

BERICHTE ZUR ELEKTRONISCHEN DATENVERARBEITUNG (1)

## **Rechenzentrum SEL Stuttgart**

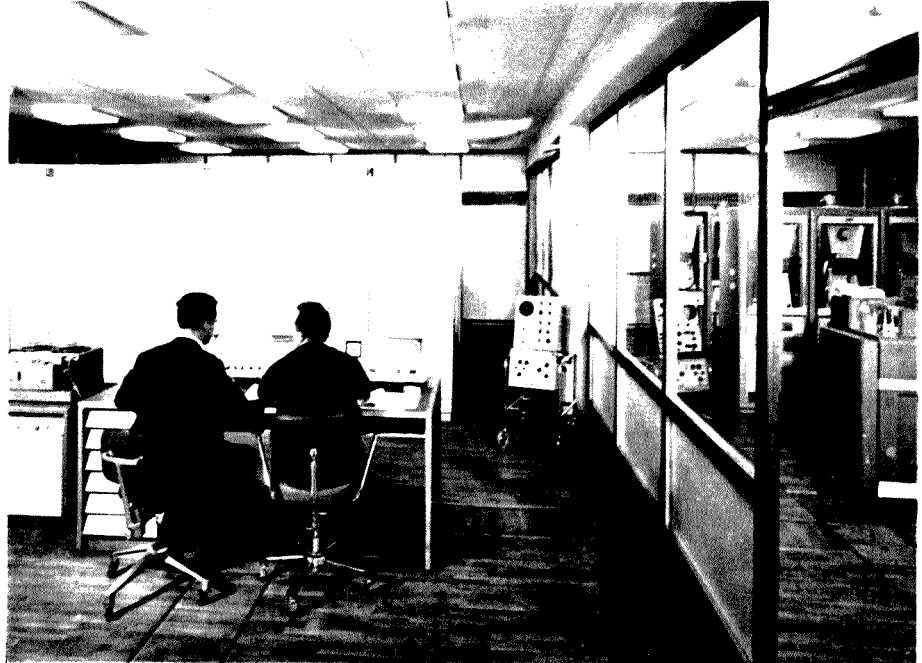
Von H. Härtl, E. Dachtler, K. Köberle, T. Lutz, G. Jung · Stuttgart

---

Sonderdruck aus SEL-Nachrichten 10 (1962), Heft 1, Seite 1...17







## Rechenzentrum SEL Stuttgart

Von H. Härtl · Stuttgart

DK 681.14-523:658

### 1. Einleitung

In den letzten zehn Jahren hat sich die Anwendung von Elektronenrechnern für technisch-wissenschaftliche, kaufmännische und betriebswirtschaftliche Aufgaben beinahe explosionsartig entwickelt. Grob geschätzt, verzehnfacht sich in den westlichen Ländern die Anzahl der ausgeführten Maschinenoperationen etwa alle drei Jahre. Die Entwicklung neuer, schnellerer und zweckmäßigerer Rechner macht so große Fortschritte, daß die Verbesserung und Erweiterung der Anwendung kaum damit Schritt zu halten vermag. Bekanntlich kann ein Elektronenrechner jede Aufgabe, die überhaupt logisch und rationell beschreibbar ist, ausführen. Gemessen an diesen Möglichkeiten steckt die Programmierung und Anwendung von Elektronenrechnern immer noch in den Kinderschuhen und man darf auf diesem Gebiet noch beträchtliche Fortschritte erwarten.

Ein mittlerer oder großer Rechner mit genügender Speicherkapazität stellt immer noch ein Millionenobjekt dar; dazu kommen noch hohe Programmierkosten. Es besteht deshalb oft die Gefahr, daß vor lauter Begeisterung über die praktisch unbegrenzten Anwendungsmöglichkeiten die Frage der Rentabilität etwas zu kurz kommt. Auf der anderen Seite muß aber eine große Firma mit vielseitigen Fertigungsaufgaben für die mannigfachen Aufgaben in Entwicklung, Verwaltung und

Fertigung Elektronenrechner einsetzen, wenn sie auf lange Sicht konkurrenzfähig bleiben will. Das trifft auch auf die SEL zu, die verschiedenartige und oft sehr komplizierte Geräte für die ganze Nachrichtentechnik erzeugt.

Anfang 1958 begannen die ersten Vorbereitungen für das Rechenzentrum. Dem Test und Lauf der ersten Programme diente zunächst das Labormodell des ER 56 beim Informatikwerk. Zwei Jahre später wurde die eigene Maschine aufgestellt. Im Folgenden sei kurz über Maschinenausstattung und Organisation des Rechenzentrums sowie über die Wirtschaftlichkeit und die bisherigen Erfahrungen berichtet.

### 2. Maschinenausstattung

Das Rechenzentrum verfügt über einen vom Informatikwerk der SEL gefertigten Elektronischen Rechenautomat ER 56 [1] mit erweiterter Ausstattung. Als Arbeitsspeicher enthält die Anlage zur Zeit sechs Ferritkern-Teilspeicher und zwar vier für 1000 Wörter und zwei für 200 Wörter, wobei ein Wort 7 Dezimalziffern bzw. 35 bit entspricht. Demnächst werden noch drei weitere 1000-Wort-Speicher eingebaut. Als Ergänzungsspeicher dienen zur Zeit eine Magnettrommel mit einer Kapazität von 12000 Wörtern und acht Magnetbandgeräte mit je 2000000 Wörtern. Bis zum Endausbau kommen noch zwei Magnettrommeln und vier Magnetband-

geräte hinzu. Zum Ansprechen der Magnetbandgeräte sind drei Steuereinheiten vorhanden. Die Ein- und Ausgabegeräte für Lochstreifen arbeiten mit einer Geschwindigkeit von 400 bzw. 50 Zeichen/s, für Lochkarten mit 400 bzw. 100 Karten/min, während der angeschlossene Drucker 15 Zeilen/s ausschreiben kann. Das Rechenwerk enthält eine Fest- und Gleitkommaarithmetik. Der elektronische Koordinatenschalter des ER 56 verbindet – vom jeweiligen Rechenprogramm durch das Kommandowerk gesteuert – die einzelnen Arbeitsteilspeicher nach Bedarf mit dem Rechenwerk, den Ergänzungsspeichern sowie den Ein- und Ausgabewerken, so daß neben der eigentlichen Rechnung gleichzeitig und unabhängig voneinander Übertragungen von und zu den Trommel- und Bandspeichern sowie verschiedene Ein- und Ausgabeoperationen ablaufen können. Die angegebene Ausstattung macht diese Anlage sowohl für technisch-wissenschaftliche als auch für kommerzielle Aufgaben sehr gut geeignet.

Der ausschließlich mit Halbleitern bestückte Rechner enthält in der jetzigen Ausbaustufe etwa 30 000 Transistoren und 50 000 Dioden. Seine durchschnittliche Arbeitsgeschwindigkeit beträgt 2000 Operationen/s. Ersatzgeräte und eine große Anzahl von Ersatz-Steckeinheiten erleichtern den vier Wartungsleuten des Rechenzentrums das routinemäßige Warten der Maschine und das Beheben von Störungen. Die Maschine läuft durchschnittlich 20 Stunden am Tag; ihre Ausfallzeit durch unvermeidliche Störungen beträgt rund 5% der Gesamtlaufzeit.

### 3. Aufgaben und Organisation

Im Laufe der Zeit soll das Rechenzentrum alle wirtschaftlich mit einem Elektronenrechner zu bewältigenden Probleme und Aufgaben aus den Entwicklungs-, Verwaltungs- und Fertigungsbereichen der SEL übernehmen. Dazu steht dort eine Programmiergruppe von derzeit 17 Personen zur Verfügung, die allmählich auf etwa 25 Personen erhöht werden muß. Dieser Gruppe gehören größtenteils Vollakademiker an und zwar Mathematiker, Ingenieure, Physiker und Wirtschaftsingenieure, die als Berufsprogrammierer alle Möglichkeiten der Maschine genau kennen und durchschnittlich zwei Jahre Programmiererfahrung besitzen.

Die Leistung der teuren Maschine wird wesentlich durch die Güte und Wirksamkeit der laufenden Programme, also durch die Leistung der Programmierer bestimmt. In einem Rechenzentrum sind deshalb die besten und damit teuersten Kräfte relativ am billigsten und wirtschaftlichsten. Eine solide mathematisch-naturwissenschaftliche Ausbildung, verbunden mit persönlicher Zuverlässigkeit und Exaktheit des Denkens, erscheint hier am wirkungsvollsten. Der ideale Programmierer muß ein erstklassiger Spezialist auf dem Programmiergebiet sein, aber zugleich auch einen umfassenden Überblick über Technik, Verwaltung und Betriebswirtschaft haben.

Das SEL-Rechenzentrum arbeitet überwiegend als Dienstleistungsbetrieb, d. h. anfallende Programmier- und Maschinenkosten gehen zu Lasten der Auftraggeber aus Entwicklung, Verwaltung und Fertigung. Nur einige grundsätzliche Untersuchungen auf dem Gebiet der Programmforschung, mit denen man gerechterweise nicht irgendeine Stelle belasten kann, werden aus einem besonderen Fond bezahlt. Der Gebührensatz für die Programmier- und die Maschinenstunde ist so bemessen, daß er die Maschinenamortisation, die Wartungskosten sowie die Nebenkosten für Strom, Raum usw. deckt. Diese Regelung gewährleistet, daß im Laufe der Zeit automatisch alle Arbeiten, die sich heute noch auf andere Weise wirtschaftlicher erledigen lassen, dem Rechner ferngehalten werden.

Das Erstellen eines Programms erfolgt in enger Zusammenarbeit mit dem Auftraggeber, der die anfallenden Kosten trägt und deshalb auch die letzte Entscheidung darüber hat, was gemacht werden soll. In der Regel versucht man zunächst, das Problem roh abzugrenzen und damit zu einer groben Abschätzung der entstehenden Programmier- und Maschinenkosten zu gelangen. Entscheidet man sich daraufhin, das Programm zu erstellen, weil entweder die Kosten relativ niedrig sind oder sonstige wichtige Gründe – z. B. Zeitgewinn oder Arbeitskräftemangel – den Ausschlag geben, so beginnt eine detaillierte Absprache zwischen Auftraggeber und federführendem Programmierer im Rechenzentrum. Sie kann je nach Umfang der Aufgabe Stunden, Tage oder Monate dauern. Als Ergebnis dieser Besprechungen entsteht dann eine allgemeinverständliche, detaillierte Beschreibung des Problems und der Aufgabenstellung, die – vom Auftraggeber unterschrieben – dem Rechenzentrum als Grundlage für die Programmplanung und Programmierung dient. Das Rechenzentrum verpflichtet sich, daß das erstellte Programm genau die festgelegten Arbeiten ausführt. Der Auftraggeber liefert die Eingabedaten sowie Daten für Testbeispiele und wertet die Ergebnisse. Befriedigen die Ergebnisse aus den Test- und Probeläufen, so werden die vollständigen Unterlagen in die Programmbibliothek eingereicht und das Programm gehört zu den laufenden Routineprogrammen bis sich Ergänzungen, Änderungen oder eine vollständige Umstellung als notwendig erweisen.

Ein offener Betrieb des Rechenzentrums, bei dem jede auftraggebende Abteilung ihre eigenen Programmierer hat, dürfte nicht so wirksam und wirtschaftlich sein wie ein geschlossener Betrieb mit in einer Gruppe zusammengefaßten Berufsprogrammierern, es sei denn, die Vorbereitungszeit vor Aufstellen der Maschine war zu kurz bemessen, um den Rechner sogleich voll auslasten zu können. Die bei der Zusammenarbeit zwischen Rechenzentrum und Auftraggeber manchmal geäußerte Befürchtung, daß der Einfluß bestimmter Abteilungen und Personen beträchtlich vermindert werde, weil schließlich der Elektronenrechner alle wesentlichen Aufgaben löse, hat sich bisher nicht bestätigt. Vielmehr zeigte sich, daß bei intensiver Zusammenarbeit mit dem Rechenzentrum die Leistungsfähigkeit und damit der Einfluß der betreffenden Abteilungen beträchtlich wuchs, zumal bei entsprechender Umorganisation die guten Mitarbeiter Zeit für schöpferische Denkarbeit gewinnen, die der Rechner heute noch nicht leisten kann.

### 4. Technisch-wissenschaftliche und kommerzielle Programme

Technisch-wissenschaftliche Programme werden fast ausschließlich für die Laboratorien und Entwicklungsabteilungen erstellt. Häufig teilt man sie in die drei folgenden Gruppen [2] ein:

#### a) Analyseprogramme

Dabei ist die Schaltung, die Anordnung oder das ganze Gerät gegeben und gesucht sind die Eigenschaften. Jede Messung an einem Gerät ist ein solcher Analysevorgang. Das SEL-Rechenzentrum verfügt über verschiedene Programme zur Analyse von Vierpolen, logischen Schaltungen, Antennen oder allgemeinen aktiven oder passiven Netzwerken. Durch Variieren der Parameter und anschließende Analyse kann man mit dem Elektronenrechner schnell die Einflußgrößen der einzelnen Elemente sowie ihre zulässigen Toleranzen bestimmen oder zielgerichtet die Parameter ändern, um neue geforderte Eigenschaften zu erzwingen.

Wird diese zielgerichtete Änderung automatisch vom Rechner gesteuert, so gehört das Programm eigentlich schon zur Gruppe b.

#### b) Syntheseprogramme

Bei der Synthese sind die (zulässigen) Eigenschaften vorgeschrieben und die Schaltung, Anordnung oder das Gerät ist gesucht. Mit Ausnahme von sehr seltenen Sonderfällen gibt es dafür keine formelmäßig geschlossenen Lösungen, weil sie entweder nicht bekannt sind, oder weil sie meist nichtlineare Beziehungen höheren Grades enthalten, für die geschlossene Lösungen grundsätzlich nicht möglich sind. In allen solchen Fällen leisten die erwähnten Analyseprogramme, bei denen die Änderung der Parameter nach irgendeinem Gradientenverfahren oder einfach abhängig vom Erfolg automatisch vom Rechner gesteuert wird, sehr gute Dienste. Solche Programme finden sich z. B. in unseren Syntheseprogrammen für Trägerfrequenzfilter und Laufzeitverzerrer.

#### c) Simulationsprogramme

Bei solchen Programmen wird im Rechner ein geeignetes Modell einer technischen Anordnung simuliert und getestet. Beispiele dafür sind das Testen von Koppelanordnungen der Vermittlungstechnik durch einen mit Zufallszahlen erzeugten, künstlichen Fernsprecherverkehr [3] oder die Ermittlung der zulässigen Schaltelementetoleranzen mit Gauß- oder Rechteckverteilung bei logischen Schaltungen [4]. Die Lösung von Problemen mit „Monte-Carlo-Methoden“, d. h. durch Würfeln mit Zufallszahlen, erfordert für eine genügende statistische Sicherheit oft die Erzeugung sehr großer Mengen von Zufallszahlen und ist dann ziemlich zeitraubend. Sie werden deshalb in der Regel nur dort angewendet, wo man keinen anderen Lösungsweg finden kann.

Als kommerzielle Programme gelten hier alle Programme, die mit Aufgaben aus dem kaufmännischen und volkswirtschaftlichen Bereich zu tun haben. Die Grenze zu den technisch-wissenschaftlichen Aufgaben ist fließend. Beide Gebiete enthalten viele gemeinsame Teile. Bei gegebener Aufgabenstellung weist die Arbeit der Programmplanung und Programmierung prinzipiell keine Unterschiede auf. Für kommerzielle Probleme ist in der Regel die Menge der Ein- und Ausgabedaten größer, der eigentliche Lösungsweg einfacher und das Programm enthält mehr Organisationsbefehle.

Die Aufgabe, eine gegebene Datenmenge mit einer gegebenen Speichergruppierung auf möglichst einfache Weise nach einem bestimmten Kriterium zu sortieren, gehört eigentlich in den Bereich der Mathematik. Die Erfahrung hat gezeigt, daß hervorragende Programmierer für technisch-wissenschaftliche Aufgaben auch immer hervorragende Arbeit leisten bei der Programmierung kommerzieller Aufgaben. Deshalb arbeiten die Programmierer des SEL-Rechenzentrums auf beiden Gebieten.

Während für technische Aufgaben seit fast hundert Jahren die Theorie ausgezeichnete Hilfsmittel liefert, wird im kommerziellen Bereich erst jetzt durch die Anwendung elektronischer Rechenmaschinen eine Theoriebildung und Methodenforschung angeregt. Die kommerziellen Programme kann man zweckmäßig einteilen in

1. „statische Programme“, bei denen eine gegebene Datenmenge nach bestimmten, gegebenen Regeln verarbeitet wird. Dazu gehören z. B. die in unserem Rechenzentrum laufenden Programme für Vertriebsstatistik, Lagerbuchhaltung, Inventur, Werksherstellkosten [5] und Lohnabrechnung.

2. „dynamische Programme“, bei welchen die Art und Anzahl der laufenden Eingabedaten den jeweiligen Programmablauf steuern. Dazu gehört z. B. ein Programm für optimale Lagerhaltung. Solche Programme sind sehr nützlich für Steuerungsaufgaben, wie etwa Produktionssteuerung. Hierbei wird jedoch eine Rückkopplung wirksam, und es können unter Umständen Instabilitäten ähnlich wie bei Regelvorgängen auftreten.

3. Modellprogramme, mit denen man betriebswirtschaftliche Zusammenhänge an (meist linearen) Modellen untersucht. Das wichtigste Hilfsmittel dafür ist das lineare Dispositionsmodell (linear programming). Mit Hilfe eines solchen, bereits erstellten Programms werden zur Zeit die Anwendungsmöglichkeiten auf verschiedenen Gebieten untersucht [6]. Selbst wenn das Modell absolut mit der Wirklichkeit nicht immer exakt übereinstimmt, so gewinnt man doch durch die Ermittlung der relativen Einflußgrößen bestimmter Faktoren wertvolle Einsichten in die allgemeinen Zusammenhänge.

Bei der elektronischen Bearbeitung kommerzieller Probleme wird meist eine integrierende Datenverarbeitung empfohlen, worunter man die Zusammenfassung aller Aufgaben vom Vertriebsauftrag bis zum Verkauf der Erzeugnisse in einem einzigen optimalen Programm versteht. Für einen großen Betrieb mit komplizierter Struktur bedeutet das jedoch ein völliges Umstellen des gesamten Organisationsablaufs; außerdem ist es fast unerreichbar, alle Details des Gesamtprogramms mit allen möglichen Pannen und Schwierigkeiten von vornherein zu übersehen. Um ein übermäßiges Risiko zu vermeiden, bleibt wohl nur der Weg, das Gesamtprogramm unter Verzicht auf ein optimales Programm so in Blöcke zu unterteilen, daß man jeden Block für sich nützlich anwenden kann und erst später die einzelnen Blöcke zusammenfügt. Dabei sollte man aber das Ideal der Integration immer im Auge behalten.

In der Mitte zwischen kommerziellen und technisch-wissenschaftlichen Programmen stehen die Programme zum Erstellen von technischen Unterlagen, wie Stücklisten, Verdrahtungsplänen, Legelisten [7] usw. Solche Programme sind sehr rentabel und werden in Zukunft als Verbindungsglieder zwischen Entwicklung und Fertigung eine besondere Bedeutung erlangen.

## 5. Wirtschaftlichkeit

Die Frage der Wirtschaftlichkeit eines Programms für den Elektronenrechner ist im allgemeinen sehr komplex und in den seltensten Fällen durch eine exakte Meßzahl oder genau in DM benennbar. Am einfachsten sind die Verhältnisse noch bei solchen technisch-wissenschaftlichen Berechnungen, die bisher mit Tischrechenmaschinen durchgeführt wurden, wie z. B. Netzwerk- und Filtersynthesen: bei Berechnung mit dem ER 56 ergibt sich eine 70...100fache Verbilligung. Die einmaligen Programmierkosten fallen bei dem häufigen Lauf solcher Programme nicht ins Gewicht. Dazu kommt noch der Vorteil, daß die Ergebnisse am gleichen Tag vorliegen können, während früher oft Wochen und Monate vergingen. In manchen Fällen kann durch diesen Zeitvorteil ein Programm selbst dann noch lohnend sein, wenn die Gesamtkosten gleich oder höher sind als bei der manuellen Bearbeitung. Bei vielen Problemen fehlt jedoch eine Vergleichsbasis, weil sie, wie z. B. das Testen von Koppelanordnungen mit künstlichem Fernsprecherverkehr, nur mit einem Elektronenrechner durchgeführt werden können. Die betreffende, mit den Kosten belastete Abteilung hat aber immer das beste Gefühl dafür,

wie lohnend für sie das Programm ist. Aus diesem Grunde stellt die volle Weiterbelastung aller Kosten ein sehr gutes Regulativ für die Wirtschaftlichkeit elektronischer Berechnungen dar.

Bei kommerziellen Programmen ist in der Regel die Anzahl der Ein- und Ausgabedaten riesengroß. Demnach sind bei der Verarbeitung die im Vergleich zur Elektronik langsamen mechanischen Geräte, wozu eigentlich auch die Bandgeräte gehören, wesentlich beteiligt. Dies führt dazu, daß sich kommerzielle Aufgaben mit dem Elektronenrechner durchschnittlich nur etwa fünfmal billiger lösen lassen, als die entsprechende Arbeit bei der Lochkartenstelle. Wenn außerdem auf Grund mangelhafter Organisation bei der Datenerfassung und Erstellung die Eingabe-Lochkarten viele Fehler enthalten, die entweder der Elektronenrechner grundsätzlich nicht zu erkennen vermag, oder die jedesmal zu einem Stop des Rechners führen, so kann auch ein sehr gut ausgearbeitetes Programm sehr schnell unwirtschaftlich werden. Allerdings ergibt sich bei der elektronischen Bearbeitung immer ein beträchtlicher Zeitgewinn.

Die Gesamrentabilität des Rechenzentrums einer Firma wie z. B. der SEL dürfte sich aber wohl immer sichern lassen, wenn ein genügender Grundstock von lohnenden technisch-wissenschaftlichen Programmen vorliegt, so daß Schritt für Schritt kommerzielle, mit größeren Risiken verbundene Programme übernommen werden können. Wenn dazu noch, wie beim ER 56, eine Parallelarbeit zwischen verschiedenen Programmen (time-sharing) möglich ist, die zwar einerseits höchste Anforderungen an Programmorganisation, Sicherheit der Maschine usw. stellt, auf der anderen Seite aber großen Nutzen bringen kann, ist bei entsprechenden Anstrengungen und enger Zusammenarbeit des Rechenzentrums mit allen beteiligten Stellen eine sehr gute Rentabilität gewährleistet.

## Rechenzentrum SEL Stuttgart

# Berechnung der Werkherstellkosten mit dem ER 56

Von E. Dachtler · Stuttgart

Das Endziel bei der elektronischen Bearbeitung kommerzieller Probleme ist immer eine integrierte Datenverarbeitung, die eine optimale Programmierung und einen optimalen Arbeitsablauf erlaubt. Da sie aber meist eine völlige Umorganisation des Betriebs erfordert, geht man zur Vermeidung eines zu großen Risikos schrittweise vor und übernimmt in sich abgeschlossene Bereiche auf datenverarbeitende Anlagen, ohne jedoch das Gesamtziel aus den Augen zu verlieren. Dabei räumt man ein, daß später Programmteile geändert werden müssen oder zeitlich nicht ganz optimal arbeiten. Letztlich kommen die bei der Erstfassung solcher Programme gesammelten Erfahrungen einer endgültigen Ausführung zugute.

Ein Teilgebiet der elektronischen Produktionssteuerung in obigem Sinne ist die dem betrieblichen Rechnungswesen zugehörige Ermittlung der Werkherstellkosten. Diese setzen sich aus den Kosten für das Fertigungsmaterial sowie aus den Fertigungslöhnen und den Produktionszuschlägen zusammen. Andererseits sind die Werkherstellkosten Ausgangspunkt für die Errechnung der Herstell- bzw. der Verkaufskosten. Nachdem die Planungsabteilung des betriebswirtschaftlichen Fabrikbereichs zusammen mit dem Rechenzentrum die notwendigen Vorarbeiten geleistet hatte, konnte ein Maschinen-

## 6. Ausblick

Die bisherige Erfahrung zeigt, daß fast alle Abteilungen, die einmal elektronische Berechnungen durchführen ließen, mit immer neuen Wünschen, Vorschlägen und Aufträgen an das Rechenzentrum herantreten. Besonders bei technisch-wissenschaftlichen Berechnungen besteht eine starke Tendenz, die Programme laufend zu erweitern und flexibler zu gestalten, während kommerzielle Programme in der Regel längere Zeit unverändert laufen. Wenn man z. B. im Trägerfrequenzgebiet heute Filter und zum Teil Umsetzer synthetisch berechnet, so dürften in Zukunft sehr wahrscheinlich ganze Geräte berechnet, aber auch Unterlagen für Vertrieb und Fertigung sowie die Planung und Steuerung der Produktion von Elektronenrechnern bearbeitet werden. Dazu werden sowohl beträchtliche Fortschritte bei der Entwicklung der Maschine als auch bei der Programmierung wesentlich beitragen.

### Schrifttumsnachweis

- [1] R. Basten und H.-J. Dreyer: Der Elektronische Rechenautomat ER 56. Elektron. Rechenanl. 1 (1951), H. 2, S. 60...67; SEL-Nachr. 7 (1959), H. 4, S. 199...208
- [2] G. Hässler: Editorial. Elektron. Rechenanl. 3 (1961), H. 5, S. 193
- [3] H. Wagner: Einfluß der Zubringerteilgruppen auf die Bemessung neuer Mischwähleranordnungen. SEL-Nachr. 9 (1961), H. 1, S. 70...72
- [4] T. Lutz: Über ein Programm zur Ermittlung des Einflusses von Schaltelementtoleranzen mit Hilfe einer Monte-Carlo-Methode. SEL-Nachr. 10 (1962), H. 1, S. 9...12
- [5] E. Dachtler: Berechnung der Werkherstellkosten mit dem ER 56. SEL-Nachr. 10 (1962), H. 1, S. 4...6
- [6] S. Vajda: Einführung in die Linearplanung und die Theorie der Spiele. Verlag R. Oldenbourg, Wien und München 1961
- [7] G. Jung: Erstellen von Verdrahtungsunterlagen mit dem ER 56. SEL-Nachr. 10 (1962), H. 1, S. 13...17

DK 681.14-523.8:518.5:657.472

programm für den Elektronischen Rechenautomaten ER 56 erstellt werden. Damit lassen sich für etwa 30...40% aller vorkommenden Bauteile die Werkherstellkosten berechnen. Dieses Programm wird seit etwa eineinhalb Jahren mit Erfolg verwendet.

Die Berechnung der Werkherstellkosten ist so organisiert, daß sich im wesentlichen vier Hauptbestandteile ergeben:

1. Der Lochkartensatz je Bauteil
2. Die Einkaufsverrechnungswerte mit Zuordnungskatalog
3. Der Schlüsselkatalog
4. Das Programm für den ER 56

Der Lochkartensatz je Bauteil ist ein Fertigungsplan in Lochkartenform, also eine Kombination aus Arbeitsgangbeschreibung und Stückliste. Er besteht aus einer Kopfkarte (Kartenart 2) und einer Ergänzungskarte (Kartenart 3); beide beinhalten auf das ganze jeweilige Bauteil bezogene Angaben, z. B. Fertigungsplannummer (Zeichnungsnummer), Planausgabe, Ausschußprozentatz und Streifenkonstante. Ferner gibt es je Fertigungsplan soviel Materialkarten (Kartenart 4), wie in dem betreffenden Bauteil verschiedene Mate-

rialien vorkommen; sie tragen außer der Fertigungsplannummer die Materialnummer, die Materialgruppe und die Mengeneinheit. Für jeden notwendigen Arbeitsgang ist außerdem eine Arbeitsgangkarte (Kartenart 5) vorhanden; sie enthält die Arbeitsgangnummer, die Kostenstelle, die Maschinengruppe, die Lohngruppe, die Einheit, die Vorgabezeit je Einheit sowie die Rüstzeit. Außerdem können noch Textkarten (Kartenart 7) vorkommen, die jedoch in diesem Zusammenhang nicht interessieren.

Die normalerweise vom Einkauf festgesetzten Einkaufsverrechnungswerte für Rohstoffe bzw. fremdbezogene Teile zieht man zur Materialbewertung der Eigenerzeugnisse heran. Da Einzelteilen fast ausschließlich solche

Der Schlüsselkatalog enthält alle vorkommenden Kostenstellen mit den zugehörigen Fertigungsgemeinkosten und den Funktionslohnsätzen, weiter die verschiedenen Lohngruppen mit den Lohnsätzen, die Maschinengruppen mit den Platzkostenfaktoren sowie außerdem Materialgemeinkosten-, Materialfunktionskosten- und Werkzeugkostensätze. Die Einkaufsverrechnungswerte mit dem Zuordnungskatalog und der Schlüsselkatalog sind auf Magnetband gespeichert, weil sie von dort bei Bedarf schnell in die Arbeitsspeicher gebracht werden können. Sie gelten für einen gewissen Zeitraum als gleichbleibend. Sollten sich jedoch die Einkaufsverrechnungswerte, z. B. wegen starker Schwankungen auf den Rohstoffmärkten, ändern, oder andere Lohnsätze einen neuen Schlüssel erfordern, so lassen sich diese Korrekturen mit einem vorhandenen Hilfsprogramm leicht durchführen. Das Programm kann solange unverändert bleiben, wie man die Berechnungsmethode beibehält.

Das Programm sei kurz an Hand eines Flußdiagramms (BILD 1) erläutert.

Der Vorlauf hat folgende Aufgaben: Suchen des in Frage kommenden Programms auf dem Magnetband, Überführung des Zuordnungskatalogs und des Programms in die Kernspeicher sowie der Einkaufsverrechnungswerte in die Trommelspeicher. Danach liest das Lochkarteneingabewerk die Lochkarten je Bauteil wechselweise in die Kernspeicher  $m_1$  und  $m_2$  ein. Hierbei wird jeweils der Inhalt der vorletzten Lochkarte verarbeitet. Dies hat den Vorteil, daß die Verarbeitungszeit wegen der Parallelarbeit des ER 56 vollständig in der Einlesezeit untergeht. Die Art der gerade verarbeiteten Lochkarte bestimmt den weiteren Ablauf. Die Kartenart 2 besagt, daß ein neues Bauteil zur Berechnung ansteht und die Berechnungsspeicher auf Null gestellt bzw. die neue Fertigungsplannummer, die Planausgabe sowie der Ausschußprozentsatz in diese übernommen werden müssen. Der Kartenart 3 entnimmt das Programm nur eine Streifenkonstante zur Lohnberechnung, dagegen der Kartenart 4 die Materialnummer zum Auffinden des zugehörigen Einkaufsverrechnungswerts. Dafür wird nach Aufsuchen der Trommeladresse im Zuordnungskatalog die entsprechende Anzahl von Trommelblöcken in den Kernspeicher gebracht und dort über eine Abfragekaskade der zugehörige Einkaufsverrechnungswert gefunden. Wenn dieser Wert fehlt, ändert sich die Prüffzahl. Ferner steuert die Materialkarte neben der Materialmenge auch die Mengeneinheit und die Materialgruppe bei. Daraus ergibt sich die Materialbewertung zu

$$\text{Materialkosten} = \frac{\text{Materialmenge}}{\text{Mengeneinheit}} \cdot \frac{\text{Preis}}{\text{Preiseinheit}}$$

Die Materialkosten werden im Berechnungsspeicher an der dafür vorgesehenen Stelle addiert.

Aus Kartenart 5, der Arbeitsgangkarte, übernimmt das Programm die Kostenstelle, die Arbeitsgangnummer, die Maschinengruppe, die Lohngruppe, die Vorgabezeit je Einheit, die Einheit und die Rüstzeit. Die Arbeitsgangnummer dient zugleich als Kriterium, ob der vorliegende Fertigungsplan vorläufige oder endgültige Daten enthält. Dies wird in einer Kennzahl vermerkt. Wenn eine indirekte Vorgabezeit vorliegt, findet eine Umrechnung dieser Zeit mit Hilfe der Streifenkonstante aus Kartenart 3 statt. Aus der Kostenstelle läßt sich entnehmen, ob eine - nur bei manchen Kostenstellen notwendige - Platzkostenrechnung vorzunehmen ist. In diesem Fall zieht das Programm auch noch die Maschinenzeit aus der Arbeitsgangkarte zur Berechnung heran. Nach dem Abfragen aller Unterscheidungsmerkmale werden die benötigten Lohn- bzw. Funktionslohn-, die Fertigungsgemein-

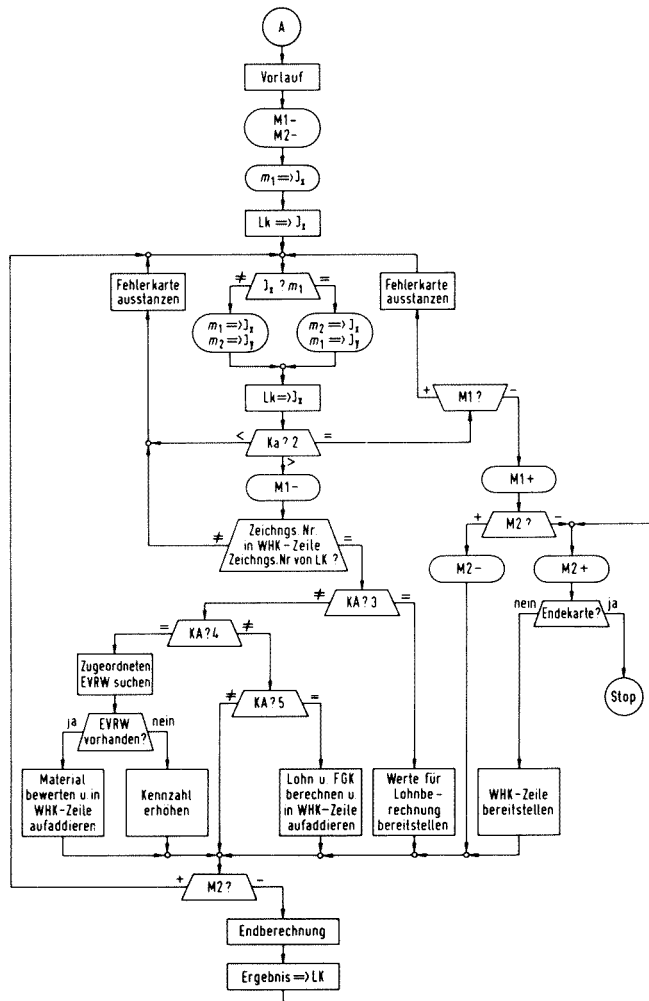


Bild 1 Flußdiagramm für die Berechnung der Werkerstellkosten

Werte zugrunde liegen, genügt es, alle vorkommenden Einkaufsverrechnungswerte für die Materialbewertung bereit zu haben. Wegen der großen Anzahl (es gibt einige tausend Einkaufsverrechnungswerte) kann man sie nicht mehr in Kernspeicher unterbringen. Deshalb wurden sie - in aufsteigender Reihenfolge nach Materialnummer sortiert - auf den Trommelspeicher gegeben. In den Kernspeicher selbst kommt nur ein Zuordnungskatalog, der die Materialnummer und die Trommeladresse jedes fünfzigsten Einkaufsverrechnungswerts ausweist. Mit dessen Hilfe läßt sich jeder Einkaufsverrechnungswert auf der Trommel schnell finden. Je Einkaufsverrechnungswert sind auf der Trommel die Materialnummer, der Preis je Mengeneinheit und die Preiseinheit gespeichert.

kosten- und Funktionsgemeinkostensätze aus dem Schlüsselkatalog geholt, danach die Lohn-, Funktionslohn-, Fertigungsgemein- und Funktionsgemeinkosten berechnet und diese schließlich in die vorgesehenen Speicherplätze addiert. Es gelten

$$\text{Lohn} = \frac{\text{Vorgabezeit je Einheit} \cdot \text{Lohnsatz} \cdot 100}{\text{Einheit}}$$

$$\text{Funktionsgemeinkosten FuGK} = \text{Satz} \cdot \text{Lohn}$$

$$\text{Funktionslohnkosten Fulok} = \text{Satz} \cdot \text{Lohn}$$

$$\text{Fertigungsgemeinkosten FGK} = \text{Satz} \cdot \text{Lohn}$$

Bei Platzkostenrechnung gilt:

$$\text{Fertigungsgemeinkosten} = \frac{\text{Platzkostenfaktor} \cdot \text{Maschinenzeit} \cdot 100}{\text{Einheit}}$$

Sämtliche Werte sind auf 100 Teile bezogen.

Textkarten mit Kartenart 6 und 7 bleiben unberücksichtigt. Sobald in der Folge eingelesener Karten wieder eine Kopfkarte ansteht, ist für den vorangegangenen Fertigungsplan die Endberechnung zu erstellen und eine Ergebniskarte zu stanzen. Bei der Endberechnung werden die noch fehlenden Materialgemein-, Materialfunktions- und Werkzeugkosten berechnet sowie alle Einzelwerte mit dem Ausschußprozentsatz beaufschlagt. Alle diese Werte ergeben als Summe die

Werkherstellkosten. In den einzelnen Lochfeldern der Ergebniskarte sind folgende Angaben enthalten:

1. Kartenart
2. Zeichnungsnummer
3. Planausgabe, Basisjahr, Materialart
4. Fertigungszeit
5. Prüfwahl
6. Fertigungsmaterialkosten
7. Materialgemeinkosten
8. Fertigungslohn
9. Fertigungsgemeinkosten
10. Werkzeugkosten
11. Materialfunktionskosten
12. Funktionslohn
13. Funktionsgemeinkosten
14. Werkherstellkosten

Das gesamte Rechnerprogramm umfaßt etwa 1000 Befehle. Außerdem sind noch zwei festverdrahtete Programme in Form von Stecktafeln für das Lochkartenein- und -ausgabewerk vorhanden. Die durchschnittliche Laufzeit beträgt für 100 Berechnungen bei etwa 20 Lochkarten je Fertigungsplan, also insgesamt rund 2000 Eingabe-Lochkarten, etwa 6...8 Minuten. Das ist praktisch die reine Einlesezeit, da die Berechnungen und das Ausstanzen der Ergebniskarten infolge interner Parallelarbeit zeitlich nicht in Erscheinung treten.

## Rechenzentrum SEL Stuttgart

# Auszählen und Auswerten von Zielbereich-Registrierungen in Fernsprechnetzen

Von K. Köberle · Stuttgart

DK 681.14-523.8:621.395.31

## 1. Einleitung

Um den Ausbau von Fernsprechnetzen nach wirtschaftlichen Gesichtspunkten planen zu können, muß man die Verkehrsbelastungen in den verschiedenen Richtungen kennen. Nur dann ist es möglich, die Leitungsbündel günstig zu verlegen und zu dimensionieren, so daß einerseits die einzelnen Leitungen genügend ausgenutzt sind und andererseits nicht unnötig viele Vermittlungsstellen und Leitungsabschnitte in den Verbindungszügen liegen.

Die Verkehrsbelastung eines Leitungsbündels wird normalerweise direkt mit der sogenannten Verkehrsmeßeinrichtung gemessen. Auf diese Weise läßt sich aber nur der Verkehr

bis zur nächsten Wahlstufe erfassen. Über die in diesem Verkehrswert enthaltenen Anteile nach bestimmten, im weiteren Verbindungszug liegenden Zielbereichen sagt diese Messung dagegen nichts aus. Deshalb wurde eine Zielbereichs-Registriereinrichtung\*) entwickelt; sie liefert einen Lochstreifen, der die Kennzahlen aller während der Registrierdauer gewählten Rufnummern enthält.

Der Lochstreifeninhalt läßt sich äußerst rationell mit Hilfe eines Digital-Rechenautomaten auswerten. Bevor das dazu notwendige Programm beschrieben wird, seien zunächst der Begriff „Zielbereich“ und die Art der Kennzahl-Registrierung – soweit zum Verständnis der Aufgabe erforderlich – erläutert.

## 2. Zielbereiche und Rufnummern-Registrierung

Ein vereinfachtes Beispiel soll die Aufteilung eines Fernsprechnetzes in Zielbereiche veranschaulichen. BILD 1 gibt schematisch ein Netz wieder, dessen Gesamtfläche in fünf Teilflächen zerlegt ist; jede von ihnen gliedert sich in mehrere Felder. In den Feldern stehen dreistellige Zahlen. Die erste Ziffer (2, 3, 4, 5 oder 6) kennzeichnet die Teilfläche, während die zweite Ziffer zwischen den Feldern einer Teilfläche unterscheidet. Das x an der dritten Stelle repräsentiert den hier uninteressanten Rest der Rufnummer. Alle Kennzahlen innerhalb eines Feldes haben also an erster und zweiter Stelle dieselben Ziffern. Das gleiche gilt sinngemäß für weitere, der Zielbereich-Kennzeichnung dienende Stellen bei noch feinerer Unterteilung des Netzes.

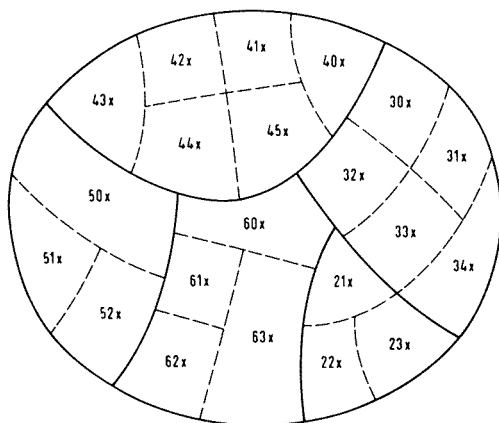


Bild 1 Vereinfachtes Beispiel der Aufteilung eines Fernsprechnetzes in Zielbereiche

\*) K. Linkeheil und G. Loewe: Zielbereich-Registriereinrichtung für Landesfernwahlnetze. SEL-Nachr. 9 (1961), H. 4, S. 230...231.



Die Zielbereich-Registriereinrichtung hält die gewählten Kennzahlen in einem Fünf-Kanal-Lochstreifen fest. Dabei sind die zehn Ziffern 0...9 im (2 aus 5)-Code dargestellt (BILD 2a). Da dieser Code nur zehn unterschiedliche Kombinationen zuläßt, erhebt sich die Frage, worin das jede Kennzahl abschließende Endezeichen bestehen soll. Um mehr unterscheidbare Zeichen zu gewinnen, wird jeder registrierten Kennzahlziffer eine 0 angehängt, so daß stets ein Ziffern-paar als Zeichen zu werten ist. Das Endezeichen kann nun irgendein Ziffern-paar sein, das an der zweiten Stelle eine von 0 abweichende Ziffer aufweist; man entschied sich für 88. BILD 2b zeigt einen Lochstreifenausschnitt mit zwei derartig zusammengesetzten vierstelligen Kennzahlen. Auch das Ende des Registrierstreifens markiert ein solches Ziffern-paar, und zwar die Zahl 99.

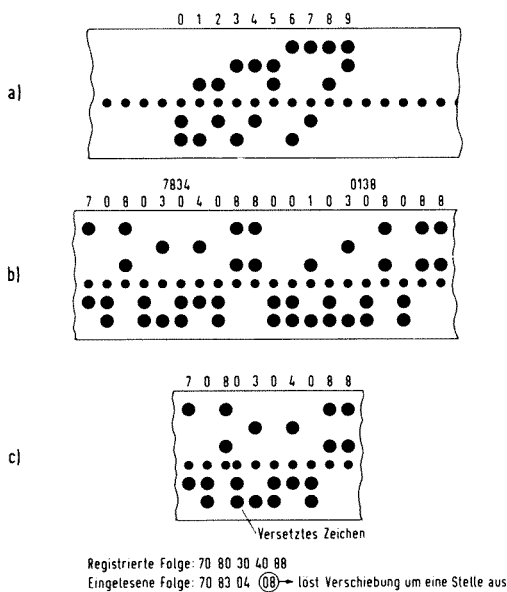


Bild 2 Lochstreifenabschnitte; Darstellung a) der Ziffern 0...9 im (2 aus 5)-Code, b) der registrierten Ziffern nach Anhängen einer 0 und Einfügen des Endezeichens und c) der Auswirkung einer versetzten Zeichenkombination auf den Lesevorgang

Die Anzahl der zu registrierenden Stellen kann man im Registriergerät festlegen. Soll die Registrierung sich beispielsweise auf fünf Stellen erstrecken, dann werden von jeder gewählten Rufnummer nur die ersten fünf Ziffern festgehalten; darauf folgt selbsttätig das Endezeichen. Wählt ein Teilnehmer weniger als fünf Stellen – sei es, daß er sich geirrt hat und deshalb den Wahlvorgang abbricht, sei es, daß er einen mit weniger als fünf Ziffern zu erreichenden Sonderdienst wünscht –, so erscheint das Endezeichen bereits nach der letzten gewählten Ziffer, eine Registrierung mit  $n$  Stellen bedeutet also, daß spätestens nach  $n$  Rufnummernziffern ein Endezeichen vorkommt.

### 3. Aufgabenstellung für das Programm

Beim Aufbau des Programms sind Forderungen verschiedener Seiten zu berücksichtigen. Zunächst liegen Wünsche von außen vor, das heißt von der Stelle, die Registrierstreifen an liefert und deren Auswertung erwartet. Des weiteren muß man die durch Eigenschaften der elektronischen Anlage bedingten Möglichkeiten und Einschränkungen beachten. Letztlich sollen Unregelmäßigkeiten auf dem angelieferten Lochstreifen nach Möglichkeit durch den Programmablauf selbst abgefangen werden.

Der in Abschnitt 2 beschriebene Registrierstreifen ist nach folgenden Regeln auszuzählen und auszuwerten:

#### a) Auszählung

1. Alle registrierten Kennzahlen werden zuerst auf ihre Stellenanzahl geprüft.
2. Die Kennzahlen mit  $n$  Stellen, wobei  $n$  der für die Registrierung festgelegten Stellenanzahl entspricht, sind dem durch ihre  $m$  ersten Ziffern bestimmten Zielbereich zuzuordnen. Die Stellenzahl  $m$  ist unter der Bedingung  $m \leq n$  frei wählbar; innerhalb eines Auszählprozesses bleibt sie dann unveränderlich und stellt den Regelfall dar. Existiert für eine vollständige Kennzahl kein Zielbereich, so fällt diese Aufzeichnung unter „Teilnehmerfehler“.
3. Bei Kennzahlen mit weniger als  $n$  Stellen wird an Hand einer Liste untersucht, ob es sich etwa um vollständig gewählte Rufnummern von Sonderdiensten (Feuerwehr, Notruf usw.) handelt. Wenn ja, dann ist nach a) 2) zu verfahren, wenn nein, diese Aufzeichnung – ohne sie zu zählen – auszuscheiden.
4. Neben der  $m$ -stelligen Zielbereichsermittlung nach a) 2) ist für einige besonders gekennzeichnete Fälle eine genauere Auflösung des Zielbereichs durch  $m'$  Stellen ( $m' > m$ ) notwendig.

#### b) Auswertung

Die Zielbereiche werden in mehreren Gruppen, z. B. Sonderdienste, Ortsverkehr und Fernverkehr, zusammengefaßt. Für jede dieser Gruppen ist eine mittlere Belegungsdauer bekannt. Nun läßt sich für alle Zielbereiche der Anteil an den Belegungen sowie – unter Berücksichtigung der für die Gruppe geltenden mittleren Belegungsdauer – am Gesamtverkehr berechnen.

Zur Bearbeitung der Aufgabe wird der Elektronische Rechenautomat ER 56 verwendet. Da er mehrere Prozesse gleichzeitig durchführen kann, ist es möglich, den bereits eingelesenen Teil des Registrierstreifens auszuzählen, während der nächste Teil gerade eingelesen wird. Der Vorteil dieses Ablaufs zeigt sich in einer nicht unerheblichen Zeitersparnis.

Die auszuwertenden Lochstreifen weisen mitunter gewisse Unregelmäßigkeiten in Form eines zu geringen Abstands der Transportlöcher und damit der Einzelzeichen auf. Das kann zum Überspringen eines Zeichens, also zum Verschieben der Information führen (BILD 2c). Dadurch würde auch das Endezeichen 88 nicht mehr als solches erkannt und die Auszählung verfälscht. Da bei einer derartigen Verschiebung die erste Stelle des nächsten Endezeichens mit der stets vorangehenden 0 ein Ziffern-paar 08 bildet, bezeichnet diese Zahl das Auftreten eines Lesefehlers. Das Programm muß auf Ziffernpaare 08 entsprechend reagieren.

### 4. Der Programmablauf

Das Flußdiagramm BILD 3 gibt eine Übersicht über das Programm. Es setzt sich aus drei Teilen zusammen, die zeitlich nacheinander ablaufen.

Der erste Teil (Bild 3a) bereitet die Auszählung vor. Zuerst wird eine Liste der interessierenden Zielbereiche eingelesen, und zwar jeder Zielbereich als siebenstelliges Rechner-„Wort“. Die Ziffer am Beginn der Worte gibt an, wie viele der restlichen sechs Stellen für die Zielbereich-Kennzeichnung Gültig-

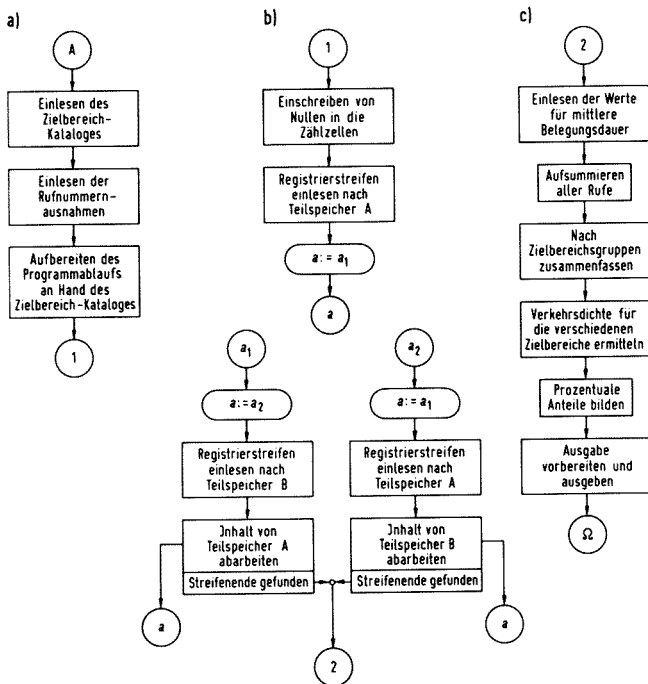


Bild 3 Flußdiagramm; a) Vorbereitung, b) Auszahlung, c) Auswertung

keit haben. Ein Beispiel möge dies erklären: In einem Netz sei mit  $n = 5$  Stellen registriert worden, die Auszahlung erfolge im Regelfall mit  $m = 2$ . Dann kann die Liste der siebenstelligen Worte folgendermaßen aussehen:

- 2 020000
- 2 060000
- 2 070000
- 5 071520
- 4 070300
- 4 071400
- 2 080000
- 2 090000
- ⋮

Es existieren demnach die zweistelligen Zielbereiche

- 02, 06, 07, 08, 09...

Ferner gibt es bei 07 noch eine feinere Auflösung in Bereiche mit mehr als zwei Stellen. Es sind dies

- 07152, 0703 und 0714.

Außerdem werden alle vollständigen Rufnummern eingelesen, die weniger als  $n$  Stellen haben. Dieser erste Programmteil ermöglicht es, den Ablauf des Programms an jedes beliebige Fernsprechnetz anzupassen.

Im zweiten Programmteil (Bild 3b) folgt das Einlesen des Registrierstreifens abschnittsweise wechselnd in die Teilspeicher A und B. Während nach B bzw. A eingelesen wird, arbeitet das Programm den Inhalt von A bzw. B ab, indem es die einzelnen Kennzahlen den Zielbereichen zuordnet und dort aufaddiert. Dieser Vorgang beansprucht etwas weniger Zeit als das Einlesen, das also die zum Bearbeiten eines Registrierstreifens erforderliche Maschinenzeit bestimmt. Ergänzend zum Flußdiagramm Bild 3b sei bemerkt, daß dieses Abarbeiten des Inhalts in dem einen Speicher sofort nach dem Anstoßen des Einleseprozesses in den anderen Speicher beginnen kann.

In den Speichern A und B steht die Information in derselben Folge wie auf dem Lochstreifen. Jeweils 14 Ziffern (zwei Rechnerworte) gelangen in das als Magazin dienende Multiplikatorregister. Von dort werden sie paarweise in den

Akkumulator geschoben und darauf geprüft, ob die Ziffernpaare 88, 99 oder 08 vorkommen. Die Zahl 88 zeigt an, daß eine Kennzahl zu Ende ist und eine neue beginnt. Von den vorliegenden Ziffernpaaren entfällt die jeweils an zweiter Stelle stehende 0. Die verbleibenden Ziffern bilden die gewählte Kennzahl; sie erhöht den Stand der Zählstelle für den angerufenen Zielbereich um den Wert 1. Tritt das Ziffernpaar 08 auf, dann liegt ein Lesefehler vor. Zur Korrektur verschiebt das Programm die Ziffernfolge um eine Stelle; dabei geht eine Kennzahl verloren, was aber für eine statistische Auswertung ohne Bedeutung ist. Ein nochmaliges Erscheinen der Zahl 08 macht diese Änderung wieder rückgängig. Das Ziffernpaar 99 am Schluß der Registrierung beendet den zweiten Programmteil.

Der dritte Teil des Programms (Bild 3c) wertet das Zählergebnis aus. Zunächst wird die Anzahl aller als richtig ermittelten Kennzahlen und, getrennt davon, aller Teilnehmerfehler festgestellt. Es folgt das Gruppieren der richtigen Kennzahlen nach Sonderdiensten, Ortsverkehr und Fernverkehr. Für jede dieser drei Gruppen ist eine mittlere Belegungsdauer eingegeben worden. Eine kleine, hier nicht näher erläuterte Rechnung ergibt die prozentualen Anteile eines jeden Ziel-

gesamtverkehr: 7030 t - teile  
anzahl der belegungen:  
gesamt: 25566 richtig: 25049 tln-fehler: 117

zielbereich	belegung	verkehr (proz.)	verkehr (t-teile)
00	33	0,13	3,14
02	200	0,80	56,24
06	177	0,70	49,21
07	1528	6,11	429,53
08	245	0,98	69,89
09	131	0,52	36,36
summe	2315	9,24	649,57
10	60	0,05	4,22
11	1195	1,35	95,51
19	35	0,03	2,11
summe	1291	1,45	101,94
24	1942	7,72	542,72
25	204	0,81	56,94
27	217	0,86	60,46
29	3747	14,89	1045,77
summe	6110	24,28	1706,88
30	115	0,45	31,64
..	....	.....	.....
..	....	.....	.....
..	....	.....	.....

07152	109	0,44	30,93
0703	179	0,71	49,91
0714	108	0,43	30,23
29976	45	0,18	12,55
29977	43	0,17	11,95
2998	39	0,15	10,55

Bild 4 Auszug aus einer Ergebnisliste

bereichs am Gesamtverkehr. BILD 4 zeigt eine Ergebnisliste im Auszug. Diesem Beispiel liegt eine Zielbereich-Kennzeichnung mit  $m = 2$  Stellen als Regelfall zugrunde. Zielbereiche mit mehr Stellen sind im Anschluß an die Regelfälle aufgeführt.

### 5. Abschließende Bemerkungen

Das beschriebene Programm arbeitet seit ungefähr einem Jahr. Der Zeitbedarf zur Auszahlung und Auswertung eines Streifens mit 25 000 fünfstellig registrierten Kennzahlen beträgt rund 20 Minuten. Bei der Anwendung zeigte sich wiederholt, daß die Variationsmöglichkeiten, die das Programm enthält, von größerem Nutzen sind, als man zunächst annahm.

# Über ein Programm zur Ermittlung des Einflusses von Schaltelementetoleranzen mit Hilfe einer Monte-Carlo-Methode

Von T. Lutz · Stuttgart

DK 681.14-523.8:518.5:519.21

## 1. Aufgabe

Komplizierte algebraische Ausdrücke und eine sehr große Anzahl von Freiheitsgraden machen im allgemeinen eine exakte Toleranzrechnung in der Elektrotechnik unmöglich. Der rechnende Ingenieur sieht sich deshalb beim Dimensionieren seiner Schaltungen weitgehend auf Vermutung und Erfahrung angewiesen. Sehr oft kann man jedoch für eine Schaltung – in Abhängigkeit von den Schaltelementen – eine kennzeichnende Größe angeben, von der gefordert werden muß, daß sie erfüllt ist. In der digitalen Schaltungstechnik trifft dies etwa auf das Niveau eines abgehenden Impulses zu, das einen bestimmten Wert nicht unterschreiten darf.

Denkt man sich eine Schaltung beschrieben durch  $n$  „unsichere“ Größen  $x_1 \dots x_n$  und durch  $m$  „sichere“ Größen  $a_1 \dots a_m$ , so muß

$$f^*(x_1 \dots x_n; a_1 \dots a_m) \geq K$$

sein, oder, was dieser Forderung äquivalent ist,

$$f(x_1 \dots x_n; a_1 \dots a_m) \geq 0 \quad (1.1)$$

Offt hat man auch den Fall, daß mehrere Forderungen bestehen:

$$\begin{aligned} f_1(x_1 \dots x_n; a_1 \dots a_m) &\geq 0 \\ \vdots \\ f_l(x_1 \dots x_n; a_1 \dots a_m) &\geq 0, \end{aligned} \quad (1.2)$$

die dazu noch in irgendeiner Und/Oder-Kombination zusammengefaßt werden, wobei das Erfülltsein der Gesamtfunktion Bedingung ist. Etwa der triviale Fall

$$F(x_1 \dots x_n; a_1 \dots a_m) \equiv f_1 \& f_2 \& \dots \& f_l \quad (1.3)$$

bedeutet, daß alle  $l$  Bedingungen (1.2) stets gleichzeitig erfüllt sein müssen.

Erwünscht sind nun Wahrscheinlichkeitsaussagen über die Forderung (1.1) bzw. (1.3) unter der Voraussetzung, daß die  $n$  „unsicheren“ Größen mit Toleranzen behaftet sind, also in ihren Werten streuen und dabei etwa einer Rechteck- oder Normalverteilung genügen. Da eine strenge Lösung dieses Problems in den seltensten Fällen gelingt, ist es sinnvoll, solche Lösungen durch Monte-Carlo-Tests anzunähern.

## 2. Monte-Carlo-Teste

Der zentrale Begriff des Monte-Carlo-Tests ist die Zufallsfolge bzw. die Zufallszahl. Unter einer Zufallsfolge versteht man eine Folge  $(x_i)$  von Zahlen, deren Elemente in einem bestimmten Intervall liegen und zu der eine Wahrscheinlichkeitsdichte  $p(x)$  definiert ist, die aussagt, mit welcher Wahrscheinlichkeit ein bestimmtes  $x_i$  beim Durchlaufen der Folge angetroffen werden kann. Man nennt die Elemente einer Zufallsfolge auch Zufallszahlen. Speziell spricht man von einer rechteckverteilten oder gleichverteilten Zufallsfolge, wenn  $p(x)$  für alle Werte  $x_i$  der Zufallsfolge einen konstanten Wert annimmt, und sagt dann, die  $x_i$  seien in  $(x_i)$  gleichverteilt, oder die  $x_i$  seien gleichverteilte Zufallszahlen. Eine solche gleichverteilte Zufallsfolge entsteht etwa über dem Intervall  $(1;6)$  der ganzen Zahlen, wenn man die Ereignisse

eines Würfelprozesses mit einem idealen Würfel fortlaufend notiert. Es ist dann

$$p(x) = \begin{cases} 0 & -\infty < x < 1 \\ 1/6 & 1 \leq x \leq 6 \quad (x \text{ ganz}) \\ 0 & 6 < x < +\infty. \end{cases}$$

Neben gleichverteilten spielen auch normalverteilte Zufallsfolgen eine wichtige Rolle. In diesem Falle drückt  $p(x)$  die Dichte einer Gaußschen Normalverteilung aus, und das Intervall, über dem  $p(x)$  definiert ist, erstreckt sich, ebenso wie das der  $x_i$ , über die ganze reelle Achse. Mittelwert und Streuung von  $p(x)$  charakterisieren in diesem Falle dann auch die Folge  $(x_i)$ .

Es liegt in der Definition des „Zufalls“ begründet, daß man bei einer „echten“ Zufallsfolge  $(x_i)$  ein Bildungsgesetz oder eine Bildungsvorschrift für die  $x_i$  nicht angeben kann. Andererseits ist man jedoch, wenn man sich nicht auf empirisch, aus echten Zufallsprozessen ermittelte Tabellen stützen will, in der Praxis der Rechenmaschine darauf angewiesen, Programme zu schreiben, die bei Aufruf eine Zufallszahl und bei wiederholtem Aufruf eine endliche Zufallsfolge liefern. Ein solches Programm ist aber eine Bildungsvorschrift. Andererseits kann eine solche „unechte“ Zufallsfolge sehr wohl den Erfordernissen der Wahrscheinlichkeitsrechnung genügen. In diesem Falle spricht man von Quasizufallsfolgen bzw. Quasizufallszahlen. Die Unechtheit einer Zufallsfolge äußert sich so, daß die Elemente der Folge weitgehend stochastisch voneinander abhängig sind [1].

Die Idee des Monte-Carlo-Tests liegt nun darin, daß man jeder der Größen  $x_1 \dots x_n$  in (1.2) eine Zufallsfolge zuordnet, deren Elemente dem Streubereich der betreffenden Größe entsprechen. Wenn etwa  $x_1$  für einen ohmschen Widerstand mit dem Wert  $R$  und eine zulässige Toleranz  $\eta$  steht, so wählt man für  $x_1$  eine Zufallsfolge  $(x_{1i})$  die über dem Intervall  $[R(1-\eta); R(1+\eta)]$  gleichverteilt ist. Entsprechend nimmt man, wenn  $x_k$  eine normalstreuende Größe mit Mittelwert  $m_k$  und Streuung  $\sigma_k$  ist, eine Zufallsfolge, deren  $p(k)$  den Mittelwert  $m_k$  und die Streuung  $\sigma_k$  hat. Jeder der Größen  $x_1 \dots x_n$  ist also damit eine charakterisierende Zufallsfolge zugeordnet:  $(x_{1i}), (x_{2i}) \dots (x_{ni})$ .

Nummehr macht man  $N$  Versuche, bei denen der  $k$ -te Versuch darin besteht, daß man aus den Zufallsfolgen die  $k$ -ten Elemente  $x_{1k} \dots x_{nk}$  entnimmt und zusammen mit den Werten  $a_1 \dots a_m$  in die Forderung (1.1) bzw. (1.2) einsetzt. Je nachdem, ob die Forderung erfüllt ist oder nicht, schaltet man die zu Beginn des Prozesses auf Null gesetzten Zähler  $Z1$  oder  $Z2$  weiter.

Bezeichnet  $Z1_N = (Z1)_N$

und  $Z2_N = (Z2)_N$

den Stand der Zähler nach  $N$  Versuchen, so bildet man daraus die relativen Häufigkeiten

$$p_1 = \frac{Z1_N}{N} \quad \text{und} \quad p_2 = \frac{Z2_N}{N}$$

wobei gilt

$$p_1 + p_2 = 1.$$

Ein solches Verfahren rechnet man zu den Monte-Carlo-Testen.

Da  $p_1$  und  $p_2$  einen Sinn nur für sehr große Werte von  $N$  erhalten ( $N \geq 10^4$ ), ist man bei der Durchführung eines Monte-Carlo-Testes aus Zeitgründen weitgehend auf elektronische Rechenanlagen angewiesen. Arbeitet man etwa mit

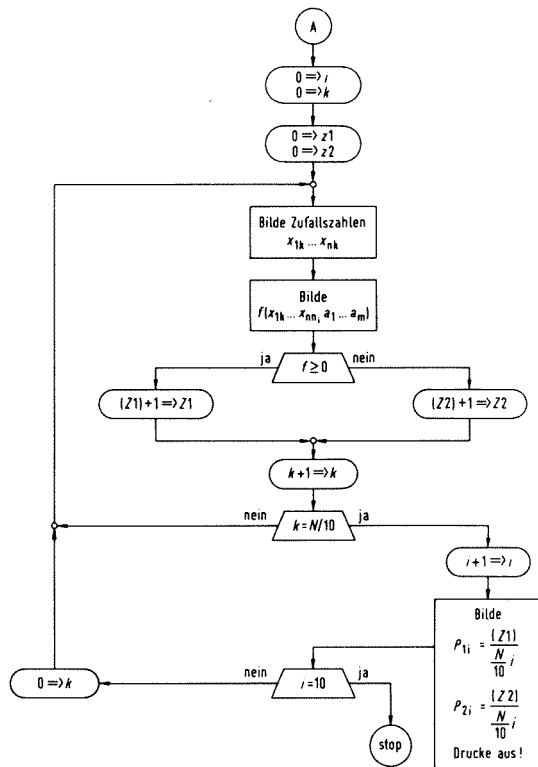


Bild 1 Flußdiagramm für den allgemeinen Ablauf eines Monte-Carlo-Testes

$N = 10^4$ , so empfiehlt es sich, die Bildung von  $p_1$  und  $p_2$  über die Dauer des Testes zu verfolgen, indem man sich etwa jeweils nach  $N/10$  Teilversuchen die Teilergebnisse betrachtet. Zeigen die Werte  $p_1$  und  $p_2$  eine Tendenz, so entsteht im allgemeinen kein großer Fehler, wenn sie als die gesuchte Wahrscheinlichkeitsaussage für die Forderung (1.1) bzw. für die zugehörige Schaltung angesehen werden.

Hat man es mit einer mehrstelligen, logisch kombinierten Forderung  $F$  im Sinn von 1.2 zu tun, so berechnet man in einem Versuch zuerst die Teilformen  $f_1...f_l$ . Wichtig ist dabei, daß für jedes  $f_i$  derselbe Satz von Zufallszahlen verwendet wird. Den erfüllten logischen Forderungen ordnet man den logischen Wert 1, den nicht erfüllten den Wert 0 zu und berechnet nach der Booleschen Algebra den zugehörigen logischen Gesamtwert; 0 bedeutet „Forderung nicht erfüllt“,

1 dagegen „Forderung erfüllt“. Für die Auswertung der Zähler  $Z1$  und  $Z2$  gilt das oben Gesagte. Diesen allgemeinen Ablauf eines Monte-Carlo-Testes veranschaulicht BILD 1.

Es sei erlaubt, darauf hinzuweisen, daß man mit der Idee der Zufallszahlen auch andere Klassen von Problemen lösen kann, die sich nicht in den Rahmen des hier Gezeigten einfügen. Beispiele dafür sind Verkehrssimulation in Verbundsystemen, die Strömung von Teilchen beschreibende partielle Differentialgleichungen usw.

### 3. Ein gleichverteilter Zufallsgenerator

Programme für elektronische Rechenanlagen, die bei Aufruf eine Zufallszahl und bei mehrmaligem Aufruf eine (endliche) Zufallsfolge erzeugen, werden in der Programmtechnik oft auch Zufallsgeneratoren genannt. Solche Zufallsgeneratoren existieren in den unterschiedlichsten Versionen für diverse Zwecke. Am bequemsten zu programmieren sind Zufallsgeneratoren für gleichverteilte Quasizufallsfolgen, die mit arithmetischen Prozessen arbeiten. Das Programm bildet dabei an Hand einer Bildungsvorschrift eine Folge von Zahlen. Jedem Element dieser Folge entnimmt man eine bestimmte Aufeinanderfolge von Ziffern und verwandelt sie in eine Zahl, die in dem gewünschten Zufallsintervall liegt.

Ein einfacher und brauchbarer Generator kommt nach [1] auf folgende Weise zustande. Es sei  $C_0$  eine reelle Zahl mit  $0,1 \leq C_0 < 1,0$  und es sei eine Folge  $(C_i)$  definiert durch

$$C_i = \begin{cases} C_{i-1}^2 & \text{für } C_{i-1} \geq 0,6 \\ C_{i-1}^2 + 0,35 & \text{für } C_{i-1} < 0,6 \end{cases} \quad (3.1)$$

Aus der Folge  $C_i$  wird eine Zufallsfolge  $(x_i)$  gewonnen, indem man aus den Zahlen  $C_i$  stets eine bestimmte Ziffernfolge von konstanter Länge durch Intersektion ausschneidet. Diese Ziffern entnimmt man zweckmäßig der Mitte oder dem letzten Drittel der Zahl  $C_i$ , um eine gute Durchmischung der Zufallszahlen zu erhalten. Aus dieser Ziffernfolge macht man dann Zahlen, die in dem gewünschten Intervall liegen. Ist dieser Prozeß mit  $x_i = I(C_i)$  bezeichnet, so hat der Generator folgende Gestalt:

$$\mathcal{G} = \begin{cases} x_i = I(C_i) \\ C_i = \begin{cases} C_{i-1}^2 \\ C_{i-1}^2 + 0,35 \end{cases} \text{ für } \begin{cases} C_{i-1} \geq 0,6 \\ C_{i-1} < 0,6 \end{cases} \\ i = 1, 2, 3, \dots \\ 0,1 \leq C_0 < 1,0 \end{cases}$$

Über weitere Eigenschaften dieses Generators wie Zyklen, Degeneration u. ä. gibt [1] erschöpfend Auskunft.

Nimmt man als Anfangswert  $C_0 = 0,1234567898765$  und ist  $I(C_i)$  so definiert, daß man aus  $C_i$  jeweils die siebente

N	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
1 000	105	85	85	98	120	90	92	113	102	110
2 000	100	105	106	96	111	83	94	103	93	109
3 000	83	108	111	95	108	99	105	87	101	103
4 000	109	88	114	108	94	105	98	97	93	94
5 000	106	86	121	99	87	114	96	104	90	97
6 000	100	83	98	106	112	107	95	99	112	88
7 000	123	103	104	97	104	92	92	104	88	93
8 000	96	105	94	110	95	96	99	103	102	100
9 000	102	114	119	111	98	97	89	79	93	98
10 000	108	102	96	89	98	99	94	111	87	116

Mittelwert:	103,2	97,9	104,8	100,9	102,7	98,2	95,4	100,0	96,1	100,8
-------------	-------	------	-------	-------	-------	------	------	-------	------	-------

Tafel 1 Absolutes Auftreten nach  $N$  Versuchen

**Tafel 2**

Relative Häufigkeiten nach  $N$  Versuchen

$N$	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
1 000	0,1050	0,0850	0,0850	0,0980	0,1200	0,0900	0,0920	0,1130	0,1020	0,1100
2 000	0,1025	0,0950	0,0955	0,0970	0,1155	0,0865	0,0930	0,1080	0,0975	0,1095
3 000	0,0960	0,0993	0,1007	0,0963	0,1130	0,0907	0,0970	0,1010	0,0987	0,1073
4 000	0,0993	0,0965	0,1040	0,0993	0,1083	0,0943	0,0973	0,1000	0,0973	0,1040
5 000	0,1006	0,0944	0,1074	0,0992	0,1040	0,0982	0,0970	0,1008	0,0958	0,1026
6 000	0,1005	0,0925	0,1058	0,1003	0,1053	0,0997	0,0967	0,1005	0,0985	0,1002
7 000	0,1037	0,0940	0,1056	0,0999	0,1051	0,0986	0,0960	0,1010	0,0970	0,0991
8 000	0,1028	0,0954	0,1041	0,1011	0,1039	0,0983	0,0964	0,1013	0,0976	0,0993
9 000	0,1027	0,0974	0,1058	0,1022	0,1032	0,0981	0,0956	0,0988	0,0971	0,0991
10 000	0,1032	0,0979	0,1048	0,1009	0,1027	0,0982	0,0954	0,1000	0,0961	0,1008
20 000	0,1010	0,1006	0,1035	0,1016	0,1014	0,0997	0,0961	0,1001	0,0981	0,0980
30 000	0,1004	0,1010	0,1045	0,0988	0,1026	0,0983	0,0973	0,0988	0,0997	0,0986

Ziffer  $C_{j,i}$  entnimmt und daraus die Zahl  $0, C_{j,i}$  macht, so liefert der Generator die zehn Ereignisse 0,0; 0,1; 0,2...0,9. Die Zufallsfolge  $x$  besteht also aus den zehn einstelligen Dezimalzahlen im Intervall (0; 0,9), die bei einem gleichverteilten Zufallsgenerator jeweils mit einer theoretischen Wahrscheinlichkeit von  $P(x_i) = 0,1$  erwartet werden müssen.

In TAFEL 1 ist die absolute Anzahl der aufgetretenen Dezimalen zusammengestellt, während TAFEL 2 die Häufigkeiten nach jeweils  $N$  Versuchen für diesen Generator ausweist.

#### 4. Ein normalverteilter Zufallsgenerator

Normalverteilte Zufallsfolgen werden im allgemeinen über gleichverteilte Zufallsfolgen erzeugt, indem man deren Elemente durch eine geeignete Formel umrechnet. Ein Verfahren dazu ist in [2] geschildert. Es seien  $x_i$  und  $x_{i+1}$  zwei Zufallszahlen einer gleichverteilten Zufallsfolge ( $x_i$ ) mit  $0 \leq x_i \leq 1$ . Berechnet man sich dazu

$$u_i = \sqrt{-2 \ln x_i} \sin 2\pi x_{i+1}$$

$$u_{i+1} = \sqrt{-2 \ln x_i} \cos 2\pi x_{i+1}$$

so bilden  $u_i$  und  $u_{i+1}$  ein Paar – allerdings stochastisch abhängiger – Zufallszahlen einer Zufallsfolge ( $u_i$ ), deren Elemente normalverteilt sind, und zwar zum Mittelwert  $m = 0$  mit der Streuung  $\sigma = 1,0$ .

Da für einen Monte-Carlo-Test sehr viele Zufallszahlen gebraucht werden und die Berechnung der  $u_i$  und  $u_{i+1}$  in großem Umfang nicht unerhebliche Rechenzeit beansprucht, ist es zweckmäßig, diese Arbeit nur einmal durchzuführen und eine große Anzahl von Zufallszahlen auf einem Magnetband abzuspeichern. Es wurden rund 30 000 normalverteilte Zufallszahlen ermittelt und in Gruppen (Blöcken) zu jeweils 99 Zahlen auf einem Magnetband gespeichert. Die Rechenzeit dafür betrug – einschließlich Bandbeschreiben und Probelesen – 27 Minuten, wobei insgesamt rund 15 000 Wurzeln und jeweils 15 000 Sinus- und Cosinuswerte zu berechnen waren.

Mittelwert  $m$  und Streuung  $\sigma$  der Zufallszahlen gingen aus einem kleinen Testprogramm hervor. Es ergaben sich die Werte

$$m = 0,000\ 566 \text{ und}$$

$$\sigma = 0,982\ 581,$$

die als ausreichend anzusehen sind.

Der Zufallsgenerator stellt nun bei Aufruf eine der Zufallszahlen und beim folgenden Aufruf die nächste Zufallszahl bereit. Das Programm (BILD 2) sorgt dafür, daß stets genügend Zufallszahlen im Arbeitsspeicher (Kernspeicher) der

Maschine zur Verfügung stehen, und es veranlaßt bei Bedarf zeitgünstig das Auslesen eines neuen Blocks vom Magnetband. Wenn das Band erschöpft ist, wird das Band zurückgespult und der Leseprozeß beginnt von vorne.

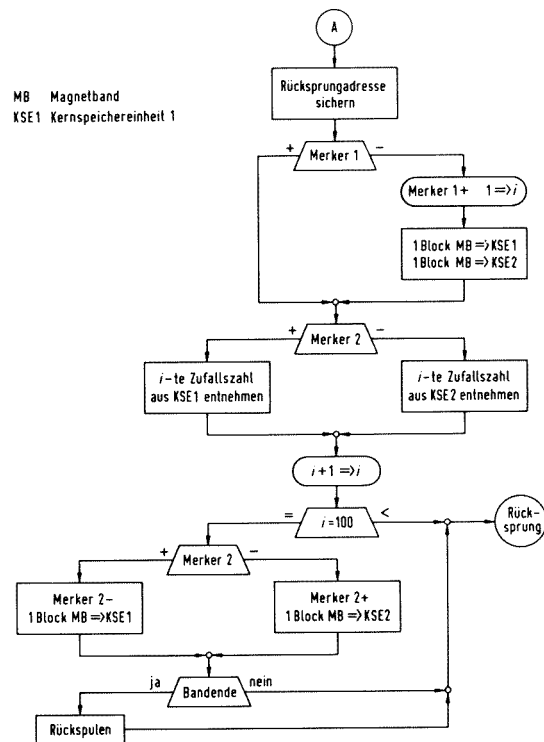


Bild 2 Flußdiagramm für den Zufallsgenerator

#### 5. Ein Beispiel

Der Aufbau eines Programms, das einen Monte-Carlo-Test organisiert, ist bereits in Abschnitt 2 erläutert. Dazu sei noch ein Beispiel gegeben, das im SEL-Rechenzentrum für das Labor für Grundschaltungen im Informatikwerk programmiert wurde. Sowohl die Aufgabenstellung als auch die Auswertung der Ergebnisse oblag dabei dem Labor für Grundschaltungen.

Die Aufgabe bestand darin, eine kombinierte Und/Oder-Schaltung mit  $n$  Oder- und  $m$  Und-Eingängen (BILD 3) zu untersuchen. Erwünscht waren Wahrscheinlichkeitsaussagen über die Spannung  $U_{Dn}$ . Dazu mußte die Funktion

$$f = a_1 - x_2 (a_2 - x_3) - x_2 \left\{ \frac{a_3 + x_1 (a_4 + x_4) - x_2 (a_2 - x_3)}{x_1 + x_2} + \frac{1}{2} \left( \frac{x_5}{x_1 + x_2} \right)^2 + \frac{x_5}{x_1 + x_2} \sqrt{\frac{a_3 + x_1 (a_4 + x_4) - x_2 (a_2 - x_3)}{x_1 + x_2} + \frac{1}{4} \left( \frac{x_5}{x_1 + x_2} \right)^2} \right\}$$

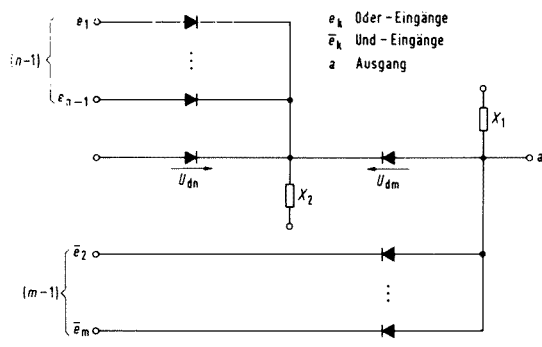


Bild 3 Kombinierte Und/Oder-Schaltung

getestet werden.  $x_1 \dots x_5$  sind dabei „unsichere“ Größen, und zwar sollten  $x_1$  und  $x_2$  rechteckverteilte ohmsche Widerstände sein, mit den Mittelwerten  $x_1$  bzw.  $x_2$  und den Toleranzen  $\eta_1$  und  $\eta_2$ .  $x_3, x_4$  und  $x_5$  waren normalverteilte Dioden-Kenn-daten mit folgenden Mittelwerten und Streuungen:

$$\begin{aligned} \bar{x}_3 &= (n-1) X_3 & \sigma_3 &= \sqrt{n-1} S_3 \\ \bar{x}_4 &= (m-1) X_4 & \sigma_4 &= \sqrt{m-1} S_4 \\ \bar{x}_5 &= X_5 & \sigma_5 &= S_5 \end{aligned}$$

$m$  ist die Anzahl der Eingänge auf der Und-Seite und  $n$  die Anzahl der Oder-Eingänge in die Schaltung. Gemäß Bild 3 wird eine  $n$ -wertige Oder-Schaltung mit dem Ausgang in eine  $m$ -wertige Und-Schaltung geführt, so daß eine Schaltung mit  $m+n-1$  Eingängen entsteht. Es interessieren  $(m, n)$  Kombinationen für die Werte  $m=2, 5, 10$  und  $n=2, 5, 10$ . Außerdem waren die Größen  $x_1$  und  $x_2$  noch zu variieren, und zwar sollte sein  $x_1=16 \text{ k}\Omega$  für  $x_2=3, 5, 8 \text{ k}\Omega$  und  $x_1=40 \text{ k}\Omega$  für  $x_2=6, 13, 20 \text{ k}\Omega$ . Die übrigen Größen sind wie folgt einzusetzen:

$$\begin{aligned} \eta_1 &= \eta_2 = 0,05 & X_3 &= 25 \cdot 10^{-3} \text{ mA} & S_3 &= 10^{-2} \text{ mA} \\ X_4 &= 25 \cdot 10^{-3} \text{ mA} & S_4 &= 10^{-2} \text{ mA} \\ X_5 &= 0,34 \frac{\text{V}}{\text{mA}} & S_5 &= 0,05 \frac{\text{V}}{\text{mA}} \\ a_1 &= 10 \text{ V} \\ a_2 &= a_4 = 0 \\ a_3 &= 38,9 \text{ V} \end{aligned}$$

Um die beiden gleichverteilten Größen  $x_1$  und  $x_2$  zu gewinnen, wurde der Würfelprozeß des in Abschnitt 3 beschriebenen Zufallsgenerators für 101 Ereignisse und das Intervall  $(0; 1,0)$  programmiert. Ist etwa  $u_k$  ein solches Ereignis (mit  $0 \leq u_k \leq 1,0$ ) sowie  $\eta$  die Toleranz und  $X$  der Mittelwert, dann stellt

$$x_k = 2 \eta u_k + X - \eta$$

eine lineare Transformation dar, die  $u_k$  in das Intervall  $X - \eta \leq x_k \leq X + \eta$  transformiert.

Ebenso kann man auch für die normalverteilten Zufallsgrößen mit einem normierten Generator arbeiten, so daß man für jede der Größen  $x_3, x_4, x_5$  mit demselben Unterprogramm auskommt. Hier wurde der in Abschnitt 4 beschriebene normalverteilte Zufallsgenerator mit dem Mittelwert 0 und der Streuung 1 verwendet. Ist  $v_k$  ein Ereignis der Zufallsreihe und sind  $m$  bzw.  $\sigma$  Mittelwert bzw. Streuung der gewünschten Verteilung, so stellt

$$x_k = v_k \sigma + m$$

wieder ein normalverteiltes Ereignis zum Mittelwert  $m$  und zur Streuung  $\sigma$  dar. Weil die normalverteilten Größen  $x_3, x_4$  und  $x_5$  Dioden-Kenngrößen waren, trat insofern eine Spezifikation ein, als  $x_k$  nicht negativ sein durfte. Es wurde deshalb beim Auftauchen eines  $x_k < 0$  der gesamte Versuch verworfen, und zwar ohne Weiterzählung.

Ein Teilttest umfaßte jeweils 1000 Versuche. Lagen nach den ersten 1000 Versuchen für eine Parameterkombination mehr als zehn Versuche vor, bei denen  $U_{Dn} < 0$  war, so galt die Schaltung als „schlecht“ und es erübrigte sich, den Test fortzusetzen. Bei allen anderen Parameterkombinationen wurden jeweils 5000 Versuche durchgeführt. Um den Zufallscharakter des Tests noch zu erhöhen, sollten die normierten Zufallsfolgen für zwei verschiedene Parameterkombinationen unterschiedlich sein, die Zufallsgeneratoren also „weiterlaufen“. Sie wurden deshalb zum Testbeginn einer neuen Parameterkombination nicht auf ihre ursprünglichen Ausgangswerte gesetzt. Dies war insofern bedeutsam, als damit dem Einfluß eventueller Zyklen in den Zufallsgeneratoren eine geringere Bedeutung zukam.

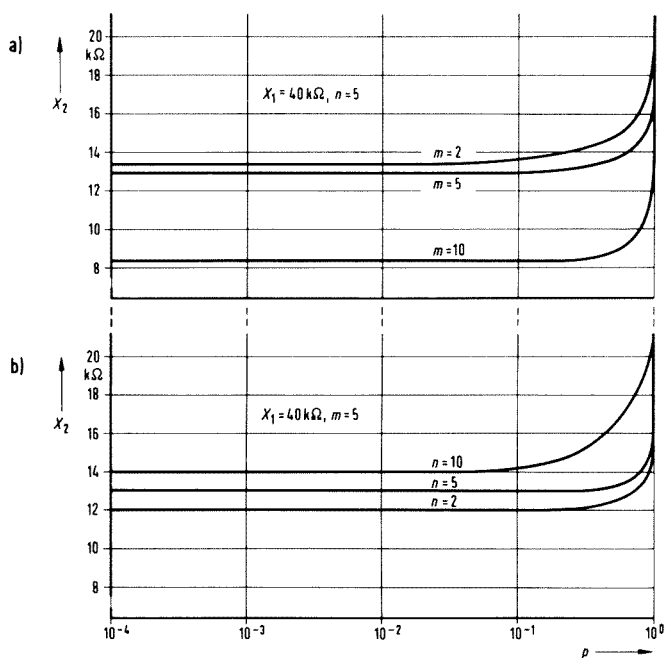


Bild 4 Ergebnis eines durchgerechneten Beispiels

Zur Auswertung trägt man die zu einer bestimmten erreichten relativen Ausfallhäufigkeit  $p$  gehörigen Werte  $X_2$  bei konstanten  $m, n$  und  $X_1$  in ein Koordinatensystem ein. Damit ergeben sich für verschiedene Werte  $m$  Kurvenscharen, die man zu einer vernünftigen Dimensionierung der Schaltung heranziehen kann. BILD 4a zeigt die Ergebnisse für festes  $n$  mit  $m$  als Scharparameter und BILD 4b für festes  $m$  mit  $n$  als Scharparameter;  $X_1$  war dabei jeweils konstant. Das Testprogramm umfaßt – mit Speicherplatz für die Parameter und Zwischenergebnisse – etwa 300 Befehle. Die Rechenzeit beträgt für 1000 Versuche rund 2,5 Minuten.

#### Schrifttumsnachweis

- [1] S. v. Hoerner: Herstellung von Zufallszahlen auf Rechenautomaten. Z. angew. Math. u. Phys. 8 (1957), Nr. 1, S. 26...52.
- [2] G. E. P. Box und M. E. Muller: A note on the generation of normal deviates. Ann. Math. Stat. 27 (1958), S. 610...611.

# Erstellen von Verdrahtungsunterlagen mit dem ER 56

Von G. Jung · Stuttgart

DK 621.3.049.73.001.1:681.14-523.8

## 1. Einleitung

Das sehr oft beim Bau von elektronischen Rechenanlagen angewandte Baukastenprinzip erfordert als Verbindung zwischen den einzelnen Bausteinen eine große Anzahl von Drähten, die in umfangreichen Listen aufgeführt werden. Diese Listen zusammensetzen und pausfähig zu schreiben ist eine langwierige und eintönige Arbeit. Wenn man sie dagegen einem elektronischen Rechenautomaten überträgt, fällt nur noch ein relativ kleiner Aufwand an menschlicher Arbeit an. Die Hauptvorteile dieses Verfahrens gegenüber dem manuellen Schreiben der Verdrahtungsunterlagen sind:

1. Sehr kurze Bearbeitungszeit,
2. Fehlerfreie Listen bei richtiger Eingangsinformation,
3. Einheitliche Schreibweise in allen vom Rechner erstellten Unterlagen.

Als das nachfolgend erläuterte Programm geschrieben wurde, stand nur die Laborausführung des Elektronischen Rechenautomaten ER 56 mit einer Speicherkapazität von 3200 Kernspeicherzellen und 12000 Speicherzellen auf einer Magnetrommel zur Verfügung. Als Eingabegerät dienten Lochstreifen- und Lochkartenleser, als Ausgabegerät ein Lochstreifen-Schnellstanzer.



Bild 1 Unverkleidete Gestelle mit Steckeinheiten und Kernspeichern im Versuchsaufbau

## 2. Programm-Voraussetzungen

### 2.1 Aufbau und Verdrahtung der Gestelle

Die im Informatikwerk gebauten Anlagen bestehen aus einzelnen Gestellen (BILD 1). Jedes Gestell bietet 120 in zehn Zeilen angeordneten Steckeinheiten Platz, die je über eine 20- und eine 30polige Messerleiste mit dem Gestell verbunden sind. An Stelle von Steckeinheiten können auch 26polige Kabelstecker, Koordinatenschalter, Magnettrommeln, Kernspeicher verschiedener Größe, Buchsenfelder für die Formatsteuerung von Ein- und Ausgabegeräten sowie Diodenplatten in die Gestelle eingebaut werden. Außerdem ist es möglich, neben den Zeilen 52polige Lötösenplatten anzubringen. Die unterschiedlichen Anschlußbedingungen aller genannten Einheiten muß das Programm berücksichtigen.

Drachtsorte: FSL-Ø,5

Kartenart:



b		a		c		b		a	
Platz		Platz		Platz		Platz		Platz	
2	1	0	0	2	1	0	0	2	1
0	S	/		/	S	/	U:1	U:0	0
9	/	W:P	2	/	A:A	3		B:0	3
8	A	L:A	/	A	L:A	/		B:0	5
7	U	P:2	/	A	A:2			9	R:3
									R:0
									1

Bild 2 Beispiel eines Formulars; Kopf des Belegungsplans für eine Steckeinheit

Für die Gestelle von Informatik-Systemen ist eine Kammverdrahtung charakteristisch. Die Drähte werden nicht in Kabelbäumen, sondern einzeln über Käme zu den Lötstiften geführt.

### 2.2 Eingangsinformationen

Jeder Leitungszug erhält einen Namen, der entweder von der Funktion des zu übertragenden Signals oder vom Entstehungsort abgeleitet und deshalb für das Wartungspersonal leicht verständlich ist. Die gleichen Namen werden auch in den Signalformeln verwendet, welche die Funktion der Steckeinheiten beschreiben. Sie dürfen aus 1...3 Zeichen (Buchstaben und Ziffern) bestehen. Dazu kommt als Negationszeichen ein Strich über den Namen, das jedoch hier, um auf Lochkarten übertragbar zu sein, die Form eines Schrägstrichs vor dem Namen annimmt.

Für jede der in Abschnitt 2.1 benannten Baueinheiten gibt es ein Formular, das vorbereitete Felder für sämtliche Lötstifte aufweist. Nach Eintragen der Platznummer in den Formulkopf ist jeder Lötstift eindeutig bezeichnet. Beispielsweise besagt die „Adresse“ 210y9a, daß dieser Lötstift in Zeile 2 und Spalte 10 auf der Steckerleiste y liegt, und zwar an neunter Stelle in deren Reihe a (BILD 2). In die Felder wird jeweils der Name des Leitungszuges eingeschrieben, der an

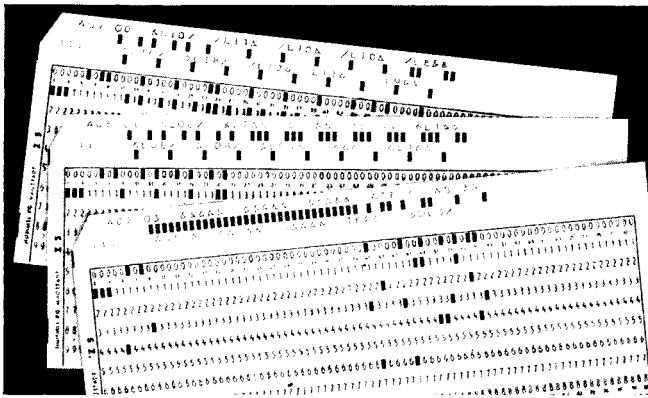


Bild 3 Lochkarten mit Belegungsangaben für Lötstift-Reihen

dem betreffenden Lötstift liegt; zugleich ist anzugeben, ob es sich um einen Ausgang (Schrägstrich hinter dem Namen) oder um einen Eingang (kein Strich) handelt.

Die Lötstifte der Baueinheiten sind so in einzelne Reihen unterteilt, daß sich pro Reihe maximal 13 Lötstifte ergeben (Kapazität einer Lochkarte), wobei sich die Aufteilung möglichst mit der Lötstift-Anordnung decken soll. Jede Reihe wird auf eine Lochkarte (BILD 3) übertragen. Am Anfang der Lochkarten steht die Adresse der betreffenden Reihe; aus ihr leitet der Rechner die Adressen der Lötstifte ab.

Das Ausfüllen dieser Formulare ist in der Regel Aufgabe des Entwicklungslabors. Gegenüber dem alten Verfahren, das neben dem Ausfüllen ähnlicher Formulare noch das Zusammenstellen der einzelnen Leitungszüge erforderte, hat sich auch der zeitliche Aufwand für diese Arbeit vermindert.

### 2.3 Erläuterung der Legeliste und des Gestellstromlaufplans

In den Legelisten eines Gestells (BILD 4) sind alle Leitungszüge dieses Gestells aufgeführt. Die Zeilen beginnen mit dem Namen, der Nummer und der Farbe des jeweiligen Drahtzugs. Damit die Werkstatt zuerst die kurzen Verbindungen (Brücken) und später die längeren Verbindungen einlegen kann, enthält die Legeliste alle Drahtzüge nach vier Verdrahtungsarten getrennt. Im Schriftkopf des Blattes ist die Drahtsorte der aufgeführten Leitungszüge angeschrieben. Für jede Drahtsorte – oft gibt es mehrere in einem Gestell – wird eine besondere Legeliste erstellt. Die Legelisten dienen als Arbeitsanweisung zum Verdraten der einzelnen Gestelle.

In dem als Prüf- und Wartunterlage benötigten Gestellstromlaufplan (Bild 4) sind alle belegten Lötstifte – nach Baueinheiten getrennt – aufgeführt. Hinter der Bezeichnung des jeweiligen Lötstifts stehen der Name und, sofern es sich bei dem Lötstift um einen Ausgang handelt, alle Punkte des Drahtzuges; ist der Lötstift jedoch ein Eingang, so folgen nur jene Punkte des Drahtzuges, die einen Ausgang darstellen. Hinter der Lötstiftbezeichnung von Ausgangspunkten steht immer ein Doppelpunkt. Über den Namen und die Adresse der Einheit gibt der Schriftkopf jedes Planes Auskunft. An Hand von Gestellstromlaufplänen kann man die Drahtzüge sehr leicht verfolgen, ohne in der Legeliste suchen zu müssen.

## 3. Das Programm zum Erstellen der Verdrahtungsunterlagen

### 3.1 Einlesen der Information

Das Flußdiagramm für den Einlesevorgang ist in BILD 5 wiedergegeben. Je nachdem, ob eine Legeliste oder ein Gestellstromlaufplan zu erstellen ist, werden die Lochkarten einer Drahtsorte oder alle Lochkarten eines Gestells eingelesen. Gibt es nur eine Drahtsorte in dem Gestell, so genügt sowohl für die Legeliste als auch für den Gestellstromlauf die gleiche Information.

Die belegten Punkte einer Lötstiftreihe, der eine Lochkarte oder mehrere Lochkarten mit gleicher Adresse entsprechen, werden in einen zuvor gelöschten Zwischenspeicher (im folgenden Lochkartenspeicher genannt) übertragen, in dem für jeden Lötstift ein bestimmtes Doppelwort reserviert ist. Dabei steht der Drahtzugname im ersten und die zugehörige, aus der Lochkartenadresse abgeleitete Stiftadresse im zweiten Teil des betreffenden Doppelwortes. Nach dem Einlesen einer Lochkarte mit neuer Adresse überträgt das Programm die Doppelworte, die ungleich Null sind, aus dem Lochkartenspeicher in einen zweiten Speicher (im folgenden Trommelblockspeicher genannt). In diesem Trommelblockspeicher werden 100 Doppelworte gesammelt und dann auf die Trommel geschrieben.

Beim Verschlüsseln der Lochkartenadressen und beim Abspeichern der Drahtzugnamen prüft das Programm, ob Fehler vorliegen (falsche Adresse, verbotenes Signal, zwei verschiedene Namen auf einem Lötstift bei mehreren Drahtsorten). Ist das der Fall, dann läuft der Rechner auf einen Stop und

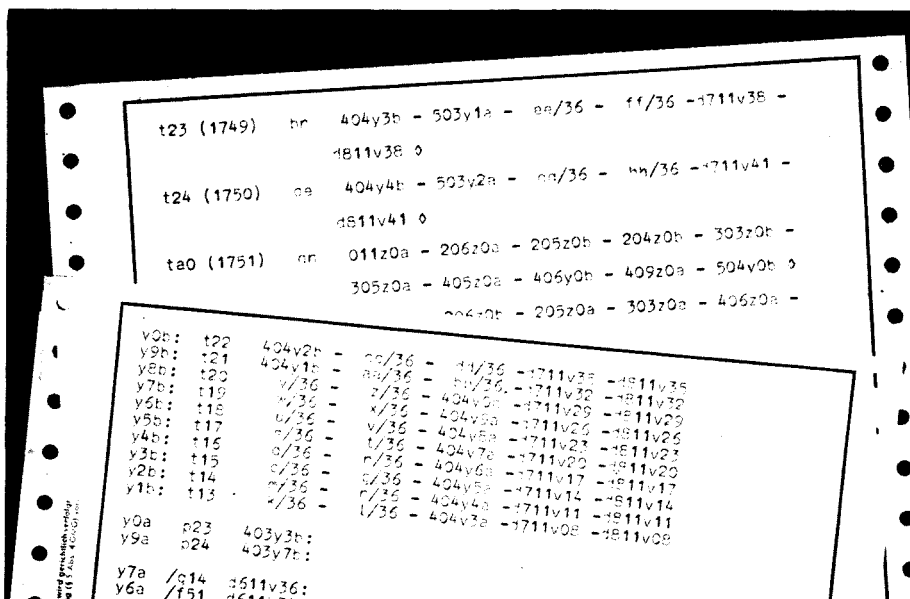


Bild 4 Ausschnitt aus einer Legeliste (im Bild unten) und aus einem Gestellstromlaufplan (oben)



zeigt Art und Spalte des Fehlers an, so daß die fehlerhafte Lochkarte schnell verbessert werden kann.

Am Ende der Lochkarten liegt eine besondere Endekarte. Wenn diese eingelesen ist, wird ein Endezeichen hinter die Information im Trommelblockspeicher geschrieben und dessen Inhalt auf die Trommel übertragen. Anschließend läuft das Unterprogramm „Trommel sortieren“ ab, wobei alle Punkte

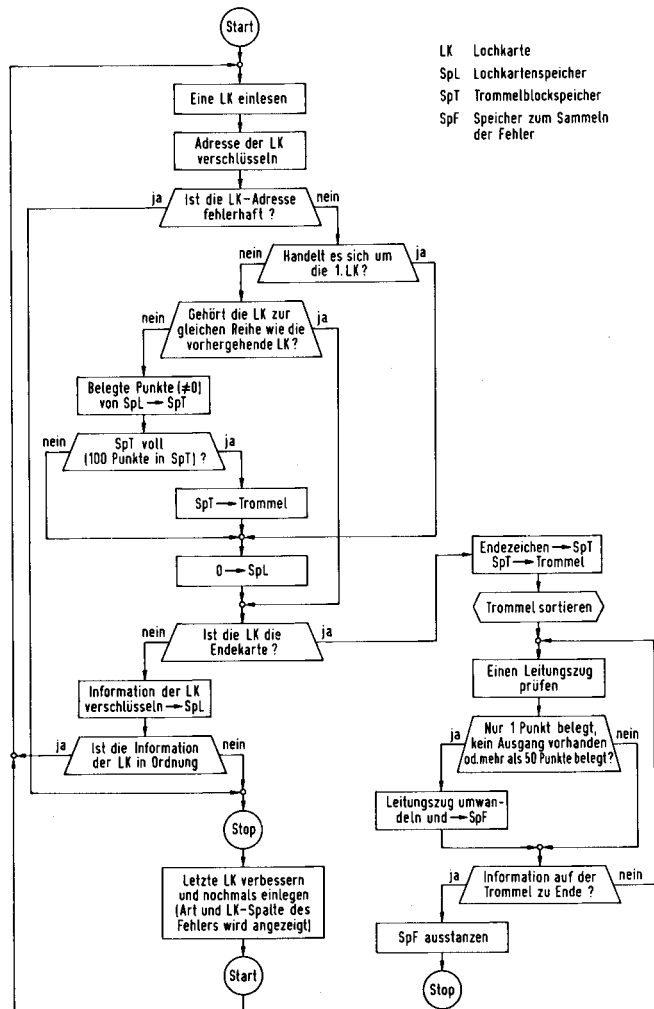


Bild 5 Flußdiagramm für das Einlesen der Information

mit gleichen Namen hintereinander zu stehen kommen. Im Programm folgt eine Prüfung auf Fehler, die beim Einlesen der Lochkarten nicht festzustellen waren (ein Leitungszug besteht nur aus einem Punkt, hat keinen Ausgang oder geht an mehr als 50 Punkte). Die festgestellten Fehler sammelt ein Speicher, um sie später auf Lochstreifen auszustanzen; über einen Fernschreiber ausgeschrieben dienen diese Angaben dem Entwicklungslabor zum Verbessern der Unterlagen. Die korrigierten Lochkarten werden in der gleichen Weise nochmals eingelesen und überprüft.

### 3.2 Ausstanzen der Legeliste

#### 3.2.1 Leitungszug-Gruppen

Beim Einteilen der Leitungszüge in vier Gruppen kommen in Gruppe 1 alle Leitungszüge, die Steckeinheiten-Lötstifte auf ein festes Potential schalten. In den Belegungsplänen trägt man bei diesen Lötstiften den Namen des betreffenden Potentials ein. Nach dem Sortieren ergibt sich für jedes Potential ein Leitungszug.

Diese Leitungszüge werden so unterteilt, daß jeweils nur die zu einer Steckeinheit gehörenden Punkte als ein Leitungszug gelten. Sie bestehen aber nur aus Eingängen, da die Potentialnamen nur an den Eingängen angeschrieben werden. Den Ausgang des betreffenden Potentials muß deshalb bei jedem Leitungszug das Programm zusetzen.

Gruppe 2 (Brückenverdrahtung) alle Leitungszüge, die nur nebeneinander bzw. um eine Steckeinheit auseinander liegende Punkte miteinander verbinden.

Gruppe 3 (Zeilenverdrahtung) alle Leitungszüge, die in einer Zeile liegende Punkte miteinander verbinden, ausgenommen die Leitungszüge der Gruppe 2.

Gruppe 4 (Restverdrahtung) alle restlichen, nicht zu den Gruppen 1, 2 oder 3 gehörenden Leitungszüge.

#### 3.2.2 Programm

Der Aufbau des Programms geht aus BILD 6 hervor. Da bei Gruppe 1 direkt die Potentialnamen auf der Trommel gesucht werden, weicht das Programm für diese Gruppe von dem dargestellten Schema ab. Die Gruppenzugehörigkeit für

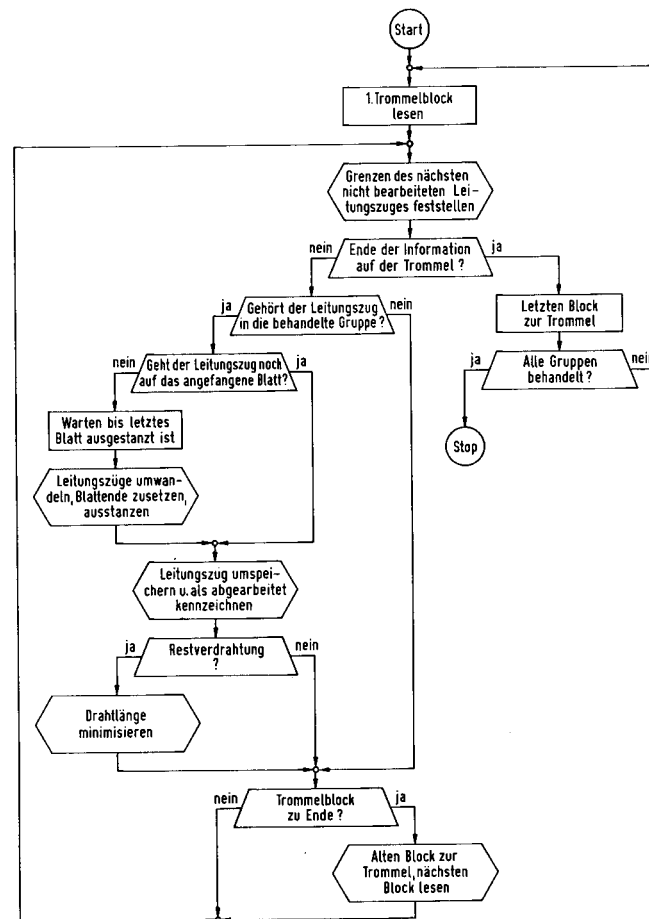


Bild 6 Flußdiagramm für das Ausstanzen der Legeliste

Gruppe 2 und 3 ergibt sich aus der jeweils berechneten Differenz zwischen aufeinanderfolgenden Adressen; die Differenz darf für die einzelnen Gruppen einen bestimmten Höchstwert nicht überschreiten. Alle letztlich noch nicht als bearbeitet gekennzeichneten Leitungszüge fallen dann unter Gruppe 4; bei diesen Leitungszügen sorgt ein besonderes Unterprogramm für das Minimieren der Drahtlänge.

Stets prüft das Programm, ob für den gerade anstehenden Leitungszug auf dem angefangenen Blatt noch Platz ist.

Trifft das zu, so wird der Leitungszug in einen Zwischenspeicher übertragen, der die Leitungszüge eines Blattes sammelt. Ist das Blatt voll oder geht ein neuer Leitungszug nicht mehr auf das Blatt, so wandelt ein Unterprogramm die gespeicherten Leitungszüge um und veranlaßt das Ausstanzen des ganzen Blattes in einem Block. Während des Ausstanzens können die nächsten Leitungszüge in dem Zwischenspeicher zusammengestellt werden. Wenn es sich um Leitungszüge der Gruppe vier handelt, läuft dabei zugleich der Minimisiervorgang ab. Beim Umwandeln setzt das Programm jedem Leitungszug, der nicht besonders gekennzeichnet ist, in zyklischer Folge eine Drahtfarbe zu. Die Leitungszüge erhalten eine Nummer. Nach Anfügen der zum Auffüllen des Blattes notwendigen Anzahl von Zeilenvor-schüben treten noch die Angaben für das Schriftfeld des Blattes hinzu.

### 3.23 Minimieren der Drahtzüge

Die Leitungszüge erhalten durch das Sortieren einen Verlauf nach BILD 7a, der im allgemeinen recht ungünstig ist. Das nachfolgend beschriebene Verfahren soll für diese Leitungszüge einen besseren Verlauf ermitteln. Um beim Minimieren der Leitungszüge mit möglichst wenig Zeit auszukommen, werden für das Minimieren aus den Adressen die Ordinaten und Abszissen der Punkte abgeleitet. Gleiche Koordinaten scheidern aus, um die Anzahl der Punkte nicht unnötig zu vergrößern (gleiche Koordinaten kommen vor, wenn ein Leitungszug mehrere Punkte einer Einheit berührt). Besteht der Leitungszug nur aus zwei verschiedenen Koordinaten, so entfällt der Minimierprozeß.

Vor dem eigentlichen Minimierprogramm werden die ungeradzahigen Zeilen und danach die geradzahigen Zeilen des Leitungszuges umgedreht, so daß sich Drahtzüge nach BILD 7b und c ergeben. Von dem Leitungszug, der sich dabei als kürzester herausstellt, geht das weitere Minimieren aus. Bei dem angewandten Verfahren wird ein Punkt nach dem anderen eingeordnet. BILD 8 führt als Beispiel das Einordnen von Punkt 6 (aus Bild 7b) vor Augen. Versuchsweise setzt das Programm ihn an das Ende, zwischen die

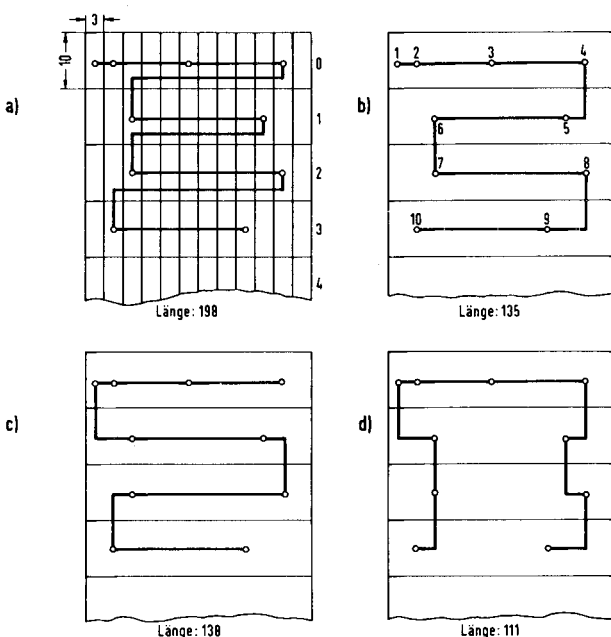


Bild 7 Beispiel eines Drahtzuges a) nach dem Sortieren, b) 2. und 4. Zeile vertauscht, c) 1. und 3. Zeile vertauscht und d) nach abgeschlossenem Näherungsverfahren

einzelnen Punkte und an den Anfang des Leitungszuges. In die kürzeste dieser Kombinationen wird der nächste Punkt eingeordnet usw. BILD 7d veranschaulicht das Ergebnis dieses Prozesses.

In einigen Fällen führt das beschriebene Näherungsverfahren zu einem längeren Drahtzug, als das einfache Vertauschen nach Bild 7a...c. Um eine schlechte Lösung zu vermeiden, gehen auch die durch Vertauschen gewonnenen Drahtzüge in den Entfernungsvergleich ein, so daß die ausgeschriebene Legeliste den kürzesten der vier Drahtzüge ausweist.

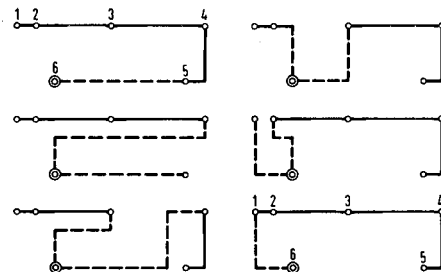


Bild 8 Näherungsverfahren; versuchsweises Einordnen von Punkt 6 in den vorgegebenen Leitungszug

Das Näherungsverfahren ließe sich noch etwas verbessern, doch lohnt sich der damit verbundene Aufwand nicht. Wie die bis jetzt erstellten Legelisten gezeigt haben, würden lediglich einige sehr selten vorkommende Leitungszüge verkürzt.

### 3.3 Erstellen des Gestellstromlaufplanes

Beim Gestellstromlaufplan ist zu dem Leitungszugnamen eines Lötstiftes der entsprechende Leitungszug auszuschreiben, damit man den Leitungszug von diesem Stift aus verfolgen kann. Dazu benötigt man eine Liste, in der die belegten Lötstifte einheitenmäßig zusammengefaßt, und eine zweite Liste, in der die Leitungszüge aufgeführt sind. Die Information auf der Trommel entspricht im sortierten Zustand der zweiten und im unsortierten, aber mit der Reihenfolge aller eingegebenen Lochkarten übereinstimmenden Zustand der ersten Liste.

Da der Speicherraum bei maximal 6000 Lötstiften nur für eine Liste ausreicht, ist es erforderlich, die Lochkarten ein zweites Mal einzulesen. Im Verlauf dieses Prozesses werden die belegten Punkte aus dem Lochkartenspeicher so umgespeichert, daß hinter jedem Lötstift Platz für maximal 50 Leitungszugpunkte bleibt, und dann die nicht belegten Punkte als erledigt gekennzeichnet. Ein Suchprogramm stellt die Leitungszüge für alle Punkte einer Lötstiftreihe zur gleichen Zeit zusammen, damit sich ein mehrmaliges Absuchen der Trommel erübrigt. Die Information wird bei diesem Suchen in Blöcken zu 100 Doppelworten gelesen und jeder Drahtzugname der belegten Lötstifte, einer nach dem anderen, mit dem letzten Drahtzugnamen dieser Blöcke verglichen. Ergibt der Vergleich bei einem Drahtzugnamen  $\leq$ , so kommt der gesuchte Leitungszug in diesem Block vor und wird hinter den betreffenden Drahtzugnamen übertragen. Wenn der betreffende Punkt ein Eingang ist, dann scheidet diese Übertragung außerdem alle Adressen, die einen Eingang darstellen, aus. Nach dem Übertragen des Leitungszuges wird der Leitungszug als erledigt gekennzeichnet und der Suchvorgang so lange fortgesetzt, bis alle Leitungszüge zusammengestellt sind.

Während nun das Ausstanzen der Leitungszüge beginnt, stellt ein zweiter Speicher die Leitungszüge der nächsten Lochkarte zusammen, wodurch sich die erforderliche Rechnerzeit we-

sentlich verkürzt. Vor dem Ausstanzen werden die Leitungszüge umgeformt, mit Formatangaben versehen und in den Ausstanzspeicher übertragen. Für die nicht belegten Punkte bleiben bei der Umformung Zeilen frei, damit man bei Änderungen eventuell einen Leitungszug von Hand nachtragen kann. Wenn eine ganze Lötstiftreihe nicht belegt ist, stanzt der Rechner, um unnötige Wartezeiten zu vermeiden, zugleich mit den Leitungszügen der nächsten Lochkarte die notwendigen Zeilenvorschübe. Hinter die Leitungszüge der letzten zu einer Einheit gehörenden Lötstiftreihe wird der Schriftkopf mit ausgestanzt. Im Schriftkopf steht neben Gestellname und Zeichnungsnummer die aus der letzten Lötstiftadresse abgeleitete Bezeichnung der Einheit und die Seitenzahl sowie bei Steckeinheiten deren Typenbezeichnung. Gehen die Leitungszüge einer Einheit nicht auf ein Blatt, was bei langen Leitungszügen der Fall sein kann, so laufen die restlichen Leitungszüge auf ein zweites Blatt über.

#### **4. Ausblick**

Es hat sich gezeigt, daß die manuelle Bearbeitungszeit für die Verdrahtungsunterlagen von 10...12 Wochen mit dem beschriebenen Verfahren auf eine Woche reduziert werden konnte. Wenn das Programm auf den Rechner des SEL-Rechenzentrums umgeschrieben würde, was jederzeit leicht möglich ist, ließe sich durch die Verwendung des Zeilendruckers die zum Ausschreiben der Lochstreifen benötigte Zeit einsparen; auch würde sich die Rechnerzeit von 4...6 Stunden auf etwa die Hälfte verringern.

Wesentlich für die Anwendung des Verfahrens ist die normierte Bauweise der Informatik-Gestelle. Durch entsprechende Änderungen im Programm dürfte es jedoch möglich sein, auch Unterlagen für die kompliziertere Verdrahtung der Gestelle innerhalb der Vermittlungs- und Übertragungstechnik über den Rechner erstellen zu lassen.