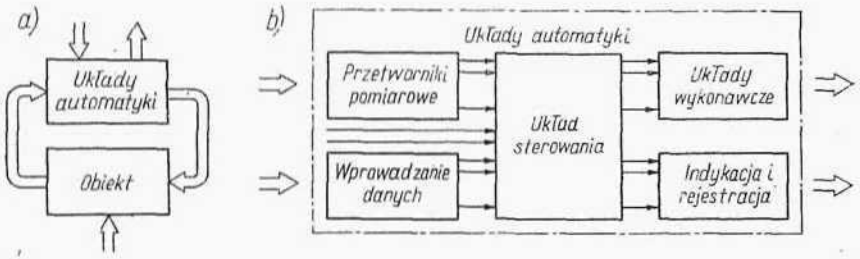
1. WIADOMOŚCI PODSTAWOWE

1.1. ZADANIA UKŁADÓW CYFROWYCH AUTOMATYKI

Zasady współpracy układów automatyki z obiektem automatyzowanym w najbardziej ogólny sposób można przedstawić schematem obiegu informacji z rys. 1-1a. Układy automatyki, na podstawie informacji o stanie obiektu (sygnałów pomiarowych i alarmowych) oraz informacji wprowadzanej z zewnątrz przez operatora, lub specjalne urządzenia



Rys. 1-1. Współpraca układu automatyki z obiektem (a) i podstawowe bloki układu automatyki (b)

(wartości zadane, program działania, sygnały rozruchowe) wytwarzają sygnały sterujące obiekt oraz sygnały wysyłane do urządzeń zewnętrznych — pokazujących i rejestrujących stan parametrów obiektu. Tak więc układy automatyki są — ogólnie biorąc — przetwornikami informacji pomiarowych i programujących w informacje wykonawcze i kontrolne (zobrazowujące). Nośnikami informacji są sygnały, których fizyczne właściwości mogą być bardzo różne.

W ogólnym, blokowym schemacie układów automatyki (rys. 1-1b) można wyodrębnić cztery grupy urządzeń, służących do obsługi czterech wymienionych rodzajów informacji, oraz centralny układ sterowania,

realizujący funkcje przetwarzania. Urządzenia czterech grup pomocniczych nie przetwarzają informacji, lecz zmieniają charakter sygnałów tak, by umożliwić współpracę obiektu i operatora z centralnym układem sterowania.

Przedstawiony na rys. 1-1 model jest bardzo ogólny — można w ten sposób opisać zarówno pracę prostego regulatora temperatury w piecu jak i sterowanie walcowni za pomocą uniwersalnej maszyny cyfrowej. Jeśli pętla obiekt-układy automatyki jest zamknięta — opisuje regulację lub sterowanie i kontrolę automatyczną (pomiary i sygnalizację). Jeśli pętla jest otwarta — występuje tylko sterowanie albo tylko kontrola. Gdy obiekt jest oddalony od stanowiska dyspozycji — układy automatyki rozbudowują się o odpowiednie układy telesterowania, telemetrii, telesygnalizacji lub teleregulacji, czyli układy telemechaniki.

Sygnały wejściowe i wyjściowe układu sterowania (rys. 1-1b), zwykle elektryczne lub pneumatyczne, mogą występować w dwóch postaciach: jako *sygnały ciągłe* (*analogowe*) lub *nieciągłe*. Sygnały mogą być nieciągłe w czasie (*próbkowane*) albo z wartością nieciągłą (*kwantowane*). Gdy występują obie te cechy — używa się nazwy *sygnały dyskretne*. Ponieważ sygnały dyskretne stanowią najważniejszy i najczęściej stosowany rodzaj sygnałów nieciągłych — często utożsamia się nazwy „dyskretny" i „nieciągły".

Sygnały ciągłe uzyskuje się np. z typowych przetworników pomiarowych temperatury, ciśnienia, z potencjometrów wartości zadanej, a trzeba je dostarczyć do ciągłego sterowania prędkości silnika, zasilania wskaźników wychyłowych lub rejestratorów itp. Sygnały nieciągłe uzyskuje się np. z łączników drogowych, przełączników sterujących, impulsowych czujników prędkości, a trzeba je dostarczyć do załączania i wyłączania silników, zasilania wskaźników cyfrowych, silników krokowych, drukarek itp. Różny charakter sygnałów wejściowych i wyjściowych wymaga stosowania w układzie sterowania różnych elementów i urządzeń — innych dla sygnałów ciągłych i innych dla sygnałów dyskretnych.

Stosowanie w procesie przetwarzania sygnałów dyskretnych ma wiele istotnych zalet:

— dokładność przetwarzania może być dowolnie duża i zależy wyłącznie od dokładności informacji wejściowych,

— znacznie większa jest odporność na zakłócenia i ogólna niezawodność urządzeń,

— informacja może być stosunkowo łatwo zapamiętana i magazynowana przez dowolnie długi czas,

— istnieje możliwość dokładnego, cyfrowego przedstawiania informacji wyjściowych za pomocą wskaźników cyfrowych i urządzeń drukujących liczby,

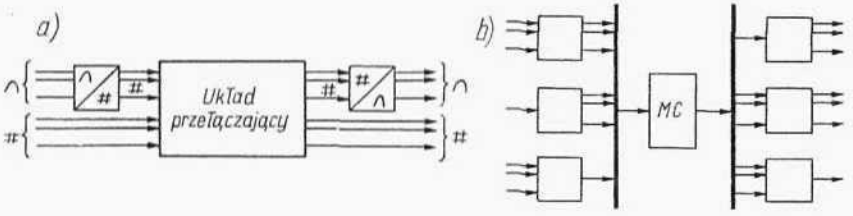
— koszt układów realizujących złożone i dokładne przetwarzanie jest względnie niski.

Wady przetwarzania sygnałów dyskretnych to:

— wyższy niż w przypadku sygnałów ciągłych koszt urządzeń przy przetwarzaniu z niewielką dokładnością,

— dłuższy czas wykonywania złożonych operacji.

Zalety urządzeń dyskretnych są tak duże, że w bardzo wielu przypadkach opłaci się zmienić charakter sygnałów wejściowych i wyjściowych (z ciągłych na dyskretne lub odwrotnie) dla zachowania jednoli-



Rys. 1-2. Przetwarzanie sygnałów dla układu przełączającego (a) i uniwersalna maszyna cyfrowa jako układ przełączający (b)

tego — dyskretnego charakteru sygnałów podlegających złożonym operacjom przetwarzania. W bloku układów sterowania pojawiają się wówczas dodatkowe człony (rys. 1-2a): przetwornik analogowo-dyskretny na wejściu i dyskretno-analogowy na wyjściu, a układ centralny staje się *układem przełączającym*, gdyż tak zwykle są nazywane przetworniki informacji dyskretnych. Budowanie układu sterującego w ten właśnie sposób jest szczególnie uzasadnione wówczas, gdy przetwarzanie jest złożone lub bardzo dokładne, bądź też, gdy liczba sygnałów dyskretnych jest większa od liczby sygnałów ciągłych. Zresztą próg opłacalności sto-

sowania układów dyskretnych z każdym rokiem obniża się, w miarę rozwoju technologii elementów.

Niniejsza książka zajmuje się tymi układami automatyki, w których podstawowe zadanie przetwarzania informacji jest realizowane za pomocą układu przełączającego. Zakres działania takich układów jest tak szeroki jak ogólne są schematy z rys. 1-1 i 1-2a. Mogą to być proste układy automatyki napędu elektrycznego, bardziej złożone układy sterowania obrabiarek, aż po skomplikowane przeliczniki i specjalizowane maszyny cyfrowe. W przyjętym modelu układem przełączającym lub jego częścią może też być uniwersalna maszyna cyfrowa, jednakże taki przypadek nie będzie dalej rozważany. Maszyny cyfrowe prawie nigdy nie współpracują z układami automatyki bez pośrednictwa pomocniczych bloków (rys. 1-2b) przygotowujących informację, wstępnie ją przetwarzających lub przystosowujących do urządzeń automatyki; synteza takich bloków wchodzi już w zakres książki. Tak więc dalej będzie przedstawiona synteza tych wszystkich dyskretnych układów automatyki, które nie są uniwersalną maszyną cyfrową.

Synteza układów automatyki, podobnie jak i innych złożonych układów, sprowadza się do szczegółowej odpowiedzi na dwa pytania:

— **jaki** algorytm trzeba zrealizować dla optymalnego działania obiektu?

— **jak** zrealizować ten algorytm w sposób optymalny?

W niektórych przypadkach znacznie trudniejszy jest problem pierwszy, w innych stopień trudności jest zbliżony, a w jeszcze innych algorytm jest trywialny, natomiast istotne trudności sprawia jego realizacja. W książce staramy się odpowiedzieć na drugie pytanie, przy założeniu, że odpowiedź na pierwsze jest znana. W przypadku układów regulacji trzeba jej szukać w teorii sterowania, przy projektowaniu układów telemechaniki — w teorii informacji i metrologii itd. Gdy już wiadomo, **co** należy zrobić, można w tej książce szukać odpowiedzi, **jak** to zrobić.

Główną rolę w dalszych rozważaniach będzie odgrywał układ przełączający — podstawowy przetwornik informacji. Jeśli jego funkcje są złożone, w pierwszym etapie syntezy dzieli się go zwykle na mniejsze bloki o jednorodnych, prostych funkcjach, łatwiejsze do projektowania lub już znane. Ten etap syntezy, zwany *syntezą blokową*, może być przeprowadzony wówczas, gdy projektujący zna już pewną liczbę blo-

ków typowych i umie ocenić stopień złożoności realizacji poszczególnych funkcji. Z tego względu dalsze rozważania nie będą prowadzone w takiej kolejności, w jakiej dokonuje się syntezy układu, lecz w takiej, która wynika z rozbudowywania układu od postaci bardzo prostej do bardzo złożonej.

Układ przełączający ma zalety opisane wyżej dzięki temu, że jego elementy są proste i niezawodne, a te cechy wynikają z faktu, że elementy mają tylko dwa wyróżnione stany: załączony-wyłączony (przekaźnik), nasycony-zablokowany (tranzystor) itp. Takie elementy mogą przetwarzać wyłącznie sygnały dwuwartościowe — jedna wartość sygnału jest związana z jednym stanem elementu, druga z drugim. Dlatego też w dalszym ciągu przez nazwę „*sygnał dyskretny*" będzie rozumiany sygnał dwuwartościowy. W realizacjach fizycznych oznacza to, że sygnał ma charakter impulsowy: zwykle wyższy poziom to jedna wartość, niższy poziom — druga wartość. Te dwie wartości są najczęściej oznaczane przez 0 i 1. W wielu przypadkach odpowiednie ciągi impulsów oznaczają cyfry lub mogą być opisane przez cyfry i liczby; mówi się wówczas, że są to *sygnały cyfrowe*, a związane z nimi metody, układy i urządzenia też nazywa się *cyfrowymi* lub *numerycznymi*. Ponieważ głównym przedmiotem rozważań będą dalej układy przełączające (automaty cyfrowe), wszystkie współpracujące z nimi bezpośrednio urządzenia przetwarzania, indykacji, rejestracji itp. będą nazywane *urządzeniami zewnętrznymi* (*peryferyjnymi*).

1.2. PODSTAWY MATEMATYCZNE

1.2.1. ALGEBRA BOOLE'A

Układy przełączające można podzielić na dwie grupy o odmiennych właściwościach. Jeśli wartość każdego sygnału wyjściowego y_i jakiegoś układu zależy wyłącznie od aktualnej wartości sygnałów wejściowych $x_1, x_2, \ldots, x_n$, to układ taki jest nazywany *układem kombinacyjnym*. Jeśli wartość y_i zależy nie tylko od aktualnego stanu sygnałów wejściowych x, lecz także od ich poprzednich wartości, to jest to *układ sekwencyjny*. Metody syntezy sprowadzają zwykle układy sekwencyjne do postaci kombinacyjnej, co sprawia, że opisanie i zbadanie właściwości układów kombinacyjnych jest bardzo ważne.

Na podstawie definicji układ kombinacyjny o n wejściach i m wyjściach można opisać rodziną równań o postaci

$$y_i = f_i(x_1, x_2, \ldots, x_n) \qquad i = 1, 2, \ldots, m$$

Problemem jest jednak wybór funkcji, oznaczonych symbolicznie przez f_i i nazywanych *funkcjami przełączającymi*. Skoro x i y mogą przyjmować tylko wartości 0 lub 1, y_i nie mogą być zwykłymi funkcjami arytmetycznymi, gdyż dodawanie wprowadza liczbę 2, odejmowanie — liczby ujemne itd. Rozwiązanie problemu znaleziono w logice matematycznej, w tzw. *dwuelementowej algebrze Boole'a*. Algebra ta wprowadza trzy nowe operacje (funkcje), których argumentami i wynikami są zawsze elementy 0 lub 1. Operacje te można zdefiniować w następujący sposób:

— *suma logiczna* (*alternatywa*, *dysjunkcja*) argumentów a i b jest równa 1, gdy $a = 1$ **lub** $b = 1$;

— *iloczyn logiczny* (*koniunkcja*) argumentów a i b jest równy 1, gdy $a = 1$ **i** $b = 1$;

— *negacja* argumentu a jest równa 1, gdy **nie** jest $a = 1$ (czyli — gdy jest $a = 0$).

Stosowane są różne oznaczenia operacji boole'owskich, np.:

suma $a+b$, $a \vee b$

iloczyn $a \cdot b$, $a \wedge b$, $a \,\&\, b$

negacja $\overline{a}$, a', $\sim a$, $\neg a$

W technice zazwyczaj stosuje się symbole zwykłej sumy i iloczynu, z uprzednim zaznaczeniem, czy chodzi o operacje arytmetyczne, czy logiczne. Kropkę iloczynu często pomija się.

Wprowadzając symbole operacji można podane wyżej definicje zapisać w postaci równań:

$$0+0=0 \qquad 0\cdot 0=0 \tag{1-1}$$

$$0+1=1 \qquad 0\cdot 1=0 \tag{1-2}$$

$$1+0=1 \qquad 1\cdot 0=0 \tag{1-3}$$

$$1+1=1 \qquad 1\cdot 1=1 \tag{1-4}$$

$$\overline{0}=1 \qquad \overline{1}=0 \tag{1-5}$$

Z zależności tych można utworzyć związki ogólniejsze, np. z wyrażeń (1-1) i (1-2) wynika, że dodanie 0 do argumentu nie zmienia jego wartości, zaś pomnożenie argumentu przez 0 daje zawsze 0, czyli:

$$0+a=a \qquad 0\cdot a=0 \tag{1-6}$$

Podobnie z zależności (1-3) i (1-4) wynika, że

$$1+a=1 \qquad 1\cdot a=a \tag{1-7}$$

Analizując wyrażenia (1-1) i (1-4) można sformułować zależności:

$$a+a=a \qquad a\cdot a=a \tag{1-8}$$

natomiast ze wzorów (1-2), (1-3) i (1-5) można otrzymać

$$a+\bar{a}=1 \qquad a\cdot\bar{a}=0 \tag{1-9}$$

Podobnie z zależności (1-2) i (1-3) wynika przemienność operacji:

$$a+b=b+a \qquad a\cdot b=b\cdot a \tag{1-10}$$

a z wyrażenia (1-5) właściwość

$$\bar{\bar{a}}=a \tag{1-11}$$

Kilka innych ważnych właściwości powyższych operacji, jak: *łączność*

$$(a\cdot b)\cdot c=a\cdot(b\cdot c) \qquad (a+b)+c=a+(b+c) \tag{1-12}$$

rozdzielność

$$a\cdot(b+c)=ab+ac \qquad a+bc=(a+b)\cdot(a+c) \tag{1-13}$$

prawa de Morgana

$$\overline{ab}=\bar{a}+\bar{b} \qquad \overline{a+b}=\bar{a}\cdot\bar{b} \tag{1-14}$$

można udowodnić podstawiając wszystkie możliwe wartości argumentów.

Podstawiając w powyższych tożsamościach w miejsce prostych argumentów — funkcje (co wolno zrobić, gdyż funkcje te są dwuwartościowe) można zakres ich działania rozszerzyć na większą liczbę argumentów. Na przykład, skoro

$$a=a+a, \quad \text{to} \quad a+a=(a+a)+(a+a)$$

czyli:

$$a+a+\ldots+a=a \quad \text{i podobnie} \quad a\cdot a\ldots a=a \tag{1-15}$$

Podstawiając w wyrażeniu (1-14) odpowiednio $b=cd$ i $b=c+d$ otrzymuje się: $\overline{a\cdot cd}=\bar{a}+\overline{cd}=\bar{a}+\bar{c}+\bar{d}$ i $\overline{a+c+d}=\bar{a}\cdot\overline{c+d}=\bar{a}\cdot\bar{c}\cdot\bar{d}$, czyli ogólnie:

$$\overline{ab\ldots n}=\bar{a}+\bar{b}+\ldots+\bar{n} \qquad \overline{a+b+\ldots+n}=\bar{a}\cdot\bar{b}\ldots\bar{n} \tag{1-16}$$

W wielu powyższych zależnościach suma i iloczyn logiczny podlegają tym samym zasadom co zwykła suma i iloczyn, ale występują też wyraźne różnice, np. (1-13), (1-15). Wzory (1-12) pozwalają pominąć nawiasy w wieloargumentowych operacjach; w złożonych funkcjach obowiązuje kolejność działań: iloczyny-sumy.

Można udowodnić, że za pomocą trzech operacji algebry Boole'a może być opisany każdy układ kombinacyjny, zadany funkcjami przełączającymi

$$y_i = f_i(x_1, x_2, \ldots, x_n) \qquad i = 1, 2, \ldots, m$$

W opisie tym niekiedy dogodnie jest stosować jedną z postaci algebry Boole'a, tzw. *rachunek zdań*.

Podane wyżej definicje sumy i iloczynu różnią się w jednym tylko miejscu — spójnikami (wydrukowanymi półgrubą czcionką) **lub** oraz **i**. W wyrażeniach algebraicznych rolę tych słów odgrywają znaki sumy i iloczynu. Podobnie w definicji negacji decydującym słowem jest **nie**, któremu odpowiada znak negacji we wzorach, co umożliwia np. zdanie

„a równa się 1 lub b nie równa się 1"

zapisać w postaci $a+\overline{b} = 1$. Jeśli zdanie „działa przekaźnik P" oznaczy się symbolicznie przez $P = 1$, zdanie „nie działa przekaźnik P" przez $\overline{P} = 1$ ($P = 0$), „naciśnięty jest przycisk W" przez $W = 1$ itd., to zdanie opisujące układ techniczny: „silnik M działa, jeśli działa przekaźnik P i nie jest naciśnięty przycisk W lub gdy naciśnięty jest przycisk Z" można przedstawić w postaci wzoru

$$M = P\overline{W}+Z$$

W podobny sposób, przypisując prostym zdaniom dwie możliwe wartości — 0 albo 1 — można zapisać zdania bardziej złożone, co w wielu przypadkach umożliwia bezpośrednie wyznaczenie realizacji technicznej.

Ścisła odpowiedniość między operacjami logicznymi iloczynu, sumy i negacji oraz wyrazami *i*, *lub*, *nie* sprawiła, że często zarówno te operacje jak i realizujące je elementy są nazywane odpowiednio I, LUB, NIE.

1.2.2. WAŻNIEJSZE FUNKCJE LOGICZNE

Obok sumy, iloczynu i negacji w zastosowaniach praktycznych duże znaczenie mają również inne funkcje dwóch argumentów, które wprawdzie można zapisać za pomocą operacji $+$, $\cdot$, $-$, ale dogodniej jest nazwać je i oznaczyć odrębnie.

Implikacja $a \to b$ ma wartość 1, jeśli $a = 0$ lub $b = 1$, czyli

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

Funkcja zakazu (lub *różnica niesymetryczna*) $a \triangle b$ (lub $a - b$) ma wartość 1, jeśli $a = 1$ i $b = 0$, czyli:

$$a \triangle b = a \cdot \overline{b} \qquad a \triangle b = \overline{a \to b}$$

Drugi argument jest nazywany zakazującym.

Funkcja Peirce'a (*Łukasiewicza, Vebba*) $a \downarrow b$ ma wartość 1, jeśli $a = 0$ i $b = 0$

$$a \downarrow b = \overline{a} \cdot \overline{b} = \overline{a + b}$$

Funkcja „negacja sumy", czyli NIE-LUB, oraz jej *funktor* (element realizujący funkcję) występują zwykle pod nazwą NOR, od angielskich słów „Not-OR".

Funcja Sheffera $a|b$ ma wartość 1, jeśli $a = 0$ lub $b = 0$

$$a|b = \overline{a} + \overline{b} = \overline{ab}$$

Operacja „negacja iloczynu", czyli NIE-I, oraz jej funktor znane są pod nazwą NAND (od angielskich „Not-AND"). Zarówno NOR jak i NAND są funkcjami przemiennymi, ale nie łącznymi:

$$a \downarrow b = b \downarrow a \qquad a \downarrow (b \downarrow c) \neq (a \downarrow b) \downarrow c$$
$$a|b = b|a \qquad a|(b|c) \neq (a|b)|c$$

Równoważność $a \equiv b$ $(a \sim b)$ ma wartość 1, gdy argumenty mają jednakowe wartości

$$a \equiv b = \overline{a}\,\overline{b} + ab$$

Nierównoważność (*suma modulo 2, różnica symetryczna*) $a \oplus b$ ma wartość 1, gdy tylko jeden argument ma wartość 1

$$a \oplus b = a\overline{b} + \overline{a}b = \overline{a \equiv b}$$

Jest to operacja przemienna i łączna, o ciekawej właściwości:

jeśli $a \oplus b = c$, to $a \oplus b \oplus c = 0$, $a \oplus c = b$ itd.

Łatwo wykazać, że:

$$a \oplus 0 = a \qquad a \oplus 1 = \bar{a}$$
$$a \oplus a = 0 \qquad a \oplus \bar{a} = 1$$

Jeśli

$$\bar{a} \oplus b = c$$

to

$$a \oplus b = \bar{a} \oplus b \oplus 1 = c \oplus 1 = \bar{c}$$

Wartości wszystkich podanych wyżej funkcji zestawiono w tabl. 1-1.

Tablica 1-1

Ważniejsze funkcje logiczne dwóch zmiennych

a	b	$a \rightarrow b$	$a \triangle b$	$a \downarrow b$	$a \mid b$	$a \equiv b$	$a \oplus b$
0	0	1	0	1	1	1	0
0	1	1	0	0	1	0	1
1	0	0	1	0	1	0	1
1	1	1	0	0	0	1	0

1.2.3. SYSTEMY FUNKCJONALNIE PEŁNE

Wprawdzie każdą funkcję przełączającą można przedstawić za pomocą argumentów i operacji logicznych suma, iloczyn, negacja, ale techniczna realizacja tych operacji nie zawsze jest prosta i dlatego interesujące jest pytanie, czy istnieją inne operacje umożliwiające przedstawienie dowolnej funkcji przełączającej.

Zbiór operacji takich, że każda funkcja przełączająca może być przedstawiona za pomocą argumentów, stałych 0 i 1 oraz tych operacji, nosi nazwę *systemu funkcjonalnie pełnego* (SFP). Funkcje logiczne sumy, iloczynu i negacji tworzą tzw. *podstawowy system funkcjonalnie pełny*. Można go wykorzystać do wyznaczenia innych SFP zauważywszy, że jeśli operacje jakiegoś systemu umożliwiają realizację sumy, iloczynu i negacji, to tym samym umożliwiają realizację każdej funkcji przełączającej, czyli rozważany system jest systemem funkcjonalnie pełnym.

Łatwo sprawdzić, że podstawowy SFP nie jest minimalny, ponieważ sumę można (korzystając z prawa de Morgana) wyrazić za pomocą iloczynu i negacji

$$\text{skoro } \overline{a+b} = \bar{a}\cdot\bar{b}, \quad \text{to} \quad a+b = \overline{\bar{a}\cdot\bar{b}}$$

Tak więc operacje I oraz NIE tworzą SFP.

Podobnie z $\overline{ab} = \bar{a}+\bar{b}$ wynika, że $ab = \overline{\bar{a}+\bar{b}}$, więc operacje LUB, NIE też tworzą SFP.

Jeśli suma i iloczyn stanowią SFP, to systemem takim będzie również funkcja Peirce'a (NOR), zawierająca obie te operacje. Rzeczywiście, skoro

$$a \downarrow b = \overline{a+b}, \quad \text{to} \quad \bar{a} = a \downarrow a \quad \text{lub} \quad \bar{a} = a \downarrow 0$$

oraz

$$a+b = \overline{a \downarrow b} \quad \text{czyli} \quad a+b = (a \downarrow b) \downarrow 0$$

a z

$$a \downarrow b = \bar{a}\cdot\bar{b} \quad \text{wynika} \quad a\cdot b = \bar{a} \downarrow \bar{b} = (a \downarrow 0) \downarrow (b \downarrow 0)$$

Za pomocą operacji NOR można więc opisać wszystkie operacje podstawowego SFP; NOR stanowi zatem SFP.

Podobnie dla funkcji Sheffera (NAND) można napisać:

$$\bar{a} = a|a \quad \text{lub} \quad \bar{a} = a|1$$

$$ab = \overline{a|b} = (a|b)|1$$

$$a+b = \bar{a}|\bar{b} = (a|1)|(b|1)$$

Z zależności powyższych wynika, że stosowanie operacji NOR albo NAND bardzo komplikuje realizację prostych funkcji sumy i iloczynu, co — wydawałoby się — niweczy ich praktyczną przydatność. Okazuje się jednak, że liczba operacji | albo ↓ nie jest wcale dużo większa od liczby operacji suma, iloczyn, negacja, natomiast zalety jednego, uniwersalnego elementu stanowiącego SFP są liczne.

Inne SFP o jednej operacji można zbudować z implikacji lub funkcji zakazu, gdyż

$$\bar{a} = 1 \triangle a \qquad \bar{a} = a \to 0$$

$$ab = a \triangle \bar{b} = a \triangle (1 \triangle b) \qquad ab = \overline{a \to \bar{b}} = [a \to (b \to 0)] \to 0$$

$$a+b = \bar{\bar{a}} \triangle \bar{b} = 1 \triangle [(1 \triangle a) \triangle b] \qquad a+b = \bar{a} \to b = (a \to 0) \to b$$

Równoważność i suma modulo 2 nie tworzą same SFP, lecz trzeba je uzupełniać dodatkową operacją, np. sumą lub iloczynem i dlatego systemy te nie mają większego znaczenia praktycznego.

W podanej wyżej definicji SFP założono, że stałe 0 i 1, podobnie jak argumenty, odpowiadają istniejącym sygnałom. Takie założenie można przyjąć, gdy realizacja techniczna sygnałów 0 i 1 jest trywialnie prosta, a tak właśnie jest we wszystkich prawie stosowanych w automatyce elementach elektrycznych i pneumatycznych. W specjalnych przypadkach tzw. techniki dynamicznej uzyskanie stałych 0 i 1 (a zwłaszcza jednej z nich) jest nieco trudniejsze, co może mieć wpływ na wybór optymalnego SFP. Z tego powodu stała 0 i 1 są niekiedy włączane w skład SFP, tworząc tzw. system funkcjonalnie pełny w mocnym sensie.

1.2.4. ALGEBRA ZBIORÓW

W procesie syntezy układów przełączających bardzo przydatny jest jeszcze jeden przykład algebr Boole'a — *algebra zbiorów*.

Element zbioru to obiekt o określonych właściwościach, np. punkt jest elementem pewnej przestrzeni, liczba 2 jest elementem zbioru liczb naturalnych itp. Elementy oznacza się zwykle małymi literami, zbiory — dużymi, np. $S = \{s_1, s_2, \ldots, s_n\}$

$x \in A$ oznacza, że element x należy do zbioru A,
$x \notin A$ x nie należy do zbioru A.

Zbiór zawierający wszystkie elementy o określonych właściwościach to *zbiór pełny*, oznaczany przez I. Zbiór nie zawierający żadnego elementu, to *zbiór pusty*, oznaczany przez $\emptyset$.

Jeśli wszystkie elementy zbioru S są też elementami zbioru T, to S jest zawarty w T ($S \subseteq T$) i S jest nazywany *podzbiorem* zbioru T. Na przykład

$$\{1, 2, 4\} \subseteq \{0, 1, 2, 3, 4, 5\}$$

Jeśli $S \subseteq T$ i $T \subseteq S$, to zbiory są równe ($S = T$).

Iloczyn (przekrój) zbiorów S oraz T ($S \cap T$) to zbiór, którego elementy należą do S i do T; np.

$$\{1, 3, 4\} \cap \{0, 3, 5\} = \{3\}$$

Suma (połączenie) zbiorów S oraz T ($S \cup T$) to zbiór, którego elementy należą do S lub do T; np.

$$\{1, 3, 4\} \cup \{0, 3, 5\} = \{0, 1, 3, 4, 5\}$$

Uzupełnienie zbioru S (S') to zbiór elementów nie należących do S, np. jeśli $I = \{0, 1, 2, 3, 4, 5\}$, to

$$\{1, 3, 4\}' = \{0, 2, 5\}$$

Zbiory S oraz T są *rozłączne*, jeśli $S \cap T = \emptyset$.

Różnica zbiorów S oraz T ($S - T$ lub $S|T$) to zbiór, którego elementy należą do S i nie należą do T; np.

$$\{1, 3, 4\} - \{0, 3, 5\} = \{1,4\}$$

Można więc napisać: $S' = I - S$.

Jeśli we wzorach (1-1) do (1-16) zastąpi się sumę logiczną — sumą zbiorów, iloczyn logiczny — iloczynem zbiorów, negację — uzupełnieniem, stałą 0 — zbiorem pustym $\emptyset$, stałą 1 — zbiorem pełnym I, to zależności te, podobnie jak i inne wzory dwuelementowej algebry Boole'a, będą obowiązywały dla algebry zbiorów.

1.3. METODY KODOWANIA

1.3.1. SYSTEMY ZAPISU LICZB

Informację na wejściu i wyjściu układu przełączającego często dogodnie jest przedstawiać w postaci liczbowej, co umożliwia dyskretny charakter sygnałów. Jednakże stosowane do tego opisu liczby nie zawsze mogą być liczbami dziesiętnymi, skoro same sygnały są opisywane za pomocą dwóch tylko symboli — 0 i 1. Powstaje więc konieczność wprowadzenia *systemu dwójkowego zapisu liczb*.

Stosowany powszechnie *system dziesiętny zapisu* operuje dziesięcioma różnymi znakami dla przedstawienia cyfr, a każdej cyfrze, w zależności od jej pozycji względem przecinka, jest przypisana *waga*, będąca odpowiednią potęgą podstawy 10. Zapis liczby jest więc umownym zapisem współczynników przy odpowiednich potęgach 10

$$L_{10} = a_n \ldots a_2 a_1 a_0 = \sum_{i=0}^{n} a_i \cdot 10^i$$