ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ
1971 год Выпуск 47

УДК 681.142.1.01

ФОРМАЛЬНЫЙ СИНТЕЗ СХЕМ УСТРОЙСТВ ПО АЛГОРИТМАМ ИХ ФУНКЦИОНИРОВАНИЯ

И.В. Иловайский

В работе исследуется этап преобразования описаний функционирования устройств (А-описаний) в схемные описания (Сх-описания), относящийся к логическому синтезу [1]. Обычно этот этап явно не выделяется (так, в языке АЛОС, разработанном для описания устройств [2], есть средства для перехода к схемным описаниям, и в одной и той же записи могут присутствовать описание работы блока и описание структуры). Подобный трансляторный подход, когда, не выходя за рамки одного языка, постепенно опускаются до полного описания синтезируемого устройства, требует мощного математического обеспечения. При синтезе ЦВМ (и других аналогичных устройств) на небольших УЦВМ этап логического синтеза требуется расчленять на отдельные независимые части. Объекты, расположенные между подэтапами, должны быть формально определены. Такими объектами могут служить описания проектируемого устройства.

Процессор ЦВМ (и любое другое устройство) может быть задан двумя группами описаний: Сх-описаниями и А-описаниями. Внутри каждой группы описаний существует естественная иерархия, и описания А и Сх могут быть (по степени подробности) попарно со-

поставлены друг другу [3].

Инженерная традиция выделяет следующие описания цифрового устройства (см.рис.1). Любая методика проектирования реализует в схеме рис.1 некоторую последовательность преобразований, идущую вниз вправо. Преобразования А-Сх однозначны, а А - А и Сх-Сх, идущие вниз, требуют добавочной информации, которая может рассматриваться как выходной объект аналогичной системы описаний для элементарных подустройств всего устройства.

Для описания структуры схем вводится понятие ориентированной сети. Для полного представления схемных описаний используется понятие ориентированной сети над каталогом систем функций алгебры логики (СФАЛ).

Для описания алгоритмов используются граф-схемы (графы алгоритмов), операторы которых являются системами функций алгебры логики.

Между граф-схемами и ориентированными сетями (через СФАЛ) устанавливается соответствие, позволяющее строить одни по другим. При этом существенным является то обстоятельство, что построение структуры схемы по граф-схеме алгоритма не требует знания СФАЛ - операторов алгоритма, а требует лишь сведений об их входном и выходном множествах переменных.

Вводятся формальные экспликанты инженерных понятий "функциональная схема устройства", "логическая схема устройства", "алгоритм работы устройства", "микропрограммное описание работы устройства" и т.п. [4].

1. Формализация схемных описаний устройств

1.1. Структура схемных описаний

Ориентированные сети. Обычно схема устройства изображается как некоторое количество прямоугольников-элементов (могут быть и другие фигуры), соединенных ориентированными связями. Элементы имеют входы (помеченные стрелками) и выходы. Один выход какого-либо элемента может быть связан с несколькими входами других элементов. Такой пучок связей назовем узлом. Дадим формальное определение ориентированной сети и свяжем его с содержательным понятием схемы устройства.

Пусть даны $W=\{w\}$ – множество вершин сети и $\mathcal{E}=\{E_i\}$ – множество наборов, составленных из вершин. Каждый набор состоит из поднаборов – входного E_{i1} и выходного E_{i2}, т.е. $E_i = (E_{i1}, E_{i2})$ – упорядоченная пара. Тогда множество W с выделенной совокупностью наборов

$$\mathcal{E}=\{(E_{i1}, E_{i2})\}$$

называется ориентированной сетью и обозначается $\langle W, \mathcal{E}\rangle$. Наборы E_i называются элементами сети.

Мощности множеств W, $\mathcal{E}$ и E разумно считать конечными, так как реальные устройства имеют конечное число элементов и связей.

Введенный нами объект может рассматриваться как конечная сеть в смысле Яблонского [5] с введенным на множестве компонент наборов сети отношением порядка.

Геометрическая (схемная) интерпретация понятия ориентированной сети. Сопоставим каждому элементу (набору) прямоугольник. От прямоугольника отведем "отводы" – выходы, и каждый из них взаимно однозначно сопоставим вершинам из выходного поднабора. К прямоугольнику подведем стрелки – " выводы " (входы) и каждый из них взаимно однозначно сопоставим вершинам из входного поднабора. Именованный ввод (вывод) будем называть полюсом элемента. Соединим полюса, соответствующие одной и той же вершине, ломаными линиями, не пересекающими элементов. Видно, что полученная фигура может рассматриваться как схема некоторого устройства, если в вершину объединяется не более одного выходного полюса. Вершина при этом интерпретируется как узел. Набор E_0 интерпретируется как "внешний" элемент, то есть его полюса рассматриваются как входы и выходы схемы.

Схемы устройства имеют ту особенность, что один выходной полюс элемента связывается с несколькими входными, то есть соединение является узлом. Узлы суть непересекающиеся наборы полюсов. Потоки информации в устройстве не попадают в "тупики" и идут в конечном счете от входов к выходам, не теряясь и не возникая. Указанные особенности схем приводят нас к понятию

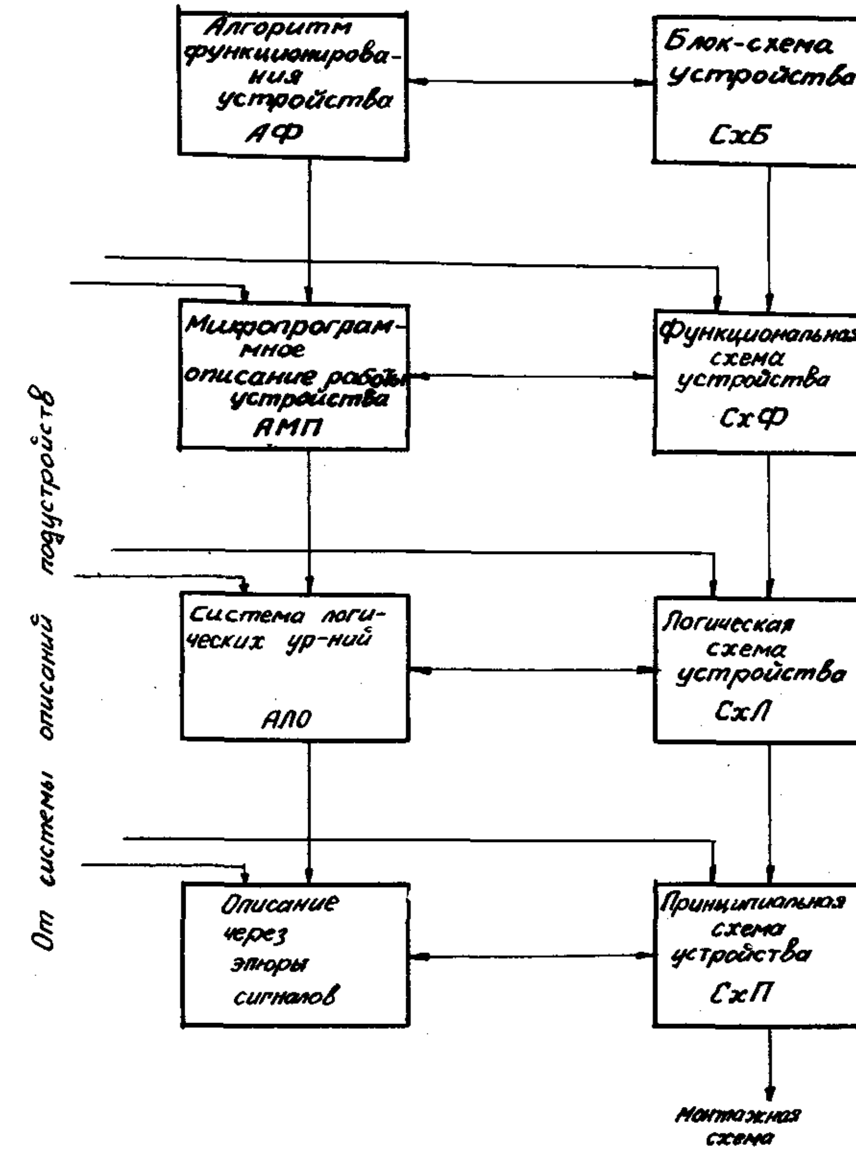

Рис. I. Система описаний устройства

правильной ориентированной сети, у которой

1) оба поднабора любого набора непусты;

2) любая вершина входит в один и только один выходной поднабор и хотя бы в один входной.

Указанное распространяется и на набор E_0, следует лишь учитывать, что поднабор E_{01} – это входы элемента E_0, то есть выходы сети, а E_{02} – это выходы элемента E_0, то есть входы сети.

На рис. 2 приведен пример правильной ориентированной сети $\langle W, \mathcal{E} \rangle$.

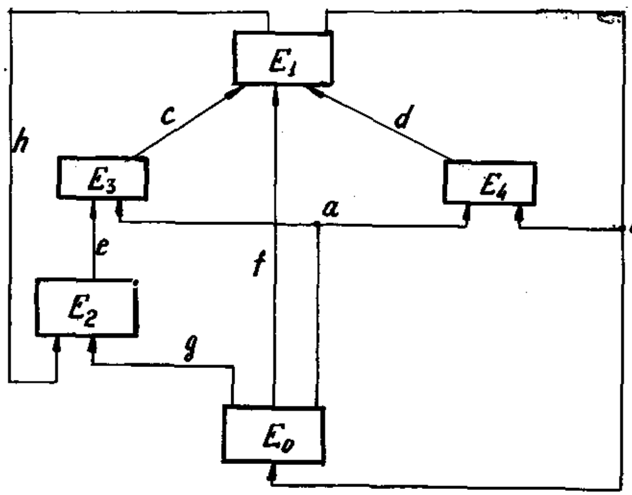

Рис. 2.

$$W = \{a, b, c, d, e, f, g, h\};$$

$$\mathcal{E} = \{E_0, E_1, E_2, E_3, E_4\};$$

$$E_0 = (b, afg);\ E_1 = (cdf, bh);\ E_2 = (gh, e);$$

$$E_3 = (ae, c);\ E_4 = (ab, d).$$

и ее геометрической реализации. Последнюю можно рассматривать как схему некоторого устройства.

В дальнейшем мы будем оперировать только с правильными

ориентированными сетями и для краткости можем употреблять просто термин "сеть".

Суперпозиция правильных ориентированных сетей. Пусть даны сети $\langle W, \mathcal{E} \rangle$ и $\langle U, \mathcal{F} \rangle$ над такими множествами вершин W и U, что $W \cap U = \varnothing$, а между F_{01} и E_{k2} с одной стороны, и F_{02} и E_{k1} с другой стороны, устанавливаются взаимно однозначные соответствия, $F_{01} \leftrightarrow E_{k2}$, $F_{02} \leftrightarrow E_{k1}$. Тогда сеть $\langle U, \mathcal{F} \rangle$ может быть подставлена в элемент E_k сети $\langle W, \mathcal{E} \rangle$. При этом множества E_k и F_0 изымаются, находится общее множество вершин

$$V = W \cup U \setminus F_0 \setminus E_k,$$

и мы получаем сеть

$$\langle V, (\mathcal{E} \setminus \{E_k\} \cup \mathcal{F} \setminus \{F_0\}) \rangle.$$

Геометрически это означает, что мы отождествляем контуры и соответствующие полюса элементов F_0 и E_k, внутри контура E_k рисуем сеть $\langle U, \mathcal{F} \rangle$, а затем стираем контур E_k.

Отметим важное свойство операции суперпозиции. Суперпозиция правильных ориентированных сетей дает правильную ориентированную сеть.

Действительно, пусть $\langle W, \mathcal{E} \rangle$ – правильная ориентированная сеть и в E_k подставляется правильная ориентированная сеть $\langle U, \mathcal{F} \rangle$. При суперпозиции ликвидируются два набора E_k и F_0, и новых наборов не вводится. Признак 2) правильности удовлетворяется, так как любая вершина, входящая в какой-либо выходной набор F_{j2}, может войти (по правилу отождествления E_k и F_0) в поднабор E_{k1}, но в силу условия

$$W \cap U \setminus E_k \setminus F_0 = \varnothing$$

ни в какой другой набор E_i $(i \neq k)$. Аналогичная картина имеет место для вершин из F_{j1}. Признак 1) выполняется, так как новых наборов не вводится и из наборов E_i и F_j $(i \neq k,\ j \neq 0)$ ничего не изымается. (Обратная суперпозиции операция также обладает этим свойством).

Соотнесенный граф и блок-схема устройства. Блок-схема обычно понимается как изображение (описание) устройства через элементы и информационные связи между ними. Содержательно блок-схему можно получить из схемы устройства, сопоставив каждому элементу последней элемент СхБ и соединив элементы СхБ, если были связаны соответствующие элементы в схеме устройства.

Формализуем приведенные рассуждения. Сопоставим каждому набору E_i ориентированной сети $<W, \mathcal{E}>$ вершину графа, а дуги определим следующим образом: рассмотрим возможные пары наборов из $\mathcal{E}$. Если у каких-либо двух наборов E_i и E_j $E_{i2} \cap E_{j1} \neq \emptyset$, вводим дугу от вершины графа, соответствующей набору E_j, к вершине, соответствующей набору E_i. Если имеет место $E_{j2} \cap E_{i1} \neq \emptyset$, вводится дуга обратной ориентации.

Граф $G(\mathcal{E})$, вершины которого соответствуют наборам (элементам) ориентированной сети $<W, \mathcal{E}>$, а дуги – вершинам указанной сети (узлам), называется соотнесенным графом сети $<W, \mathcal{E}>$ (с-графом).

На рис. 3 изображен с-граф ориентированной сети рис.2.

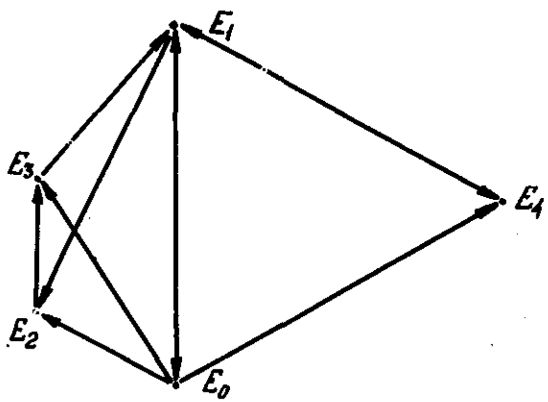

Рис. 3.

Заметим, что не всегда C-граф может быть интерпретирован как блок-схема. Так, C-граф сети, изображающей логическую схему, заведомо не будет блок-схемой.

Отметим два важных свойства C-графов.

При суперпозиции сетей вершина C-графа, соответствующая элементу E_k (в котором происходит подстановка), изымается вместе с инцидентными ей дугами. Из C-графа подставляемой сети удаляется вершина F_0 с инцидентными ей дугами, и с-графы объединяются в один путем введения дуг, которые соответствуют непустым пересечениям пар наборов вида $(E_{i1} \cap F_{j2}), (E_{i2} \cap F_{j1})$.

Обратная суперпозиции операция приводит к стягиванию в точку с-подграфа изымаемой подсети.

I.2. Представление схемных описаний правильными ориентированными сетями над каталогами систем функций алгебры логики

В инженерной практике описание устройства наряду со схемой (структурой соединений) предусматривает описание функционирования элементов и их взаимодействия. Любое А-описание несет достаточную информацию о работе устройства, поэтому нашей целью является связать А-описания со Сх-описаниями.

Связующим звеном между А и Сх-описаниями мы выбираем системы функций алгебры логики (СФАЛ), то есть один из видов АЛО (см.рис.I). Такой выбор объясняется тем, что, как мы покажем, с одной стороны, СФАЛ просто сопоставляются схемным описаниям любого уровня иерархии, а с другой – являются языком А-описаний. Поэтому задача связывания А и Сх – описаний сводится к задаче трансляции А-описаний в СФАЛ. При соответствующем выборе способов задания А-описаний способ трансляции не зависит от конкретного языка А-описания, а вся процедура перехода от А к Сх-описаниям строится так, что СФАЛ нигде явно не выписываются (до последнего этапа работы), достаточно сведений об их входных и выходных множествах переменных. Это обстоятельство позволяет не привязывать способы синтеза схем к конкретным базисам элементов. Вообще, при синтезе структуры схемы, мы можем не интересоваться функциями элементов схемы.

Для наших целей удобно определить СФАЛ как множество функций алгебры логики $\{f_1, \dots, f_\sigma\}$, рассматриваемое как отображение множества двоичных векторов

$$X = \{<x_1, x_2, \dots, x_m>\}$$

во множество двоичных векторов

$$Y = \{<y_1, y_2, \dots, y_n>\}.$$

Множества $X = \{x_i\}$ и $Y = \{y_j\}$ будем называть соответственно множествами входных и выходных переменных.

Две СФАЛ будем считать эквивалентными, если они реализуют одно и то же отображение $\mathcal{X}\to\mathcal{Y}$. (Естественно, должны выполняться соответствия $X_1\leftrightarrow X_2$ и $Y_1\leftrightarrow Y_2$).

Множество $\mathcal{B}$ элементов-представителей классов эквивалентности $\mathcal{T}=\{\varphi_i\}$ систем ФАЛ будем называть каталогом отображений, реализуемых СФАЛ из $\mathcal{T}$, или, короче, каталогом.

Введенные определения распространяются и на СФАЛ, в которых есть временные ФАЛ. При этом мы предполагаем, что временные ФАЛ заданы в форме [6];

$$\varphi=\varphi_0\cdot\tau_0\vee\varphi_1\cdot\tau_1\vee\ldots\vee\varphi_{s-1}\cdot\tau_{s-1},\quad \text{где}$$

$$\tau_\alpha=\begin{cases}0, & \text{если } t\neq\alpha,\\ 1, & \text{если } t=\alpha,\end{cases}\quad \text{а } \varphi_0,\ldots,\varphi_{s-1} \text{ есть ФАЛ.}$$

В связи с этим будем различать два непересекающихся множества входных переменных СФАЛ: обычное X и множество временных переменных $T=\{\tau_\alpha\}$. Тогда, вообще, СФАЛ реализует отображение $\mathcal{X}\cup T\to\mathcal{Y}$, где $T=\{<\tau_\alpha>\}$. T будем называть множеством управляющих переменных.

Ориентированные сети над каталогами СФАЛ. Сопоставим набору E_i правильной ориентированной сети $<W,\mathcal{E}>$ СФАЛ φ_i таким образом, чтобы установились взаимно однозначные соответствия

$$T_i\cup X_i\leftrightarrow E_{i1} \quad\text{и}\quad Y_i\leftrightarrow E_{i2}.$$

Будем говорить, что элемент E_i на своих выходах (выходных полюсах) E_{i2} реализует значения СФАЛ φ_i из $\mathcal{Y}_i$, если на его входы подаются наборы значений входных переменных из $\mathcal{X}_i\cup T_i$, или, короче, что E_i реализует φ_i.

Пусть каждый элемент E_i $(i\neq 0)$ правильной ориентированной сети $<W,\mathcal{E}>$ реализует некоторую СФАЛ φ_i. Имеем множество СФАЛ $\mathcal{T}=\{\varphi_i\}$ с каталогом $\mathcal{B}$. Тогда сеть реализует СФАЛ φ_W, полученную следующим образом. Имеется соответствие $Y_i\leftrightarrow E_{i2}$, но компоненты поднабора E_{i2} входят в какие-то поднаборы E_{j1}, E_{t1} и т.д. (в силу правильности сети), а это означает, что соответствующие $y\in Y_i$ отождествляются с соответствующими $x\in X_j$, $x\in X_t$ и т.д. Поэтому для каждой СФАЛ $\varphi_i\in\mathcal{T}$ можно произвести отождествление ее выходных переменных с входными переменными некоторых других СФАЛ $\varphi_j\in\mathcal{T}$. Объеди-

нив все φ_i в одну СФАЛ φ_W, получим для нее $X_W=\bigcup_i X_i$, $Y_W=\bigcup_i Y_i$ $(i\neq 0)$. В силу описанного построения путем суперпозиций ФАЛ из φ_W, последнюю можно привести к такому виду, что все промежуточные переменные будут удалены. Очевидно, останутся лишь переменные, сопоставленные поднаборам из E_0. Тогда $X_W=Y_0$, а $Y_W=X_0$.

Правильную ориентированную сеть $<W,\mathcal{E}>$, каждому набору которой (кроме E_0) взаимно однозначно сопоставлена СФАЛ $\varphi_i\in\mathcal{T}$ (где $\mathcal{T}=\{\varphi_i\}$ с каталогом $\mathcal{B}$) так, что $X_i\leftrightarrow E_{i1}$, а $Y_i\leftrightarrow E_{i2}$, будем называть правильной ориентированной сетью $<W,\mathcal{E}>$ над каталогом $\mathcal{B}$ и обозначать $<W,(\mathcal{E}[\mathcal{B}])>$.

Если выразить каждую ФАЛ из $\mathcal{B}$ через ФАЛ какого-либо базиса $\{f_1,\ldots,f_\sigma\}$, получим правильную ориентированную сеть $<W,(\mathcal{E}[\{f_1,\ldots,f_\sigma\}])>$ над базисом $\{f_1,\ldots,f_\sigma\}$. Очевидно, любые сети над каталогами можно преобразовать в сети над одним и тем же базисом. Отсюда следует, что любая правильная ориентированная сеть над каталогом эквивалентна некоторой логической сети, а потому реализует некоторый конечный автомат.

Суперпозиция сетей над каталогами. Пусть даны правильные ориентированные сети $<W,\mathcal{E}>$ и $<U,\mathcal{F}>$ и пусть сеть $<U,\mathcal{F}>$ может быть подставлена в сеть $<W,\mathcal{E}>$. Подстановка сети $<U,\mathcal{F}[\mathcal{D}]>$ в элемент E_i сети $<W,\mathcal{E}[\mathcal{B}]>$ в этих условиях возможна, если, кроме того, $\varphi_i\in\mathcal{T}$ эквивалентна φ_U и $X_i=X_U$, $Y_i=Y_U$. Естественно, операция нетривиальна, если φ_i и φ_U заданы разными способами.

При суперпозиции каталог $\mathcal{B}$ пополняется, $\mathcal{B}'=\mathcal{B}\cup\mathcal{D}$. Если в $\mathcal{T}$ φ_i была представителем одноэлементного множества, из $\mathcal{B}'$ изымается φ_i. СФАЛ φ_0, очевидно, в $\mathcal{B}'$ не войдет. (Вообще, СФАЛ, реализуемая сетью, в ее каталог не входит. В противном случае сеть рассматривается как элемент, то есть, иными словами, поскольку все СФАЛ φ_i содержатся в $\varphi_W=\varphi_0$, последняя их поглотит и каталог будет одноэлементным множеством. Отсюда элемент сети есть сеть над одноэлементным каталогом).

Нетрудно сформулировать процедуру проведения обратной операции.

2. Формализация алгоритмических описаний устройств

2.1. Формализация алгоритмов функционирования устройств

В инженерной практике принято описание функционирования устройства выполнять с помощью языков, содержащих операции передачи информации и удобных для регистрации движения информации в устройстве. Обычно такие языки просты по структуре, легко представляются граф-схемами. Примерами языков такого типа является язык регистровых передач [7], специальное подмножество адресного языка [8], ЯЯПАС [9], Ф-язык [10], язык APL [11].

Мы будем полагать, что АФ-описание (см.рис.1) устройства есть такой алгоритм, определяемый граф-схемой Γ ($\mathfrak{A}-\mathfrak{R}$ - алгоритм) [12], что каждый $\mathfrak{A}$ - оператор $\mathfrak{A}_i$ есть СФАЛ φ_i, определенная на некотором множестве двоичных векторов $\mathcal{U}$ и принимающая значения из того же множества, а каждый $\mathfrak{R}$ - распознаватель $\mathfrak{R}_j$ есть СФАЛ Π_j, определенная на том же множестве $\mathcal{U}$ и принимающая значения из некоторого множества $\mathcal{Z} = \{<\beta_{je}>\}$, компоненты которого определяются следующим образом. Пусть каждый распознаватель $\mathfrak{R}_j$ в зависимости от значения его входов и состояний осуществляет переход по какому-либо из k условий $\alpha_{j1},\dots,\alpha_{jk}$ к другим k операторам (распознавателям). На условия перехода налагаются ограничения

$$\forall s \forall t \, |s \neq t| \rightarrow \beta_{js} \wedge \beta_{jt} = 0; \quad \bigvee_s \beta_{js} = 1,$$ где

$$\beta_{je} = \begin{cases} 0, & \text{если } \alpha_{je} \text{ не выполнено,} \\ 1, & \text{если } \alpha_{je} \text{ выполнено.} \end{cases}$$

Исполнение $\mathfrak{A}-\mathfrak{R}$ алгоритма состоит в последовательном вычислении СФАЛ φ_i и Π_j, что возможно, если эти СФАЛ суть компоненты временной СФАЛ, описывающей весь алгоритм, то есть $\varphi_q = \bigcup_i \varphi_i$; $\Pi_r = \bigcup_j \Pi_j$. Тогда φ_q есть некоторое отображение множества $\mathcal{U}$ в себя, а Π_r - отображение множества $\mathcal{U}$ в $\mathcal{Z}$ (то есть обобщенный предикат алгоритма). Множество $\mathcal{Z}$ есть результат работы СФАЛ Π_r, множество выходных переменных которой $B = \{\beta_{je}\}$.

Исполнение алгоритма зависит от того, какой набор значе -

ний переменных $\tau \in T$ будет осуществлен, что, в свою очередь, зависит от того, какие значения приняли переменные из $\mathcal{Z}$, то есть должно существовать отображение $\mathcal{Z} \rightarrow T$.

Так как T и B суть множества двоичных переменных, указанное отображение может быть реализовано СФАЛ. Эта СФАЛ обычно задается эквивалентным ей конечным автоматом, который просто строится по графу граф-схемы Γ [13].

Поскольку временные СФАЛ эквивалентны логическим сетям [6], а последние - автоматам, $\mathfrak{A}-\mathfrak{R}$-алгоритм можно представить парой автоматов $<A, Z>$, как показано на рис. 4. Именно такую

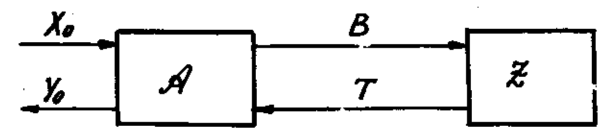

Рис. 4.

структуру, согласно [14], имеет процессор ЦВМ, то есть мы всегда можем интерпретировать любое АФ-описание (см.рис.1.) устройства как описание некоторой однопроцессорной ЦВМ (универсальной или специализированной).

Мы видим, что при интерпретации $\mathfrak{A}-\mathfrak{R}$-алгоритмов СФАЛ появляется в явном виде время. К отношению порядка на множестве T предъявляется требование, чтобы оно не противоречило отношению порядка на множестве ($\mathfrak{A}$, $\mathfrak{R}$), вводимом самой граф-схемой Γ.

2.2. Формализация микропрограммных описаний работы устройств

Понятие микропрограммного описания строится на основе понятия алгоритма, определяемого граф-схемой, следующим образом. Введем множество $\{t_j\}$ тактов времени. Пусть для каждого такта составлен неупорядоченный список, в который входят те и только те операторы $\mathfrak{A}-\mathfrak{R}$-алгоритма, которые выполняются в этом такте. Этот список будем называть микрокомандой. Отношение порядка на множестве тактов $\{t_j\}$ (а значит, и на множестве управляющих переменных T) определяется порядком следования микрокоманд в ходе реализации микропрограммного описания.

Сопоставим каждой микрокоманде вершину графа, а дуги графа определим следующим образом. Если микрокоманда μ_j может осуществиться (условно или безусловно) за микрокомандой μ_i, то в графе вводим дугу $\mu_i \to \mu_j$. Каждой вершине сопоставим $t \in \{t_j\}$ – такт, в котором выполняется соответствующая микрокоманда. Полученный граф G вместе с множествами $\mathfrak{A}, \mathfrak{R}, \{t_j\}$ и $\{u_j\}$ назовем граф-схемой микропрограммного описания работы устройства.

3. Связь между $\mathfrak{A}$-$\mathfrak{R}$ алгоритмами и ориентированными сетями

3.I. $\mathfrak{A}$-$\mathfrak{R}$ алгоритмы, реализуемые сетями, и определение понятия "схема устройства"

Ранее мы связали СФАЛ с ориентированными сетями и дали определение правильной ориентированной сети над каталогом. В силу этого определения и данной нами интерпретации $\mathfrak{A}-\mathfrak{R}$- алгоритма СФАЛ, мы можем сказать, что граф-схема Γ (для АМП соответственно граф-схема G) реализуется правильной ориентированной сетью $\langle W, (\mathcal{E}[\mathcal{B}])\rangle$, если.

I) множеству переменных $X \cup T \cup Y \cup B$ сопоставлено множество вершин сети W;

2) каждому оператору $\mathfrak{A}_i$ сопоставлена СФАЛ $\varphi_i \leftrightarrow E_i$;

3) каждому распознавателю $\mathfrak{R}_j$ сопоставлена СФАЛ $\Pi_j \leftrightarrow H_j$ $(H_j \in \mathcal{E};\ \{H_j\} \cap \{E_i\} = \varnothing)$;

4) $\mathcal{B}$ есть каталог отображений из $\{\varphi_i\} \cup \{\Pi_j\}$.

Ориентированная сеть $\langle W, (\mathcal{E}[\mathcal{B}])\rangle$ в таком случае реализует СФАЛ $\varphi_q \cup \varphi_r = \varphi_w$. Эту сеть мы будем называть операционной частью полной сети, реализующей Γ (соответственно G). В силу того, что граф-схемы Γ и G отличаются только отношением порядка на множества $(\mathfrak{A}, \mathfrak{R})$, но не элементами, операционная часть будет одной и той же в обоих случаях. Если построить сеть A_c, реализующую СФАЛ φ_o, получим совокупную сеть, реализующую Γ (соответственно G). Эта совокупная сеть будет функциональной схемой устройства, реализующего $\mathfrak{A}-\mathfrak{R}-$ алгоритмы. Действительно, под функциональной схемой принято понимать описание устройства в виде элементов и связей между ними, охватывающих все информационные и управляющие сигналы.

Дадим формальное определение функциональной схемы.

Правильную ориентированную сеть $\langle W, (\mathcal{E}[\mathcal{B}] \cup A_o[\varphi_o])\rangle$ над каталогом $\mathcal{B} \cup \{\varphi_o\}$ назовем функциональной схемой устройства, реализующего $\mathfrak{A}-\mathfrak{R}$-алгоритм (либо МП-описание), A_o будем именовать генератором (распределителем) тактов, если выполнены соотношения:

$$\mathcal{E} = \{E_0, E_1, \dots, E_i, \dots, E_n, H_1, \dots, H_j, \dots, H_m\};$$

$$\cup E_{i1} \leftrightarrow X_q \cup T_q; \qquad \cup H_{j1} \leftrightarrow X_r \cup T_r;$$

$$\cup E_{i2} \leftrightarrow Y_q; \qquad \cup H_{j2} \leftrightarrow Y_r = B;$$

$$A_{01} \leftrightarrow B; \qquad A_{02} \leftrightarrow T_q \cup T_r.$$

и условия п п. I–4, перечисленные на стр. 68. Обобщенная функциональная схема приведена на рис.5.

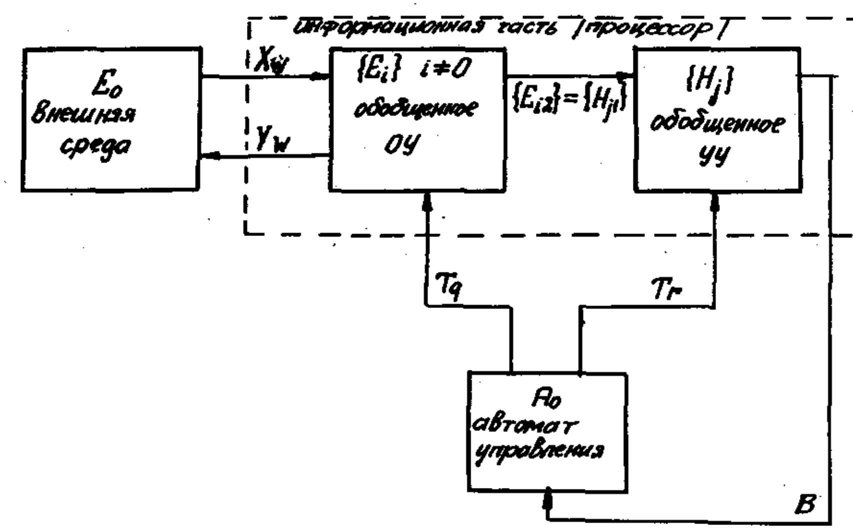

Рис. 5.

С-граф функциональной схемы будем называть блок-схемой.

Если от СхФ путем суперпозиции перейти к логической сети в каком-либо базисе, получим логическую схему устройства, реализующего алгоритм, определенный граф-схемой Γ (соответственно G). Такое определение не расходится с содержательным представлением о логической схеме как описании устройства в виде логических узлов, каждый из которых является наименьшей схемной единицей в данной системе элементов.

3.2. Связь между подстановками граф-схем и суперпозицией ориентированных сетей

Подстановке граф-схемы Γ_2 в другую граф-схему Γ_1 [I2] на место некоторого оператора $\mathfrak{A}_i$ соответствует подстановка СхФ устройства, реализующего Γ_2 , в элемент E_i функциональной схемы, реализующей Γ_1 . Однако существует некоторое различие между СхФ, реализующей совокупную граф-схему, и СхФ, полученной суперпозицией. Операционные части обеих СхФ, естественно, совпадают, однако если в первом случае СхФ имеет общий генератор, то во втором - в схеме два генератора и две системы тактностей: медленная - у основной СхФ и быстрая - у СхФ , подставленной в элемент E_i . Путем преобразования тактностей и композиции автоматов два генератора могут быть сведены в один.

3.3. Применение метода к случаю задания алгоритма на конкретном языке

В семантике таких языков, как Φ -язык, ЛЯПАС и т.д., переменная, которой присвоено значение, сохраняет его, пока ей не присвоится новое значение. При реализации граф-схемы сетью это соответствует тому, что входным и выходным переменным оператора должны сопоставляться некоторые элементы сети, то есть переменные операторов нельзя отождествлять со входными переменными СФАЛ, реализующих эти операторы. Фактически переменные СФАЛ в языковой записи явно не присутствуют. Тогда, рассматривая переменную в записи алгоритма (и в соответствующей записи граф-схеме) как оператор, мы приходим к следующему представлению. Каждый явно выписанный оператор $\mathfrak{A}_i$ (распознаватель $\mathfrak{R}_j$) есть последовательность операторов, начинающаяся с $\mathfrak{A}_i^o$ (соответственно $\mathfrak{R}_j^o$), реализующих требуемое преобразование, и

группы операторов запоминания результата $\mathfrak{A}_i^1$,..., $\mathfrak{A}_i^n$ (соответственно $\mathfrak{R}_j^1$,..., $\mathfrak{R}_j^n$), которым соответствуют выходные переменные оператора $\mathfrak{A}_i$ ($\mathfrak{R}_j$). Входные переменные $\mathfrak{A}_i$ ($\mathfrak{R}_j$)-это перечень тех операторов запоминания, результаты работы которых использует оператор $\mathfrak{A}_i^o$ (соответственно $\mathfrak{R}_j^o$). При интерпретации граф-схемы операторам $\mathfrak{A}_i^o$ и $\mathfrak{R}_j^o$ сопоставляются СФАЛ, не зависящие от времени, а операторам запоминания (переменным)-СФАЛ вида

$$f_i = \neg(x_{i_1}\tau \vee \neg(x_{i_2}\tau \vee f_i)),$$

то есть регистры с раздельными входами.

Подробное изложение алгоритмов синтеза СхБ и СхФ по АФ-описанию и соответственно АМП-описанию приведено в работах [I5] и [I6].

3.4. Синтез логической схемы по функциональной ведется методом суперпозиции сетей. Для упрощения этой процедуры сети задаются в форме $\langle \mathcal{E}, \mathcal{V} \rangle$ через множество наборов $\mathcal{E}$ и узлов $\mathcal{V}$. Синтез логической схемы производится путем объединения $\langle \mathcal{E}, \mathcal{V} \rangle$- описаний СхФ с $\langle \mathcal{E}, \mathcal{V} \rangle$- описаниями из библиотеки логических схем элементов(каталога). Если в библиотеке даны принципиальные схемы элементов, результатом синтеза будет СхП устройства.

Литература

I. ГЛУШКОВ В.М. Проблемы синтеза цифровых автоматов.-"Теория конечных и вероятностных автоматов".-Труды Международного симпозиума, "Наука", М., I965.

2. ГЛУШКОВ В.М., КАПИТОНОВА Ю.В., ЛЕТИЧЕВСКИЙ А.А. Об автоматизации проектирования вычислительных машин.-"Кибернетика", I967, №5.

3. ИЛОВАЙСКИЙ И.В. О некоторых однозначных преобразованиях описаний процессора ЦВМ. -"Тезисы докладов Всесоюзной межвузовской конференции по алгоритмическим методам проектирования цифровых систем",Л.,I96I.

4. ИЛОВАЙСКИЙ И.В. Об одном методе автоматического синтеза схем цифровых устройств по алгоритмам функционирования . Автореферат диссертации на соискание ученой степени канд.техн. наук, Новосибирск, I970 г.

5. ЯБЛОНСКИЙ С.В. Основные понятия кибернетики,-"Проблемы кибернетики", 1959, № 2.

6. ПОСПЕЛОВ Д.А. Логические методы анализа и синтеза схем. "Энергия", 1968.

7. BARTEE T.C., LEBOW I.L., REED I.S. Theory and design of digital machines, N.Y.,1962.

8. ИЛОВАЙСКИЙ И.В., ЛОЗОВСКИЙ В.С., ФЕТ Я.И. Применение адресного языка для автоматизации синтеза цифровых вычислительных устройств. - "Вычислительные системы", Новосибирск, "Наука" СО, 1965, вып.18.

9. ЗАКРЕВСКИЙ А.Д. Алгоритмический язык ЛЯПАС и автоматизация синтеза дискретных автоматов, Томск, Изд.ТГУ, 1966.

10. ПИСКУНОВ С.В., СЕРГЕЕВ С.Н., СИДРИСТЫЙ Б.А. Язык для описания алгоритмов функционирования ЦВМ.- "Вычислительные системы", Новосибирск, "Наука" СО, 1969, вып.34.

11. IVERSON K. A Programming Language, N.Y.,1962.

12. КАЛУЖНИН Л.А. Об алгоритмизации математических задач.- "Проблемы кибернетики", 1969, № 2.

13. МИЩЕНКО А.Т. Формальный синтез автомата по микропрограмме.-"Кибернетика", 1968, №3, II."Кибернетика",1968, №5.

14. ГЛУШКОВ В.М. Теория автоматов и формальные преобразования микропрограмм. -"Кибернетика", 1965, №5.

15. ИЛОВАЙСКИЙ И.В., СЕРГЕЕВА Э.Е. Построение блок-схемы ЦВМ по алгоритму ее функционирования.- "Вычислительные системы", Новосибирск, "Наука" СО, 1968, № 31.

16. ИЛОВАЙСКИЙ И.В. Построение функциональной схемы ЦВМ по алгоритму ее работы. -"Вычислительные системы", Новосибирск, "Наука" СО, 1969, № 34.

Поступила в редакцию
22. I. 1970