W p. 7.3 będzie przedstawiona metoda postępowania, ułatwiająca określenie funkcji bloku sterowania.

Synteza arytmometru jest trudniejsza, gdyż w bloku tym są przetwarzane słowa, często będące liczbami o konkretnym znaczeniu fizycznym, a działania na słowach nie mają tak sformalizowanego aparatu matematycznego, jakim jest algebra Boole'a dla działań na sygnałach binarnych. Przy złożonych problemach przetwarzania, bywają pomocne metody numeryczne, ale w specjalizowanych układach automatyki problem najczęściej nie polega na tym, jak rozbić zadany proces przetwarzania na operacje arytmetyczne, lecz na tym — jak, w jakiej kolejności i na jakich sygnałach te operacje wykonać, aby realizacja była prosta. W przeciwieństwie do uniwersalnych maszyn cyfrowych w urządzeniach specjalizowanych bardzo często szybkość przetwarzania nie jest ważna, a dążenie do prostoty rozwiązań wymaga rozważenia różnych wariantów, w tym również tak powolnego jak przetwarzanie sygnałów unitarnych.

Ponieważ nie istnieje algorytmiczna metoda syntezy układu optymalnie przetwarzającego słowa, najprostszą drogą do nabrania doświadczenia jest zapoznanie się z różnymi wariantami przetwarzania. W opisanych niżej układach pominięto rozwiązania złożone, które są dokładnie przedstawione w bogatej literaturze dotyczącej maszyn cyfrowych.

7.2. UKŁADY ARYTMOMETRU

7.2.1. PRZETWARZANIE W INNĄ POSTAĆ

Na etapie syntezy blokowej nie rozważa się zazwyczaj takich szczegółów jak kod stosowanych sygnałów, czy ich postać fizyczna, gdyż te cechy rzadko wpływają na strukturę blokową urządzenia, ale dwie cechy sygnałów muszą być wzięte pod uwagę:

— sposób zapisu liczb (jedynkowy, dwójkowy),

— sposób przekazywania (szeregowy, równoległy),

Połączenia tych wariantów daje 4 możliwości, ale sygnały równoległe jedynkowe nie są stosowane, pozostają więc trzy typy sygnałów:

X^{1S} — jedynkowy szeregowy,

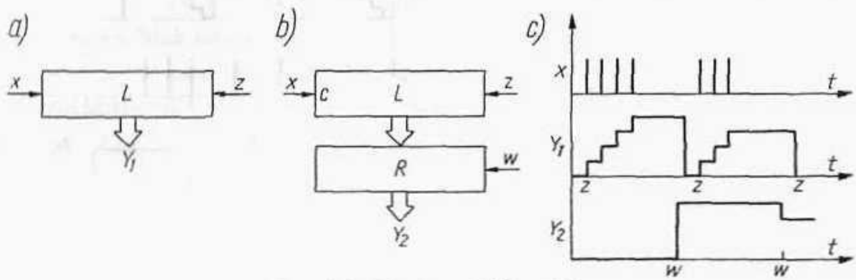
X^{2S} — dwójkowy szeregowy,

X^{2R} — dwójkowy równoległy.

Wykonywanie działań na sygnałach tych trzech rodzajów wymaga stosowania urządzeń o różnej złożoności, a czasy realizacji działań są również bardzo różne i dlatego rozwiązanie optymalne urządzenia wymaga często zamiany sygnałów jednego rodzaju na inne.

Dla uproszczenia zapisów przez X, Y, Z, ... będą oznaczane zarówno ciągi sygnałów, jak i liczby im odpowiadające. Dodawanie, odejmowanie, mnożenie i dzielenie liczb będzie zapisywane w postaci: X_1+X_2, X_1-X_2, $X_1\times X_2$, X_1/X_2. Dla liczbowego ujęcia sygnałów unitarnych zastosowano zapis $L_T(x)$, oznaczający liczbę impulsów na wejściu (wyjściu) x w okresie T. Zmiana postaci sygnałów (liczb) będzie oznaczana przez: $X\to Y$.

1. Zamiana $X^{1S}\to Y^{2R}$ może być przeprowadzona za pomocą licznika (rys. 7-2a) pracującego w kodzie przewidzianym dla Y. Przy takim rozwiązaniu odczytywanie odpowiedniej liczby wyjściowej jest możliwe dopiero po zakończeniu serii impulsów wejściowych (rys. 7-2c). Gdy stan Y musi być dostępny w sposób ciągły, należy zastosować rejestr (rys. 7-2b). Oczywiście jego wyjście podaje zawsze wartość z poprzedniego cyklu zliczania. Na rys. 7-2c zaznaczono chwile czasowe, w których powinny pojawiać się impulsy sterujące z i w. Działanie układu opisuje zależność: $Y=L_T(x)$, gdzie T jest okresem zliczania jednej serii wejściowej.



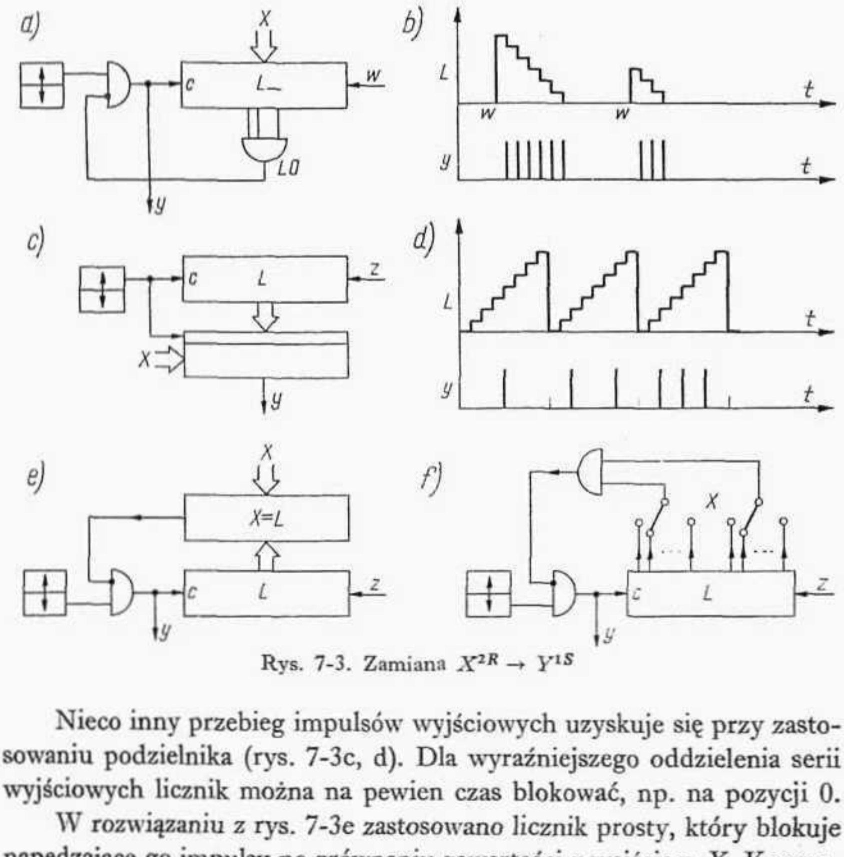
Rys. 7-2. Zamiana $X^{1S}\to Y^2$

Zapis jedynkowy jest często stosowany nie do przenoszenia całych liczb (opisujących np. mierzoną wielkość), lecz przyrostów, tzn. różnic pomiędzy wartością aktualną i poprzednią. Zmniejsza to długość potrzebnych serii impulsów, ale przy wartościach rosnących i malejących w czasie jest potrzebny dodatkowy sygnał określający znak przyrostu. Jeśli sygnał X^{1S} jest sygnałem przyrostowym (ΔX^{1S}), to jego zamiana na Y^{2R} wymaga zastąpienia licznika (rys. 7-2) licznikiem rewersyjnym i zlikwidowania sygnału zerującego.

2. Zamiana $X^{2R} \rightarrow Y^{1S}$ może być zrealizowana jednym z układów z rys. 7-3. W wersji a) liczba X jest wpisana do licznika odwrotnego (odejmującego), co uruchamia układ impulsowania, zmniejszający stan licznika. Po osiągnięciu stanu 0 ($LO = 1$) impulsowanie jest przerywane, a liczba impulsów jest równa liczbie X (w kodzie licznika)

$$L_T(y) = X$$

Uproszczony wykres czasowy, ilustrujący zasadę działania, przedstawiony jest na rys. 7-3b (zmieniający się sygnał X pominięto).

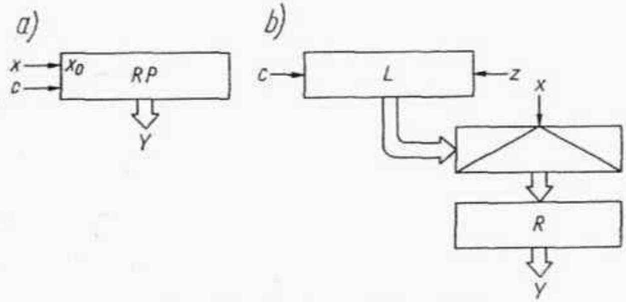


Rys. 7-3. Zamiana $X^{2R} \rightarrow Y^{1S}$

Nieco inny przebieg impulsów wyjściowych uzyskuje się przy zastosowaniu podzielnika (rys. 7-3c, d). Dla wyraźniejszego oddzielenia serii wyjściowych licznik można na pewien czas blokować, np. na pozycji 0.

W rozwiązaniu z rys. 7-3e zastosowano licznik prosty, który blokuje napędzające go impulsy po zrównaniu zawartości z wejściem X. Komparator do tego potrzebny może się bardzo uprościć, gdy X jest nastawiane przełącznikami lub przyciskami (rys. 7-3f). Oczywiście licznik musi pracować w odpowiednim kodzie (np. kod „1 z 10", zapis dwójkowo-dziesiętny).

3. Zamiana $X^{2S} \to Y^{2R}$ wymaga zastosowania rejestru przesuwającego (z wejściem na x_0) — rys. 7-4a. lub komutatora — rys. 7-4b. Druga wersja, chociaż bardziej złożona, może znaleźć zastosowanie, gdy np. licznik i rejestr są potrzebne z innych jeszcze powodów. Gdy kody



Rys. 7-4. Zamiana $X^{2S} \to Y^{2R}$

sygnałów wejściowych i wyjściowych mają być różne, należy wprowadzić dodatkowy blok konwersji. Zwykle jest to konwerter równoległy od strony wyjścia gdyż konwersja szeregowa (po stronie wejścia) jest możliwa tylko dla nielicznych kodów.

4. Zamiana $X^{2R} \to Y^{2S}$ jest bardzo podobna do poprzedniej (rys. 7-5). W rozwiązaniu b) sygnał X musi trwać niezmiennie przez cały czas przetwarzania.

5. Zamiana $X^{1S} \to Y^{2S}$ może być realizowana przez zamianę pośrednią

$$X^{1S} \to V^{2R} \to Y^{2S}$$

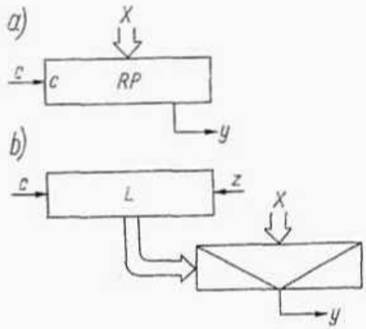
za pomocą metod opisanych wyżej.

6. Zamiana $X^{2S} \to Y^{1S}$, podobnie jak poprzednia, odbywa się za pomocą V^{2R}

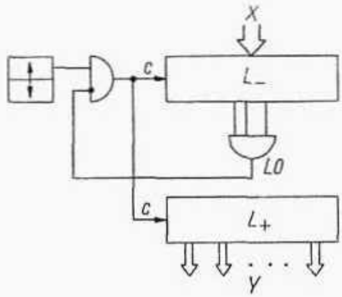
$$X^{2S} \to V^{2R} \to Y^{1S}$$

7. Opisane przejścia między sygnałami wyczerpują już możliwości przedstawione na wstępie, ale warto jeszcze zwrócić uwagę na sposób

tworzenia zapisu dwójkowo-dziesiątkowego. Konwertery kombinacyjne kodu, zmieniające zapis dwójkowy na dwójkowo-dziesiętny, są bardzo skomplikowane i dlatego, jeśli czas pozwala, chętnie stosuje się rozwiązanie z rys. 7-6 (dla zamiany $X^{2R} \rightarrow X^{2-10R}$), w którym wykorzystuje się pośrednictwo sygnałów jedynkowych. Napełnianie drugiego licznika w trakcie zerowania pierwszego może służyć do zamiany dowolnych



Rys. 7-5. Zamiana $X^{2R} \rightarrow Y^{2S}$



Rys. 7-6. Zamiana $X^{2R} \rightarrow Y^{2-10R}$

kodów, także zapisu dwójkowo-dziesiętnego w dwójkowy. Inne metody zamiany kodu dwójkowego naturalnego na dwójkowo-dziesiętny (i przeciwnie) wykorzystują rejestry przesuwające (rys. 5-36).

8. Jeśli przetwarzaniu podlegają liczby ze znakiem, to w przypadku zapisów dwójkowych zamiany dokonuje się zazwyczaj na liczbach równoległych, w myśl zasad opisywanych wyżej. W przypadku sygnałów typu X^{1S} znak jest przekazywany odrębnym sygnałem, a szczególnie łatwa jest zamiana X^{1S} na X^{2R} w zapisie dopełniającym; w układzie z rys. 7-2a występuje wówczas licznik rewersyjny.

7.2.2. KONTROLA SŁOWA

Kontrola słowa, przeprowadzona w arytmometrze, może dotyczyć różnych cech tego słowa. Najczęściej jest to:

a) kontrola wartości liczby,

b) kontrola zmiany liczby,

c) kontrola kodu (postaci) słowa.

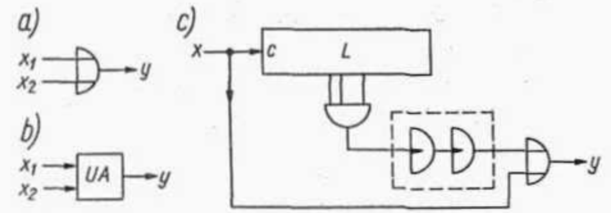
Kontrola wartości polega na porównaniu słowa, reprezentującego liczbę, z jednym lub dwoma słowami (liczbami), określającymi ograniczenia. Kontrola zmiany polega na porównaniu aktualnej wartości jakiegoś parametru z wartością poprzednią; różnica podlega często kontroli wartości, gdy jest zadany maksymalny dopuszczalny przyrost. Wszystkie te czynności sprowadzają się do działań arytmetycznych porównania i odejmowania, opisanych niżej.

Kontrola kodu lub postaci słowa polega na badaniu wartości poszczególnych bitów lub relacji między tymi wartościami, a więc sprowadza się do działań logicznych. Syntezę odpowiedniego układu wykonuje się metodami z rozdz. 3 i 4. Warto pamiętać, że w wielu przypadkach kontrolować można słowa zarówno w postaci równoległej jak i szeregowej, więc należy wybrać wersję prostszą. Na przykład — kontrola parzystości słowa 8-bitowego w wersji równoległej wymaga zastosowania piramidy z siedmiu funktorów równoważności, a w wersji szeregowej — jednego przerzutnika typu *t*.

7.2.3. OPERACJE ARYTMETYCZNE

A. Dodawanie

1. $X_1^{1S}+X_2^{1S}$. Jeśli wynik ma mieć postać Y^{1S}, to najprostszym układem dodającym jest element sumy logicznej (rys. 7-7a). Gdy impulsy wejściowe mogą na siebie nachodzić sumę trzeba zastąpić układem anty-



Rys. 7-7. Dodawanie: a,b) $X_1^{1S}+X_2^{1S} = Y^{1S}$; c)$X^{1S}+k = Y^{1S}$

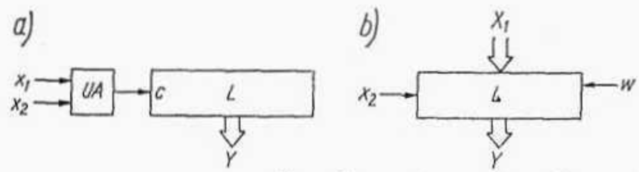
koincydencyjnym (rys. 7-7b), separującym te impulsy. Przypadkiem szczególnym dodawania rozważanego typu sygnałów jest układ dodający stałą liczbę *k* impulsów do każdych *n* impulsów wejściowych, co na ogół można sprowadzić do dodawania 1 impulsu do każdych *n*/*k* impulsów wejściowych. Odpowiedni układ jest przedstawiony na rys. 7-7c. Licznik o okresie *n*/*k* powoduje dodanie do przebiegu wejściowego jednego impulsu na okres. Generatorem impulsu jest tu zespół dwóch elementów impulsowych (jeden — dla uzyskania przerwy).

Jeśli wynik dodawania ma mieć postać Y^{2R}, do układów z rys. 7-7a, b należy dodać prosty przetwornik $X^{1S} \to Y^{2R}$, czyli licznik (rys. 7-2a). Otrzymuje się

$$Y = L_T(x_1) + L_T(x_2)$$

2. $X_1^{2R} + X_2^{1S}$. Jeśli do licznika wpisze się liczbę X_1^{2R}, a następnie uruchomi jego wejście taktujące (rys. 7-8b), to wynik będzie sumą

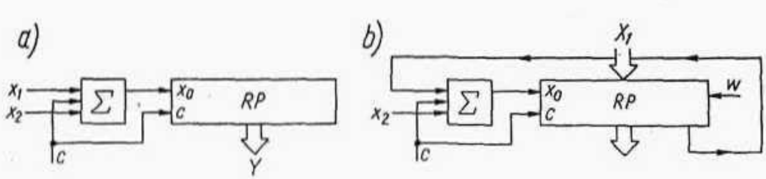
$$Y = X_1 + L_T(x_2)$$



Rys. 7-8. Dodawanie: a) $X_1^{1S} + X_2^{1S} = Y^{2R}$; b) $X_1^{2R} + X_2^{1S} = Y^{2R}$

3. $X_1^{2R} + X_2^{2R}$. Tego typu działania wykonuje się za pomocą sumatora równoległego (jeśli obie liczby są dane równocześnie) lub akumulatora (jeśli liczby występują kolejno). Jeśli czas pozwala, można jeden ze składników zamienić na postać X^{1S} (rys. 7-3), a następnie wykorzystać prosty układ z rys. 7-8b.

4. $X_1^{2S} + X_2^{2S}$. Jeśli wynik ma mieć postać Y^{2S}, realizuje go sumator szeregowy. Wynik typu Y^{2R} uzyskuje się przez dodanie do sumatora



Rys. 7-9. Dodawanie: a) $X_2^{1S} + X_2^{2S} = Y^{2R}$, b) $X_1^{2R} + X^{\ S} = Y^{2R}$

przetwornika $X^{2S} \to Y^{2R}$, czyli rejestru przesuwającego (rys. 7-9a). Rejestr taki jest również przydatny, gdy liczby wejściowe są podawane kolejno; w pierwszym cyklu wprowadza się X_1^{2S} do rejestru, a następnie —

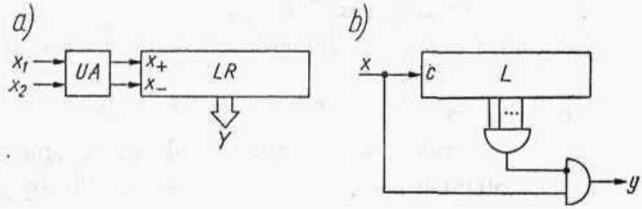
w drugim cyklu — zawartość rejestru wprowadza się na wejście sumatora równocześnie z X_2^{2S}, otrzymując w rejestrze sumę (układ jak na rys. 7-9b, ale bez X_1).

5. $X_1^{2R}+X_2^{2S}$. W układzie z rys. 7-9b wpisuje się równolegle X_1, a następnie dodaje zawartość rejestru szeregowo z X_2. Możliwa jest też druga wersja: zamiana $X_2^{2S} \to X_2^{2R}$ w rejestrze przesuwającym i dodanie zawartości do X_1^{2R} w sumatorze równoległym, ale ta realizacja jest droższa.

W układach cyfrowych całkowanie realizuje się jako wielokrotne dodawanie.

B. Odejmowanie

1. $X_1^{1S}-X_2^{1S}$. Podstawowy układ przedstawiono na rys. 7-10a. Układ antykoincydencyjny (UA) zapobiega nakładaniu się impulsów (w tym wypadku impulsy nakładające się można pominąć). Jeśli serie nie pokrywają się, UA można usunąć, a jeśli są przesunięte w czasie można je wprowadzać na jedno wejście c licznika (rys. 5-52b), odpowiednio sterując sygnałami b_+ i b_-. W rezultacie otrzymuje się



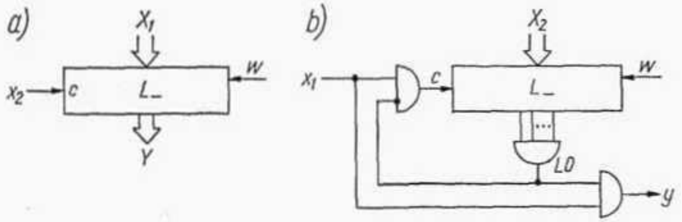
Rys. 7-10. Odejmowanie: a) $X_1^{1S}-X_2^{1S} = Y^{2R}$; b) $X^{1S}-k = Y^{1S}$

$$Y = L_T(x_1)-L_T(x_2)$$

Gdy odjemnik jest większy od odjemnej, wynik jest przedstawiony w zapisie dopełniającym.

Odejmowanie k impulsów od każdych n impulsów wejściowych (czyli jednego od każdych n/k) może być zrealizowane podobnie jak przy dodawaniu, za pomocą licznika o okresie n/k (rys. 7-10b), który na jednej ze swoich pozycji blokuje bramkę wyjściową, wycinając jeden impuls.

2. $X_1^{2R}-X_2^{1S}$. Działanie to można uzyskać w liczniku odejmującym (rys. 7-11a) do którego wstępnie wpisano X_1. Wynik ujemny będzie przedstawiony w zapisie dopełniającym. Można też zrobić odwrotnie, do licznika dodającego wpisać X_1 w zapisie dopełniającym, a wówczas wynik ujemny będzie w zapisie modułowym, natomiast dodatni będzie miał postać X_1 czyli dopełniającą. Jest to właściwie realizacja działania $X_2^{1S}-X_1^{2R}$.



Rys. 7-11. Odejmowanie: a) $X_1^{2R}-X_2^{1S} = Y^{2R}$; b) $X_1^{1S}-X_2^{2R} = Y^{1S}$

Jeśli $L_T(x_1) > X_2$, a wynik odejmowania $X_1^{1S}-X_2^{2R}$ ma mieć postać Y^{1S}, można zastosować układ z rys. 7-11b, w którym pierwsze impulsy w liczbie X_2 napędzają licznik, a dopiero pozostałe przedostają się na wyjście.

3. Pozostałe działania typu $X_1^{2R}-X_2^{2R}$, $X_1^{2R}-X_2^{2S}$, $X_1^{2S}-X_2^{2S}$ itp. wykonuje się tak jak dodawanie, ale zastępując bloki sumowania blokami odejmowania, albo zastępując odejmowanie dodawaniem liczby ze zmienionym znakiem.

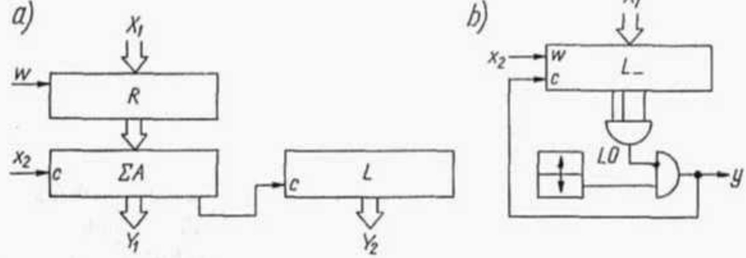
W układach cyfrowych różniczkowanie jest sprowadzane do odejmowania.

C. Mnożenie

1. $X_1^{2R}\times X_2^{1S}$. Najłatwiej można to działanie wykonać przez X_2-krotne dodawanie mnożnej X_1, np. w układzie jak na rys. 7-12. Każdy impuls X_2 powoduje dodanie do zawartości akumulatora liczby X_1 z rejestru wejściowego. Aby uniknąć budowania sumatora akumulującego o liczbie bitów wyniku mnożenia, zastosowano tu akumulator o długości jak X_1, a każde jego przepełnienie jest liczone za pomocą dodatkowego licznika. Wynik mnożenia (o postaci Y^{2R}) składa się z dwóch części: mniej znaczącej Y_1 z akumulatora i bardziej znaczącej Y_2 z licznika.

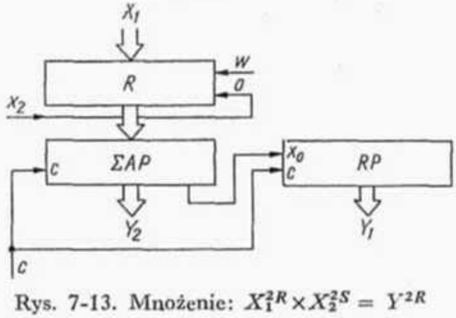
Jeśli impulsy X^{1S} mogą być podawane w dowolnym czasie, miejsce sumatora akumulującego z rozwiązania na rys. 7-12a może zająć układ z rys. 7-12b. Wpisywanie X_1 następuje tu w czasie, gdy licznik osiąga stan 0 ($L0 = 1$).

2. $X_1^{2R} \times X_2^{2S}$. Mnożenie dwóch liczb binarnych, wykonywane metodą stosowaną zwykle przy mnożeniu „na papierze" wielocyfrowych liczb



Rys. 7-12. Mnożenie: a) $X_1^{2R} \times X_2^{1S} = Y^{2R}$; b) $X_1^{2R} \times X_2^{1S} = Y^{1S}$

dziesiętnych, sprowadza się do dodawania mnożnej, pomnożonej przez ostatni bit mnożnika, do przesuniętej o jeden bit w lewo mnożnej pomnożonej przez drugi (z prawej) bit mnożnika, dodania wyniku do przesuniętej o jeszcze jeden bit mnożnej, pomnożonej przez trzeci bit mnożnika itd. Mnożenie sprowadza się tu więc do dodawania i przesuwania liczb, co można zrealizować za pomocą sumatora równoległego i rejestru przesuwającego, albo za pomocą sumatora akumulującego z przesuwaniem.



Rys. 7-13. Mnożenie: $X_1^{2R} \times X_2^{2S} = Y^{2R}$

Zamiast dodawać do wyniku dodawania pośredniego przesuniętą w lewo mnożną, można mnożną dodawać do przesuniętego w prawo wyniku dodawań pośrednich. W odpowiednim układzie (rys. 7-13) bit X_2 o wartości 1 powoduje dodanie X_1 do zawartości akumulatora, która następnie zostaje przesunięta w prawo (impulsy c są zsynchronizowane z taktami X_2). Jeśli kolejny bit X_2 jest zerem, zawartość akumulatora nie ulega zmianie, zostaje tylko przesunięta. Ze względów oszczędnościowych przesuwający sumator akumulujący jest krótki, a mniej znaczące bity wyniku są umieszczane w rejestrze przesuwającym.

3. $X_1^{2R} \times X_2^{2R}$. Do wykonania tego działania można wykorzystać układ z p. 2, wprowadzając X_2 równolegle do rejestru przesuwającego (rys. 7-14a) i kolejnymi sygnałami z tego rejestru sterując wejście akumulatora. Mnożenie będzie zakończone, gdy wszystkie bity X_2 wyjdą z rejestru, ustępując miejsca wynikowi.

Układ mnożący z wykorzystaniem podzielnika jest pokazany na rys. 7-14b. Po wpisaniu liczby X_1 licznik L_+ wykonuje X_1 pełnych cykli, a w czasie każdego cyklu na wyjściu podzielnika uzyskuje się X_2 impulsów, wobec czego — po wyzerowaniu licznika L_- otrzymuje się

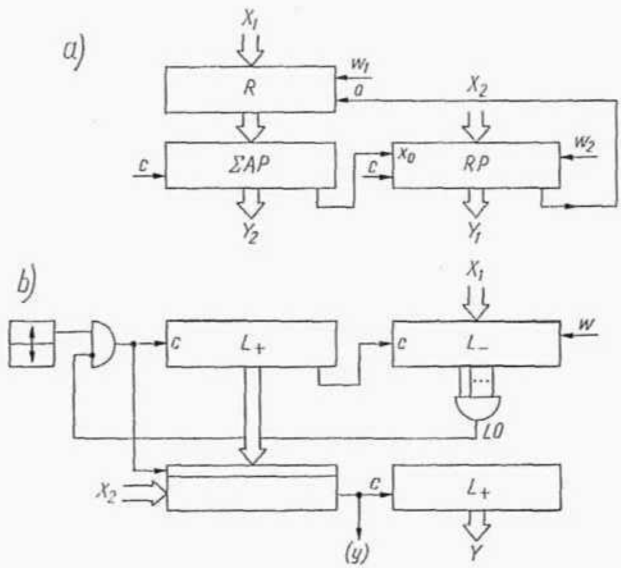
$$L_T(y) = X_1 \dot{\times} X_2$$

Jeśli potrzebne jest wyjście typu Y^{2R}, impulsy y są zliczane w dodatkowym liczniku. Jeśli początkowym stanem tego licznika było X_3, to $Y = (X_1 \times X_2) + X_3$.

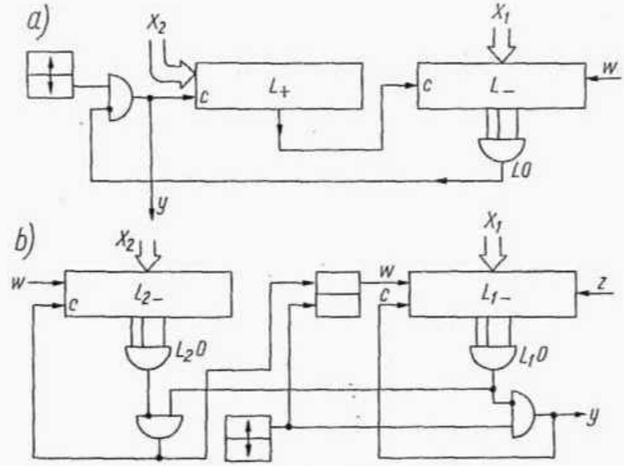
Układ mnożący z wykorzystaniem dzielnika częstotliwości przedstawiono na rys. 7-15a. Dzielnik częstotliwości zastąpił tu licznik i podzielnik z poprzedniego rozwiązania, tworząc układ szybszy, gdyż licznik o okresie X_2 szybciej wyzeruje L_- niż poprzedni licznik L_+ o okresie max X_2. Jeśli wynik ma być w postaci Y^{2R}, do układu trzeba dodać licznik.

Jeszcze inny układ mnożący można utworzyć przez rozbudowę rozwiązania z rys. 7-12b, w sposób pokazany na rys. 7-15b. Przerzutnik zapewnia właściwy czas trwania impulsów X_1.

Działanie opisanych wyżej układów mnożących z wyjściowym sygnałem o postaci Y^{1S} można przyspieszyć, jednocześnie zmniejszając ilość potrzebnego sprzętu, przez podział jednego z czynników na grupy o stałej liczbie bitów. Zamiast mnożyć dwie liczby X_1 i X_2 — można mnożyć X_1 przez grupy X_2, a następnie dodawać wyniki. Przykład układu dla dwudekadowej liczby X_2 jest przedstawiony na rys. 7-16. Układ mnożący jest przystosowany do działania na poszczególnych dekadach liczby X_2. Przełączanie kolektora następuje po zakończeniu mnożenia. Wyjście y_p



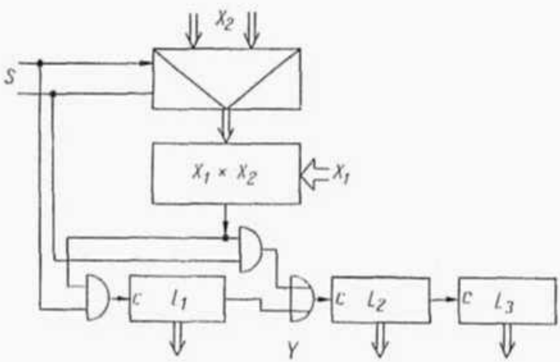
Rys. 7-14. Mnożenie: a) $X_1^{2R} \times X_2^{2R} = Y^{2R}$; b) $X_1^{2R} \times X_2^{2R} = Y^{2R}$ lub $X_1^{2R} \times X_2^{2R} = Y^{1S}$



Rys. 7-15. Mnożenie: $X_1^{2R} \times X_2^{2R} = Y^{1S}$

licznika L_1 ma wartość 1 tylko na jednej pozycji licznika i przy $c = 1$, przez co nie blokuje taktowania licznika L_2. Układ można oczywiście rozbudować dla większej liczby dekad.

W tego typu rozwiązaniach grupę może stanowić dowolna liczba bitów, w szczególności jeden bit. Upraszcza to bardzo układ mnożenia, ale rozbudowuje układy komutacji.



Rys. 7-16. Mnożenie z podziałem na grupy

Jeśli mnożone liczby nie są duże (np. w ramach jednej dekady), można układ mnożący wykonać w postaci układu kombinacyjnego (matrycy). Szczególnie proste jest mnożenie liczby X przez stałą $A = 2^n$, sprowadzające się do traktowania każdego i-tego bitu liczby X (x_i) jako bitu $i+n$ wyniku (y_{i+n}). Przy przekształceniach szeregowych mnożenie takie można zrealizować za pomocą rejestru przesuwającego.

Do mnożenia liczb w postaci X^{1S} przez stałe liczby ułamkowe można wykorzystać układy z rys. 7-7c i 7-10b, które dodają (odejmują) k impulsów do (od) n impulsów wejściowych, czyli

$$L(y) = L(x) + \frac{k}{n} L(x) = \frac{n+k}{n} L(x)$$

oraz

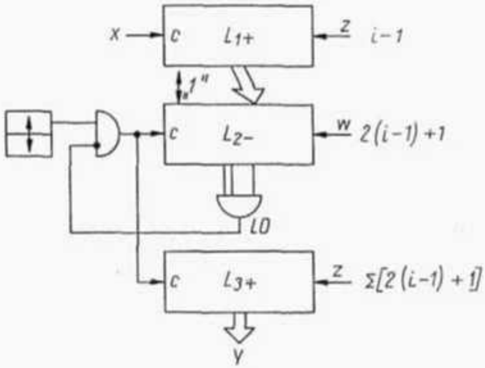
$$L(y) = L(x) - \frac{k}{n} L(x) = \frac{n-k}{n} L(x)$$

Wartość n może być zmieniana, jeśli licznik zastąpi się dzielnikiem częstotliwości.

Szczególny przypadek mnożenia — potęgowanie, realizuje się zazwyczaj korzystając z zależności

$$n^2 = \sum_{i=1}^{n}(2i-1) = \sum_{i=1}^{n}[2(i-1)+1]$$

W układzie z rys. 7-17 sygnał $z = 1$ ustawia w L_1 same jedynki, więc zawartość licznika wynosi $L(x)-1$. Stan licznika L_1 po każdym



Rys. 7-17. Potęgowanie: $X^{1S} \times X^{1S} = Y^{2R}$

impulsie x jest przepisywany do licznika L_2, z przesunięciem o jeden bit i stałą jedynką na pozycji mniej znaczącej. Wyzerowanie tego licznika za pośrednictwem wejścia c wymaga więc dostarczenia impulsów w liczbie $2\,[L(x)-1]+1$. Te same impulsy są wprowadzane do licznika L_3, który nie jest zerowany przez cały czas zliczania, a zatem gromadzi sumę stanowiącą kwadrat liczby $L(x)$.

Przez niewielką modyfikację układu z rys. 7-17 można uzyskać układ do obliczania pierwiastka kwadratowego. Trzeba w tym celu zadaną liczbę wpisać do odejmującego licznika L_3, a do licznika L_1 wprowadzać (wejściem c) impulsy z generatora. Gdy stan licznika L_3 osiągnie 0 — w liczniku L_1 będzie liczba o 1 mniejsza od wyniku pierwiastkowania.

D. Dzielenie

1. X_1^{2R}/X_2^{2R}. Algorytmy bezpośredniego dzielenia liczb całkowitych są dosyć złożone i dlatego często korzysta się z możliwości zastąpienia dzielenia mnożeniem. Zamiast obliczać Y takie, aby było $X_1/X_2 = Y$, szuka się takiego Y, aby otrzymać $X_1 = X_2 \times Y$. Przystosowany do tego celu układ z rys. 7-14 jest przedstawiony na rys. 7-18a. Generator impulsów napędza tu licznik L_+ przez tyle cykli, aby ich liczba pomnożona przez X_2 dała X_1. Liczba cykli jest więc szukanym ilorazem, który może być odczytany w postaci Y^{2R} ze specjalnego licznika. Liczba impulsów, pozostała z dzielenia jako reszta, jest zapisana w pierwszym liczniku L_+, ale nie w postaci jawnej.

Przystosowany do dzielenia układ z rys. 7-15a przedstawiono na rys. 7-18b. Do wyzerowania licznika L_- — licznik L_+ wykona tyle cykli, ile razy X_2 mieści się w X_1. Reszta z dzielenia jest umieszczona w liczniku L_+, w postaci jawnej.

2. X^{1S}/X^{2R}. Najprostszą realizacją tego działania jest dzielnik częstotliwości (rys. 7-19a), ewentualnie z dodatkowym licznikiem, gdy wynik ma mieć postać Y^{2R}. Reszta z dzielenia jest zapisana w liczniku dzielnika. Łatwo zauważyć, że dodając do tego rozwiązania układ zamiany $X_1^{2R} \to X_1^{1S}$, otrzymuje się układ z rys. 7-18b.

Inna prosta realizacja dzielenia jest przedstawiona na rys. 7-19b. Gdy okres licznika L wynosi n, otrzymuje się

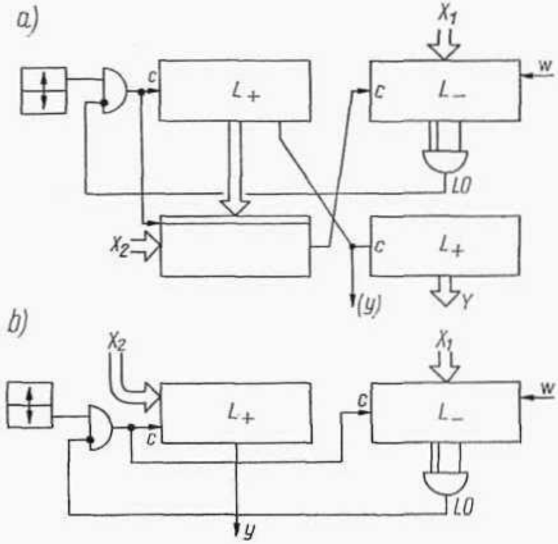
$$L_T(y) = L_T(x_1)/\frac{n}{X_2}$$

Dzielenie liczby X przez stałą $A = 2^n$ jest równie proste jak mnożenie. Do dzielenia przez stałą liczbę ułamkową można wykorzystać układy z rys. 7-7c i 7-10b, gdyż uzyskuje się w nich:

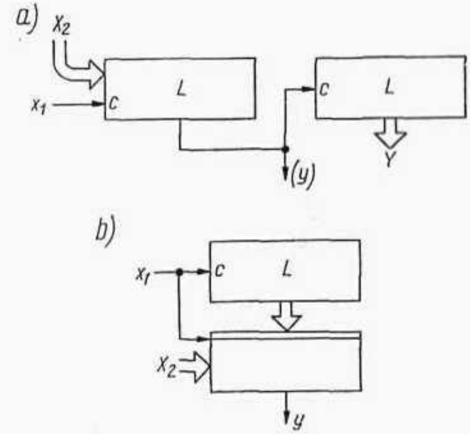
$$L_T(y) = L_T(x)/\frac{n}{n+k}$$

oraz

$$L_T(y) = L_T(x)/\frac{n}{n-k}$$



Rys. 7-18. Dzielenie: $X_1^{2R}/X_2^{2R} = Y^{2R}$ albo Y^{1S}



Rys. 7-19. Dzielenie: $X_1^{1S}/X_2^{2R} = Y^{2R}$ albo Y^{1S}

Dzielenie typu $X^{1S}/B = Y^{1S}$ można łatwo zrealizować za pomocą licznika o stałej pojemności B.

E. Porównywanie

1. $X_1^{1S} \geqslant X_2^{1S}$. Do porównywania można wykorzystać układy odejmowania, z odpowiednio uproszczonymi sygnałami wyjściowymi. Układ z rys. 7-20a (por. rys. 7-10a) generuje dwa sygnały wyjściowe:

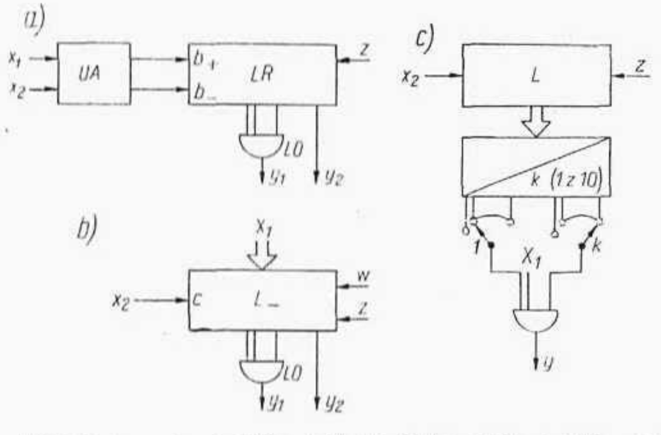
$$y_1 = 1 \quad \text{gdy} \quad L_T(x_1) = L_T(x_2)$$

oraz

$$y_2 = 1 \quad \text{gdy} \quad L_T(x_1) < L_T(x_2)$$

a więc stan wyjść (y_1, y_2) przybiera następujące wartości:

$$\begin{array}{ll} (00) \text{— gdy} & X_1 > X_2 \\ (10) \text{— gdy} & X_1 = X_2 \\ (01) \text{— gdy} & X_1 < X_2 \end{array}$$



Rys. 7-20. Porównanie: a) $X_1^{1S} \geqslant X_2^{1S}$; b) $X_1^{2R} \geqslant X_2^{1S}$; c) $X_1^{2R} = X_2^{1S}$

Zakłada się tu, że przy n stopniach licznika różnica między X_1 a X_2 jest mniejsza niż 2^{n-1}, tzn. ostatni przerzutnik licznika może określać znak różnicy.

2. $X_1^{1S} \geqslant X_2^{2R}$. Układ porównujący tego typu działa na podobnej zasadzie jak poprzedni i stan jego wyjść znaczy to samo (rys. 7-20b), ale fakt że jedna z liczb jest podana w postaci równoległej umożliwia zastąpienie licznika rewersyjnego licznikiem odejmującym,

Gdy liczba X_1 jest ustawiana za pomocą przełączników lub przycisków, równość liczb

$$X_1 = L_T(x_2)$$

może być sygnalizowana w układzie z rys. 7-20c (przystosowanym do kodu dwójkowo-dziesiętnego). Jest to wprawdzie układ bardziej złożony niż poprzedni, ale za to może być wykorzystany do wyznaczenia momentów, w których liczba X_2 jest równa kilku innym liczbom typu X_1, czyli

$$y_i = 1 \quad \text{gdy} \quad X_{1i} = L_T(x_2) \qquad i = 1, 2, \ldots$$

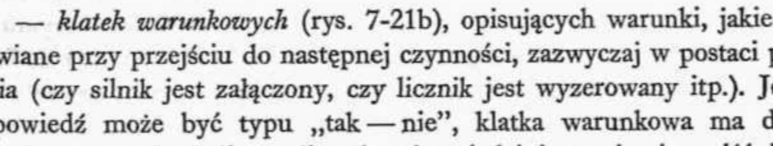
3. $X_1^{2R} \geqslant X_2^{2R}$, $X_1^{2S} \geqslant X_2^{2S}$. Do porównania liczb w tych przypadkach służą komparatory.

7.3. STEROWANIE

7.3.1. SIEĆ DZIAŁAŃ

Pracę złożonego układu cyfrowego wygodnie jest opisywać za pomocą tzw. *sieci działań* (*schematów czynności*), budowanych z dwóch podstawowych elementów:

— *klatek operacyjnych* (rys. 7-21a), opisujących jedną lub kilka równocześnie lub w dowolnej kolejności wykonywanych czynności, zwykle w formie polecenia (załącz, wykonaj, drukuj itp.),



Rys. 7-21. Elementy sieci działań: a) klatka operacyjna; b) klatka warunkowa

— *klatek warunkowych* (rys. 7-21b), opisujących warunki, jakie są stawiane przy przejściu do następnej czynności, zazwyczaj w postaci pytania (czy silnik jest załączony, czy licznik jest wyzerowany itp.). Jeśli odpowiedź może być typu „tak — nie", klatka warunkowa ma dwa wyjścia (T — N), jeśli możliwych odpowiedzi jest więcej, wyjść jest również więcej (np. określenie ruchu: w górę — w dół — stop).