

3.9.1.3. Anfang einer Sektion

LEVEL LEVEL at which to stop jump.



F2 = festes Zeichen, das den Pseudobefehl darstellt.

K = 2 hexadezimale Zeichen, die das Etikett des Niveaus darstellen
(die Kombinationen F1 bis FF sind verboten).

Der Pseudobefehl LEVEL erfüllt in einem in die Maschinensprache umgewandelten Programm keine Aufgabe und erzeugt darum auch keinen Befehl.

Während des Ladens eines Programmes in den Kernspeicher wird das Bezugszeichen jedes Niveaus in allen Sprungbefehlen durch die reelle Adresse des Befehls ersetzt, der dem betreffenden Pseudobefehl unmittelbar folgt.

Die Pseudobefehle werden anschließend unterdrückt; das Programm entsprechend zusammengeschoben, wobei bei der Bestimmung der reellen Adresse dieser Vorgang berücksichtigt wird.

Beispiel:

Anfangsprogramm		Programm in Maschinensprache	
Adr. A	Befehle	Adr. B	Befehle
0800	JRT 02 F1 1A	0800	02 09 54
0954	LEVEL F2 1A	0954	80 12 60 60 90 40
0956	MVC 80 12 60 60 90 40	0956	Fortsetzung des Programmes
	Fortsetzung des Programmes		

Adressen A: Adressen bei Laden vor dem Zusammenschieben

Adressen B: endgültige reelle Adressen.

3.9.1.4. Nullbefehl

NØP

No OPeration



Dieser Befehl erzeugt zwei gepackte Nullen an der Stelle, an der er im Programm steht.

Wenn das Programm auf diese Stelle stößt, wird keine Funktion ausgelöst. Das PAR erhöht sich aber um 1. Dadurch wird der Programmablauf nicht unterbrochen und der nächste Befehl kommt zur Durchführung.

Anmerkung:

Wenn dieser Pseudobefehl innerhalb einer Befehlsfolge steht, darf diese Stelle kein anderes Zeichen erhalten, da dieses von der Zentraleinheit als Operationstyp gewertet wird und ein nicht vorherzusehendes Resultat ergibt.

Beispiel:

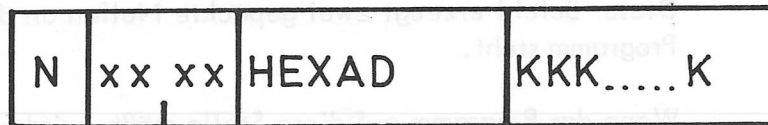
			PAR
800	NØP	} vorher nachher	0800
801	MVC		0801
	etc.		

3.9.2. Erstellung von Konstanten

Die Pseudobefehle zur Erstellung von Konstanten haben einen besonderen Aufbau, so daß sie nicht innerhalb eines Programmes auftreten dürfen. Ihr Platz ist entweder unmittelbar vor dem BEGIN oder nach dem END.

3.9.2.1. Hexadezimale Konstanten

HEXAD HEXADezimal data to load.



N = Zahl der Bytes, die von den einzelnen Zeichen belegt werden (max. 32);

xxxx = absolute Adresse, an der das 1. Zeichen von links eingespeichert werden soll;

KK...K = hexadezimale Zeichen, die eingesetzt werden sollen (2-64 Zeichen, K = 2 Zeichen).

Dieser Pseudobefehl bewirkt das Erstellen von Zeichen K an den Stellen, die der absoluten Adresse xxxx folgen.

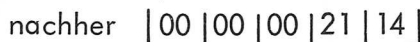
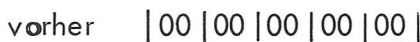
Die Zeichen K werden von links nach rechts eingespeichert, so daß je 2 ein Byte in aufsteigender Reihenfolge belegen.

Beispiel:

Erstellen einer Basisadresse im Register 8



Register 8



0140

3.9.2.2. Alphanumerische Konstanten

AN AlphaNumeric data to load.



- N = Zahl der Bytes, die von den einzelnen Zeichen belegt werden (max. 64);
- xxxx = absolute Adresse, an der das 1. Zeichen von links eingespeichert werden soll;
- AA...A = alphanumerische Zeichen, die eingesetzt werden sollen (1-64; A = 1 Zeichen).

Dieser Pseudobefehl bewirkt das Erstellen von Zeichen A an den Stellen, die der absoluten Adresse xxxx folgen.

Die Zeichen A werden von links nach rechts eingespeichert, so daß 1 Zeichen ein Byte in aufsteigender Reihenfolge belegt.

Beispiel:

Erstellen von konstanten Daten



vorher:

00	00	00	00	00	00	00	00
----	----	----	----	----	----	----	----

nachher:

00	20	31	30	38	25	20	00
----	----	----	----	----	----	----	----

0830

3.9.2.2. Alphanumerische Konstanten
Alphanumeric data to load



N = Zahl der Bytes, die von den einzelnen Zeichen belegt werden
 (max. 64)
 xxx = absolute Adresse, an der das 1. Zeichen von links einge-

speichert werden soll;
 AAA...A = alphanumerische Zeichen, die eingesetzt werden sollen
 (1-64; A = 1 Zeichen)

Dieser Pseudobefehl bewirkt das Erstellen von Zeichen A an den Stellen,
 die der absoluten Adresse xxx folgen.

Die Zeichen A werden von links nach rechts eingezeichnet, so dass 1 Zeichen
 ein Byte in aufsteigender Reihenfolge belegt.

Beispiel:

Erstellen von konstanten Daten



0830

4. Simultaneität und Multiprogramming beim GE-55

4.0. Vorbemerkungen

Der erste Abschnitt dieses Kapitels hat es sich zur Aufgabe gestellt, anhand eines ganz allgemeinen Plans eine Vorstellung der Simultaneität und des Multiprogramming, die wiederum nur eine erweiterte Form der Simultaneität ist, zu geben. Die verschiedenen Stufen des Simultanarbeitens werden dargelegt, mit denen eine Optimierung des Nutzen eines Rechners, der nur mit einem Processor ausgestattet ist, wie der GE-55 erreicht werden kann. Es wird gezeigt:

- wo jede Stufe gerechtfertigt ist und was sie für Folgen hat;
- wie man sie anwendet;
- wo die kritische Grenze ist, nach der eine Optimierung illusorisch ist.

Dieser erste Abschnitt nimmt darum auch keinen Bezug auf die Art und Weise, mit der diese Stufen der Simultaneität des GE-55 realisiert werden. Das ist Aufgabe der Abschnitte 4.2. und 4.3. Im besonderen soll man keine Schlüsse in bezug auf den GE-55 aus den Diagrammen ziehen, die ganz allgemein das in 4.1. gesagte illustrieren sollen.

4.1. Allgemeine Einführung in die Simultaneität

Im Normalfall hat ein Programm folgenden Aufbau:

Lesen einer Reihe von Daten
und Übertrag in die Lesezone

|
Verarbeitung in der ZE

|
Schreiben der Ergebnisse aus
einer Ausgabezone

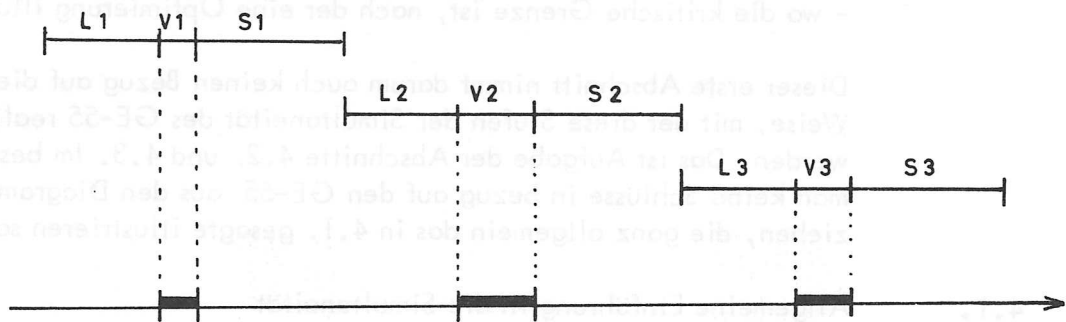
|
Lesen einer neuen Reihe
von Daten

|
etc.

Bezeichnet man mit:

- L 1 die Dauer des Lesens der Daten 1,
- V 1 die Dauer der Verarbeitung dieser Daten,
- S 1 die Dauer des Schreibens der Ergebnisse,
- L 2, V 2, S 2, die Dauer dieser Arbeiten für die Daten 2 etc.

So entsteht folgendes Diagramm der Operationen für eine Maschine, die keine Möglichkeit der Simultanarbeit hat:



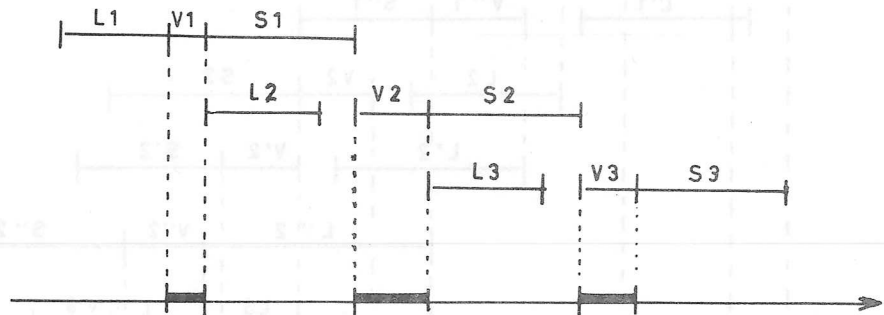
Belegung der Zentraleinheit (schraffiert dargestellt).

Aufgrund dieses Diagramms erkennt man, daß das Programm nur einen kleinen Bruchteil der Zeit beansprucht, die zur Bearbeitung durch die Zentraleinheit mit ihrer Elektronik zur Verfügung steht, während die Funktionen des Lesens und Schreibens durch relativ langsame elektromechanische Geräte erfolgen.

1. Abschnitt

Eine erste Verbesserung dieses Ablaufs erreicht man, wenn man das Lesen der zweiten Serie von Daten nach der Verarbeitung der Daten 1 vornimmt, d.h., gleichzeitig mit dem Schreiben der Ergebnisse 1. Da man nach der Verarbeitung V₁ die Daten 1 nicht mehr benötigt, kann man neue Daten in die Lesezone übertragen.

Das Diagramm sieht dann folgendermaßen aus:



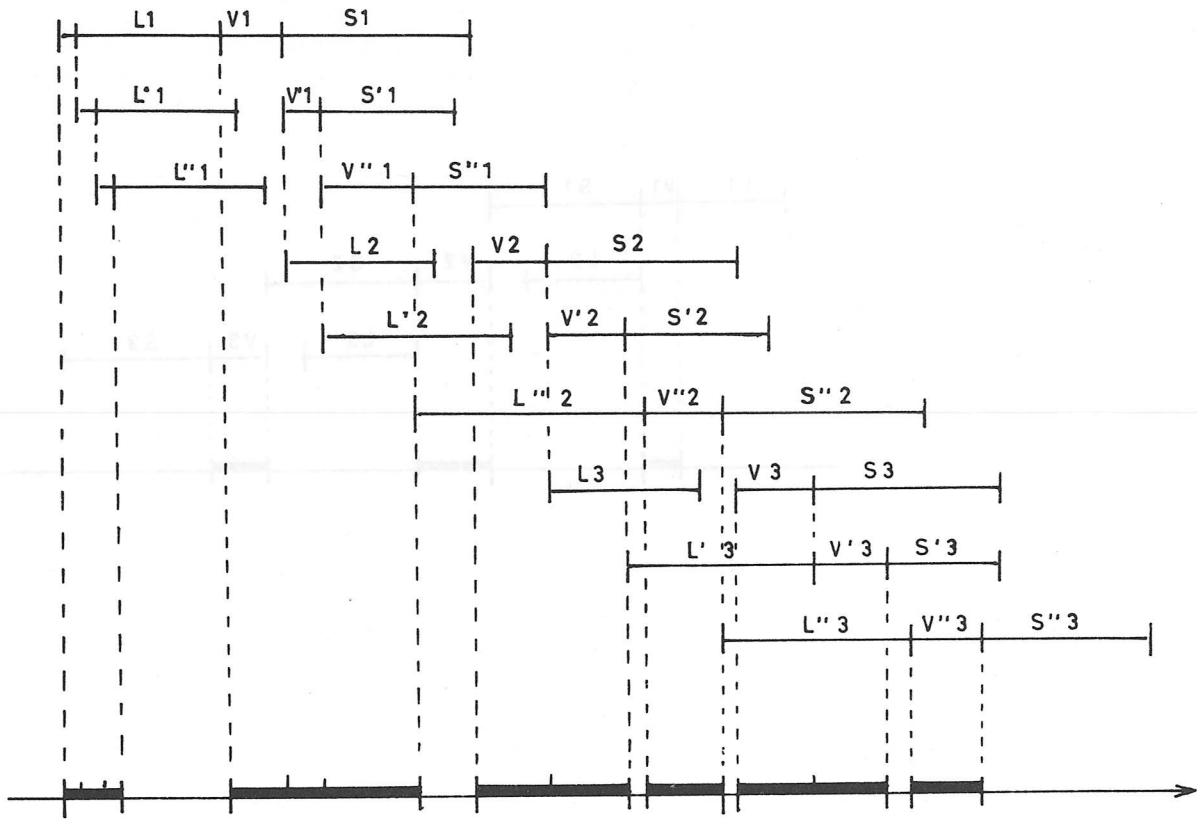
In diesem angenommenen Beispiel ist der Schreibvorgang länger als der Lesevorgang. Die Verarbeitung V 2 kann also nicht vor dem Ende des Schreibens S 1 beginnen, weil sonst die Ausgabezone zerstört würde.

Diese Simultaneität der Ein/Ausgaben entspricht der Hardware des GE-55. Die Gesamtdauer eines Programms ist also gleich der Summe der eigentlichen Verarbeitungszeit plus der Zeit für die längste Ein/Ausgabeoperation.

2. Abschnitt

Trotzdem ist man weit davon entfernt, alle Möglichkeiten der Verarbeitung innerhalb der Maschine voll auszuschöpfen. Um in dieser Richtung Fortschritte zu erzielen, öffnet das Multiprogramming einen Weg, der, gut angewendet, es erlaubt, fast das Optimum zu erzielen.

Das Multiprogramming besteht daraus, parallel mehrere Programme, die sich zur gleichen Zeit im Zentralspeicher befinden, auszuführen. Das folgende Diagramm zeigt den Ablauf von drei Programmen P, P' und P'' im Zusammenhang mit Multiprogramming. L 1 bezeichnet die Leseoperation der ersten Daten des Programms P; V3'' die Verarbeitung der dritten Daten des Programms P'' etc.



Anfang der Arbeit

Dieses Schema zeigt, daß bei einer solchen Organisation die drei Programme im ganzen genommen simultan, praktisch in der gleichen Zeit arbeiten, die sie auch allein brauchen würden.

Das Ergebnis ist wie folgt:

- 1) Die Möglichkeiten der Simultanarbeit der Ein/Ausgabeeinheiten werden auf breiter Basis angewendet.
- 2) Die elektronischen Organe werden zu einer Verarbeitung benutzt, die wechselweise den verschiedenen Programmen zur Verfügung steht.

Untersucht man das Diagramm, so ist folgendes festgehalten:

- a) in einem bestimmten Augenblick können die simultan ablaufenden Ein- und Ausgaben verschiedene Programme betreffen.
- b) Die Ein- und Ausgaben eines Programms werden simultan zur Verarbeitung der anderen Programme ausgeführt.
- c) Es erfolgt keinerlei Simultaneität in bezug auf die eigentliche Verarbeitung. Diese erfolgte durch die gleichen Schaltkreise der Zentraleinheit und zwar in der Reihenfolge, in der diese Schaltkreise zur Verfügung stehen. Das hat gewisse Wartezeiten im Ablauf der Programme zur Folge.

Zusammenfassung:

Es ist also letztlich nicht exact, zu sagen, daß die Programme simultan immer in der gleichen Zeit arbeiten, in der sie auch allein laufen würden. Dafür sind aber die Schaltkreise fast ständig in Arbeit. Je näher sie der Vollbelegung kommen, um so weniger treten Wartezeiten auf und die oben gezeigte Formel erhält ihren vollen Wert. Aber in dem Augenblick, in dem die Verarbeitungsmöglichkeiten der Maschine erschöpft sind, ist es evident, daß das Multiprogramming eine Täuschung sein muß. Für eine gegebene Maschine bestimmt das Verhältnis von internen Verarbeitungsgeschwindigkeiten zu der Geschwindigkeit der Ein/Ausgaben die maximale Zahl von Programmen, die man tatsächlich simultan und genau so schnell, als wenn sie allein wären, ablaufen lassen kann. Für den GE-55 ist diese Zahl 5. Das ist der Grund dafür, daß - wie man im Abschnitt über das Multiprogramming sehen wird - die logische Organisation der Maschine die mögliche Zahl der simultan arbeitenden Programme auf 5 begrenzt.

4.2. Die Simultaneität der Ein/Ausgaben des GE-55

4.2.1. Darstellung

Es ist bekannt, daß mehrere Randeinheiten zur gleichen Zeit im gleichen Ein/Ausgabebefehl (IØC oder IØIC - s.3.6.) unter der Voraussetzung, daß sie an verschiedenen Kanälen angeschlossen sind, angerufen werden können. Sie können aber auch durch mehrere Befehle in verschiedenen Programmen unter dem gleichen Vorbehalt angerufen werden. Diese Möglichkeit wird im Abschnitt über Multiprogramming beschrieben.

Die Aufgabe dieses Kapitels ist es, festzulegen, unter welchen Bedingungen die Randeinheiten arbeiten, wenn die Befehle wie folgt gegeben werden. Vorher müssen jedoch drei Typen von Ein/Ausgabeeinheiten festgelegt werden:

- 1) Normale Randeinheiten, die auf jedem Zeichen halten können, wie Drucker MB 50 und Stanzer PS 40.
Der Übertrag des Inhalts einer Speicherzone nach einer solchen Einheit kann zu jedem Zeitpunkt unterbrochen und später fortgesetzt werden.
- 2) Normale Randeinheiten, die nicht auf jedem Zeichen halten können, wie der Kartenleser.
Der Übertrag der Zeichen nach oder von einer solchen Einheit kann nicht unterbrochen werden.
- 3) Schnelle Randeinheiten wie die Trommel und der Drucker I 41.

Der Pufferspeicher, der im gleichen Rythmus mit der Zentraleinheit selbst arbeitet, wird nicht berücksichtigt.

4.2.2. Anwendung

Wie bekannt belegt ein Ein/Ausgabebefehl eine bestimmte Zone im Zentralspeicher, die IØC-Zone von Stelle 31 - 59. Infolgedessen steuert der verdrahtete Supervisor der Zentraleinheit aufgrund des Inhalts dieser Zone vollständig den Anschluß der Randeinheiten ohne weiteren Befehl durch das Programm.

4.2.2.1. Die IØC - Zone

Sie besteht aus zwei Teilen:

- 24 Bytes (31 - 54) sind unterteilt in 6 Gruppen zu je 4 Bytes, die "IØC-Register" genannt werden. Jede dieser Gruppen enthält alle notwendigen Angaben zur Steuerung einer Einheit.
- 5 Bytes (55 - 59) werden entsprechend den 5 Programmen belegt, die simultan arbeiten können (Zusammenhang mit Multiprogramming).

Sie enthalten zu jedem Zeitpunkt die Anzahl der Randeinheiten, die für jedes Programm effektiv zur Verfügung stehen.

4.2.2.2. Die IØC - Register

Sie enthalten für jede in Betrieb befindliche Einheit:

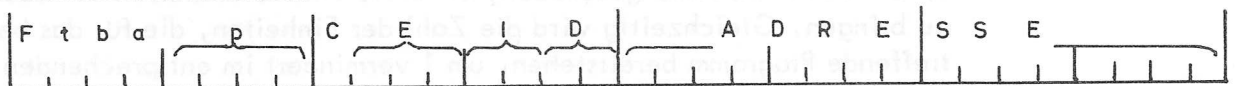
- in den ersten zwei Bytes:

- . die Programmnummer für das sie arbeitet,
- . die Nummer des Kanals, an den sie angeschlossen ist,
- . die Nummer der Einheit an diesem Kanal,
- . den Typ der Einheit (langsam, schnell, ohne oder mit Halt auf dem Zeichen),
- . die durchzuführende Operation (Lesen, Schreiben etc.);

- in den anderen zwei Bytes:

- . die Kernspeicherstelle des nächsten zu übertragenden Zeichens. Am Anfang ist diese Adresse gleich dem Inhalt des im IØC angegebenen Registers.

IØC - Register



- F = 1 Freie Gruppe
- t = 0 Einheit ist angeschaltet; = 1 Einheit ist abgeschaltet
- b = 0 Einheit kann auf der Stelle halten oder (1) nicht
- a = 0 Einheit ist anzuschliessen; = 1 ist oder war angeschlossen
- p = Nummer des Programms
- C = 0 schnelle und (1) langsame Randeinheit
- E = Nummer der Einheit des Kanals
- I = Nummer des Kanals
- D = Aufgabe (Lesen, Schreiben etc.)

Zu einem gegebenen Zeitpunkt kann man also bis zu 6 Randeinheiten in dieser Zone haben, die simultan arbeiten und durch einen oder mehrere IØC (oder IØIC) Befehle angerufen wurden.

Vorstehendes ist die Beschreibung der Steuerung der Randeinheiten nach Belegung der IØC - Zone.

4.2.3. Ablauf der Organisationen

Man hat drei Möglichkeiten zu unterscheiden:

- Alle Einheiten, die in der IØC - Zone gespeichert sind, sind langsame Einheiten mit Halt auf jedem Zeichen;
- Es sind langsame Einheiten ohne Halt auf jedem Zeichen;
- Es sind schnelle Einheiten.

4.2.3.1. Alle Einheiten sind langsam mit Halt auf jedem Zeichen.

Die Zentraleinheit tastet alle Steuergruppen ab, die effektiv belegt sind. Im Verlauf dieser Abtastung gibt sie einen Übertragungsbefehl für ein Zeichen an alle Einheiten, die, mechanisch gesehen, bereit sind. Am Ende der Abtastung wird ein Programm gestartet, das sich in der Bearbeitungsphase befindet (soweit eins vorhanden ist. Das ist nur in Verbindung mit Multiprogramming möglich) um einen Befehl durchzuführen, bevor eine neue Abtastung erfolgt und evtl. neue Befehle zur Übertragung eines Zeichens gegeben werden. Man sieht also, daß im Zusammenhang mit dem Multiprogramming eine effektive Simultaneität von mehreren Ein/Ausgaben und Verarbeitung erzielt werden kann. Ist eine Bearbeitung etwas länger, kann es sein, daß eine Einheit etwas auf den Befehl zum Übertrag eines Zeichens warten muß. Das hat jedoch keine Bedeutung, weil die Einheit zwischen zwei Zeichen halten kann. Die einzige Folge ist nur ein kurzer Aufenthalt.

Bei jedem von oder nach einer Randeinheit übertragenen Zeichen wird die in den Steuergruppen enthaltene Adresse um 1 weitergerechnet. Am Ende des Übertrags, der durch das Erkennen einer Trennmarke festgestellt wird, wird die Steuergruppe dieser Einheit gelöscht und die IØC - Zone zusammengeschoben, um evtl. andere Einheiten in Arbeit zu bringen. Gleichzeitig wird die Zahl der Einheiten, die für das betreffende Programm bereitstehen, um 1 vermindert im entsprechenden Byte des 2. Teiles der IØC - Zone.

4.2.3.2. Die IØC - Zone enthält Steuergruppen langsamer Randeinheiten ohne Halt auf jedem Zeichen

Die langsamen Randeinheiten, mit oder ohne Halt an jeder Stelle, können wegen der Abtastung der Steuergruppen der IØC - Zone simultan arbeiten. Aber im Gegensatz zu dem vorher Gesagten wird durch die Zentraleinheit kein evtl. in der Bearbeitungsphase befindliches Programm gestartet, weil Einheiten ohne Halt auf jeder Stelle sich in Arbeit befinden. Es erfolgt eine ständige Abtastung der IØC - Zone, um in der erforderlichen Zeit die Befehle zur Übertragung eines Zeichens an diese Randeinheit geben zu können. So könnte z. B. das Lesen einer Karte nicht unterbrochen werden, um einen Befehl abzuwarten, ohne daß Zeichen verlorengehen.

Die unterbrochene Abtastung der IØC - Zone bewirkt also, daß ein Simultanarbeiten in Verbindung mit einer Randeinheit ohne Halt auf dem Zeichen : nicht möglich ist.

Wie im vorigen Fall erfolgt auch hier nach Beendigung eines Übertrags ein Zusammenschieben der IØC - Zone. Es ist übrigens möglich, daß bei dieser Gelegenheit auf den vorhergehenden Fall genau zurückgekommen wird. (Durch die Möglichkeit der Simultaneität der Verarbeitung und der Ein/Ausgaben).

4.2.3.3. Die IØC - Zone enthält Steuergruppen für schnelle Randeinheiten (wie die Magnettrommel oder der Drucker I 41)

Die Natur dieser Randeinheiten erfordert es, daß Überträge nur ohne Unterbrechung zwischen den Zeichen erfolgen können. Weiterhin erlaubt es der Rhythmus des Übertrags der Zentraleinheit nicht, wie in den vorigen Fällen, mehrere Randeinheiten zu steuern. Wenn im Laufe der Abtastung der IØC - Zone die Zentraleinheit gleichzeitig auf folgende drei Bedingungen trifft:

- die IØC - Zone enthält Steuergruppen schneller Einheiten,
- diese schnellen Einheiten sind bereit, Überträge auszuführen,
- keine langsame Einheit ohne Halt auf dem Zeichen ist in Arbeit,

erfolgt der Übertrag in einem Zug. Die Abtastung der IØC - Zone wird unterbrochen. Das bedeutet, daß der Anschluß einer schnellen Randeinheit jede Simultanarbeit ausschließt, sowohl mit anderen Ein/Ausgaben als auch mit Bearbeitungen. Sobald der Übertrag beendet ist, beginnt die Abtastung erneut nach dem Zusammenschieben und man kommt auf einen der drei vorher genannten Fälle zurück.

4.2.3.4. Der Pufferspeicher

Wird im Laufe der Abtastung der IØC - Zone eine Steuergruppe ange-
troffen, die sich auf den Pufferspeicher bezieht, erfolgt der Übertrag von 6 Zeichen in einem Zug, wie bei einer schnellen Randeinheit. Man kann aber nicht von einer Unterbrechung der Simultaneität sprechen, weil der Übertrag im Rhythmus des Zentralspeichers erfolgt und auf dieser Stufe keinerlei Simultaneität möglich ist.

4.2.4. Simultaneität des I 41

4.2.4.1. Simultaneität des I 41 mit einer langsamen Randeinheit.

Die Schaltkreise des GE-55 erlauben eine Simultaneität des I 41 mit einer einzelnen langsamen Randeinheit, gleich, ob diese Randeinheit auf dem Zeichen hält oder nicht.

4.2.4.2. I 41 und eine Randeinheit mit Halt auf dem Zeichen. Die Einheiten arbeiten voll simultan.

Beispiel: IØC mit I 41 und Stanzer

Die Placierung der Einheiten innerhalb des IØC-Befehls (I 41 - Stanzer oder umgekehrt) ist ohne Einfluß auf die Simultanarbeit.

4.2.4.3. I 41 und eine langsame Randeinheit ohne Halt auf dem Zeichen

Entsprechend dem unter 2.7.1. aufgezeigten Verfahren ist eine Simultanarbeit nur dann möglich, wenn das Register AK im Augenblick der Verbindung des I 41 Null enthält.

Das bedeutet, daß der I 41 zuerst gestartet werden muß, wenn man eine Simultaneität erreichen will.

Beispiele: IØC mit Leser und I 41
IØC mit I 41 und Leser

In dem ersten Beispiel wird der Leser zuerst gestartet. Das Register enthält dann "1", und die Verbindung des I 41 ist nicht möglich. Er wird erst dann gestartet, wenn das Lesen beendet ist.

In dem zweiten Beispiel ist eine volle Simultaneität gegeben. In dem Augenblick, in dem der I 41 zu verbinden ist, enthält das Register AK Null. Der I 41 wird gestartet. Die Schaltkreise des GE-55 erlauben eine Simultanarbeit des I 41 mit einer langsamen Randeinheit und somit wird auch der Leser gestartet.

4.2.4.4. I 41 und mehrere langsame Randeinheiten

Das kann sich in den folgenden 3 Fällen ergeben:

- ein IØC spricht mehrere Randeinheiten an (maximal 6)
- in der vertikalen Simultaneität
- im Multiprogramming

Grundsätzlich tasten die Schaltkreise des GE-55, wenn der I 41 einmal gestartet worden ist, den ersten Bereich der IØC - Zone ab und ermitteln von links nach rechts die erste langsame Randeinheit, die schon verbunden ist oder noch zu verbinden ist. Das führt zu einer Simultaneität zwischen dem I 41 und dieser langsamen Einheit.

Beispiel: IØC - Stanzer - I 41 - Leser

Der Stanzer wird zuerst verbunden, dann der I 41 (AK = 0), und die erste langsame Randeinheit wird ermittelt. Die erste Einheit, die von links gefunden wird, ist der Stanzer. PS 40 und I 41 arbeiten also simultan. Wenn der I 41 seine Arbeit beendet, kommt es zu einer Simultaneität zwischen Stanzer und Leser. Wenn das Stanzen vor dem Drucken beendet ist, wird der Ausdruck allein weitergeführt, und erst danach wird der Leser verbunden.

Beispiel: (vertikale Simultaneität)
53 11 AA 09

simultanes Programm
53 12 1E 08 96 06

Durch den ersten IØC wird das Stanzen gestartet, und das anschließende Verarbeitungsprogramm läuft simultan dazu ab. In dem Fall, daß das Stanzen bei der Aufnahme des doppelten IØC I 41 - Leser noch nicht beendet ist, befinden sich zu diesem Zeitpunkt in der IØC - Zone:

PS 40 - I 41 - Leser

Der Stanzer ist verbunden.
Das Register AK ist Null.

Der I 41 wird verbunden, und das Abtasten im ersten Bereich der IØC - Zone wird aufgenommen. PS 40 und I 41 arbeiten simultan, und wir kommen zurück auf den vorangegangenen Fall.

Bei Beendigung des Stanzens vor der Aufnahme des IØC I 41 - Leser kommt es zu einer Simultanarbeit I 41 - Leser.

4.2.4.5. Im Multiprogramming ist das Problem im Prinzip genau so wie bei der vertikalen Simultaneität; die einzelnen IØC in der IØC-Zone kommen hier lediglich aus verschiedenen Programmen.

4.3. Multiprogramming beim GE-55

4.3.1. Darstellung

Es ist beim GE-55 möglich, maximal 5 Programme, die sich gleichzeitig im Zentralspeicher befinden, nach den Regeln des Multiprogramming ablaufen zu lassen.

Wie am Anfang dieses Kapitels gezeigt wurde, bestehen diese Programme aus Verarbeitungsserien, die durch Ein/Ausgabebefehle ($I\Phi C$) getrennt werden. Trifft eines von ihnen im Laufe der Arbeit einen $I\Phi C$ - Befehl, setzt dieser Befehl eine oder mehrere Randeinheiten nach den im vorigen Abschnitt beschriebenen Regeln in Arbeit. Solange die Ein/Ausgabeoperationen dieses Programms nicht beendet sind, wird mit der Weiterverarbeitung dieses Programms ausgesetzt. Man sagt, daß ein solches Programm "erstarrt" ist. Das ist der Fall, wenn der Inhalt des PAR konstant bleibt.

Ein anderes Programm, das sich in der Verarbeitungsphase befindet, wird also zur Ausführung gebracht. Es läuft simultan mit den Ein/Ausgabeoperationen des "erstarrten" Programms ab, die Befehle schieben sich in die Abtastung der $I\Phi C$ - Zone unter den vorher gesagten Bedingungen ein. Sobald das zweite Programm in seinem Ablauf auf einen Ein/Ausgabebefehl trifft, "erstarrt" es ebenfalls und es wird zu einem weiteren Programm übergewechselt. Kann allerdings der Ein/Ausgabebefehl nicht in die $I\Phi C$ - Zone eingespeichert werden, weil die Zahl der zur Verfügung stehenden Steuergruppen nicht ausreicht, wird der Programmwechsel solange aufgeschoben, bis der Befehl effektiv zur Ausführung gelangt.

Nachdem die Ein/Ausgabeoperationen eines "erstarrten" Programms durchgeführt wurden, kann dieses seinen Ablauf wieder aufnehmen. Aber genau wie zu einem bestimmten Zeitpunkt mehrere Programme "erstarrt" sein können, gibt es auch mehrere Programme, die ablaufen können. Das freiwerdende Programm wird darum in eine Warteschleife eingereiht, um zum Laufen zu kommen, wenn es an der Reihe ist. Man sagt, ein solches Programm ist "gestartet".

Das Multiprogramming des GE-55 besteht also in einer Folge von Abläufen "gestarteter" Programme simultan mit den Ein/Ausgaben "erstarrter" Programme.

Außer den "gestarteten" und "erstarrten" Programmen gibt es noch ein oder mehrere "anhaltende" Programme. Das sind Programme, bei denen keine Operationen erfolgen, sondern die darauf warten, von einem anderen Programm gestartet zu werden. Der Halt eines solchen Programms kann durch das gleiche Programm oder durch die Einrichtung einer "Sicherung" (Schutz) erfolgt sein, der dazu bestimmt ist, zu vermeiden, daß Sequenzen, die unvereinbar mit den verschiedenen Programmen sind, gleichzeitig mit ihnen ablaufen. Das ist besonders der Fall, wenn mehrere

Programme gleiche Zonen im Zentralspeicher oder gleiche Randeinheiten, oder zwei am gleichen Kanal angeschlossene Einheiten benutzen. Es ist bekannt aus Abschnitt 4.2.1., daß zwei Einheiten, die an dem gleichen Kanal angeschlossen sind, nicht gleichzeitig von der Zentraleinheit gesteuert werden können. Das bedeutet also, daß nur $I\Phi C$ - Befehle übernommen werden können, die den gleichen Kanal benutzen, wenn sie einzeln in die $I\Phi C$ - Zone eingehen.

Das Zwischenspiel zwischen den Programmen wird durch Spezialbefehle gesteuert, die in Abschnitt 4.3.3. beschrieben werden.

Anmerkung:

Die Gesamtheit der simultan durchgeführten Programme ergibt eine "Verarbeitungseinheit". Die Programme werden numeriert von 1 - 5. Am Anfang der Arbeit sind alle Programme, außer einem, "angehalten". Nur das Programm 1 ist gestartet und kommt unmittelbar zur Durchführung. Dieses Programm muß die anderen im gewünschten Augenblick in der vom Programmierer gewünschten Reihenfolge starten.

4.3.2. Anwendungsweise

Es handelt sich hierbei im wesentlichen um verschiedene Zonen im Zentralspeicher, die von der Zentraleinheit ausgewertet werden, um den reibungslosen Ablauf der gemeinsam im Speicher befindlichen Programme zu steuern.

4.3.2.1. Die Registerzonen

Von den in der Standardregisterzone (Bytes 85-595) befindlichen Registern ist eine gewisse Zahl mit genau definierten Aufgaben für das gerade ablaufende Programm belegt.

Das sind:

- die spezialisierten Register (85-95), die notwendige Angaben zur Steuerung des ablaufenden Programms enthalten: PAR, Vergleichsergebnisse etc..
- die numerischen Register 0-9, in denen die Basisadressen, speziell die für die Mehrfachüberträge, gespeichert sind.

Bei Multiprogramming spielen diese Stellen die gleiche Rolle für das gerade "in Ausführung" befindliche Programm, wie bei Monoprogramming. Wenn aber in der Folge dieses Programm auf einen $I\Phi C$ - Befehl stößt und "erstarrt" und ein anderes Programm "gestartet" wird, nimmt es den Platz des vorherigen ein. Man muß also sowohl den Inhalt dieser Register fort speichern als auch sie mit den Angaben für dieses neue Programm belegen. Daher braucht man 5 Registerzonen, von denen eine bei Stelle 85 beginnt und die anderen nach Angabe des Programmierers. Jedesmal, wenn ein Programmwechsel erfolgt, wird durch die Zentraleinheit automatisch ein Austausch der zwei entsprechenden Zonen durchgeführt.

Was die numerischen Register betrifft, so können zwei Möglichkeiten vorliegen: Entweder übersteigt die Gesamtzahl aller von den verschiedenen Programmen benutzten Register die Zahl 100 nicht, dann werden die Register einfach auf die einzelnen Programme aufgeteilt, wie auch der Normalspeicherbereich; oder sie übersteigt die Zahl 100, dann muß eine Anzahl Register verschiedenen Programmen dienen und es muß bei jedem Programmwechsel der gleiche Austausch vorgenommen werden, wie auch bei den spezialisierten Registern.

Zusammenfassend dienen also die Registerzonen nicht nur zur Aufnahme der spezialisierten Register, sondern auch für eine gewisse Zahl von numerischen nach Angabe des Programmierers. Diese letzteren bestimmen also die Größe der Registerzonen und es ist klar, daß demnach die Länge der Registerzone die gleiche ist für alle im Speicher befindlichen Programme. Die Registerzone für das in Arbeit befindliche Programm beginnt also an Stelle 85 bis zu einer Stelle, an der sich eine Trennmarke FE befindet, die vom Programmierer gesetzt wird.

Diese Trennmarke begrenzt die Länge des Austausches im Augenblick eines Programmwechsels. Es versteht sich, daß diese Zone so klein wie möglich sein soll, damit die Dauer des Austausches verringert wird.

4.3.2.2. Die Zonen zur Steuerung der Randeinheiten und der Programme

Das Multiprogramming benutzt folgende drei Zonen:

- 1) Die Zone der Ein/Ausgaben (IØC - Zone - Input - Output-Control von Stelle 31 - 59)

Diese Zone wurde bereits in Abschnitt 4.2.2. beschrieben. Sie steuert im wesentlichen die Simultaneität der Ein- und Ausgaben, und zwar sowohl bei Mono - als auch bei Multiprogramming.

Im folgenden soll daher nur noch eine Beschreibung der letzten 5 Stellen (55 - 59) dieser Zone gegeben werden, deren Aufgabe sich bei Multiprogramming zeigt.

Diese Stellen werden in bezug auf die Programme 1 - 5 belegt. Jedes von Ihnen enthält in seinem rechten Halbbyte die Zahl der im Laufe des Programms verwendeten Randeinheiten zum Abzählen durch das Programm, dessen Byte belegt ist.

In dem Augenblick, wo ein IØC - Befehl in einem Programm "P" zum Zuge kommt, wird die Zahl der durch den Befehl angeschalteten Randeinheiten an der Stelle (54 + P) gespeichert, und zwar im rechten Halbbyte. Jedesmal wenn der Rechner die auf eine dieser Einheiten bezogenen Überträge beendet, wird der Inhalt dieses Halbbytes um 1 vermindert. Wenn der Inhalt "0" erreicht, wird das entsprechende Programm "wieder gestartet". (s. Zone PRC unten).

- 2) Die Zone der "Registerzonenadressen" (RFA-Register Field Adresses
Stelle 60 - 69)

Diese Zone enthält 5 Register zu je 2 Bytes, die entsprechend von rechts nach links für die Programme 1 - 5 belegt werden. Diese Register zeigen in jedem Moment die Adressen an, wohin die besonderen Register jedes Programms übertragen werden.

Nachdem vorher bereits Gesagten, ist eine dieser Adressen 85 und entspricht dem laufenden Programm. Jeder Wechsel der Registerzonen ruft daher die entsprechenden Adressen in der RFA - Zone an.

- 3) Die Zone der "Programmsteuerung" (PRC-PRograms control, 70-84)

Diese Zone enthält zwei Teile, die durch eine Trennmarke "00" getrennt werden.

- a) Der linke Teile der PRC - Zone enthält die Warteschleife der "gestarteten" Programme (max.5). Jedesmal wenn ein Programm gestartet wird, empfängt das erste freie Byte von links:

- in seinem rechten Halbbyte die Nummer des Programms,
- in seinem linken Halbbyte den Wert 1, der anzeigt, daß es sich um das letzte "gestartete" Programm handelt. Das linke Halbbyte des vorhergehenden Bytes wird auf Null gesetzt.

Das Programm, dessen Nummer an Stelle 70 gespeichert ist, ist in Arbeit. Wenn dieses Programm auf einen IØC, STØP oder PRØ - Befehl (s.4.3.3.) trifft und ihm in der Warteschleife ein "gestartetes" Programm folgt, erfolgt Programmwechsel. Die Warteschleife wird um eine Stelle nach links versetzt und das Programm, dessen Nummer sich an Stelle 71 befand, kommt zur Ausführung (Stelle 70). Das "erstarrte" oder "angehaltene" Programm verschwindet aus diesem Teil der PRC - Zone. Die Programme werden so in der Reihenfolge, in der sie "gestartet" wurden, zur Ausführung gebracht.

Anmerkung:

Befindet sich ein arbeitendes Programm allein in der Warteschleife in dem Augenblick, in dem es auf einem IØC-Befehl "erstarrt" (Wert 1 im linken Halbbyte der Stelle 70), wird die Versetzung ausgesetzt bis dieses eine Programm erneut "gestartet" wird, das dann wiederum das Programm an Stelle 70 ist. Diese Besonderheit wird verwendet bei Monoprogramming.

- b) Der rechte Teil der PRC - Zone dient der Steuerung des Zusammenwirkens der Programme.

Er nimmt die Zeichen auf, die die Bedingungen dieses Zusammenwirkens darstellen. Diese Zeichen werden nacheinander von rechts nach links in der Reihenfolge, in der sie von den "Halt" oder "Schutz" - Befehlen erstellt werden, eingesetzt (s.4.3.3.)