

MINCAL 4E

MINCAL 4N

SPEICHERPROGRAMMIERTE DIGITALRECHNER  
INFORMATIONSBESCHREIBUNG

- vorläufig -

HEINRICH DIETZ INDUSTRIE-ELEKTRONIK  
MUELHEIM/PUHR KOELNER STRASSE 115  
TELEFON 48 85 41 TELEX 08 56770

Bachmeier

	<u>Inhalt</u>	<u>Seite</u>
<u>Vorbemerkung</u>		2
1. <u>Technische Daten</u>		3
2. <u>Konzeption</u>		5
2.1) Gemeinsame Merkmale		6
2.2) Unterschiedliche Merkmale		7
2.3) Wortlänge		8
2.4) Komprimiertes Speichern, Doppelwort- und Gleitkommarechnung		8
2.5) Speicherkapazität		8
2.6) Geschwindigkeit		8
2.7) Befehlsumfang		9
2.8) Technologie		10
3. <u>Baugruppen</u>		11
3.1) Zentrale Recheneinheit		11
3.2) Feste Register		11
3.3) Prioritätssystem		12
3.4) Puffersystem		13
3.5) Parallel-Ein/Ausgang		13
3.6) Bedienung und Anzeige		14
4. <u>Programmfunktionen</u>		15
4.1) Darstellung von Daten		15
4.2) Darstellung von Instruktionen		16
4.3) Adressen		17
4.4) Indirekte Adressierung und Adreßmodifikation		17
4.5) Registeroperationen		17
4.6) Verzweigungen		17
4.7) Unterprogramme		18
5. <u>Programmbefehle</u>		19
5.1) Befehlsliste		19
5.2) Formatbefehle		30
6. <u>Programmierhilfen</u>		31
6.1) Umwandlungsprogramm		31
6.2) Weitere Programme		31
7. <u>Peripherie</u>		32
7.1) Schreibmaschine		32
7.2) Lochstreifengeräte		32
7.3) Puffer vom E-Typ		32
7.4) Weitere Puffer		34
7.5) Periphere Systemkomponenten		35

Vorbemerkung

Diese Informationsbeschreibung soll Interessenten und künftigen Benutzern der Digitalrechner MINCAL 4E und MINCAL 4N einen Überblick über deren Eigenschaften geben; sie ist also weder eine Programmierfibel noch ein "interface manual".

Für eine Kurzinformation genügt es zu lesen:

zur Einordnung in das Spektrum heutiger Digitalrechner	Kapitel 1 und 2
zum Verstehen des inneren Aufbaus	Kapitel 3
für den Programmierer	Kapitel 4, 5 und 6
für den Techniker, der einen MINCAL 4 richtig dimensionieren und ihn für seine Zwecke einsetzen will	Kapitel 7

Weitere Informationen stehen auf Wunsch gern zur Verfügung.

1. Technische Daten

MINCAL 4E

MINCAL 4N

Typ:	Speicherprogrammierter Einadreß-Digitalrechner mit fester Wortlänge und Festkomma	
Kernspeicher-Zugriff:	parallel	parallel
Interne Verarbeitung:	seriell	seriell
Taktfrequenz:	2.9 MHz	2.9 MHz
Wortzeit:	20 µs	20 µs
Wortlänge:	18 bit + Vorzeichenbit + Prüfbit	18 bit + Vorzeichenbit + Prüfbit
Speicherkapazität:	4096 Worte (4K)	min. 4096 Worte (4 K) max. 32768 Worte (32K)
Befehle:	60 Grundbefehle	60 Grundbefehle
Programmniveaus:	2 Ebenen mit unterschiedlicher Priorität	8 Ebenen mit unterschiedlicher Priorität
Arbeitsregister:	2 je Niveau	2 je Niveau
Indexregister:	4 je Niveau	4 je Niveau
Sensor-Eingänge:	7 gemeinsame + 14 niveaugebundene	7 gemeinsame + 56 niveaugebundene
Gepufferter Ein/Ausgang:	1 Wort-Puffer	min. 1 Wort-Puffer max. 8 Wort-Puffer
Parallel-Ein/Ausgang:	(Erweiterung)	(Erweiterung)
Standard-Schreibmaschine:	IBM 735 BCD 15 Z/s, 130 Z/Zeile	IBM 735 BCD 15 Z/s, 130 Z/Zeile
Standard-Streifenleser:	8-Kanal 30 Z/s	8-Kanal 300 Z/s
Standard-Streifenlocher:	8-Kanal 20 Z/s	8-Kanal 75 Z/s
Rechenkörper:	19"-Einschub Größe K 10 Einheiten = 445 mm hoch ca. 450 mm tief (einschl. Kernspeicher)	19"-Einschub Größe K 10 Einheiten = 445 mm hoch ca. 450 mm tief

	<u>MINCAL 4E</u>	<u>MINCAL 4N</u>
Stromversorgung:	19"-Einschub Größe C 3 Einheiten = 134 mm hoch ca. 450 mm tief	19"-Einschub Größe D 4 Einheiten = 178 mm hoch ca. 450 mm tief
Kernspeichereinheit (4K):		19"-Einschub Größe B 2 Einheiten = 89 mm hoch ca. 450 mm tief
Lochstreifengeräte:	19"-Einschub Größe F 6 Einheiten = 268 mm hoch ca. 450 mm tief  oder  in Gehäuse als Tischgerät	19"-Einschub Größe K 10 Einheiten = 445 mm hoch ca. 450 mm tief
Besondere Ausführungsformen:	Rechner mit Stromversorgung in Tischgehäuse ca. 500 mm breit x 650 mm tief x 600 mm hoch	Standard-Konsole mit Rechen- körper, Stromversorgung, Kernspeichereinheiten, Lochstreifeneinheit, Schreib- maschine und Raum für weitere Einschübe 19"-Doppelschrank ca. 1100 mm breit x 660 mm tief x 1560 mm hoch
Netzanschluß:	220 V 50 Hz einphasig ca. 800 VA	220 V 50 Hz einphasig ca. 1500 VA
Netzausfall-Schutz:	vorgesehen	vorgesehen
Verwendete Halbleiter:	Monolithische integrierte Schaltkreise und Silizium-Epitaxial- Planartransistoren	
Kernspeicher-Material:	Ferritkerne mit weitem Temperaturbereich	

## 2. Die Konzeption der Digitalrechner MINCAL 4E und MINCAL 4N

### 2.1) Gemeinsame Merkmale

Das Bedürfnis nach rascher und vollständiger Sammlung von Informationen und ihrer unmittelbaren Auswertung, der Wunsch nach Aufdeckung unbekannter Zusammenhänge und schneller Gewinnung von Resultaten und insbesondere die verfeinerte Steuerung von Arbeitsprozessen erfordern in zunehmendem Maße den Einsatz von datenverarbeitenden Geräten.

Besonderes Interesse wird kleineren Rechnern entgegengebracht, die einerseits nicht zu teuer, einfach in Bedienung und Programmierung und problemlos hinsichtlich der Wartung sind, auf der anderen Seite aber auch hinreichend schnell arbeiten, eine leistungsfähige Befehlsorganisation besitzen, sich den jeweiligen Aufgaben organisch anpassen und mit deren Vergrößerung erweitern lassen.

Das gilt - im technischen Bereich - für recht unterschiedliche Anwendungsgebiete, die von Rechenproblemen wissenschaftlicher Natur über die Sammlung und Verarbeitung von Meßdaten bis zur Steuerung von Prozessen reichen. Dank der technologischen Fortschritte, insbesondere in der Entwicklung von schnellen, kompakten Halbleiterbauteilen, auf den Gebieten der Speicher und der Periphergeräte, sind heute digitale Rechner verfügbar, die diesen Forderungen weitgehend genügen.

Die Digitalrechner MINCAL 4E und MINCAL 4N sind universelle, frei programmierbare Datenverarbeitungsgeräte, die für den beschriebenen Anwendungsbereich zugeschnitten sind. Als besondere Merkmale, die sie von anderen Rechnern gleicher Größenordnung unterscheiden, verdienen hervorgehoben zu werden:

18 bit Wortlänge - ausreichend groß für Meßdaten nahezu jeder Art und deren Verarbeitung,  
echte Darstellung positiver und negativer Zahlen durch zusätzliches Vorzeichen-bit,  
sehr klarer Befehlscode - gestattet leichte Interpretation,  
umfassende und wirkungsvolle Rechen-, Ablauf-, Ein/Ausgabe- und Formatbefehle - daher sowohl raumsparendes als auch einfaches, leicht erlernbares Programmieren,  
Befehlsfunktionen, die programmierte Doppelwort- und Gleitkommarechnungen vereinfachen - für das Rechnen mit höherer Genauigkeit,  
ausreichende Zahl von echten Indexregistern - zur Modifikation von Speicher- und externen Adressen,  
mehrere Programm niveaus mit getrennten Instruktionszählern, Index-, und Arbeitsregistern - für echte Vorrangverarbeitung in getrennten Programmabläufen,  
zahlreiche statische Sensor-Eingänge - für die Modifikation von Programmabläufen,  
eingebaute Wortpuffer für langsame Ein- und Ausgabe-Vorgänge - ermöglichen echte Parallelverarbeitung in der Peripherie und reduzieren den Programmaufwand sehr erheblich,  
zu den Puffern passende, teilweise eingebaute Anschlußschaltungen - machen die Verbindung mit Periphergeräten und zahlreichen externen Informationsträgern problemlos und kostensparend,

leistungsfähige und schnelle Standard-Periphergeräte - für betriebsmäßige Daten-Ein/Ausgabe in Klartext und über Lochstreifen,  
bequeme Bedienung und vielfältige manuelle Eingriffsmöglichkeiten - wichtig für Programm- und Anlagentests,  
eingebaute Steuerung für Programmierabläufe - nützlich für Programmierung, Tests, Fehlersuche und Änderungen,  
eingebaute Logik für Kernspeichertests,  
19"-Einschubbauweise - universell verwendbar und räumlich mit anderen Geräten zu kombinieren.

Alle diese Eigenschaften machen die Rechner der MINCAL 4-Baureihe zu wirklich komfortablen Datenverarbeitungsgeräten; ein gewisser Mehraufwand an "hardware", mit dem dieser Komfort erkauft wird, macht sich durch einfachere Programmierung, durch kostensparende Anpassung an die Peripherie und durch bequeme Bedienung, Überprüfung und Wartung bezahlt.

Diese Merkmale sowie Kapazität, Arbeitsgeschwindigkeit und Technologie erschließen den Digitalrechnern MINCAL 4E und MINCAL 4N viele Anwendungsmöglichkeiten, sei es als selbständige Geräte für wissenschaftlich-technische Berechnungen, sei es für Aufgaben der Meßwertverarbeitung oder als Prozeßrechner.

## 2.2) Unterschiedliche Merkmale

Beide Typen von Digitalrechnern sind weitgehend gleichartig aufgebaut, insbesondere in den verwendeten Baugruppen sowie hinsichtlich Wortlänge, Befehls- und Adressstruktur, Geschwindigkeit, Ein/Ausgängen, Bedienung usw. Die Programmsprache und damit die "software" sind für beide Typen völlig identisch.

Unterschiede bestehen - abgesehen von Baugröße und Bauform - jedoch hinsichtlich der Kapazität des Kernspeichers, der Prioritätslogik und des Puffersystems sowie der Standard-Lochstreifen-Geräte:

<u>MINCAL 4E</u>	<u>MINCAL 4N</u>
1 Kernspeichereinheit mit 4096 Worten, in den Rechenkörper eingebaut	1 bis 8 Kernspeichereinheiten zu je 4096 Worten, als getrennte Einschübe
2 Programm niveaus	8 Programm niveaus
1 eingebauter Puffer	1 bis 8 eingebaute Puffer
Stromversorgung für Rechner ausreichend	Stromversorgung für Rechner und ausgedehnte Peripherie ausreichend
Lochstreifeneinheit (Einschub oder Tischgerät) 30 Z/s Lese- und 20 Z/s Lochgeschwindigkeit	Lochstreifeneinheit (als Einschub) 300 Z/s Lese- und 75 Z/s Lochgeschwindigkeit
Ausführung in Einschubform oder als Tischgerät	Ausführung vorzugsweise in Schrankform (Standard-Konsole)

Der Rechner MINCAL 4E kann auf Wunsch auch mit der schnelleren Lochstreifeneinheit des MINCAL 4N ausgerüstet werden.

Damit ergeben sich für den Einsatz der beiden Rechner-Typen unterschiedliche Schwerpunkte:

MINCAL 4E

Betrieb als off-line-Rechner, insbesondere als Tischgerät, für wissenschaftlich-technische Datenverarbeitung mit beschränktem Programmumfang bzw. Speicherbedarf

Einsatz im on-line-Betrieb für nicht zu umfangreiche Aufgaben der Meßwert erfassung und -verarbeitung sowie zur Steuerung von Abläufen

MINCAL 4N

Betrieb als off-line-Rechner für Aufgaben mit erhöhtem Speicherbedarf, für wissenschaftlich-technische Aufgaben und im technisch-organisatorischen Bereich

Einsatz als on-line-Rechner in Anlagen zur Meßwertverarbeitung und Prozeßsteuerung, insbesondere dort, wo der Programmumfang oder die Zahl der Vorrangprogramme, der gleichzeitig zu bedienenden Peripheriegeräte und der Daten-Ein/Ausgänge die Möglichkeiten des MINCAL 4E übersteigen.

2.3) Wortlänge

Die Rechner MINCAL 4E und MINCAL 4N arbeiten mit fester Wortlänge. Das bedeutet, daß sie Informationen konstanten Formats verarbeiten, im Speicher aufheben und mit der Peripherie austauschen. Jede Programminstruktion und jeder numerische Wert erscheinen in Gestalt eines Wortes, dessen Länge und Aufbau Befehlsvielfalt und Rechengenauigkeit bestimmen.

Numerische Werte liegen in binärer Form vor; jedes Wort umfaßt 18 Binärstellen (18 bit) mit der Wertigkeit  $2^{-1}$  bis  $2^{-18}$ . Die Rechengenauigkeit beträgt  $2^{-18}$  bezogen auf eine numerische Operation, was einem Rechenfehler von weniger als  $5 \cdot 10^{-6}$  entspricht. Bedenkt man, daß z.B. technische Meßwerte häufig mit einer relativen Unsicherheit von  $10^{-3}$  bis  $10^{-4}$  behaftet sind, so kann man erwarten, daß die zusätzlichen Fehler, die bei anschließenden, langwierigen Rechenvergängen auftauchen, vernachlässigbar klein sein werden.

Bei der Ein- und Ausgabe haben Zahlenwerte im allgemeinen dezimale Form. Zur Darstellung einer Dezimalstelle werden 4 Binärstellen benötigt; ein 18-bit-Wort kann eine 4 1/2-stellige Dezimalzahl aufnehmen, entsprechend einer Auflösung von  $4 \cdot 10^{-5}$ . Für die Aufnahme von 3 alphanumerischen Zeichen (zu je 6 bit) ist ein Wort nötig.

Den 18 Binärstellen jedes Wortes ist ein Vorzeichen-bit zugeordnet; positive und negative Zahlen werden echt unterschieden. Im Kernspeicher hat jedes Wort eine weitere Binärstelle, die zur Kontrolle der Gesamtzahl aller Binärstellen mit Inhalt "1" (Prüfbit) dient. Ein Kernspeicherwort hat also 20 bit.

Die Wortlänge ist auch für die Vielfalt der Programmbefehle entscheidend. Eine Instruktion hat die Länge eines Wortes; sie ist außerordentlich klar aufgebaut: 6 Binärstellen charakterisieren den Befehl und gestatten die Unterscheidung der 60 verschiedenen Operationsarten. 3 weitere bits ergänzen den Befehl, z.B. in seiner Anwendung auf bestimmte Register im Kernspeicher. Die restlichen 9 Binärstellen geben die Adresse an, auf die sich der Befehl bezieht.

#### 2.4) Komprimiertes Speichern, Doppelwort- und Gleitkommarechnung

Die Wortlänge von 18 bit ist für genaue numerische Verarbeitung sehr nützlich. Informationen wesentlich geringerer Länge kann man aber auch zu mehreren in einem Wort unterbringen. Der Befehlscode erlaubt einfache Programmabläufe zum Zusammenfügen und Trennen der komprimiert abgespeicherten Informationen.

Andererseits kann, insbesondere bei wissenschaftlichen Programmen, die Wortlänge nicht ausreichen, um eine genügende Rechengenauigkeit zu erzielen. In diesem Fall wird mit doppelter Wortlänge gearbeitet, d.h. mit zwei Wörtern zu je 18 bit, die eine zusammenhängende Zahl mit einer Auflösung von  $2^{-36}$  enthalten. Die Rechenbefehle enthalten bereits Ansätze hierfür, so daß man einfache und relativ schnelle Programmabläufe für das Rechnen mit doppelter Wortlänge gestalten kann.

Die Geräte der MINCAL 4-Baureihe sind Festkomma-Rechner; bei Multiplikation und Division ist das Komma stets am gleichen Ort, und zwar links von der Binärsielle mit der höchsten Wertigkeit; die 18 bit eines Wörtes haben die Bedeutung  $2^{-1}$  bis  $2^{18}$ . Wo die dadurch gegebene relative Genauigkeit zwar ausreicht, die zu verarbeitenden Zahlen aber sehr verschiedene Größenordnung haben, empfiehlt sich das Rechnen mit gleitendem Komma. Hierbei wird eine Zahl durch eine Mantisse (zwischen 0,5 und 1) und einen Exponenten (Pctenz von 2) dargestellt. Die Rechner sind mit Befehlen ausgerüstet, die den Aufbau verhältnismäßig einfacher Programme für Gleitkomma-Rechnung zulassen. Für noch höhere Ansprüche werden Doppelwort- und Gleitkomma-Rechnung kombiniert.

#### 2.5) Speicherkapazität

Der Rechner MINCAL 4E enthält einen Kernspeicher mit 4096 Wörtern Kapazität. Da jedes Wort einen relativ großen Zahlenwert über einen fein gegliederten Programmbefehl enthalten kann, lassen sich mit 4096 Wörtern bereits recht umfangreiche Programme beherrschen.

Die Speicherkapazität des MINCAL 4N beträgt in der Grundausführung ebenfalls 4096 Worte, läßt sich jedoch um bis zu 7 weitere Einheiten von je 4096 Plätzen erweitern, entsprechend einer Gesamtkapazität von 32768 Wörtern.

Der Kernspeicher ist in einzelne Bereiche zu je 512 Wörtern unterteilt. Mit Hilfe der 9 Adreßbits einer Instruktion können, je nach Vorzeichen, die 512 Worte des laufenden Bereichs (dem die Instruktion angehört) bzw. die 512 Worte des ersten Bereichs (Bereich  $\emptyset$ ) adressiert werden. Zugang zu anderen Bereichen wird durch indirekte Adressierung geschaffen.

Man kann den Kernspeicher mit Instruktionen, Festwerten oder Variablen in beliebiger Weise belegen. Allerdings haben bestimmte Speicherplätze feste Funktionen (Startadresse, 4 Indexregister, Programmstand, Arbeits- und Hilfsregister); je Programmlevel sind 8 Plätze für diese Funktionen reserviert. 10 weitere Plätze sind reserviert für Festwerte im Zusammenhang mit Code-Umwandlungsbefehlen.

#### 2.6) Geschwindigkeit

Ein wichtiges Kriterium für die Leistungsfähigkeit eines Rechners ist seine Arbeitsgeschwindigkeit bzw. die Dauer der einzelnen Operationen. Als Maß hierfür dient die "Wortzeit", d.h. die Zeit für Lesen, Verarbeiten und Zurückschreiben eines Wortes. Jede Operation setzt sich aus einer Reihe von Wortzeiten zusammen.

Die Wortzeit der Rechner MINCAL 4E und MINCAL 4N beträgt 20 Mikrosekunden. In dieser Zeit wird z.B. ein 18-bit-Wort aus dem Kernspeicher gelesen, zu einem anderen Wort addiert und in den Kernspeicher zurückgeschrieben. Das Lesen und Schreiben eines Wortes erfolgt bit-parallel; dagegen wird das gelesene Wort bit für bit (seriell) verarbeitet. Diese Anordnung ergibt ein Optimum von Aufwand und Arbeitsgeschwindigkeit; sie bietet sich durch die Kombination eines langsameren, temperaturstabilen Kernspeichers mit sehr schnellen integrierten Schaltkreisen an.

Die Operationszeiten liegen zwischen 60 µs (z.B. Festwert-Addition) und 1600 µs (Division). Aus dieser Sicht ist die Einstufung der MINCAL 4-Baureihe als mittelschnelle Rechner nach dem heutigen Stand der Technik gerechtfertigt.

Allerdings darf man dabei nicht außer Acht lassen, daß die reelle Arbeitsgeschwindigkeit einer Datenverarbeitungsanlage nicht allein von der Zeitdauer interner Abläufe bestimmt wird, sondern auch von Geschwindigkeit der Peripheriegeräte und des Datenaustauschs mit der Umwelt. Will man die reelle Arbeitsgeschwindigkeit erhöhen, so muß man dafür sorgen, daß langsame Ein-Ausgabevorrichtungen gleichzeitig und unabhängig vom eigentlichen Rechner arbeiten und doch so lückenlos wie möglich vom Rechner bedient werden. Durch Anwendung eingebauter Puffer und durch die Prioritätslogik wird dieser Forderung entsprochen.

## 2.7) Befehlsumfang

Die Rechner der MINCAL 4-Baureihe besitzen eine Operationslogik für 60 verschiedene Befehle.

Sie umfassen Vorzeichenbehandlung, Schiften um eine angegebene Zahl von Binärstellen in beiden Richtungen und Linksschiften zwecks Normierung; Transportoperationen zwischen den festen, niveau-zugeordneten Registern und beliebigen Plätzen im Kernspeicher; Addition, Subtraktion und logische Operationen; Multiplikation, Division sowie spezielle Befehle zur Umwandlung von Dezimal- in Binärzahlen und umgekehrt; ferner Programmverzweigungen, die entweder unbedingt oder abhängig von Bedingungseingängen, von Vorzeichen oder vom Inhalt der festen Register ablaufen; Verzweigungen in Unterprogramme, Prioritätswchsel, Anhalten und Programmende; außerdem Ein-Ausgabebefehle in Verbindung mit externen Kreisen, insbesondere den Puffern.

Bei allen numerischen, logischen und Transportoperationen kann die Adresse über eines der 4 Indexregister (additiv) verändert werden; anstelle der Adresse kann man beim Lade- und Addierbefehl einen Festwert von 9 Binärstellen Länge angeben. Bei allen speicherbezogenen Operationen ist indirekte Adressierung möglich, d.h. es wird ein Speicherplatz benutzt, dessen Adresse in dem in der Instruction angegebenen Platz steht.

Die Gruppe der Ein/Ausgabebefehle unterscheidet zwischen der Auslösung von Eingabe- und Ausgabevorgängen und der Übernahme des Pufferinhalts; Anhalten oder Weiterlaufen des Programms sowie die Prioritätsstart-Berechtigung ist programmierbar. In der Adresse ist der angeschlossene Datengeber bzw. -empfänger angegeben; die Adresse ist über Indexregister modifizierbar.

In Verbindung mit bestimmten Puffern enthält die Adresse alle Hinweise für die Geräteauswahl und das Ein/Ausgabeformat (Vorzeichen, Zeichendarstellung, Stellenzahl und Endzeichen). Diese Art von Instructions erleichtert die Programmierung außerordentlich; mit einem einzigen Befehl kann ein Wort in beliebigem Format aufgenommen bzw. ausgegeben werden.

Der Rechner MINCAL 4E ist in der Grundausführung ohne Operationslogik für Multiplikation, Division und Umwandlungsbefehle lieferbar; nachträgliche Erweiterung ist durch Einsticken einer Baugruppe in einen verdrahteten Platz möglich.

Die einzelnen Programmbefehle sind in Kapitel 5 beschrieben.

## 2.8) Technologie

Die Rechner der MINCAL 4-Baureihe sind vorwiegend aus monolithischen integrierten Schaltkreisen aufgebaut, die sich durch kompakte Bauweise, kurze Schaltzeiten und hohe Zuverlässigkeit auszeichnen. Bei der Anpassung an die Peripherie werden, wo der Leistungspegel dies erfordert, Schaltungen aus diskreten Silizium-Halbleitern verwendet.

Der Kernspeicher enthält Ferritkerne mit weitem Arbeitstemperaturbereich.

Alle Schaltungen sind auf steckbaren, gedruckten Platten aus Epoxydharz-Glasgewebe angeordnet; Rechenkörper, Stromversorgung, Kernspeicher und die Anpassung an die Peripherie sind in 19"-Einschüben enthalten.

### 3. Die Baugruppen der Digitalrechner MINCAL 4E und MINCAL 4N

#### 3.1) Zentrale Recheneinheit

Die zentrale Recheneinheit enthält 2 Werke A und C mit je 18 Flipflops und ein Werk B mit 9 Flipflops, die als Schieberegister ausgeführt sind. Das C-Werk übernimmt wortweise (bit-parallel) die Informationen aus dem Kernspeicher und gibt sie in gleicher Weise in den Speicher zurück. Die Adresse des Kernspeicherplatzes wird von 9 Binärstellen des A-Werks bestimmt, das im Übrigen auch zum zwischenzeitlichen Aufheben von Daten benutzt wird. Die internen Verarbeitungsoperationen bestehen aus einem fortlaufenden Austausch von Daten zwischen Kernspeicher, C- und A-Werk, wobei die Übertragung von C nach A und zurück sowie der Werke in sich selbst seriell erfolgt und von einer Verknüpfungslogik bestimmt wird. Das B-Werk übernimmt zu Anfang jeder Operation Befehl und Ergänzung (9 bits).

Die Stellen 1 bis 18 der Werke A und C entsprechen den Binärstellen  $2^{-1}$  bis  $2^{-18}$ ; für Vorzeichen- und Prüfbit sind dem C-Werk zwei Flipflops zugeordnet, die jedoch nicht Bestandteile des Schieberegisters sind.

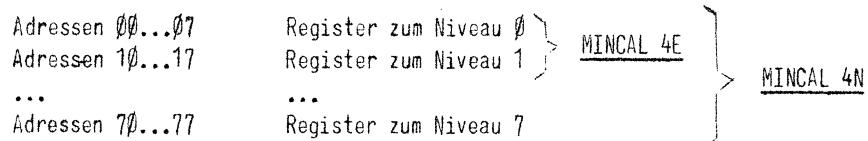
#### 3.2) Feste Register

Die 3 Werke A, B und C haben nur während der Dauer einer Operation Bedeutung, wobei der Ablauf von ihrem Inhalt und ihrer Verknüpfung abhängt. Am Ende der Operation sind alle für den weiteren Programmverlauf wichtigen Informationen in festen Plätzen des Kernspeichers enthalten, die bei der nächsten Operation nach Bedarf erneut abgefragt werden. Diese "festen" Register sind vom Standpunkt der Programmierung allein interessant. Je Niveau sind folgende 8 Register vorgesehen:

<u>Adresse</u>	<u>symbol.Bez.</u>	<u>Funktion</u>
Ø	Ø	Startadresse
1	1	Indexregister 1
2	2	Indexregister 2
3	3	Indexregister 3
4	4	Indexregister 4
5	N	Programmstand
6	W	Arbeitsregister
7	X	Hilfsregister

Im Platz Ø steht die Adresse der ersten Instruktion eines Programms, während Register N die Adresse der jeweils folgenden Instruktion enthält. Das Arbeitsregister W nimmt das Ergebnis der jeweiligen Operation auf; eventuelle Nebenergebnisse stehen im Hilfsregister X. Die 4 Indexregister dienen zur additiven Adreßmodifikation.

Der Rechner arbeitet stets mit dem Satz von Registern, die dem jeweiligen Programmlevel zugeordnet sind. Die ersten Plätze des Kernspeichers sind mit den festen Registern belegt, wobei (in oktaler Schreibweise) die "1er-Adresse" die Art des Registers (siehe oben), die "10er-Adresse" das Niveau bezeichnen, also:



Niveau 7 ist dasjenige mit der höchsten Prioritätsstufe.

Jedes dieser Register ist wie alle übrigen Kernspeicherplätze über seine Adresse erreichbar. Bei einer Reihe von Befehlen kann ein Register (des betreffenden Niveaus) jedoch auch speziell ange- sprochen werden, und zwar durch Einsetzen der 1er-Adresse in die Ergänzungsstelle der Instruktion. Im übrigen werden insbesondere die Register  $\emptyset$ , N, W und X von der Operationslogik selbsttätig an- gewählt.

### 3.3) Prioritätssystem

Zeitsparende Programme in Verbindung mit Peripheriegeräten, die über die Puffer angeschlossen sind, erfordern die Möglichkeit von Programmunterbrechungen. Das gleiche gilt für den häufig vor- kommenden Fall, daß während eines laufenden Programms dem Rechner von außen Daten angeboten werden, die sofort übernommen werden müssen, oder ganz allgemein dann, wenn mehrere Programme, die langsame Periphergeräte und Daten-Ein/Ausgänge benutzen, gleichzeitig laufen.

Das Prioritätssystem des MINCAL 4E sieht 2 Programm niveaus vor gegenüber 8 Niveaus beim MINCAL 4N. Sie werden von außen über Startleitungen (oder auch vom Programm her) erregt. Wird das höhere Niveau gestartet, so wechselt das Programm nach Ablauf der jeweiligen Operation, um nach beendetem Vorrang- programm das unterbrochene Programm an der gleichen Stelle fortzusetzen.

Jeder Programmablauf wird durch einen Prioritätsstart begonnen, d.h. einer der Starteingänge, die den Niveaus zugeordnet sind, erhält ein Signal. Läuft kein Programm oder keines mit höherem Rang, so wird das gestartete Programm sofort begonnen; andernfalls wird der Start gespeichert, bis kein Programm mit höherem Rang mehr läuft.

Der Programmablauf in einem Niveau wird unterbrochen

durch Start eines höheren Niveaus, wobei die laufende Operation noch zu Ende geführt wird; bei Anhalten oder Ende des Vorrangprogramms läuft das unterbrochene Programm mit der nächsten Instruktion automatisch weiter,

durch einen Halt-Befehl im betreffenden Programm; mit einem neuen Start wird das Programm von der unterbrochenen Stelle an weiterverfolgt,

durch einen End-Befehl; der nächste Start bewirkt, daß das Programm von vorn beginnt, d.h. mit der im Register  $\emptyset$  angegebenen Instruktionsadresse.

Statt über einen Starteingang kann jedes Programm auch durch einen Befehl ("Prioritätswechsel"), der in einem anderen Programm niveau liegt, gestartet werden, oder durch den zugehörigen Puffer, wenn dessen Arbeitszyklus beendet ist.

Die Rechner sind für das Arbeiten mit Vorrangprogrammen konzipiert. Die Anzahl der Steuer- und Rechenwerke ist auf ein Minimum reduziert, und da deren Inhalt (Programm-Stand und Dateninhalt) am Ende einer Operation ohnehin in festen Plätzen des Kernspeichers enthalten ist, erfolgt der Niveauwechsel unverzögert, d.h. sofort mit der nächsten Operation des Vorrangprogramms.

Für jedes Programm niveau ist ein getrennter Satz von Speicherplätzen mit fester Funktion vorgesehen (Startadresse, 4 Indexregister, Programm-Stand, Arbeits- und Hilfsregister); der Rechner bedient sich automatisch der Register des jeweiligen Niveaus. Dadurch sind völlig unabhängige Programmabläufe in den einzelnen Vorrangebenen möglich.

Falls Puffer und externe Einrichtungen ein gemeinsames Programm niveau starten, ermittelt der Rechner die Herkunft des Starts über Bedingungseingänge. Die Bedingungseingänge ("Sensoren"), die auf Programmverzweigungen wirken, bilden auch ein bequemes Hilfsmittel zur Modifikation von Verarbeitungsabläufen.

#### 3.4) Puffersystem

Der Datenaustausch mit der Umwelt erfolgt vorzugsweise über eingebaute Puffer, die Informationen im Umfang eines Wortes vom zentralen Rechteil übernehmen und nach außen weitergeben oder von der Peripherie aufnehmen und dem Rechner übermitteln. Die Puffer haben eigene, vom Rechner unabhängige Arbeitszyklen. Während der Daten-Ein/Ausgabe, deren Zeitdauer den äußeren Verhältnissen angepaßt ist, kann der Rechner interne Programmabläufe durchführen oder weitere Puffer bedienen.

Jeder Puffer enthält ein 18-bit-Datenregister mit einer Vorzeichen-Stelle und ein Adreßregister, welches den externen Geber oder Empfänger der Information auswählt bzw. den Ablauf des Ein/Ausgabe-Vorgangs bestimmt. Am Ende des Arbeitszyklus' löst der Puffer einen Programmstart aus, wenn dies programmiert war. Die Eigenschaften der Puffer sind in (7.3) und (7.4) beschrieben.

Der Puffer des MINCAL 4E sowie der in der Grundausführung des MINCAL 4N stets enthaltene erste Puffer (Puffer Ø) sind vom E-Typ und gestatten den Anschluß der Standard-Periphergeräte (Schreibmaschine und Lochstreifeneinheit), je eines dritten Eingabe- und Registriergeräts sowie von je 512 externen digitalen Datengebern und -empfängern.

Im Rechner MINCAL 4N können bis zu 7 weitere Puffer verschiedenen Typs untergebracht werden. Sie sind den Programm niveaus 1 bis 7 zugeordnet, werden also von Befehlen in den jeweiligen Niveaus ausgelöst und geben bei Ende des Pufferzyklus' einen Startbefehl an das betreffende Niveau zurück. Der Puffer Ø dagegen kann von jedem Niveau aus gestartet werden (in der betreffenden Instruktion durch eine 1 in der Vorzeichenstelle gekennzeichnet); erfolgt am Ende des Pufferzyklus' ein Start, so betrifft er das auslösende Programm niveau.

Der Puffer des MINCAL 4E kann von beiden Programm niveaus bedient werden; der von ihm ausgelöste Start geht jedoch stets auf das Niveau 1.

#### 3.5) Parallel-Ein/Ausgang

Der Informationsaustausch mit der Umgebung kann auch über den Parallel-Ein/Ausgang des Rechners erfolgen. Für Ausgabe und Übernahme von Daten ist je ein 19-bit-Kanal vorgesehen sowie für die Identifikation des Empfängers bzw. Geberts ein 9-bit-Adreßkanal, der maximal 512 externe Adressen anwählen kann.

Der Parallel-Ein/Ausgang ist vor allem für schnelle Daten-Ein/Ausgabevorgänge bestimmt; für Übernahme oder Ausgabe eines Wertes stehen 20 µs zur Verfügung. Sind die peripheren Einrichtungen dieser Geschwindigkeit nicht gewachsen, so kann der Rechner bis zur vollständigen Übernahme oder Bereitstellung der Information anhalten.

Die Befehle für den Parallel-Ein/Ausgang entsprechen denen des Puffersystems; man kann außerhalb einen Parallelpuffer vorsehen; ein von diesem ausgehender Prioritätsstart wird einem der Programm niveaus fest zugeordnet.

Die Rechner werden auf Wunsch mit Parallel-Ein/Ausgang ausgerüstet; nachträgliche Erweiterung um diese Eigenschaft ist möglich.

### 3.6) Bedienung und Anzeige

Es sind umfangreiche Möglichkeiten für Programmierung und Überwachung vorgesehen; alle zugehörigen Bedienungs- und Anzeigeelemente sind in der Frontplatte des Rechenkörpers zusammengefaßt.

Der Rechner enthält fest verdrahtete Steuerfunktionen zum Einlesen, Auslochen, Ausdrucken und manuellen Eingeben von Programmen sowie zum manuellen Erstellen, Duplizieren und Übersetzen von Programmstreifen, wobei Schreibmaschine, 8-Kanal-Locher und 8-Kanal-Leser benutzt werden.

Eingabe und Registrierung auf der Schreibmaschine erfolgen oktal, d.h. es werden jeweils 3 bit eines Wortes zusammengefaßt und mit Hilfe der Ziffern Ø...7 ausgedrückt. Diese Darstellung entspricht exakt dem Aufbau der Programminstruktionen und ist daher leicht zu interpretieren. Der Programmausdruck auf der Schreibmaschine gibt die oktale Adreßstruktur wieder: 8 Worte je Zeile, nach jeweils 64, 512 bzw. 4096 Werten werden 1, 2 bzw. 3 Leerzeilen vorgesehen.

Die 8-Kanal-Programmstreifen enthalten je Wort ein Vor- bzw. Zwischenraumzeichen sowie 3 alphanumerische Zeichen.

Es besteht die Möglichkeit, den Inhalt jedes Speicherplatzes anzuzeigen, auszudrucken oder über die Schreibmaschine zu ändern. Die Adresse des Platzes wird oktal entweder an einem Schaltersatz oder über die Schreibmaschine vorgewählt.

Der Inhalt der Werke C und A kann einschließlich Vorzeichen an 18 + 1 oktal eingeteilten Binär anzeigen abgelesen werden, das Befehlsregister B an 9 Binäranzeigen. Prüfbit, Fehlermeldungen und einige wichtige interne Zustände werden ebenso angezeigt wie der Zustand der Puffer, des Prioritätssystems und der Sensoreingänge.

Jedes Programm niveau läßt sich von Hand starten oder fest vorwählen, wobei im letzteren Falle alle anderen unwirksam sind; jedes Vorrangprogramm kann angehalten oder beendet werden. Die Sensoreingänge können unwirksam gemacht und durch 7 Schalter nachgebildet werden.

Besondere Vorwahlschalter gestatten zum Zwecke des Programmtests Anhalten des Ablaufs zu Beginn jeder Operation, nach der Instruktionsabfrage, am Ende jeder Wortzeit und bei einer vorgewählten Instruktionsadresse. Über 8 Tasten kann man den Inhalt der 8 festen Register des jeweiligen bzw. vorgewählten Niveaus zur Anzeige bringen.

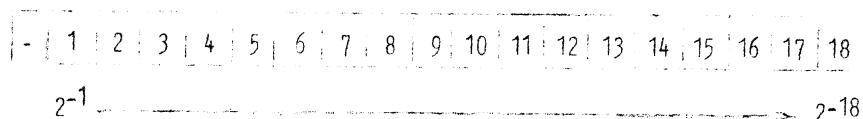
Einige Anzeigelampen für Betriebsspannung, Ruhezustand, Oszillator-Lauf und Fehlermeldung sowie Tasten und Schalter für Nullstellung, Langsamgang und Normal-Einstellung der Schreibmaschine vervollständigen das Bedienungsfeld.

Der Rechner besitzt ein verdrahtetes Prüfprogramm für den Kernspeicher, dem alle Plätze unterworfen werden können. Bei einem Ausfall bleibt das Programm stehen und zeigt Adresse und Inhalt des fehlerhaften Platzes an.

#### 4. Die Programmfunctionen der Digitalrechner MINCAL 4E und MINCAL 4N

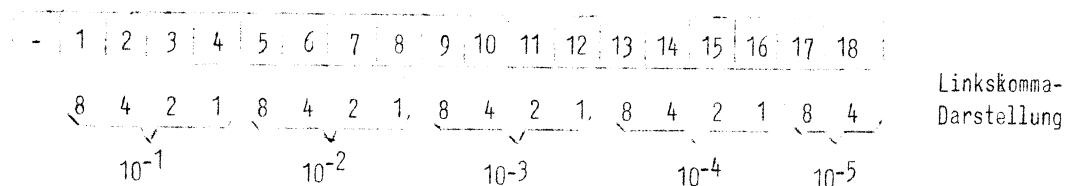
##### 4.1) Darstellung von Daten

Ein 18-bit-Wort kann eine 18-stellige Binärzahl aufnehmen; ein zusätzliches bit kennzeichnet das Vorzeichen:

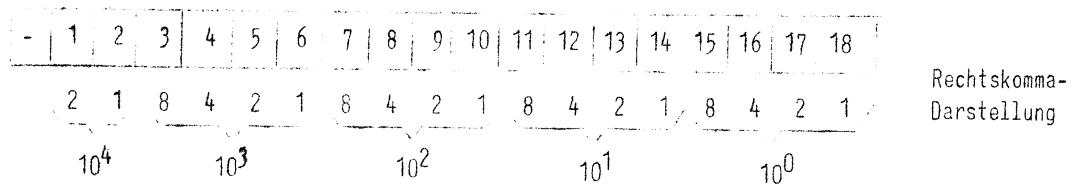


In dieser Form müssen numerische Werte stets dann vorliegen, wenn man sie addieren, subtrahieren, multiplizieren oder dividieren will. Bei Multiplikation und Division ist zu beachten, daß das Komma links von der höchsten Binärstelle ( $2^{-1}$ ) steht.

Dezimalzahlen, die vor allem beim Datenaustausch mit der Peripherie des Rechners auftreten, sind 4 1/2-stellig im Code 8-4-2-1 (BCD-Code) in einem 18-bit-Wort enthalten:

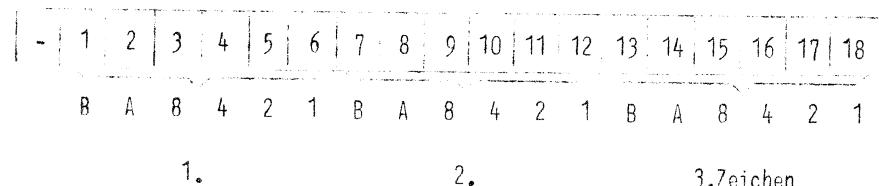


oder:



Zur numerischen Verarbeitung müssen Dezimalzahlen in Binärzahlen umgeformt werden; dafür stehen Umwandlungsbefehle zur Verfügung, und zwar sowohl für Linkskomma- (Bruch-) Darstellung als auch für Rechtskomma (als ganze Zahl).

Eine dritte Möglichkeit sind 3 alphanumerische Zeichen zu je 6 bit, die in einem Wort Platz finden:



Die Darstellung folgt dem 6-bit-Code; die Stellen B-A-8-4-2-1 entsprechen den Spuren 7-6-4-3-2-1 auf dem 8-Kanal-Lochstreifen. Mit diesem Code lassen sich insgesamt 63 verschiedene Charaktere (10 Ziffern 0...9, 26 Buchstaben A...Z und 27 Symbole) sowie das Zeichen "Zwischenraum" darstellen.

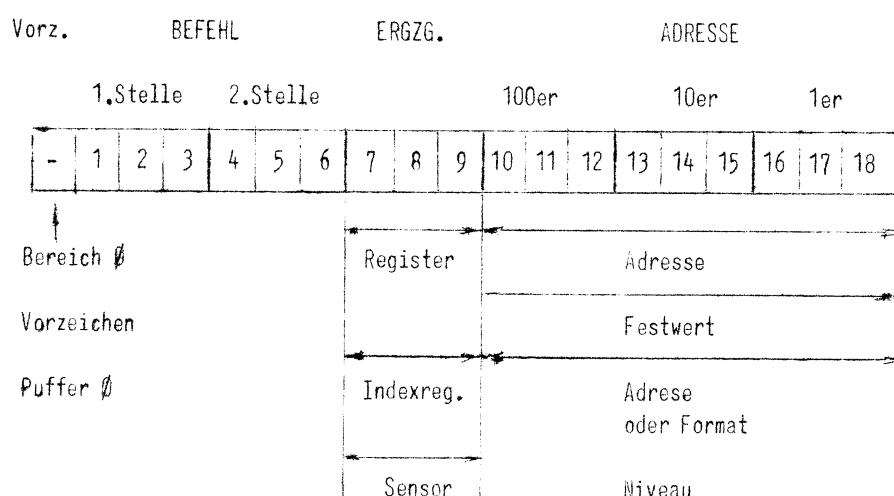
#### 4.2) Darstellung von Instruktionen

Eine Programminstruktion nimmt stets ein 18-bit-Wort sowie dessen Vorzeichenstelle in Anspruch. Das Wort wird dabei in 6 Stellen zu je 3 bit unterteilt aufgefaßt; eine in "Maschinencode" geschriebene Instruktion besteht daher aus einer 6-stelligen Oktalzahl (jede Stelle kann die Ziffern 0...7 enthalten) sowie einem Vorzeichen.

Die Instruktion ist wie folgt gegliedert:

Vorzeichen-Stelle	Vorzeichen/Speicherbereich/Puffer
Binärstellen 1...3	1. Stelle
" 4...6	2. Stelle Befehl
" 7...9	Ergänzung
" 10...12	100er-Adresse
" 13...15	10er-Adresse
" 16...18	1er-Adresse

Je nach Befehl kann den einzelnen Stellen einer Instruktion folgende Bedeutung zukommen:



#### 4.3) Adressen

Kernspeicher- und externe Adressen umfassen 9 bit und werden im Maschinencode als 5-stellige Oktalzahlen geschrieben. Sie sind in den Stellen 10 bis 18 der Instruktion enthalten. Die 4096 Kernspeicherplätze des MINCAL 4E sind in oktaler Schreibweise mit 000...7777 nummeriert, die maximal 32768 Plätze des MINCAL 4N mit 0000...77777, wobei in einer Instruktion jeweils die 3 letzten Oktalstellen der Adresse angegeben sind. Bei bestimmten Puffern haben diese Stellen sowie die Ergänzungsstelle die Bedeutung von Auswahl- und Formatbefehlen:



#### 4.4) Indirekte Adressierung und Adreßmodifikation

Bei allen speicherbezogenen Operationen ist neben der direkten auch indirekte Adressierung möglich; der Operand wird aus einem Platz geholt bzw. in einen Platz gebracht, dessen Adresse in dem Platz steht, der in der Instruktion angegeben ist. *Die indirekte Adressierung ist auch mehrfach möglich.*

Bei allen Transport-, numerischen, logischen und Ein/Ausgabebefehlen kann die Adresse über eines der 4 Indexregister modifiziert werden, indem dessen Inhalt zur Adresse addiert wird. Die Nummer des Registers wird in der Ergänzung angegeben. Auch die externen Adressen des Parallelausgangs bzw. der Puffer sind durch Indexregister modifizierbar.

Bei indirekter Adressierung wird die indirekte Adresse modifiziert, nicht die angegebene.

#### 4.5) Register-Operationen

Beim Addier- und beim Ladebefehl kann statt der Adresse ein positiver oder negativer 9-bit-Festwert programmiert werden ( $2^{-10} \dots 2^{-18}$ ), der als Operand aufgefaßt wird. Die Festwertoperation ist auf alle 8 festen Register anwendbar; das Register wird in der Ergänzung angegeben. Diese Befehle sind vor allem für das Setzen, Erhöhen und Vermindern der Indexregister sowie der Register W und X nützlich.

Besondere Operationen erlauben die Übertragung des W-Register-Inhalts in jedes der 7 anderen festen Register des gleichen Niveaus und zurück; diese Operationen sind vor allem deshalb nützlich, weil sie unabhängig vom Programm niveau das richtige Register ansprechen.

#### 4.6) Verzweigungen

Programmverzweigungen bewirken, daß der Programmablauf nicht mit der nächsten Instruktion fortgesetzt wird, sondern an einer beliebigen anderen Stelle; hierzu wird das Register N auf einen neuen Wert gesetzt (Verzweigungsadresse). Bei der direkten Verzweigung wird dieser Wert in den Adreßstellen angegeben; die indirekte Verzweigung dagegen findet die neue Adresse in einem Platz, dessen Adresse programmiert wird.

Verzweigungen finden entweder unbedingt statt oder nur dann, wenn der programmierte Bedingungseingang (Sensor) erregt ist. Eine Reihe von Verzweigungsbefehlen ist inhaltsbedingt; der Programmprung findet nur statt, wenn das in der Ergänzungsstelle angegebene Register Nullinhalt, positiven oder negativen Inhalt hat.

#### 4.7) Unterprogramme

In Unterprogramme kann direkt oder indirekt verzweigt werden; das Aufrufen eines Unterprogramms ist unbedingt oder von Bedingungseingängen abhängig programmierbar. Das Register, in dem die Rückkehradresse gespeist werden soll, muss programmiert werden. Dieses Register ist der, ausgetauschte Register.

Beim Unterprogrammsprung wird die um 1 erhöhte Instruktions-Nummer des Befehls in den adressierten Platz übertragen (Rückkehradresse), und das Programm läuft mit dem nächstfolgenden Platz weiter.

Am Ende des Unterprogramms ist ein indirekter, auf die Rückkehradresse bezogener Verzweigungsbefehl vorzusehen, der zum Hauptprogramm zurückführt. Ist es nötig, dem Inhalt des Registers, in dem die Rückkehradresse abgespeist wird, das N. Register zu aktivieren.

### 5.2) Formatbefehle

Bei bestimmten Puffern, die mit Registratur- und Eingabegeräten verbunden sind, hat die Adresse die Bedeutung eines Formatbefehls.

Um die Programmierung zu erleichtern, haben diese Befehle besondere Symbole:

61	IFG	(statt IBG)
62	IFR	(statt IBR)
63	IFP	(statt IBP)
65	OFW	(statt OBW)
66	OFR	(statt OBR)
67	OPP	(statt OBP)

Nach dem symbolischen Befehl ist in ebenfalls symbolischer, leicht merkbarer Schreibweise anzugeben:

- die Nummer des Eingabe- oder Registriergeräts,
- das Vorzeichen (ohne, mit),
- die Darstellung (oktal, dezimal mit Links- oder Rechtskomma, alphanumerisch),
- die Stellenzahl (0 bis 6),
- das Endzeichen (ohne Zwischenraum, Tabulator, Zeilenwechsel, Rücklauf, Punkt, Schwarz-, Rot-Umschaltung).

Die an dem Puffer des MINCAL 4E bzw. den Puffer Ø des MINCAL 4N angeschlossenen Standard-Peripher-Geräte können mit besonders einfachen symbolischen Formatbefehlen angesprochen werden:

	<u>Schreibmaschine</u>	<u>Leser/Locher</u>	<u>Ø.Gerät</u>
61	KG	RG	ZG (statt IBG)
62	KR	RR	ZR (statt IBR)
63	KP	RP	ZP (statt IBP)
65	WW	PW	XW (statt OBW)
66	WR	PR	XR (statt OBR)
67	WP	PP	XP (statt OBP)

Nach dem Befehl werden wiederum Vorzeichen, Darstellung, Stellenzahl und Endzeichen symbolisch angegeben.

## 6. Programmierhilfen für die Digitalrechner MINCAL 4E und MINCAL 4N

### 6.1) Umwandlungsprogramm

Trotz ihres übersichtlichen Aufbaus ist die binäre Instruktionsform ("Maschinensprache") nicht das geeignete Mittel, um längere Programme bequem, schnell und frei von Irrtümern zu formulieren. Daher wurde für die Rechner der MINCAL 4-Baureihe eine "symbolische Sprache" mit mnemotechnischen Bezeichnungen für die einzelnen Befehle ausgearbeitet (siehe auch Kapitel 5).

Jedem Benutzer steht ein Umwandlungsprogramm (Assembler) zur Verfügung, das ein symbolisch geschriebenes Programm in Maschinensprache übersetzt. Dieses Programm übersetzt nicht nur die Befehle; es ordnet auch Daten- und Verzweigungsadressen, Festwertdefinitionen und Variablenplätze, die als "Merkmale" programmiert werden, automatisch zu und erleichtert dadurch die Programmierarbeit weiter.

Das symbolisch geschriebene Programm wird in einen 8-Kanal-Streifen abgelocht und in den Rechner eingelesen, der vorher mit dem Umwandlungsprogramm geladen wurde. Der Rechner erzeugt bei der Umwandlung den eigentlichen, im Maschinencode gehaltenen Programmstreifen.

Das endgültige Programm liegt in Form eines 8-Kanal-Lochstreifens vor, der die Instruktionen, Festwerte und Variablenplätze in der Reihenfolge der Kernspeicheradressen enthält (je Wort 3 Zeichen und ein Vorzeichen bzw. Zwischenraum). Dieser Programmstreifen wird eingelesen und der Kernspeicher mit seinem Inhalt geladen.

Der symbolische Lochstreifen kann entweder auf einem Programmierplatz oder auch auf dem MINCAL-Rechner unter Benutzung von Schreibmaschine und Locheinheit erstellt werden. Für den letzteren Fall steht ein spezielles Eingabeprogramm zur Verfügung, das den Eingabevorgang erleichtert, Korrekturen ermöglicht und gewisse Kontrollen durchführt.

### 6.2) Weitere Programme

Jedem Benutzer steht ein Satz diagnostischer Programme zur Verfügung, mit deren Hilfe die einzelnen Funktionsgruppen des Rechners, die Befehlsabläufe usw. getestet werden können.

Unterprogramme für mathematische Funktionen, Doppelwort- und Gleitkommarechnungen werden auf Wunsch geliefert.

Für technisch-wissenschaftliche Aufgaben ist ein FORTRAN-Compiler (Basic FORTRAN) in Vorbereitung.

## 7. Die Peripherie der Digitalrechner MINCAL 4E und MINCAL 4N

### 7.1) Schreibmaschine

Zur Programmierung des Rechners ist eine Ein/Ausgabe-Schreibmaschine unerlässlich. Als Standardgerät wird der Typ IBM 735 BCD benutzt (Kugelkopfmodell mit leicht modifizierter Verdrahtung). Die Schreibgeschwindigkeit beträgt ca. 15 Z/s; in einer Zeile finden maximal 130 Zeichen Platz. 63 verschiedene Charaktere können eingetastet und gedruckt werden; die Funktionen Zwischenraum, Tabulator, Zeilenwechsel, Schreibkopfrücklauf (mit Zeilenwechsel) und Farbbandumschaltung sind vom Programm ansteuerbar.

Die Standard-Schreibmaschine wird direkt an den Rechenkörper angeschlossen, der die Anpassungsschaltung in Verbindung mit dem ersten Puffer und dessen Steuerlogik enthält.

Außer für die Programmierung ist die Schreibmaschine als wichtiges Daten-Ein- und vor allem als Ausgabegerät zu verwenden; da sie mit dem Rechner über einen Puffer verbunden ist, arbeitet sie unabhängig vom Programmverlauf, d.h. Drucken und Rechnen können gleichzeitig erfolgen.

### 7.2) Lochstreifengeräte

Die Rechner der MINCAL 4-Reihe werden über 8-Kanal-Lochstreifen programmiert.

Als Standardgeräte sind für den MINCAL 4E ein 8-Kanal-Locher und ein 8-Kanal-Leser vorgesehen. Die Lese- bzw. Lochgeschwindigkeit beträgt 30 Z/s bzw. 20 Z/s. Locher und Leser sind in ein Gehäuse (als Tischgerät) oder in einem 19"-Einschub eingebaut. Sie werden direkt mit dem Rechner verbunden und sind an den Puffer angeschlossen.

Als Standardgerät für den MINCAL 4N ist eine Lochstreifeneinheit vorgesehen, die einen 8-Kanal-Locher und einen 8-Kanal-Leser enthält, jeweils mit Auf- und Abwickel- bzw. Vorratsspule. Die Lesegeschwindigkeit beträgt 300 Z/s, die Lochgeschwindigkeit 75 Z/s. Die Lochstreifeneinheit ist als 19"-Einschub ausgeführt. Sie enthält ein Netzteil für die Stromversorgung von Locher und Leser und wird direkt mit dem Rechenkörper, d.h. dem ersten Puffer verbunden.

Der Rechner MINCAL 4E kann auf Wunsch auch mit der leistungsfähigeren Lochstreifeneinheit des MINCAL 4N ausgerüstet werden.

Die Lochstreifen werden im üblichen BCD-Code gelocht und gelesen; der Locher kann auf den (geringfügig abweichenden) IBM-8-Kanal-Code umgeschaltet werden.

Auch die Lochstreifeneinheit kann - außer zum Programmieren - für den laufenden Betrieb als Eingabe- und Ausgabemedium benutzt werden.

### 7.3) Puffer vom E-Typ

Der Puffer des MINCAL 4E und der Puffer Ø des MINCAL 4N sind vom E-Typ. Diese Puffer-Ausführung, die beim MINCAL 4N auch für die übrigen 7 Puffer-Plätze nach Bedarf verwendet werden kann, erlaubt folgende Ein/Ausgabefunktionen:

- (a) stellenserielle, bit-parallele Eingabe und Ausgabe dezimaler Informationen von Wortlänge (4 1/2 Stellen) in Verbindung mit max. 512 vom Programm direkt adressierbaren externen digitalen Datengebern bzw. Empfängern,
- (b) stellenserielle, bit-parallele Eingabe und Ausgabe dezimaler Informationen von Wortlänge in jeweils 3 Eingabe- und Registriergeräte mit vom Programm gesteuertem Format.

Der serielle Daten-Ein/Ausgang (a) dient zur Abfrage von Kontaktstellungen, Schaltern, digitalen Potentialen und Zuständen aller Art in der Peripherie sowie zur Übertragung von Informationen aus dem Rechner in externe Flipflop-Register und zur Ausgabe von Steuerbefehlen. Abfrage und Ausgabe erfolgen Stelle für Stelle, wobei jeweils 4 bit zusammengefaßt werden. Für dezimale Informationen bedeutet dies den Code 8-4-2-1; jedoch ist jede andere Darstellung einschließlich rein binärer Informationen möglich, wenn im Programm darauf Rücksicht genommen wird.

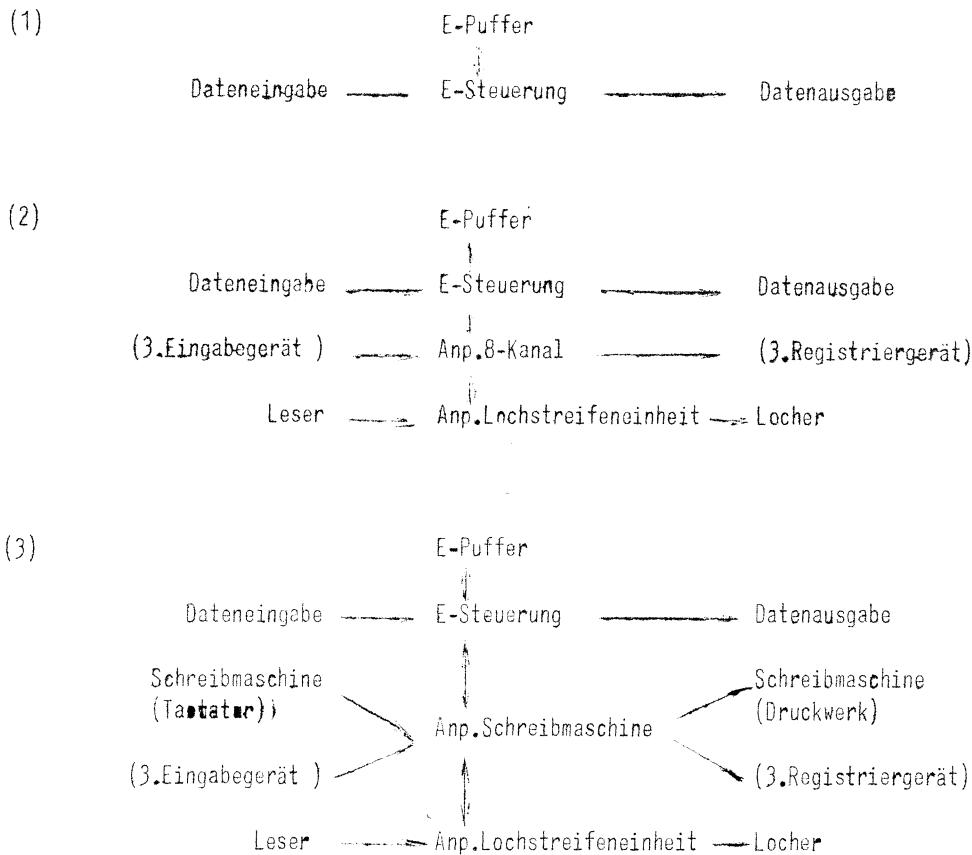
Die Stellen werden, mit der höchsten beginnend, nacheinander durch einen Stellenkanal (5 Leitungen) angewählt; die Ein/Ausgabegeschwindigkeit kann den äußeren Verhältnissen angepaßt werden. Durch einen Halt-Eingang kann der Ein/Ausgabezyklus verhindert werden, bis der adressierte Informationsträger seine Bereitschaft zur Datenabgabe bzw. -übernahme meldet. Rechts- oder Linkskomma-Ein/Ausgabe ist ebenfalls fest vorwählbar. Für Ein- und Ausgabe sind je 4 Datenanschlüsse (Code 8-4-2-1) sowie je eine Vorzeichen-Leitung vorgesehen; außerdem gibt es einen dezimalen Dateneingang mit 9 Anschlüssen (Ziffern 1...9) sowie für die Datenausgabe in externe Register einen Lösch- und einen Ladeimpuls-Ausgang.

Die zum Ein/Ausgabebefehl programmierte Adresse wird über 3 mal 8 Leitungen (100er/10er/1er) oktal entschlüsselt ausgegeben; sie identifiziert die maximal 512 (oktal mit 000...777) Geber bzw. Empfänger der Informationen. Jede Adresse ist, wenn bestimmte Voraussetzungen in der Peripherie erfüllt sind, für Eingabe und Ausgabe verwendbar, so daß insgesamt 1024 externe Adressen für den E-Puffer existieren. Ist die Anzahl von 512 Eingabe-Adressen nicht ausreichend, so kann in der Peripherie ein Register vorgesehen werden, das vom Programm über den Datenausgang mit ergänzenden Adreßstellen bedient wird und die externen Geber zusätzlich adressiert, wodurch sich deren Zahl vervielfachen läßt.

Serielle Daten-Ein/Ausgabe wird ausgelöst, wenn in der Ergänzung des Ein/Ausgabebefehls  $\emptyset$  oder 1...4 programmiert ist; im letzteren Fall wird die Adresse durch eines der Indexregister 1...4 modifiziert.

Ist in der Ergänzung dagegen 5, 6 oder 7 angegeben, so steuert der Puffer das erste, zweite oder dritte Eingabe- bzw. Registriergerät an (b) und interpretiert die Adresse als Formatangabe (siehe auch 5.2). Die Ein/Ausgabe geschieht Zeichen für Zeichen, wobei das angeschlossene Gerät die Geschwindigkeit bestimmt. Durch feste Zuordnung kann erreicht werden, daß gleichzeitige Ausgabe in 1. und 2. Registriergerät erfolgt, wenn "3. Gerät" programmiert ist; das langsame der beiden Geräte bestimmt dann die Geschwindigkeit.

Zum E-Puffer gehört stets eine Baugruppe, die den seriellen Ein/Ausgabezyklus steuert (E-Steuerung), sowie spezielle Anpassungsschaltungen für die formatgesteuerten Eingabe- und Registriergeräte. Typische Anordnungen sind z.B.:



Die unter (3) angegebene Anordnung ist für den Puffer des MINCAL 4E bzw. den Puffer Ø des MINCAL 4N **zutreffend**. E-Steuerung, Anpassung Schreibmaschine und Anpassung Lochstreifeneinheit (in 2 Ausführungen) sind im Rechenkörper enthalten. Für das 3. Eingabe- bzw. Registriergerät existiert je ein 8-Kanal-Ein- bzw. -Ausgang (BCD-Lochstreifencode) sowie je eine Auslöse- und eine Rückmeldeleitung; außerdem ist für das 3. Ausgabegerät ein Funktionskanal mit 7 Anschlüssen vorgesehen, der die 7 möglichen Endzeichen auswählt. Im allgemeinen ist für ein drittes Gerät eine spezielle Anpassungsschaltung erforderlich, die in jedem Fall außerhalb des Rechenkörpers unterzubringen ist.

#### 7.4) Weitere Puffer

Im MINCAL 4N können weitere 7, den Programm niveaus 1...7 zugeordnete Puffer vorgesehen werden. Sie befinden sich im Rechenkörper; die zugehörigen Steuer- und Anpassungsschaltungen sind jedoch außerhalb untergebracht (z.B. in einem besonderen Einschub der MINCAL 4N-Standardkonsole, der auch eventuelle Netzteile für die Versorgung der Peripheriegeräte enthält).

Nach Bedarf können entweder E-Puffer (s. 7.3) oder auch folgende spezielle Puffer-Typen in beliebiger Anordnung verwendet werden:

**R-Puffer** (mit R-Steuerung und entsprechenden Anpassungsschaltungen)  
für formatgesteuerte Ein- und Ausgabe wie beim E-Puffer, jedoch für 8 Eingabegeräte und 8 Ausgabegeräte (bzw. 8 Kombinationen von Ausgabegeräten);

D-Puffer (mit D-Steuerung)

für serielle Ein- und Ausgabe aus bzw. in 512 externe Informationsträger wie beim E-Puffer, jedoch zusätzlich für Ausgabe von 3 Schaltbefehlen programmierbarer Dauer auf 512 verschiedene (nicht modifizierbare) Adressen;

A-Puffer für parallele Datenabfrage über 18 + 1 Eingangsleitungen aus 512 verschiedenen, über Indexregister modifizierte Adressen.

7.5) Periphere Systemkomponenten

Das Peripheresystem der MINCAL 4-Rechner umfaßt eine Reihe weiterer Baugruppen zur Anpassung von Prozeßgrößen, Eingabemedien, Steuerelementen und Organisationsmitteln an die Rechner, insbesondere in Verbindung mit den Puffern, z.B.:

Analog-Digital-Umsetzer (z.B. Stufenverschlüsseler mit ca. 2 ms Meßzeit) und Meßstellenumschalter für analoge Signaleingänge,

Abfrageschaltungen für Kontakte, Schalter, Potentiale und andere digitale Informationen,

Register, Anzeigeregister, Relaischalter usw. zur Aufnahme von Daten und Befehlen aus dem Rechner,

Anpassungsschaltungen für Fernschreibsysteme und Datenübertragungen, Kartenlocher, Kartenleser, verschiedene Drucker usw.

Besondere Anpassungsgruppen werden auf Wunsch geliefert.

7.7.1967

PD/Sn

## Ausbauungen Befehlsablauf Manual 4N/4E:

1.) Die indirekte Adressierung ist zukünftig mehrmals möglich (multi level indirekt adressierung). Ist die indirekte Adresse negativ, so wird erneut indirekt adressiert. Die indirekten Adressen kann dabei beliebig oft negativ sein.

2.) Befehle LDR und TRR sind in ihrer Bedeutung verfeinert worden.

LDR = Laden in Register (Inhalt von W  $\rightarrow$  Register)

TRR = Transfer aus Register (Inhalt von Register  $\rightarrow$  W)

3.) Bei Befehlen Multiplikation, Subtraktion und Addition Konstante wird der Überlauf zu dem Inhalt des X-Registers addiert. Innerhalb des Überlaufs muss X weiter gelöscht werden.

4.) Die Division wird nicht abgebrochen, wenn der Quotient größer oder gleich 1 wird. In diesem Fall wird lediglich das Vorzeichen des X-Registers auf Minus gesetzt. Es werden auf diese Weise nach Ergebnisse bis 1,999 ermöglicht.

5.) Die Unterprogrammmpfang ist nicht mehr vom Sensor abhängig. Die Programmstelle gibt das Register an, in das die Rückkehradresse geschrieben wird. Da die Rückkehradresse nicht mehr im ersten Platz des Unterprogramms abgespeichert werden muss, beginnt das Unterprogramm schon bei der angegebenen Adresse.

7.6.68 Ba

### Aenderung Befehlskode Micro 4E, Micro 4N

Die Bedeutung des Befehls 15 (TKR) ist darüber erweitert worden, daß der Inhalt des angegebenen Registers nicht nur in das W-Register humperiert werden kann, sondern in jeder beliebige Register.

Das verlangt aber, daß diese Register in den einer Adresse bestimmt werden müssen. Wie über und über können müssen wirbleiben, ob die Adressen durch das System oder gegeben werden.

Codel:

15 E 0 OR

E = Register, aus dem die Daten geholt werden

R = Register, in das die Daten humperiert werden.

# Korrigierte Zeiten MC 4N, 4E

- 19 -

## 5. Die Programmbefehle der Digitalrechner MINCAL 4E und MINCAL 4N

### 5.1) Befehlsliste

Die 60 Befehle der Rechner MINCAL 4E und MINCAL 4N sind in 8 Gruppen unterteilt, entsprechend der Ziffer in der ersten Oktalstelle der Instruktion:

Gruppe 0	Nulloperation, Vorzeichen, Schiften
1	Transportoperationen
2	Einfache numerische und logische Operationen
3*	Umfangreiche numerische Operationen
4	Verzweigung
5	Unterprogramm, Prioritätswechsel, Halt, Ende
6	Ein/Ausgabe über Puffer
7	Ein/Ausgabe über Parallelkanal

Die mit \* gekennzeichneten Befehle der Gruppe 3 sind bei der Grundausführung des MINCAL 4E nicht möglich (keine eingebaute Multiplikation, Division und Umwandlung).

Im folgenden sind die einzelnen Programmbefehle mit oktaler und symbolischer Bezeichnung, deren Erklärung in deutsch und englisch (letztere liegt der symbolischen Bezeichnung meist zugrunde) und Angaben über Verlauf und Dauer der Operation beschrieben:

00 NOP      Keine Operation  
                 No Operation  
                 Diese Instruktion wird übersprungen.  
                 Dauer: 40 µs

01 MSP      Vorzeichen auf Plus  
                 Make Sign Plus  
                 Das Vorzeichen des W-Registers wird auf Plus gesetzt.  
                 Dauer: 60 µs

02 MSM      Vorzeichen auf Minus  
                 Make Sign Minus  
                 Das Vorzeichen des W-Registers wird auf Minus gesetzt.  
                 Dauer: 60 µs

03 MCS      Vorzeichen wechseln  
                 Make Change of Sign  
                 Das Vorzeichen des W-Registers wird in sein Gegenteil verkehrt.  
                 Dauer: 60 µs

- 04 SR Schiften Rechts  
Shift Right  
Der Inhalt des W-Registers wird um die angegebene Zahl von Binärstellen nach rechts verschoben. Der rechts überlaufende Teil steht linksbündig im X-Register.  
Dauer: 100  $\mu$ s
- 05 SL Schiften Links  
Shift Left  
Der Inhalt des W-Registers wird um die angegebene Zahl von Binärstellen nach links verschoben. Der links überlaufende Teil steht rechtsbündig im X-Register.  
Dauer: 100  $\mu$ s
- 06 SRR Schiften rechts mit Runden  
Shift Right and Round  
Der Inhalt des W-Registers wird um die angegebene Zahl von Binärstellen nach rechts verschoben. Zum Absolutbetrag des Inhalts wird vorher eine 1 in der höchsten, rechts herauslaufenden Stelle addiert.  
Dauer: 100...120  $\mu$ s
- 07 SLN Schiften links zwecks Normierung  
Shift Left and Normalize  
Der Inhalt des W-Registers wird so weit nach links verschoben, bis in der höchsten Stelle eine 1 steht. Die Größe der Stellenverschiebung steht rechtsbündig im X-Register.  
Dauer: 100  $\mu$ s
- 10 LDC Laden Festwert  
Load Constant  
Das angegebene Register wird gelöscht und mit dem programmierten positiven oder negativen Festwert geladen.  
Dauer: 60  $\mu$ s
- 11 LDR Laden aus Register  
Load from Register  
Das W-Register wird gelöscht und mit dem Inhalt des angegebenen W-Registers geladen.  
Dauer: 80  $\mu$ s

- 12 LD Laden  
Load  
Das W-Register wird gelöscht und mit dem Inhalt des adressierten Speicherplatzes geladen.  
Dauer: 80...100  $\mu$ s
- 13 LDY Laden indirekt  
Load Indirectly  
Das W-Register wird gelöscht und mit dem Inhalt eines Platzes geladen, dessen Adresse im angegebenen Speicherplatz steht.  
Dauer: 120...140  $\mu$ s
- 14 ADC Addieren Festwert  
Add Constant  
Der angegebene positive oder negative Festwert wird zum Inhalt des angegebenen Registers addiert; die Summe steht im angegebenen Register, ein eventueller Überlauf als rechtsbündige 1 im X-Register.  
Dauer: 60...80  $\mu$ s
- 15 TRR Transfer im Register  
Transfer to Register  
Der Inhalt des W-Registers wird in das angegebene Register übertragen, das vorher gelöscht wird.  
Dauer: 80  $\mu$ s
- 16 TR Transfer  
Transfer  
Der Inhalt des W-Registers wird in den adressierten Speicherplatz übertragen, der vorher gelöscht wird.  
Dauer: 80...100  $\mu$ s
- 17 TRY Transfer indirekt  
Transfer Indirectly  
Der Inhalt des W-Registers wird in einen Platz übertragen, dessen Adresse im angegebenen Speicherplatz steht und der vorher gelöscht wird.  
Dauer: 120...140  $\mu$ s
- 20 AD Addieren  
Add  
Der Inhalt des adressierten Speicherplatzes wird zum Inhalt des W-Registers addiert; die Summe steht im W-Register, ein eventueller Überlauf als rechtsbündige 1 im X-Register wird als rechtsbündige 1 zum Inhalt des X-Registers addiert.  
Dauer: 80...120  $\mu$ s

- 21 ADY Addieren indirekt  
Add Indirectly  
Der Inhalt eines Platzes, dessen Adresse im angegebenen Speicherplatz steht, wird zum Inhalt des W-Registers addiert; die Summe steht im W-Register, ein eventueller Überlauf als rechtsbündige 1 im X-Register. Inhalt des X-Registers addiert.  
um Dauer: 100...140  $\mu$ s
- 22 SB Subtrahieren  
Subtract  
Der Inhalt des adressierten Speicherplatzes wird vom Inhalt des W-Registers subtrahiert; die Differenz steht im W-Register, ein eventueller Überlauf als rechtsbündige 1 im X-Register. Inhalt des X-Registers subtrahiert.  
Dauer: 80...120  $\mu$ s
- 23 SBY Subtrahieren indirekt  
Subtract Indirectly  
Der Inhalt eines Platzes, dessen Adresse im angegebenen Speicherplatz steht, wird vom Inhalt des W-Registers subtrahiert; die Differenz steht im W-Register, ein eventueller Überlauf als rechtsbündige 1 im X-Register. Inhalt des X-Registers subtrahiert.  
Dauer: 100...140  $\mu$ s
- 24 FO Boole'sches Oder  
Boolean Or Function  
Alle Stellen des W-Registers werden mit den Stellen des adressierten Speicherplatzes in Oder-Verknüpfung gebracht; das Ergebnis steht im W-Register.  
Dauer: 80...100  $\mu$ s
- 25 FOY Boole'sches Oder indirekt  
Boolean Or Function Indirectly  
Alle Stellen des W-Registers werden mit den Stellen eines Platzes, dessen Adresse im angegebenen Speicherplatz steht, in Oder-Verknüpfung gebracht; das Ergebnis steht im W-Register.  
Dauer: 100...120  $\mu$ s
- 26 FA Boole'sches Und  
Boolean And Function  
Alle Stellen des W-Registers werden mit den Stellen des adressierten Speicherplatzes in Und-Verknüpfung gebracht; das Ergebnis steht im W-Register.  
Dauer: 80...100  $\mu$ s

- 27 FAY Boole'sches Und indirekt  
Boolean And Function Indirectly  
Alle Stellen des W-Registers werden mit den Stellen eines Platzes, dessen Adresse im angegebenen Speicherplatz steht, in Und-Verknüpfung gebracht; das Ergebnis steht im W-Register.  
Dauer: 100...120  $\mu$ s
- \*30 MP Multiplizieren  
Multiply  
Der Inhalt des W-Registers wird mit dem Inhalt des adressierten Speicherplatzes multipliziert; die bedeutenden Stellen des Produkts stehen im W-Register, die unbedeutenden im X-Register.  
Dauer: 1060...1080  $\mu$ s  
840...860
- \*31 MPY Multiplizieren indirekt  
Multiply Indirectly  
Der Inhalt des W-Registers wird mit dem Inhalt eines Platzes, dessen Adresse im angegebenen Speicherplatz steht, multipliziert; die bedeutenden Stellen des Produkts stehen im W-Register, die unbedeutenden im X-Register.  
Dauer: 1080...1100  $\mu$ s  
860...880
- \*32 DV Dividieren  
Divide  
Der Inhalt des W-Registers wird durch den Inhalt des adressierten Speicherplatzes geteilt; der Quotient steht im W-Register, der Rest im X-Register. Bei Quotient  $\geq 1$  bleibt der ursprüngliche Inhalt des W-Registers erhalten, und das Vorzeichen des X-Registers wird auf Minus gesetzt. (Dauer in diesem Fall 140...160  $\mu$ s)  
Dauer: 1580...1600  $\mu$ s  
1220...1240
- \*33 DVY Dividieren indirekt  
Divide Indirectly  
Der Inhalt des W-Registers wird durch den Inhalt eines Platzes, dessen Adresse im angegebenen Speicherplatz steht, dividiert; der Quotient steht im W-Register, der Rest im X-Register. Bei Quotient  $> 1$  bleibt der ursprüngliche Inhalt des W-Registers erhalten, und das Vorzeichen des X-Registers wird Minus gesetzt.  
Dauer: 1600...1610  $\mu$ s (Dauer in diesem Fall 160...180  $\mu$ s)  
1240...1260
- \*34 VBL Binär-Umformung mit Linkskomma  
Convert to Binary Left-point  
Der dezimale Inhalt des W-Registers wird in eine Binärzahl umgeformt, und zwar als Bruch.  
Dauer: 600...1160  $\mu$ s  
900  $\mu$ s

- 24 -

- \*35 VBR      Binär-Umformung mit Rechtskomma  
                 Convert to Binary Right-point  
                 Der dezimale Inhalt des W-Registers wird in eine Binärzahl umgeformt, und zwar als ganze Zahl.  
                 Dauer: ~~600...1160~~  $\mu$ s  
                     ~~900~~, $\mu$ s
- \*36 VDL      Dezimal-Umformung mit Linkskomma  
                 Convert to Decimal Left-point  
                 Der binäre Inhalt des W-Registers wird in eine Dezimalzahl umgeformt, und zwar als Bruch.  
                 Dauer: ~~1160...1640~~  $\mu$ s  
                     ~~900...1260~~, $\mu$ s
- \*37 VDR      Dezimal-Umformung mit Rechtskomma  
                 Convert to Decimal Right-point  
                 Der binäre Inhalt des W-Registers wird in eine Dezimalzahl umgeformt, und zwar als ganze Zahl.  
                 Dauer: ~~1160...1640~~  $\mu$ s  
                     ~~900...1260~~, $\mu$ s
- 40 BR      Verzweigen  
                 Branch  
                 Das Programm verzweigt auf den angegebenen Platz. Ist eine der Ziffern 1...7 als Ergänzung programmiert, so wird nur dann verzweigt, wenn der betreffende Bedingungseingang erregt ist.  
                 Dauer: 40...60  $\mu$ s
- 41 BRY      Verzweigen indirekt  
                 Branch Indirectly  
                 Das Programm verzweigt auf einen Platz, dessen Adresse im angegebenen Speicherplatz steht. Ist eine der Ziffern 1...7 als Ergänzung programmiert, so wird nur dann verzweigt, wenn der betreffende Bedingungseingang erregt ist.  
                 Dauer: 40...80  $\mu$ s
- 42 BZ      Verzweigen wenn Null  
                 Branch if Zero  
                 Das Programm verzweigt auf den angegebenen Platz, wenn der Inhalt des angegebenen Registers Null ist.  
                 Dauer: 60...80  $\mu$ s

- 43 BZY Verzweigen wenn Null indirekt  
Branch if Zero Indirectly  
Das Programm verzweigt auf einen Platz, dessen Adresse im angegebenen Speicherplatz steht, wenn der Inhalt des angegebenen Registers Null ist.  
Dauer: 80...100 µs  
80
- 44 BP Verzweigen wenn Plus  
Branch if Plus  
Das Programm verzweigt auf den angegebenen Platz, wenn der Inhalt des angegebenen Registers positiv ist.  
Dauer: 60...80 µs
- 45 BPY Verzweigen wenn Plus indirekt  
Branch if Plus Indirectly  
Das Programm verzweigt auf einen Platz, dessen Adresse im angegebenen Speicherplatz steht, wenn der Inhalt des angegebenen Registers positiv ist.  
Dauer: 80...100 µs  
80
- 46 BM Verzweigen wenn Minus  
Branch if Minus  
Das Programm verzweigt auf den angegebenen Platz, wenn der Inhalt des angegebenen Registers negativ ist.  
Dauer: 60...80 µs
- 47 BMY Verzweigen wenn Minus indirekt  
Branch if Minus Indirectly  
Das Programm verzweigt auf einen Platz, dessen Adresse im angegebenen Speicherplatz steht, wenn der Inhalt des angegebenen Registers negativ ist.  
Dauer: 80...100 µs  
80
- 50 CS Aufrufen Unterprogramm  
Call Subroutine  
Die um 1 erhöhte Instruktionsadresse wird im angegebenen Platz gespeichert, und das Programm verzweigt zu dem ~~ihm folgenden~~ Platz. Ist eine der Ziffern 1...7 als Ergänzung programmiert, so läuft diese Operation nur dann ab, wenn der entsprechende Bedingungsein-gang erregt ist.  
Dauer: 80...100 µs

- 51 CSY Aufrufen Unterprogramm indirekt  
Call Subroutine Indirectly *ausgeführte Raster*  
Die um 1 erhöhte Instruktionsadresse wird im einem Platz gespeichert, dessen Adresse im angegebenen Speicherplatz steht, und das Programm verzweigt zu dem ihm folgenden Platz. Ist eine der Ziffern 1...7 als Ergänzung programmiert, so läuft diese Operation nur dann ab, wenn der entsprechende Bedingungseingang erregt ist.  
Dauer: ~~ca.~~ 120  $\mu$ s *eltern ohne ausgesuchten Platz*
- 53 NCP Normaler Prioritätswechsel  
Normal Change of Priority  
Das angegebene Programm niveau wird gestartet.  
Dauer: 60  $\mu$ s
- 54 HLT Halt  
Halt  
Das Programm hält an; ein neuer Start lässt es von der gleichen Stelle an weiterlaufen.  
Dauer: 60  $\mu$ s
- 55 HCP Halt mit Prioritätswechsel  
Halt and Change Priority  
Das angegebene Programm niveau wird gestartet; das Programm hält an; ein neuer Start lässt es von der gleichen Stelle an weiterlaufen.  
Dauer: ~~ca.~~ 60  $\mu$ s
- 56 END Ende  
End  
Das Programm hält an; bei einem neuen Start beginnt es von vorn.  
Dauer: 60  $\mu$ s
- 57 ECP Ende mit Prioritätswechsel  
End and Change Priority  
Das angegebene Programm niveau wird gestartet, das Programm hält an; bei einem neuen Start beginnt es von vorn.  
Dauer: ~~ca.~~ 60  $\mu$ s
- 60 GB Übernehmen aus Puffer  
Get from Buffer  
Der Inhalt des Puffers wird ins W-Register übernommen, das vorher gelöscht wurde.  
Dauer: 80  $\mu$ s

- 61 IBG Eingabe mit Übernehmen aus Puffer  
Input by Buffer and Get  
Die Adresse wird in den Puffer übertragen; der Rechner hält an, bis die externe Information im Puffer ist; dann wird sie ins W-Register übernommen, das vorher gelöscht wurde.  
Dauer: ~~100 µs + Wartezeit~~  
*100...120µsec + Wartezeit*
- 62 IBR Eingabe über Puffer ohne Anhalten  
Input by Buffer and Run  
Die Adresse wird in den Puffer übertragen, das Programm läuft weiter. Der Puffer beginnt die externe Information zu übernehmen.  
Dauer: ~~60 µs - 80 µsec~~
- 63 IBP Eingabe über Puffer ohne Anhalten mit Prioritätsberechtigung  
Input by Buffer and Enable Priority  
Die Adresse wird in den Puffer übertragen, das Programm läuft weiter. Der Puffer beginnt die externe Information zu übernehmen. Wenn die externe Information im Puffer ist, wird ein Prioritätsstart ausgelöst.  
Dauer: ~~60 µs - 80 µsec~~
- 65 OBW Ausgabe über Puffer mit Anhalten  
Output by Buffer and Wait  
Die Adresse und der Inhalt des W-Registers werden in den Puffer übertragen; der Rechner hält an, bis die Information ausgegeben ist.  
Dauer: ~~80 µs + Wartezeit~~  
*80...100µsec + Wartezeit*
- 66 OBR Ausgabe über Puffer ohne Anhalten  
Output by Buffer and Run  
Die Adresse und der Inhalt des W-Registers werden in den Puffer übertragen; das Programm läuft weiter. Der Puffer beginnt den Dateninhalt auszugeben.  
Dauer: ~~80 µs - 100 µs !~~
- 67 OBP Ausgabe über Puffer ohne Anhalten mit Prioritätsberechtigung  
Output by Buffer and Enable Priority  
Die Adresse und der Inhalt des W-Registers werden in den Puffer übertragen; das Programm läuft weiter. Der Puffer beginnt den Dateninhalt auszugeben. Wenn die Information ausgegeben ist, wird ein Prioritätsstart ausgelöst.  
Dauer: ~~80 µs - 100 µs~~

- 70 GX Übernehmen extern  
Get External  
Der adressierte, externe Geber wird parallel abgefragt und sein Inhalt ins W-Register übertragen, das vorher gelöscht wurde.  
Dauer: 80  $\mu$ s
- 71 IXG Eingabe und Übernehmen extern  
Input External and Get  
Der externe Geber wird angewählt; der Rechner hält an, bis der Geber die Bereitstellung der Information meldet; sie wird parallel abgefragt und ins W-Register übertragen, das vorher gelöscht wurde.  
Dauer: 100  $\mu$ s + Wartezeit  
100... 120  $\mu$ s + Wartezeit
- 72 IXR Eingabe extern ohne Anhalten  
Input External and Run  
Die Adresse wird parallel in einen externen Speicher übertragen, der den externen Geber anwählt; das Programm läuft weiter.  
Dauer: 60  $\mu$ s - 80  $\mu$ s
- 73 IXP Eingabe extern ohne Anhalten mit Prioritätsberechtigung  
Input External and Enable Priority  
Die Adresse wird parallel in einen externen Speicher übertragen, der den externen Geber anwählt; das Programm läuft weiter. Wenn die externe Information bereitgestellt ist, wird ein Prioritätsstart ausgelöst.  
Dauer: 60  $\mu$ s - 80  $\mu$ s
- 75 OXW Ausgabe extern mit Anhalten  
Output External and Wait  
Der Inhalt des W-Registers wird parallel in den adressierten externen Empfänger übertragen; der Rechner hält an, bis die Übernahme der Information gemeldet ist.  
Dauer: 80  $\mu$ s + Wartezeit  
80... 100  $\mu$ s + Wartezeit
- 76 OXR Ausgabe extern ohne Anhalten  
Output External and Run  
Der Inhalt des W-Registers wird parallel in den adressierten, externen Geber übertragen, das Programm läuft weiter.  
Dauer: 80  $\mu$ s - 100  $\mu$ s

77

0XB

Ausgabe extern ohne Anhalten mit Prioritätsberechtigung  
Output External and Enable Priority

Der Inhalt des W-Registers wird, zugleich mit der Adresse, parallel in einen externen Geber übertragen, der in diesem Falle ein Speicher ist und einen Arbeitszyklus startet. Am Ende des Zyklus wird ein Prioritätsstart ausgelöst.

Dauer: 80  $\mu$ s - 100  $\mu$ s