

niesienie jest zanegowane i dlatego potrzebny jest negator. Negowanie wejść członów parzystych może być realizowane poza układem (rys. 5-23b) lub wewnątrz układu (rys. 5-23c). Człony wg rys. 5-20 umożliwiają bramkowanie sygnałów wejściowych (oddzielne lub łączne); w innych rozwiązaniach takie sygnały również można wprowadzić. Do pierwszego członu wprowadza się przeniesienie 0, albo zastępuje się go innym układem, tj. półsumatorem (o dwóch wejściach).

Struktury układów różnicy przedstawiono na rys. 5-24. Jeśli członem układu iteracyjnego jest jednobitowy układ różnicy — połączenia są oczywiste (rys. 5-24a). Jeśli stosuje się jednobitowe układy sumy, trzeba negować bity odjemnika i korygować zmiany sumowania. W przypadku gdy człon generuje przeniesienia bez negacji (rys. 5-24b), wynik pojawi się na wyjściach \bar{s} , jeśli natomiast człony negują przeniesienie (rys. 5-24c) — tylko co drugie wyjście będzie z \bar{s} .

5.3.3. SUMATORY DWÓJKOWO-DZIESIĘTNE RÓWNOLEGŁE

Dodawanie liczb przedstawionych w kodach dwójkowo-dziesiętnych wymaga, zgodnie z rozważaniami z p.1.4.2, wprowadzenia korekcji przesuwającej na skali liczbowej wyniki trafiające w niewykorzystywany zakres.

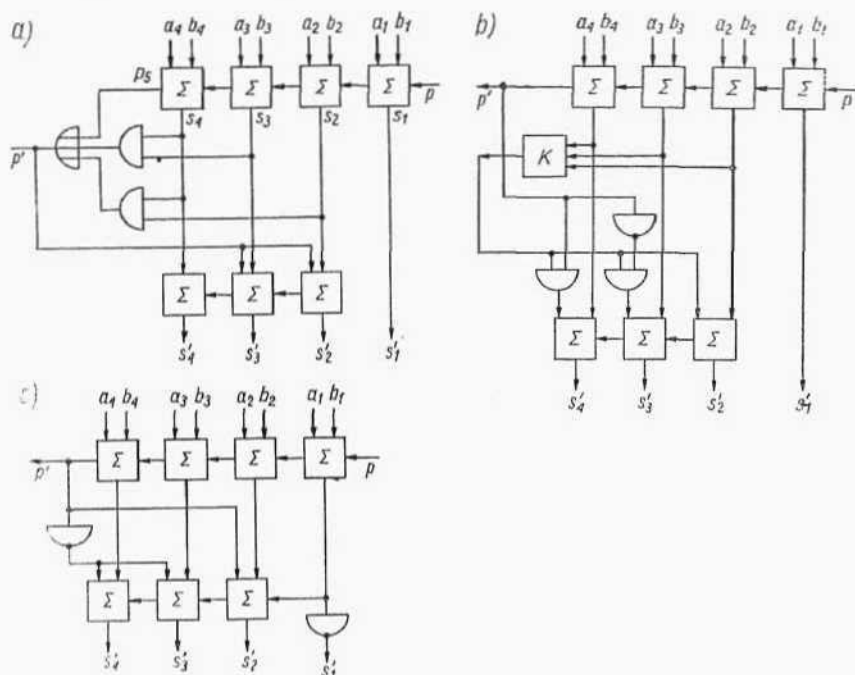
W przypadku kodu 8421 korekcja polega na dodaniu liczby 6, gdy pierwotna suma jest większa od 9. Wtedy również przeniesienie do następnej tetrady jest równe 1, co można wyrazić funkcją

$$p' = p_5 + s_4s_3 + s_4s_2 = (p_5 + s_4)(p_5 + s_3 + s_2)$$

Dodawanie liczby 6 realizuje się łatwo dodatkowym sumatorem, a więc ogólna struktura całego układu jest taka jak na rys. 5-25a. Dwa sumatory jednobitowe z dolnej warstwy można uprościć.

W przypadku kodu Aikena (IV) korekcja jest potrzebna, gdy pierwotna suma jest większa od (0100) i mniejsza od (1011), a polega na dodaniu liczby 6, gdy $p' = 0$, oraz odjęciu liczby 6 — gdy $p' = 1$. Zamiast odejmować 6 można dodać 10, gdyż sumator 4-bitowy bez przeniesienia działa modulo 16 (a $16 - 6 = 10$). Struktura całego układu jest przedstawiona na rys. 5-25b. Blok K generuje sygnał o wartości 1, gdy są spełnione podane wyżej warunki korekcji.

W przypadku kodu „+3” pierwotną sumę należy powiększyć o 3, jeśli przeniesienie jest 1, albo zmniejszyć o 3, gdy przeniesienie jest 0. Zamiast odejmować 3 można dodać $13 = 16 - 3$, uzyskując schemat jak na rys. 5-25c. Najmniej znaczący bit sumy podlega korekcji zawsze i dlatego odpowiedni półsumator zdegenerował się do postaci negacji.



Rys. 5-25. Sumatory dwójkowo-dziesiętne: a) kod 8421; b) kod Aikena; c) kod „+3”

W podobny sposób można budować sumatory równoległe i dla innych kodów dwójkowo-dziesiętnych. Układy odpowiadające tetradzie łączą się ze sobą iteracyjnie.

Schematy z rys. 5-25 można łatwo przystosować do zestawu posiadanych elementów. Wprowadzone podane struktury sugerują wykorzystanie półsumatorów i sumatorów jednobitowych, ale niekiedy może być celowe zastąpienie dolnej warstwy sumatorów (w większości — niepełnych) specjalizowanym układem kombinacyjnym. Przykłady takich

rozwiązań dla kodu 8421 są przedstawione na rys. 5-26. W pierwszym układzie każdy zespół elementów NAND ze zwartymi wyjściami można zastąpić elementem realizującym negację sumy iloczynów (jak na rys. 5-11f,g). W drugim rozwiązaniu ostatni sumator jednobitowy redukuje się do samego układu przeniesienia.

5.3.4. SUMATORY AKUMULUJĄCE

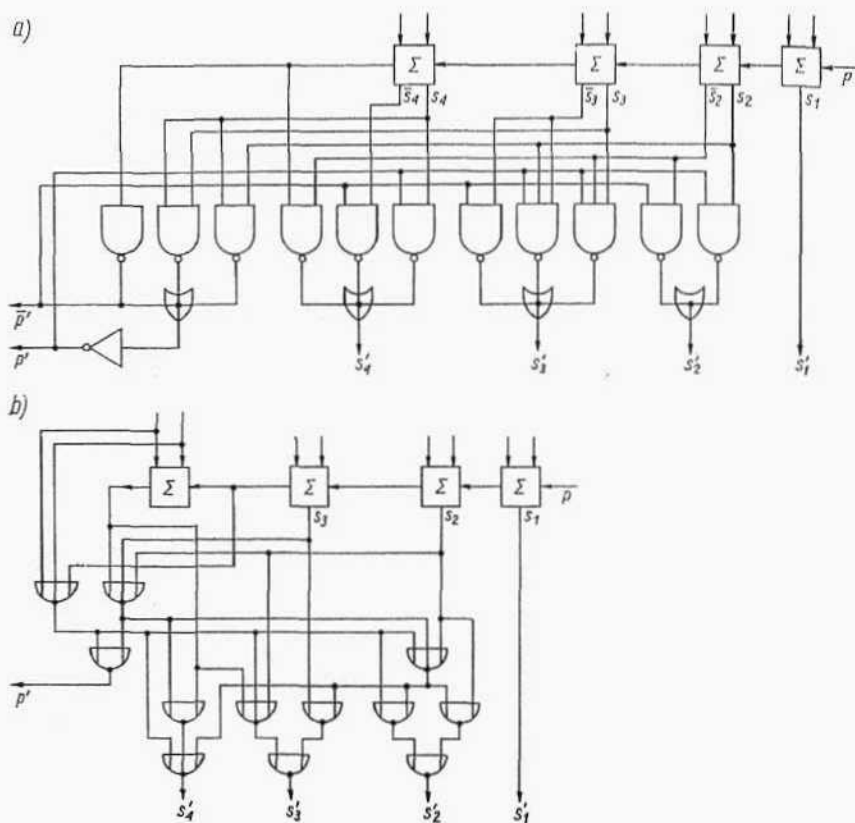
W opisanych wyżej sumatorach równoległych obydwie dadawane liczby pojawiały się na wejściach sumatora równocześnie. W wielu przypadkach jednak wskazane jest dodawanie liczb występujących w postaci równoległej, ale jedna po drugiej. Najbardziej oczywista realizacja odpowiedniego układu polega na wykorzystaniu sumatora równoległego, którego wyjścia s są (z odpowiednim opóźnieniem) przekazywane na jedno z wejść. Schemat takiego układu jednobitowego pokazano na rys. 5-27. Impulsy taktujące wpisują do przerzutnika sygnał $s = x \oplus y \oplus p$, a więc w układzie iteracyjnym złożonym z takich członów każdy impuls taktujący będzie powodował dodanie liczby X do zawartości pamięci Y i ulokowanie wyniku w tej pamięci.

Zastąpienie przerzutnika D przerzutnikiem T upraszcza nieco realizację, gdyż $T = x \oplus p$ i nie jest potrzebny sumator jednobitowy pełny. Projektowanie sumatora jako asynchronicznego układu dynamicznego prowadzi np. do wersji z rys. 5-27c.

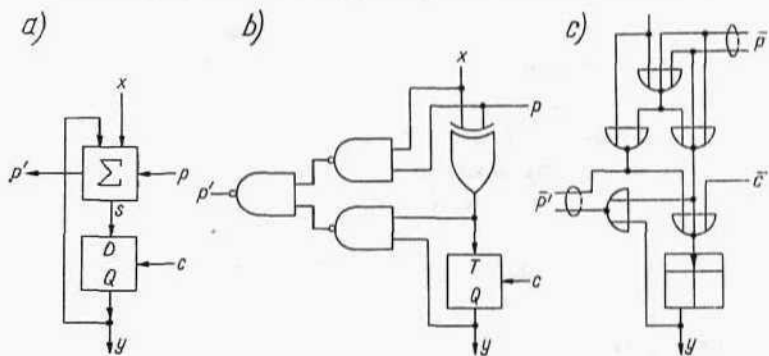
5.3.5. SUMATORY SZEREGOWE

Jeśli dodawane liczby wprowadzane są do sumatora bit po bicie (oczywiście — poczynając od bitu najmniej znaczącego), to cały sumator sprowadza się do sumatora jednobitowego i układu opóźnienia (zapamiętywania) przeniesienia. Ogólny schemat takiego układu jest przedstawiony na rys. 5-28; zmieniając realizację obydwu bloków można uzyskać wiele wersji. Synteza takiego układu była opisywana wyżej (rys. 4-28). Sumatory szeregowo budowane są wyłącznie dla naturalnego kodu dwójkowego.

Dodawanie równoległe trwa krótko, ale jest kosztowne; dodawanie szeregowo — przeciwnie — jest tanie ale długotrwałe. Optymalnym rozwiązaniem jest niekiedy droga pośrednia: *dodawanie równoległo-szeregowo*. Bity obydwu składników dzieli się wówczas na grupy; w ra-

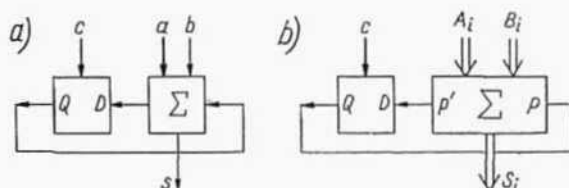


Rys. 5-26. Przykłady zmodyfikowanych sumatorów z kodem 8421



Rys. 5-27. Jednobitowe sumatory akumulujące z przełącznikami D, T i t

mach grupy dodawanie jest równoległe, a grupy następują po sobie sekwencyjnie. Szczególnie często takie rozwiązanie jest wykorzystywane przy dodawaniu liczb w kodach dwójkowo-dziesiętnych, gdyż tetrada tworzy naturalną grupę, a równoległe dodawanie w ramach dekady



Rys. 5-28. Sumator szeregowy (a) i równoległo-szeregowy (b)

umożliwia zastosowanie dowolnego kodu dwójkowo-dziesiętnego. Schemat odpowiedniego sumatora jest przedstawiony na rys. 5-28b, przy czym i oznacza kolejny numer dekady.

Elementy pamięciowe z rys. 5-27 i 5-28 mają jeszcze zawsze wejście z ustawiające ich stan początkowy, od którego zależy poprawne działanie układów.

5.3.6. KOMPARATORY RÓWNOLEGŁE

Układy porównujące liczby, czyli komparatory, występują w trzech różnych odmianach, sygnalizując występowanie relacji:

- 1) $A = B$; $A \neq B$
- 2) $A \geq B$; $A < B$
- 3) $A > B$; $A = B$; $A < B$

Pierwszy typ można zbudować z zestawu elementów równoważności (rys. 5-29ab), drugi jest szczególnym przypadkiem trzeciego, a więc najistotniejszy jest typ trzeci. Komparatory równoległe są najczęściej budowane jako układy iteracyjne, przy czym przeniesienia mogą mieć dwa kierunki: od mniej znaczących bitów do bardziej znaczących, albo przeciwnie. Pierwsza wersja jest zazwyczaj prostsza. Są stosowane dwa rodzaje kodów przeniesień (przeniesienia p_1, p_2 ostatniego członu są wyjściami y_1, y_2 układu)

	I — $\overline{p_1 p_2}$	II — $\overline{p_1 p_2}$
$A > B$	0 0	0 1
$A = B$	0 1	0 0
$A < B$	1 0	1 0

Dla obydwu kodów

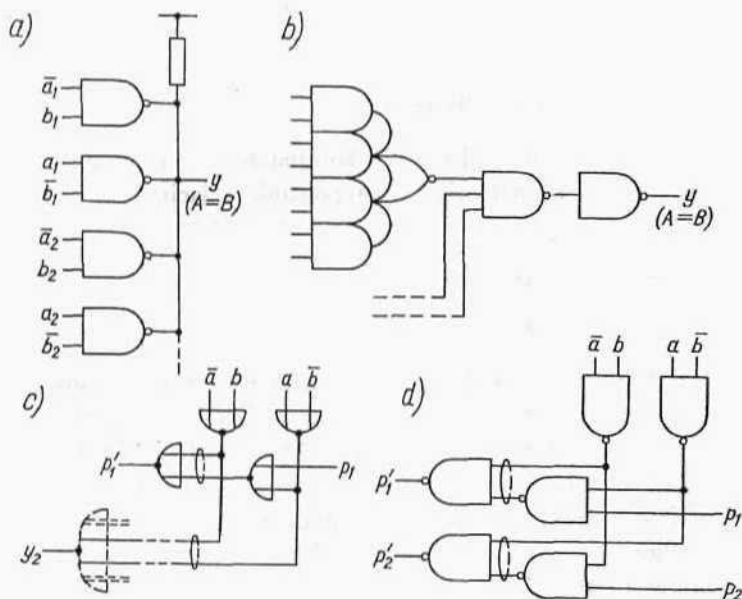
$$p'_1 = \bar{a}b + (\bar{a} + b)p_1 = (\bar{a} + b)(\bar{a}b + p_1)$$

natomiast

$$p'_2 = (ab + \bar{a}\bar{b})p_2 = p_2(a + \bar{b})(\bar{a} + b) \quad \text{kod I}$$

$$p'_2 = \bar{a}\bar{b} + (a + \bar{b})p_2 = (a + \bar{b})(\bar{a}\bar{b} + p_2) \quad \text{kod II}$$

Układ komparatora z kodem I był projektowany wyżej (rys. 3-20). Zastępując przeniesienie p_2 wiązkami, uzyskuje się bezpośrednią realizację $p_2 = y_2$ i iteracyjną realizację p_1 . Schemat układu z elementami NOR (równoważnego układowi z rys. 3-20b) jest przedstawiony na rys. 5-29c. W podobny sposób można zbudować układ z elementów NAND.



Rys. 5-29. Człony komparatorów równoległych

Przy stosowaniu drugiego kodu układy przeniesień są podobne, a człon komparatora ma postać np. jak na rys. 5-29d. W obydwu układach dalsze uproszczenie można uzyskać przez realizację wszystkich przeniesień za pomocą wiązki przewodów (tych które wchodzi na elementy generujące p'). Wprawdzie wiązki te odpowiadają sygnałom \bar{p}' , ale rozważane funkcje mają tę własność, że po zanegowaniu wszystkich zmiennych uzyskuje się negację funkcji, wobec czego wystarczy w parzystych członach układu iteracyjnego wprowadzić zanegowane wejścia, a działanie układu będzie poprawne.

Jeśli w komparatorach z rys. 5-29c,d usunie się obwody generujące p_2 (i y_2), to sygnał p_1 określa relacje: $A \geq B$, $A < B$. Zachowanie tylko obwodów y_2 w rozwiązaniu z rys. 5-29c umożliwia określenie relacji: $A = B$, $A \neq B$.

W rozpatrywanych wyżej sumatorach i komparatorach równoległych brano pod uwagę tylko rozwiązania najbardziej typowe i najprostsze. Niekiedy jest konieczne wprowadzenie dodatkowych wymagań lub ograniczeń, a wówczas układy odpowiednio zmieniają się. Najczęściej spotykanym wymaganiem jest zwiększenie szybkości pracy, szczególnie ważne w dużych przelicznikach i maszynach uniwersalnych. Stosowane są w tym celu dwie metody:

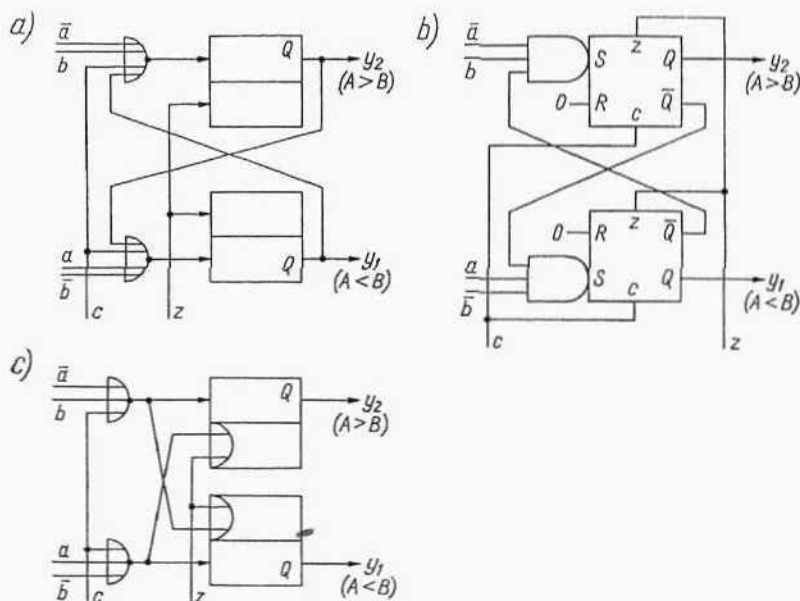
- przyspieszanie przeniesień i
- częściowe rezygnowanie z układu iteracyjnego.

Dokładne opisanie tych metod wykracza poza zakres pracy.

5.3.7. KOMPARATORY SZEREGOWE

Jeśli liczby są wprowadzane do komparatora poczynając od największego (najbardziej znaczącego) bitu, to już pierwszy takt, w którym bity nie są równe, określa relację między całymi cyframi. Wykorzystując przerzutniki do zapamiętania relacji między największymi niejednakowymi bitami, otrzymuje się układy z rys. 5-30a,b. Jediną istotną różnicą między tymi układami jest sposób taktowania impulsami c . W pierwszym rozwiązaniu jest to bramkowanie układu statycznego i zmiany sygnałów a, b dokonywane są przy $c = 1$. W drugim rozwiązaniu sygnał c jest doprowadzany do impulsowych wejść przerzutnika (taktujących), a zakres czasu, w którym nie mogą się zmieniać sygnały a, b (czyli S) zależy od realizacji przerzutnika.

Jeśli liczby są wprowadzane do komparatora od najmniejszego bitu, to wynik porównania ustala się dopiero w ostatnim takcie o różnych bitach, a każda nowa decyzja likwiduje poprzednią. Statyczny układ takiego komparatora przedstawiono na rys. 5-30c. Realizacja dynamiczna z przerzutnikami synchronizowanymi nie jest w tym przypadku oplacalna.



Rys. 5-30. Komparatory szeregowo z wprowadzaniem liczb od największego bitu (a,b) i od najmniejszego bitu (c)

W rozwiązaniach z rys. 5-30 jest $y_1 = 1$ gdy $A < B$, natomiast $y_2 = 1$ gdy $A > B$. Jeśli potrzebny jest odrębny sygnał gdy $A = B$ można go utworzyć za pomocą funkcji $y_3 = \bar{y}_1 \bar{y}_2$

5.4. REJESTRY

5.4.1. REJESTRY RÓWNOLEGŁE

Zespół przerzutników służący do przechowywania informacji w postaci liczby dwójkowej jest nazywany *rejestrem równoległym*. Typowy układ takiego rejestru przedstawiono na rys. 5-31a. Sygnał zerujący