\underset{6\ \mathrm{NAND}}{a\bar{b}+a\bar{c}+d} = \underset{4\ \mathrm{NAND}}{a(\bar{b}+\bar{c})+d}$$

albo

$$\underset{5\ \mathrm{NOR}}{(a+\bar{b})(\bar{a}+b)} = \underset{4\ \mathrm{NOR}}{(a+\bar{a}\bar{b})(\bar{a}\bar{b}+b)} \quad \text{gdyż} \quad a+\bar{a}b = a+b$$

Znalezienie takiej funkcji, by jej realizacja z elementów NOR albo NAND była najprostsza, jest zadaniem trudnym. Jedna z możliwych metod postępowania polega na przekształceniach typu algebraicznego.

3.3.2. SYNTEZA ALGEBRAICZNA

Niech zadanie polega na zaprojektowaniu układu z elementów NOR. Pojedynczy taki element realizuje funkcję, która jest albo negacją argumentu, albo iloczynem negacji argumentów, a więc każda funkcja zadana