

**СИСТЕМА ЗА АВТОМАТИЧНОТО ЧЕРТАНЕ
НА ФУНКЦИОНАЛНИ СХЕМИ****С. Сребрев, Т. Величков, В. Лазаров и К. Боянов****УВОД**

Основната цел на системите за автоматизация на проектирането на цифрови устройства е същественото намаляване на разхода на време и човешки ресурси. Документирането на логическите схеми може да се счита като един от първите етапи при изграждането на една такава система, тъй като те са основният продукт от работата на инженера. Изработването на логически схеми по автоматичен път дава възможност да се спазват определените стандарти и извличат всички необходими за по-нататъшното проектиране данни с минимална загуба на време. Освен това в процеса на проектирането се налагат непрекъснати изменения в направените схеми и една система за автоматично чертане позволява на конструктора да разполага с историята на проектирането, което в редица случаи е от изключително значение. В [1] и [2] са описани системи за автоматично чертане на логически схеми, без да се показва използваният алгоритъм.

В настоящата работа се разглежда една система за автоматично чертане на функционални схеми (САЧ), с помощта на която могат да се получат стандартни изображения на работните чертежи на конструктора при оптимално трасиране на съединителните линии.

ОБЩИ ПОЛОЖЕНИЯ

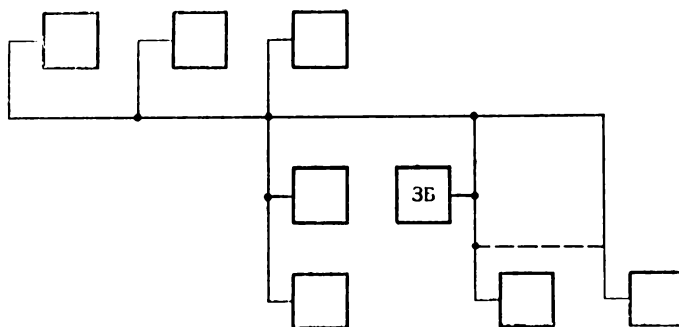
Разглежданата система реагира на символичното описание на приетото от инженера разположение на логическите елементи (логически блокове — ЛБ) в стандартна бланка. Прекарването на връзките между тях (трасирането) е основната задача на предлагания алгоритъм.

Направени са следните допускания:

1. Инженерът изготвя желания чертеж на ръка върху специална мрежа (бланка), в която съществуват определени позиции за разполагане на ЛБ. Графичната координатна мрежа се състои от елементарни графични клетки, образувани от пресичането на хоризонтални и вертикални зони, наричани по-долу полета и колони. Логическите блокове се раз-

полагат така, че да заемат една или няколко цели елементарни графични клетки.

2. Всеки блок има един изход и няколко входа, чийто брой се определя от типа на елемента. Положението на входовете е фиксирано за



Фиг. 1

всеки тип ЛБ, а положението на изхода може да се изменя в границите на ЛБ. Елементи с повече от един изход се разглеждат като съставени от отделни едноизходни ЛБ, положението на чиито изходи е фиксирано. Мястото на ЛБ в стандартната мрежа се оказва с координати, посочени от инженерта.

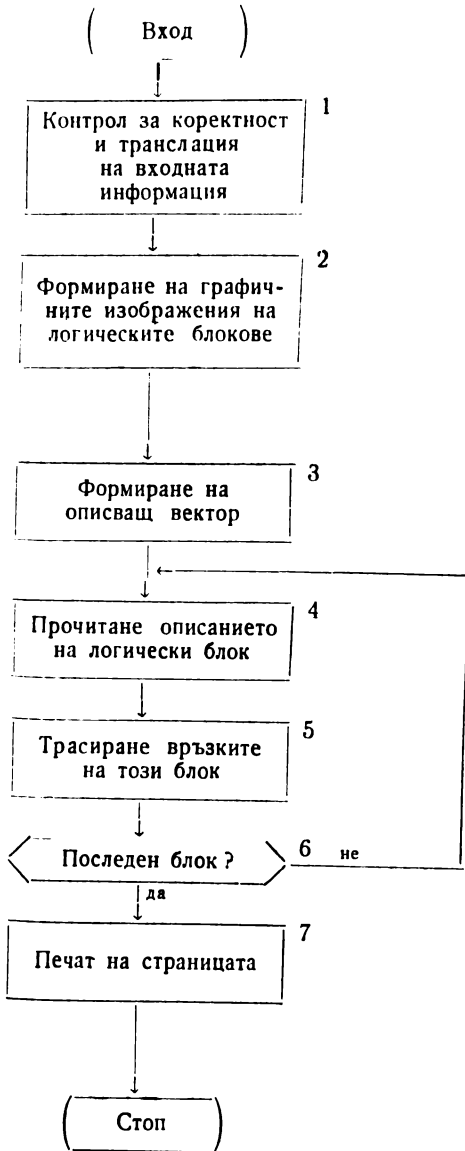
3. Връзките между логическите блокове се чертаят с помощта на хоризонтални и вертикални линии, които могат да пресичат други връзки, но не и да лягат върху тях. Вертикалните съединителни линии могат да се поставят само в определени колони, наречени междини, в които е недопустимо разполагането на ЛБ. Хоризонталните участъци от съединителни линии може да се чертаят във всички полета. В полетата и междините линиите се прекарват по определени трасета. Освен това логическите блокове позволяват трасиране на хоризонтални съединителни линии по граничната си контурна линия. Такива позиции ще наричаме специални трасета.

При трасирането са използвани два критерия за оценка на оптималността на съединителните линии: сборът от дължините на линиите за реализация на една връзка да е минимален и броят на ъглите в нея да е също минимален. Изискванията, залегнали в критериите, са до голяма степен противоречиви: използването на една хоризонтална линия в съответствие с втория критерий нарушава първия, тъй като общата дължина на връзката се увеличава (фиг. 1).

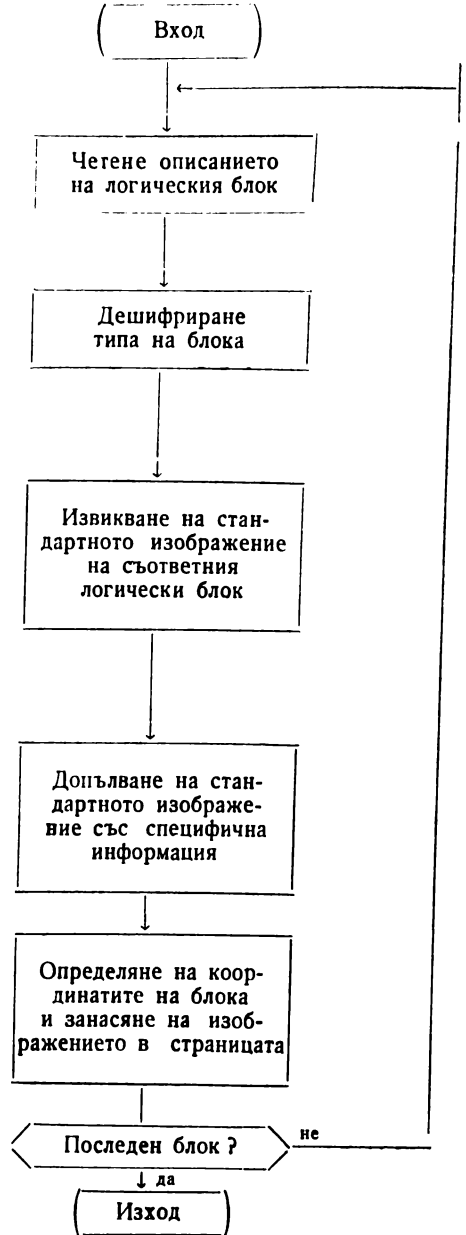
Направените експерименти наложиха избирането на компромисно решение, при което се използват две хоризонтални линии, определени като лява и дясна, лежащи съответно от лявата и от дясната страна на изхода на разглеждания блок, който ще наричаме задаващ (ЗБ). В случай, че алгоритъмът изчерпи възможностите, предоставени му от двете хоризонтални линии, без да достигне до сполучливо разрешение, той разпада линията, която не е била трасирана, на две части и прави наново опит за трасиране.

ПРОГРАМНА СИСТЕМА

Информацията за САЧ (описание на типа, разположението в графичната координатна мрежа и във връзките на всички логически блокове от схемата) се въвежда на символичен език, като се извършва контрол за коректност и трансляция в нужния машинен формат. Освен това се въвеж-



Фиг. 2

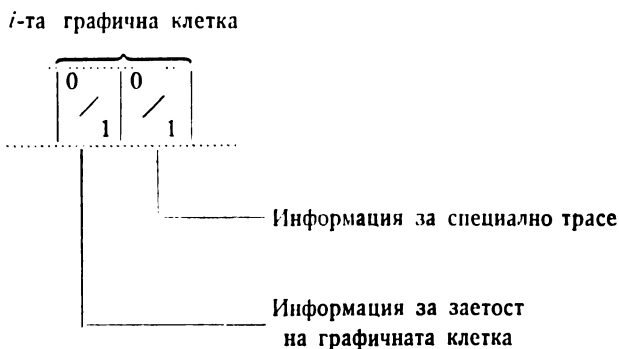


Фиг. 3

да един масив, който съдържа кодовете на символите, очертаващи тъй наречените стандартни изображения (т. е. контурите на ЛБ без поясняваща информация). Съдържанието на този масив определя вида и броя на типовете елементи, които могат да бъдат включени в изчертаваните функционални схеми. При работата на програмната система в паметта на машината се формира масивът В, наречен страница, който съдържа кодовете на символите, очертаващи ЛБ и връзките между тях. Програмната система (фиг. 2) се състои от две части: програми, реализиращи графичните изображения на ЛБ; и програми, извършващи трасирането на съединителните линии.

На фиг. 3 е показана блок-схемата на първата част от алгоритъма — формиране графичните изображения на логическите блокове. Като използва входната информация, програмата занася в масив В кодовете на всички графични символи, необходими за очертаване на геометрическите контури на логическите блокове. На този етап графичните изображения се допълват със специфична за всеки блок информация (монтажна координата, номер на изводите на интегралния модул и т. н.).

Състоянието на всяка елементарна графична клетка се характеризира с описващ вектор (ОВ), който е показан на фиг. 4.



Фиг. 4

За всяка графична клетка се записват два бита:

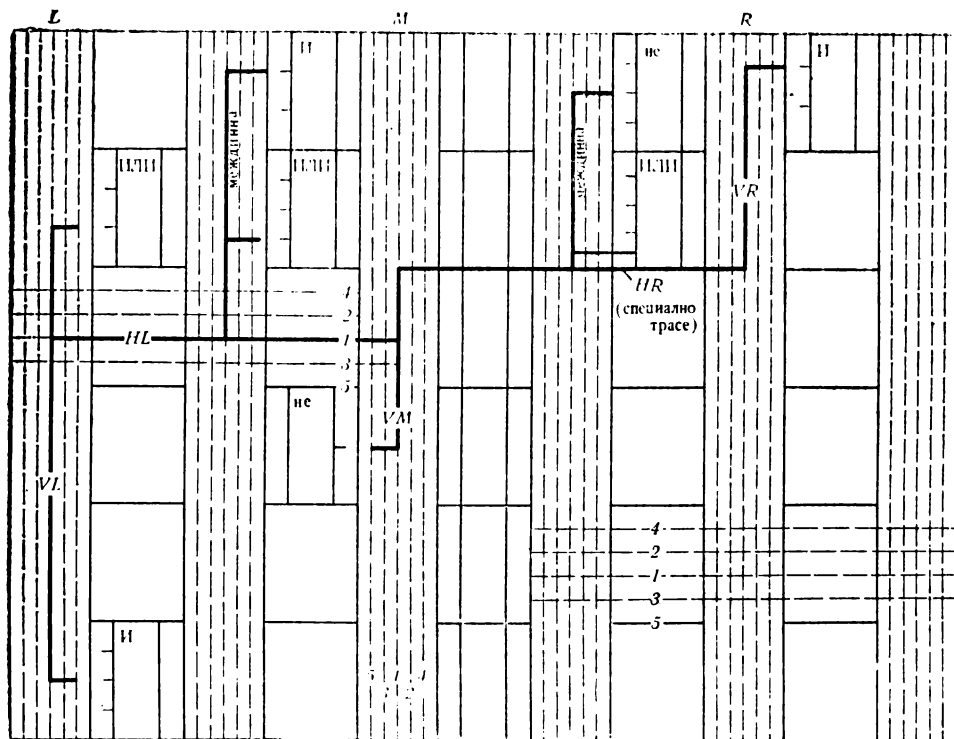
$$\begin{array}{l}
 \text{първият бит е} \\
 \text{вторият бит е}
 \end{array}
 \left\{ \begin{array}{l}
 = 1, \text{ ако в дадената графична клетка е разположено} \\
 \text{цяло или част от изображението на логически} \\
 \text{елемент;} \\
 = 0, \text{ ако в елементарната графична клетка не се} \\
 \text{съдържа изображение на логически елемент;} \\
 \\
 = 1, \text{ ако в последния ред на елементарната графична} \\
 \text{клетка е възможно прекарването на съедини-} \\
 \text{телна линия (специално трасе);} \\
 = 0, \text{ ако прекарването на такава линия е невъзможно.}
 \end{array} \right.$$

В следващия етап от алгоритъма трасиране на връзките в подходящи адреси на страницата се нанасят кодовете на символите, чрез които се очертават връзките.

При трасирането връзките се изчертават, като последователно се проследяват съществуващите електрически съединения между изходите на

всеки ЛБ и входовете на всички ЛБ върху страницата. Редът на анализ на връзките на отделните ЛБ е от горе на долу и от ляво на дясно. В процеса на работа на програмата този ред може да бъде изменен. Тази част от алгоритъма е оформена като подпрограма и при всяко активизиране резултатът от нейната работа е определяне и нанасяне върху страницата връзките на текущия ЗБ. Като входна информация подпрограмата използва координатите върху чертежа на ЗБ, координатите и номерата на входовете на ЛБ, които подлежат на свързване със дадения ЗБ.

Трасирането се извършва на два етапа. В първия етап се приема условно, че текущият ЗБ е пръв в реда на изчертаване, т. е. предполага се, че върху страницата не са нанесени никакви съединения от други ЗБ. Тази абстракция позволява да се определи една независима от останалите връзки система от линии, подчиняващи се на приетите критерии за оптималност. Тази система ще наричаме идеално трасе (ИТ). Във втория етап програмата се стреми да съвмести определеното ИТ със съществуващите вече връзки върху страницата. Измененията, които се налагат, предизвикват осцилиране на хоризонталните и вертикалните линии около



Фиг. 5

положението на ИТ, което създава гаранция за спазване на избрания критерий за оптималност. При определени критични случаи подобно съвместяване е невъзможно и тогава програмата предприема различни стъпки, увеличаващи възможността за решение. След намиране на точното поло-

жение на всяка линия програмата нанася в страницата кодовете на съответните символи, описващи линиите, при което се отчитат възможните пресичания с линии, определящи по-рано трасирани връзки. При описанието на алгоритъма се въвеждат някои величини. V_{\min} и V_{\max} определят началната и крайната точка на хоризонталните линии. V_{\min} показва отстоянието на лявата крайна точка до началото на страницата (изразено в брой елементарни позиции — символи), а V_{\max} — на дясната крайна точка. Аналогично за всяка вертикална линия се дефинират A_{\min} (горната крайна точка) и A_{\max} (долната крайна точка). Използването на описаните величини свежда системата от линии до система от точки.

За по-лесно определяне положението на линиите се приема номериране на вертикалните и хоризонталните трасета в междините и полетата. Както е показано на фиг. 5, за трасе № 1 е прието централното, а нарастващите номера осцилират в двете посоки.

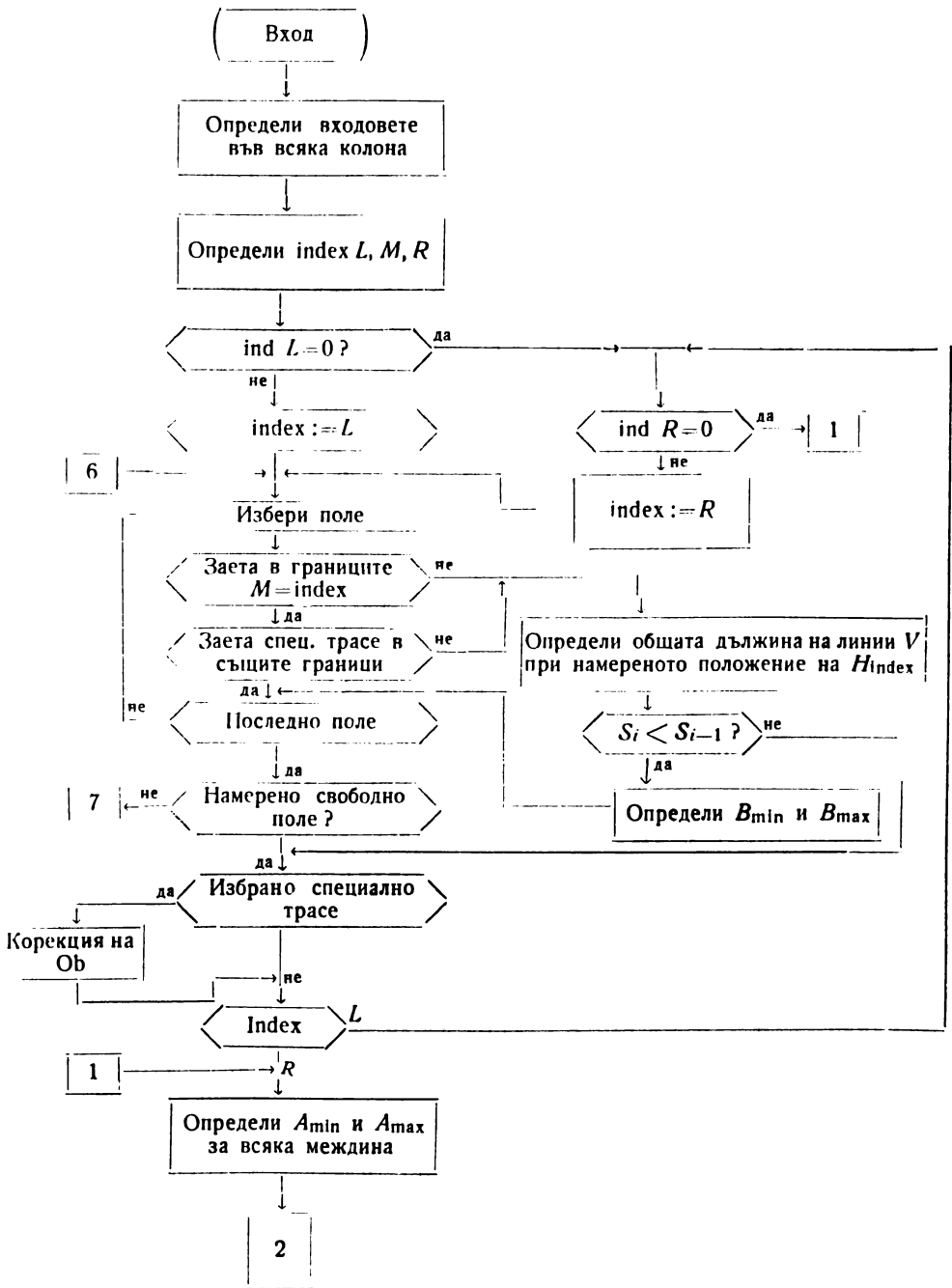
Индексът M показва номера на колоната (междината), в която се намира изходът на ЗБ (фиг. 5). Индексите L и R показват номерата на най-лявата и най-дясната колона (междина), в която има входи, подлежащи на свързване с изхода на ЗБ. Нулевата стойност на тези два индекса изразява отсъствие на ЛБ съответно наляво ($\text{ind } L=0$) или надясно ($\text{ind } R=0$) от колоната с индекс M , т. е. липсва на лява (HL) или дясна (HR) хоризонтална линия. При описанието на алгоритъма въведените индекси имат по-обобщен смисъл. Присвояването на някой от тях на една текуща величина, наречена index , означава, че се предвижда обработка или на съответната вертикална (VL, VM или VR), или на съответната хоризонтална линия (HL или HR).

На фиг. 6 е показана блок-схемата на алгоритъма за трасиране на връзките на даден ЗБ. Отначало се намира положението на точките от страницата, които трябва да бъдат свързани (в случая под точка разбираме вход на ЛБ или изход на ЗБ). Положението на всеки вход или изход се фиксира чрез отстоянието му от началото на страницата, изразено в елементарни позиции. За намирането на тази система от точки програмата използва входното описание.

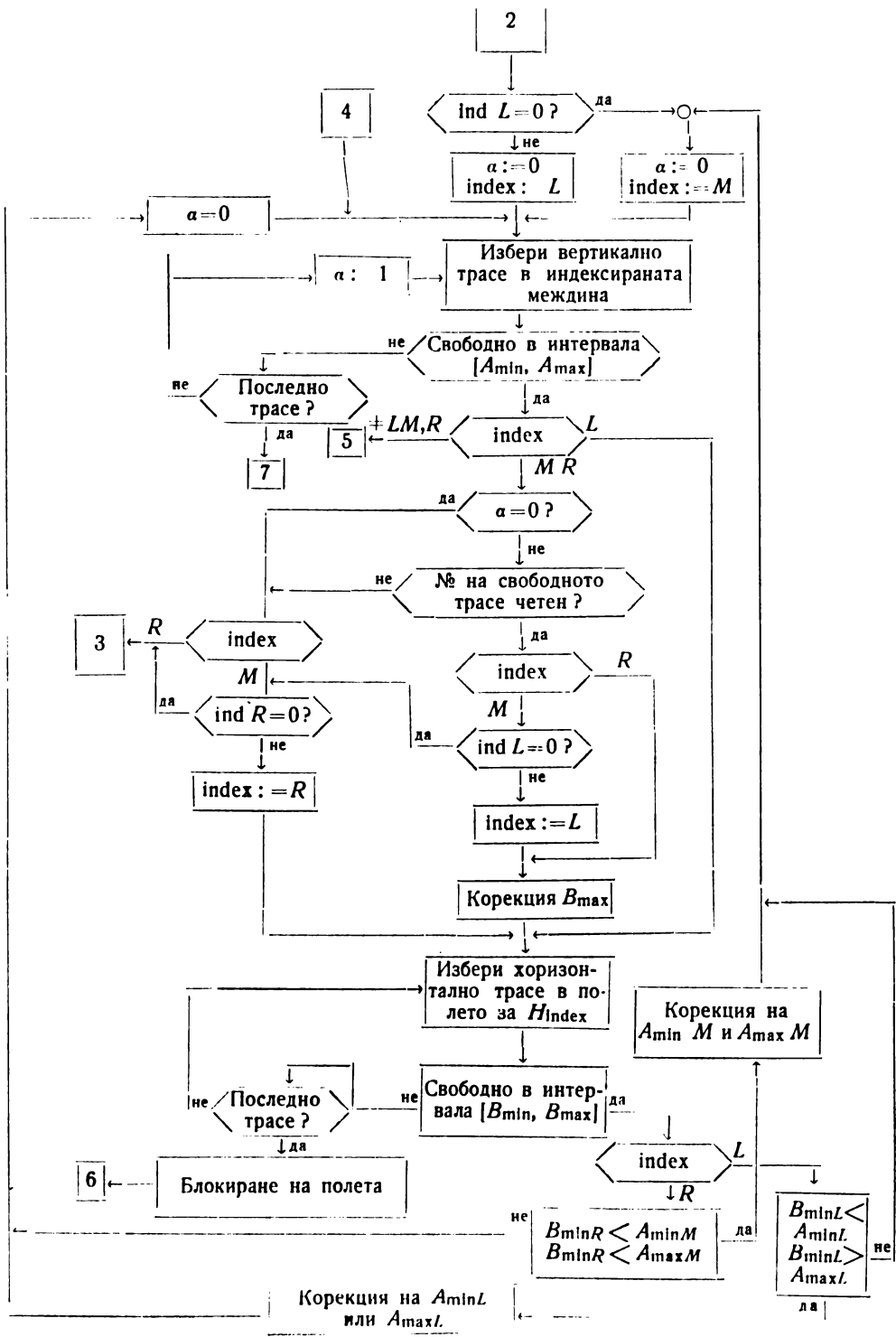
Положението на хоризонталните линии се определя при спазване критерия за минимална обща дължина на прекараните линии.

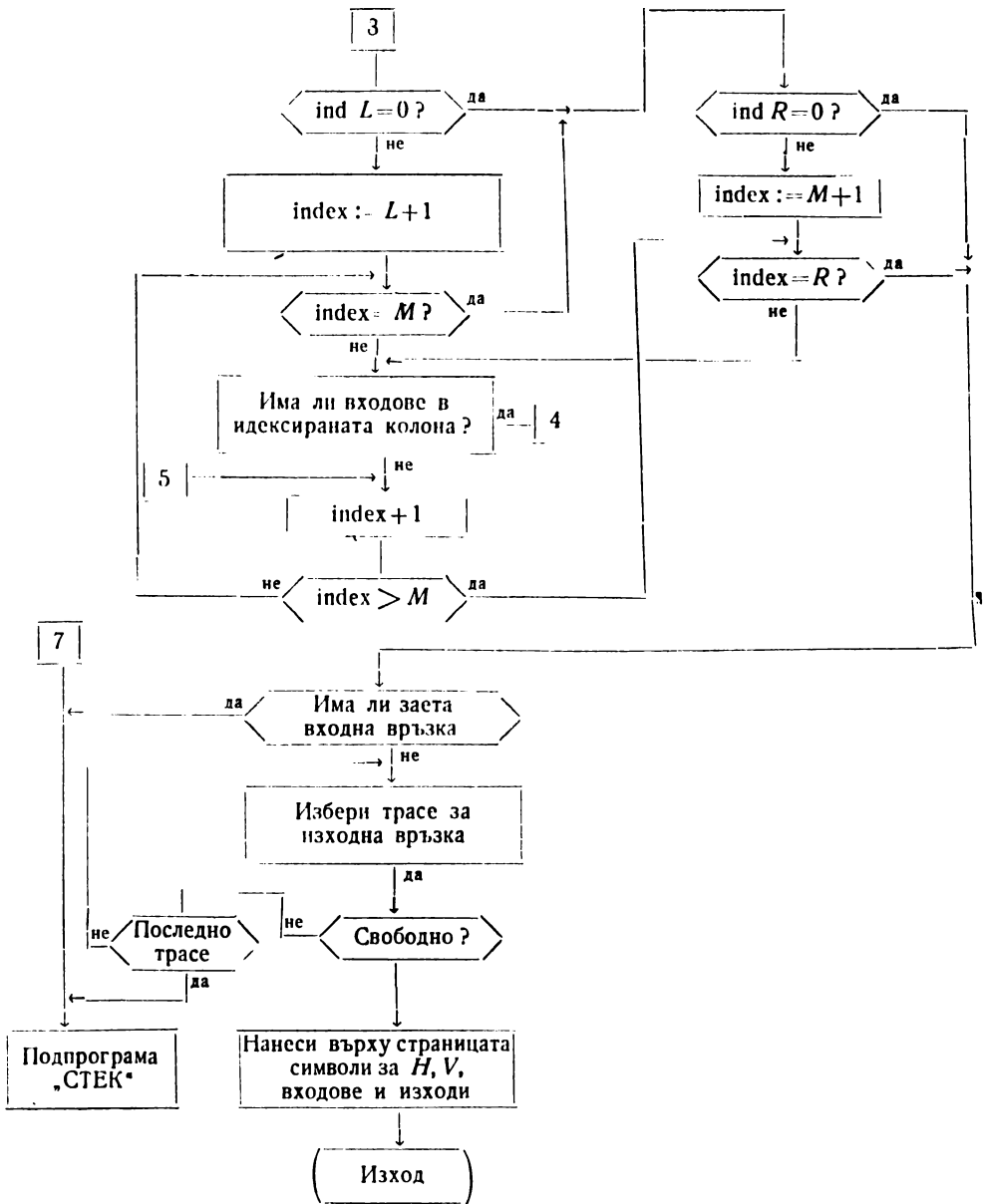
По експериментален път се установява, че съществуват три области във функцията, която описва зависимостта между дължината на връзките (S) и положението на N линия (номера на полето): строго намаляваща, зона на минимум и строго нарастваща. При конкретната конфигурация широчината на зоната на минимума е променлива, а така също е възможно отсъствието на първата или третата област.

Програмата, използвайки описващия вектор, проверява полетата от горе на долу. Когато се намери поле, свободно в границите от L до M за HL или от M до R за HR (в тези граници на полето не се среща ЛБ от страницата или е възможно използването на специално трасе), се определя дължината S на вертикалните участъци на връзките, която би се получила, ако се прекара съответната N линия в това поле. Процесът на изследване за дадена N линия се преустановява, когато определената дължина S_i е равна или по-голяма от определената за предшестващото свободно поле S_{i-1} .



Фиг. 6а





Фиг. 6в

Избраното положение на дадена H линия се фиксира чрез стойностите V_{\min} и V_{\max} . Ако приетата H линия минава по специално трасе, се налага корекция в описващия вектор, с което се избягва повторното пресмятане на S за същото трасе при трасиране връзките на другите ЗБ. След избиране положението на H линии програмата определя V линии за всяка междина, в която има входове. (Прието е, че всички входове на ЛБ, които трябва да се свържат в една междина, използват една вер-

тикална линия.) Това се постига чрез определяне на стойностите A_{\min} и A_{\max} за всяка V линия. С това се завършва първият етап — определяне на идеалното трасе. При него за всяка H и V линия са избрани централните трасета (№ 1) в съответните полета и междини (фиг. 5) (освен за случаите на „специални трасета“).

Идеалното трасе може да се определи и аналитично, но това е неудобно, тъй като се затруднява съгласуването на избраното оптимално трасе със заетостта на полетата.

При реализиране на втория етап програмата търси свободни трасета за определените V и H линии в съответстващите им междини и полета, като отчита съществуващите вече връзки. За всяка линия се проверява дали в определените граници (от A_{\min} до A_{\max} или от B_{\min} до B_{\max}) съществува трасе, което не е било използвано. Проверката започва от централното трасе (№ 1) и следва реда, показан на фиг. 5 (нарастващите номера на трасетата). Намереното свободно положение се фиксира чрез нови стойности на A_{\min} и A_{\max} , съответно B_{\min} и B_{\max} . Процесът на съвместване се извършва в следния ред: VL, HL, VM, HR, VR, междинни (фиг. 2) вертикални линии. Този ред на следване поставя в по-особено положение трите главни вертикални линии (VL, VM и VR) по отношение на междинните. Това се определя от обстоятелството, че при търсенето на свободно трасе за главните вертикали може да се наложи повторна проверка за линиите, за които вече е било намерено свободно място. Действително да приемем, че при трасирането на VR (фиг. 5) се е наложило тя да бъде изместена надясно ($\alpha=1$ е признак за изместване на вертикалите). Това удължава HR (увеличава B_{\max}) и налага нова проверка за най-дясната ѝ част. Ако тази проверка е неудовлетворителна, трябва да се намери ново положение за HR. Това ново положение може да удължи съответно VM (намаляване на A_{\min} или увеличаване на A_{\max}) и да наложи нова проверка за тази линия. Съответно едно евентуално ново положение на VM надясно от предишното би могло да предизвика нова проверка за HL, а тя от своя страна аналогично на HR да коригира A_{\min} или A_{\max} за VL, което да върне процеса за проверка в началото. Подобно удължаване на проверката не може да се предизвика от изменения на положението на междинните вертикали.

След като се намерят свободни трасета за вертикалните и хоризонталните линии, прекарват се връзките между входовете, подлежащи на свързване, и съответните вертикални линии (входни връзки), както и между изхода на ЗБ и VM (изходна връзка). В случай, че разглежданият ЗБ има само един изход, предвижда се възможност за изместване на изходната връзка в рамките на блока.

Накрая програмата нанася в масива страница кодовете на избраните символи за изчертаване на линиите.

В определени етапи при изпълнение на алгоритъма е възможно да се получат ситуации, неразрешими със средствата, описани по-горе. Такива ситуации са: заетост на всички полета в желаните граници, заетост на всички трасета от A_{\min} до A_{\max} за някоя междина, или от B_{\min} до B_{\max} за някое поле, заетост на трасето на някоя входна връзка, невъзможност за трасиране на изходната връзка. В тези случаи програмата предвижда активизиране на допълнителни програмни средства. Ако всички

трасета в някое поле са заети в границите от V_{\min} до V_{\max} , програмата блокира избирането на това поле и предава управлението на блока за определяне на оптимално положение на N линии. При повторното преминаване през този блок блокираното поле не съществува в списъка на възможните. Експериментите потвърждават, че новото положение на N линия няма да нарушава изискванията на критерия за оптималност поради съществуването на зона на минимум. При останалите от изброените по-горе ситуации се възбужда подпрограмата, която условно ще наричаме стек. Целта е да се смени редът на изчертаване на връзките на отделните ЗВ. Експериментите показаха, че съществува определена несъвместимост между някои ЗВ и че тя може да се преодолее само при спазване на определен ред на изчертаване. Стектът се стреми да осигури този ред чрез използване на един прост алгоритъм: всички линии върху страницата се „изтриват“; текущият ЗВ, чиито връзки не са били реализирани, се поставя на първо място и трасирането започва отново. Ако стекът не може да осигури търсения ред, стига се до цикличност в обръщението към него. В този случай трябва да се задействуват помощни алгоритмични средства: друг тип стекова организация, разпадане на хоризонтални линии, прекарване на вертикални линии и в колоните, заемани от ЛВ (при положение, че те са свободни) и т. н.

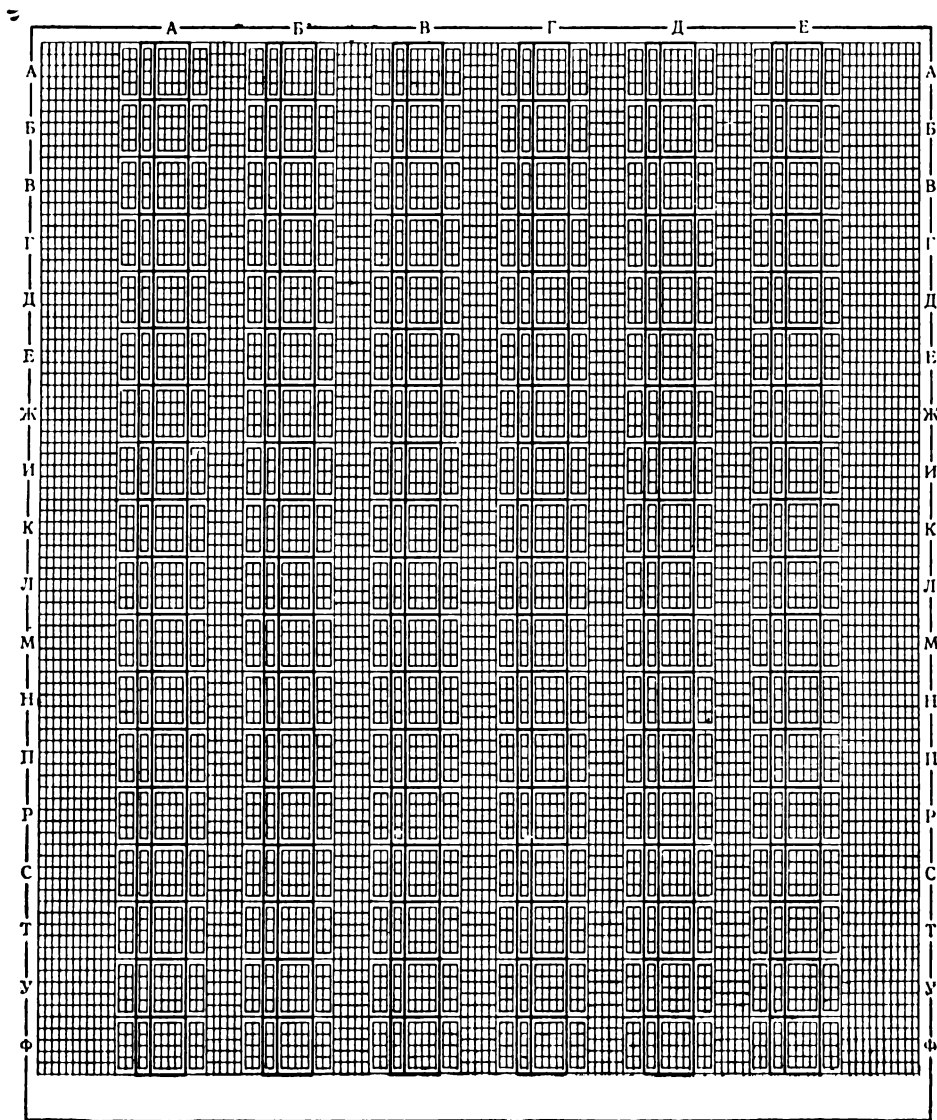
След като всички връзки на схемата бъдат трасирани, започва заключителният етап на алгоритъма, който отпечатва масива страница ред по ред на устройството за широк печат.

ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

Описаният алгоритъм и програмната система бяха реализирани на машина „МИНСК-22“. Програмата позволява чертането на функционални схеми, съставени от логически блокове, заемащи не повече от 108 елементарни графични клетки. Цялата схема се разполага върху поле, съставено от 94 реда. Всеки ред съдържа 126 символа. Във възприетата графична мрежа (фиг. 7) съществуват 6 колони, в които е възможно разполагането на логически елементи, и 7 междини, позволяващи очертаването на връзки (всяка от тях притежава 6 вертикални трасета). Съществуват 18 хоризонтални полета, във всяко от които са определени по 4 хоризонтални трасета, като се допуска използването на 18 специални трасета.

Контурите на логическите блокове се чертаят с помощта на символите $[$, $]$, и \langle с помощта на символа \rangle се означава инверсен изход. На всеки вход и изход на логически блок се присвояват по два символа, описващи поредните номера на изводите от интегралната схема, в която е разположен. В процеса на началното чертане на схемата тези номера не се поставят. Тяхното определяне се извършва в следващите етапи на работата по проектираното устройство. Ето защо в паметта на машината е предвидено място за нанасянето им. Очертаването на вертикалните участъци от връзките между елементите се извършва с помощта на символа $;$, а на хоризонталните със символа $-$. На местата на преход от вертикална към хоризонтална линия се поставя символът $*$. Електрическите съединения между отделните участъци се означават със символа 0 .

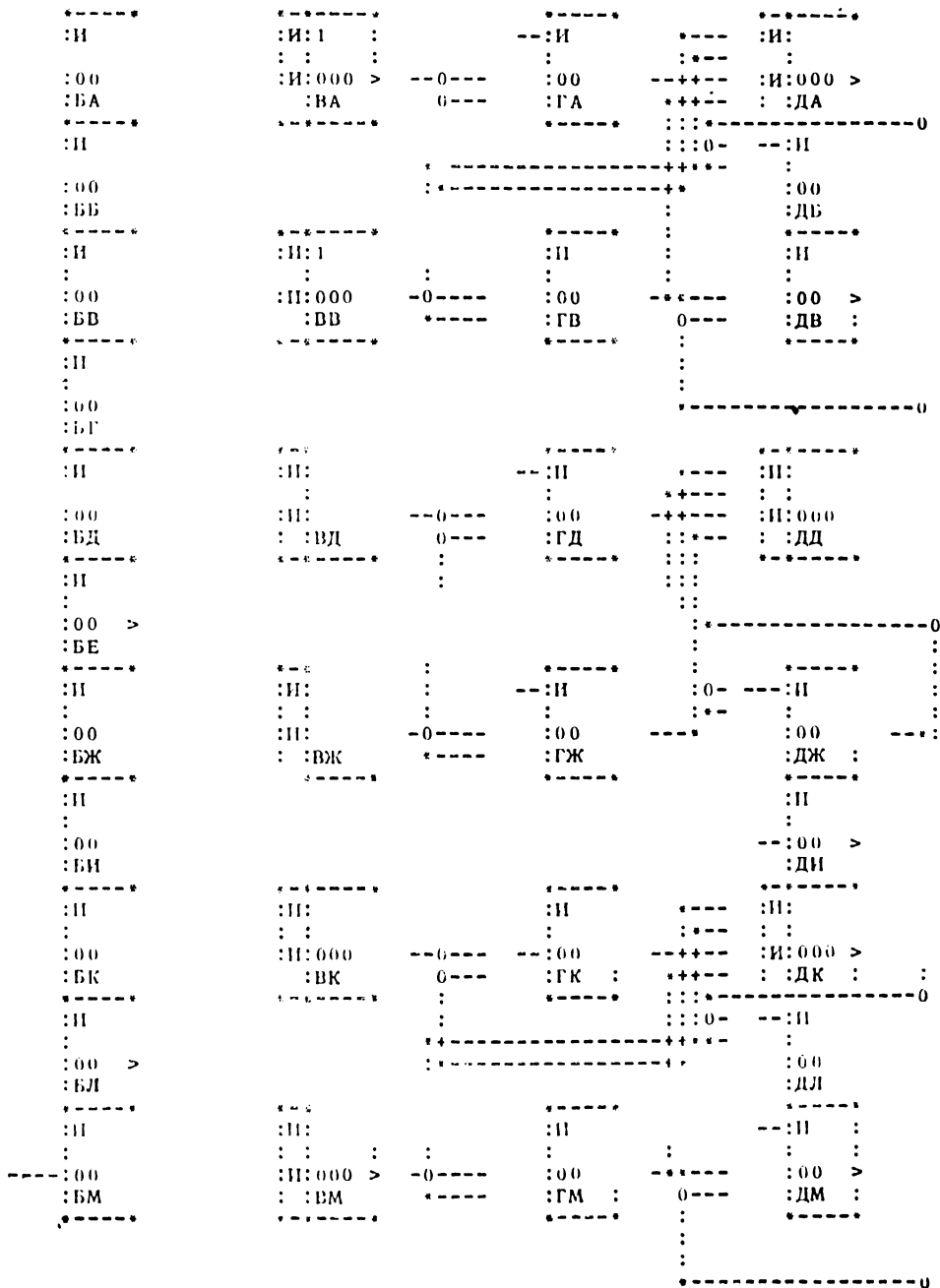
Допуска се възможност за трасиране на връзки между елементи, разположени на различни страници. Връзките, входни за дадена страница, се разполагат в лявата ѝ част, а изходните — в дясната. Тези връзки се идентифицират със съответните символи, като за входните връзки сим-



Фиг. 7

волите се намират отляво на първия символ на линията, а за изходните — отдясно на последния символ.

На фиг. 8 е показана конкретна схема, начертана от използваната програма.



Фиг. 8

ЗАКЛЮЧЕНИЕ

Описаната система за автоматично чертане на функционални схеми дава възможност за бързо получаване на стандартни документи при проектирането на логически устройства. При това системата е построена на принципа на постепенно въвеждане на по-мощни програмни средства, ако алгоритъмът не постигне разрешение при първото трасиране, което, както показаха многобройните експерименти, е изключително рядък случай. Наличността на подобна система облекчава много инженера конструктор по отношение на оперативното съставяне на логическите схеми и тяхното усъвършенстване. Освен това САЧ позволява лесно документирание при различни варианти на схемите, използвани от потребителите.

Включването на САЧ към система за автоматично проектиране ще даде възможност да се нанесат в графичната мрежа допълнителни данни в ЛБ за разположението на интегрални схеми по панели, платки и т. н., което е голямо удобство за експлоатационния персонал. При работа програмата класифицира и отбелязва използваните ЛБ, което позволява да се получат сведения за техническа спецификация на използваните интегрални елементи.

Описаната система е задължителна съставна част от системите за автоматично конструиране и документирание на ЦЕИМ.

ЛИТЕРАТУРА

1. IBM Data Processing Application. Three Dimensional Placement and Routing. New York, 1963.
2. Friedman, T. D. Methods Used in an Automatic Logic Design Generator. IEEE Transaction on computers, V. C. 18, No. 7 1969.

Постъпили на 7. IV 1970 г.

СИСТЕМА АВТОМАТИЧЕСКОГО ЧЕРЧЕНИЯ ФУНКЦИОНАЛЬНЫХ СХЕМ

С. Сребрев, Т. Величков, В. Лазаров и К. Боянов

(Резюме)

Рассматривается система автоматического черчения функциональных схем (САЧ), при помощи которой возможно получение стандартных изображений рабочих чертежей конструктора при оптимальной трассировке соединяющих линий.

Подключение САЧ к системе автоматического проектирования даст возможность нанести в графической сети дополнительные данные логических блоков о расположении интегральных схем на панелях, платах и т. п., что является большим удобством для эксплуатационного персонала.

Описанная система является обязательной составной частью систем автоматического конструирования и документирования цифровых электронных вычислительных машин.

A SYSTEM FOR AUTOMATIC DRAWING OF FUNCTIONAL CIRCUITS

S. Srebrev, T. Veličkov, V. Lazarov and K. Bojanov

(Summary)

A system for automatic drawing of functional circuits enabling the creation of standard images of the designer's working drawings by achieving an optimal tracing of the lines of connection is considered.

The system under discussion added to the system for automatic design will help us to give the supplementary data for the allocation of the integrated circuits on panels, plates etc. in the logical blocks on the graphic network which will be a great facility for the maintenance personnel.

The system described appears to be an obligatory component of the systems for automatic design and documentation of computers.