

BYTESTRING-STEUERWERK

Das Bytestring-Steuerwerk stattet das Terminal 52 mit der Möglichkeit aus, komplexe Datentransport-Befehle mit Mehrfach-Operanden auszugeben. Datentransporte werden durchgeführt, indem man Folgen von sequentiellen Bytes aus der QUELLE im Hauptspeicher über den EA-Prozessor in das Bytestring-Steuerwerk bringt. Das Bytestring-Steuerwerk legt die Daten im ZIEL im Hauptspeicher ab. Mit Hilfe von geeigneten Befehlen kann man also Bytestrings aus irgendeiner Hauptspeicher-QUELLE in ein beliebiges Hauptspeicher-ZIEL umspeichern. Die Umspeicherung kann sowohl bedingt als auch unbedingt erfolgen. Das Steuerwerk verfügt über eine Drucker-Schnittstelle.

Der Datenverkehr wird über 2 Kanäle abgewickelt; der QUELLE-Kanal liest die Daten aus dem Speicher und übergibt sie dem ZIEL-Kanal, der sie wegspeichert. Es gibt drei Arten von Datentransport-Befehlen:

- DRUCKE; an diesem Vorgang ist nur der QUELLE-Kanal beteiligt. Er holt die Daten byteweise aus der spezifizierten Hauptspeicher-QUELLE und überträgt sie zum Drucker.
- TRANSPORTIERE; bei jedem Befehl bringt der QUELLE-Kanal ein Daten-Byte aus dem Speicher in einen 1-Byte-Puffer. Der ZIEL-Kanal legt dieses Byte im ZIEL-Speicher ab (siehe Abb. 6E-1). Die TRANSPORT-Befehle lassen sich spezifizieren, so daß dem Programmierer durch Spezifikation von Maskenbedingungen eine ganze Reihe von verschiedenartigen Befehlen zur Verfügung steht, z. B.:
 - Transportiere Daten von einer QUELLE an ein ZIEL
 - Transportiere gekennzeichnete Daten von einer QUELLE an ein ZIEL
 - Transportiere ungekennzeichnete Daten von einer QUELLE an ein ZIEL
 - Verkürze einen Bytestring durch Löschen eines Zeichens
 - Verwandle ein bestimmtes Zeichen in ein anderes Zeichen
 - Ersetze alle ungekennzeichneten Zeichen durch ein anderes Zeichen (z. B. SPACE)
 - Zähle, wie oft ein bestimmtes Zeichen in einer Meldung vorkommt
 - Stelle fest, wo ein bestimmtes Zeichen erstmalig in der Meldung vorkommt
 - Lösche eine beliebige Anzahl von sequentiellen Zeichen.
- WORD WRAPAROUND; bei diesen Befehlen bringt der QUELLE-Kanal 256 Bytes aus dem Speicher in einen Hilfsspeicher (256-Byte-Puffer). Von dort gelangen die Zeichen gruppenweise in einen Sekundärspeicher (32-Byte-Puffer). Je nach Befehlsspezifikation ist die Gruppe entweder ein Wort (ein von einer Leerstelle gefolgter Zeichenstring) oder ein aus 16 Zeichen bestehender String. In Übereinstimmung mit den Vorgängen auf dem Bildschirm gehen die Daten aus dem Sekundärspeicher durch den ZIEL-Kanal in den Hauptspeicher (siehe Abb. 6E-4).

Diese Befehle sind besonders nützlich bei umfangreichen EDITING Funktionen, die sie mit großer Geschwindigkeit ausführen:

- Zeile in den Text einfügen
- Zeichen in einen Textabsatz einsetzen, ohne daß Wörter auf dem Bildschirm verstümmelt werden
- Zeichen oder Wort oder Satz in einem Textabsatz löschen, ohne daß Wörter verstümmelt werden.

Alle Vorgänge werden durch den EA-Prozessor gesteuert. Daten-Übertragungen werden nach dem 'cycle steal' Verfahren abgewickelt, sie sind für die Zentraleinheit transparent. Die ZE erhält ein Unterbrechungssignal, wenn ein EA-Vorgang beendet ist. Befehle werden an den EA-Prozessor und über den EA-Bus geleitet.

Die folgende Besprechung der einzelnen Befehle bezieht sich auf die bei T52 übliche Zuordnung des Bytestring-Steuerwerks als Gerät Nr. 4 am EA-Prozessor.

6E.1 Anweisungen

6E.1.1 Anweisungen an den EA-Prozessor

Der EA-Prozessor erhält in reservierten Speicherzellen Anweisungen der Zentraleinheit, die folgendes definieren:

- Daten-QUELLE im Speicher
- Daten-ZIEL im Speicher
- Bedingungen, welche einen TRANSPORT-Befehl beenden.

Ebenfalls in reservierten Speicherzellen findet die ZE Angaben über QUELLE- und ZIEL-Daten (die Adressen des letzten aus der QUELLE geholten sowie des letzten an das ZIEL gebrachten Zeichens).

QUELLE-Kanal								ZIEL-Kanal									
Adr. (Hex)	7	6	5	4	3	2	1 0	Adr. (Hex)	7	6	5	4	3	2	1 0		
0820								SOSL	0860							DESL	} Start-Adresse
0821								SOSH	0861							DESH	
0822								SOCL	0862							DECL	} Laufende Adresse
0823								SOCH	0863							DECH	
0824								SOTL	0864							DETL	} End-Adresse
0825								SOTH	0865							DETH	
0826								SOTC	0866							DETC	ENDE-Zeichen

Abb. 6E.1 Versorgungsblock für Befestigungs-Steuerwerk

Start-Adressen

- Speicherzelle 0820 — SOSL = Source (QUELLE) Starting address (Low)
- Speicherzelle 0821 — SOSH = Source (QUELLE) Starting address (High)
- Speicherzelle 0860 — DESL = Destination (ZIEL) Starting address (Low)
- Speicherzelle 0861 — DESH = Destination (ZIEL) Starting address (High)

Das erste aus der QUELLE zu holende Zeichen liegt in Zelle [(SOSH) □ (SOSL)] + 1; das erste zum ZIEL zu übertragende Zeichen liegt in Zelle [(DESH) □ (DESL)] + 1 oder danach (je nach Markenbedingung).

QUELLE und ZIEL Startanweisungen sind voneinander unabhängig. Der Programmierer kann sie auf eine beliebige Speicherzelle setzen, auch auf überlappende Adressen.

Laufende Adressen

Speicherzelle 0822 — SOCL	=	<u>S</u> ource (<u>Q</u> UELLE) <u>C</u> urrent address (<u>L</u> ow)
Speicherzelle 0823 — SOCH	=	<u>S</u> ource (<u>Q</u> UELLE) <u>C</u> urrent address (<u>H</u> igh)
Speicherzelle 0862 — DECL	=	<u>D</u> estination (<u>Z</u> IEL) <u>C</u> urrent address (<u>L</u> ow)
Speicherzelle 0863 — DECH	=	<u>D</u> estination (<u>Z</u> IEL) <u>C</u> urrent address (<u>H</u> igh)

SOCL, SOCH, DECL und DECH sind Register, die für die Benutzung durch den EA-Prozessor reserviert sind. In ihnen stellt der EA-Prozessor der Zentraleinheit Statusinformationen über den EA-Zyklus zur Verfügung:

[(SOCL) □ (SOCH)]	=	letzte Zelle der QUELLE, aus welcher Daten ausgelesen wurden
[(DECL) □ (DECH)]	=	letzte Zelle im ZIEL-Bereich, in welche Daten übertragen wurden.

Zu Beginn der Befehle DRUCKE, TRANSPORTIERE, WORD WRAPAROUND werden die Register durch den EA-Prozessor aktiviert.

End-Adressen

Speicherzelle 0824 — SOTL	=	<u>S</u> ource <u>T</u> erminating address (<u>L</u> ow)
Speicherzelle 0864 — DETL	=	<u>D</u> estination <u>T</u> erminating address (<u>L</u> ow)
Speicherzelle 0825 — SOTH	=	<u>S</u> ource <u>T</u> erminating address (<u>H</u> igh)
Speicherzelle 0865 — DETH	=	<u>D</u> estination <u>T</u> erminating address (<u>H</u> igh)

Bit 7 = 1 (SOTH und DETH)
TRANSPORTIERE Daten bis zum Puffer-Ende

Bit 7 = 0
TRANSPORTIERE Daten bis zum Puffer-Ende oder bis ein ENDE-Zeichen erkannt wird.

ENDE-Zeichen

Speicherzelle 0826 — SOTC	=	<u>S</u> ource <u>T</u> erminating <u>C</u> haracter
		Die Anweisung ist beendet, wenn in SOTH Bit 7 = 0 und das aus dem QUELLE-Kanal empfangene Zeichen zu SOTC paßt.
Speicherzelle 0866 — DETC	=	<u>D</u> estination <u>T</u> erminating <u>C</u> haracter
		Die Anweisung ist beendet, wenn in DETH Bit 7 = 0 und das zum ZIEL-Kanal übertragene Zeichen zu DETC paßt.

Der Programmierer kann die Ende-Bedingungen für QUELLE- und ZIEL-Kanal unabhängig voneinander festsetzen. Jedenfalls wird eine Anweisung bei Vorliegen der Ende-Bedingung sofort beendet, und die Zentraleinheit erhält eine Unterbrechungs-Anforderung.

6E.1.2 Anweisungen über den EA-Bus

Über den EA-Bus gehen folgende Anweisungen:

- Adressierung des Bytestring-Steuerwerks
- Masken-Bedingungen für die TRANSPORT-Befehle
- Abfrage des Steuerwerks-Status.

Bei Beendigung eines TRANSPORT-Befehls wird ein Status-Flag gesetzt und der Zentraleinheit eine Unterbrechungs-Anforderung übermittelt.

Anweisungen an das Bytestring-Steuerwerk werden nur ausgeführt, wenn es als das aktive EA-Gerät adressiert wurde. Das Steuerwerk bleibt aktiv, bis die Zentraleinheit ein anderes Gerät adressiert.

6E.1.2.1 Druck-Befehle

Selektiere (SELECT)

Kommando: SEL

Kommando-Byte: B4 (Hex)

Adressiert das Bytestring-Steuerwerk.

Stop

Kommando: IFL

Status-Byte: Bit 7 'NOT BUSY' (Bereit)

Bit 6 'PRINTER SELECTED' (Drucker adressiert)

Bit 5 'SOURCE TERMINATION' (Quelle Ende; die Befehle TRANSPORTIERE DATEN, SCHREIBE CODE, WRAPAROUND werden bei Erkennen der 'QUELLE ENDE' Anweisungen beendet).

Bit 4 'DESTINATION TERMINATION' (Ziel Ende; die Befehle TRANSPORTIERE DATEN, SCHREIBE CODE, WRAPAROUND werden bei Erkennen der 'ZIEL ENDE' Anweisungen beendet).

Bit 3 Wird während der Ausführung eines WRAP-Befehls gesetzt, wenn der Sekundärspeicher Daten enthält.

Bit 2 Wird während der Ausführung eines WRAP-Befehls gesetzt, wenn ein Wort mit einer Länge von mehr als 32 Bytes in den ZIEL-Speicher übertragen wurde.

Bit 1 'PRINTER NOT READY' (Drucker nicht bereit)

Bit 0 'PRINTER BUSY' (Drucker arbeitet).

Bringt das Status-Byte aus dem Bytestring-Steuerwerk in den Akku.

WCODE

Kommando: OFL

Kommando-Byte: Gewünschter Code (beim Kommando 'WRITE CODE') oder gewünschte Länge der Bildschirmzeile (1 . . . 80 in Dezimalzahl, beim Kommando 'TRANSPORTIERE DATEN MIT WRAPAROUND').

Drucke (PRINT)

Kommando: COM1

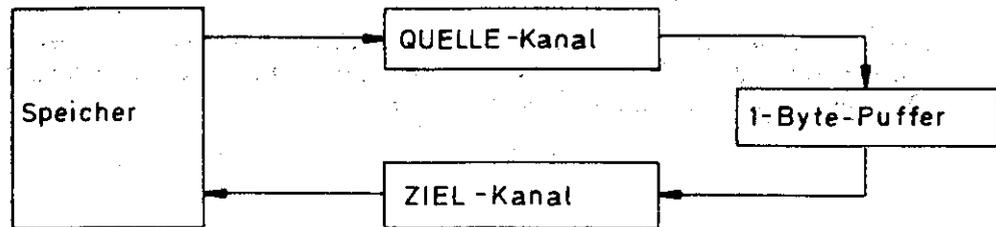
Kommando-Byte: entfällt

Überträgt Daten von der QUELLE (im Speicher) zum Drucker. Die Übertragung beginnt bei der QUELLE Start-Adresse + 1 und endet bei Vorliegen der QUELLE-ENDE Bedingung. Das ZIEL wird nicht berührt.

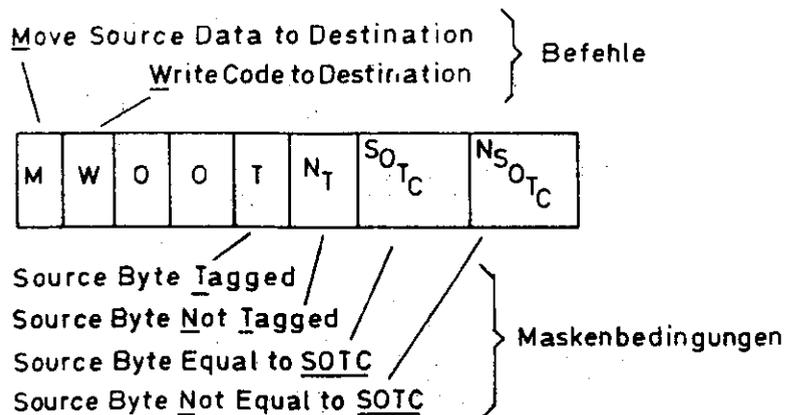
6E-1.2.2 TRANSPORT-Befehle

Kommando: COM2
 Kommando-Byte: siehe unten

Die Befehle dieser Gruppe transportieren Daten aus dem QUELLE-Bereich in den ZIEL-Bereich des Speichers, wobei ein 1-Byte-Puffer verwendet wird.



Transport-Befehle werden unter Berücksichtigung der beschriebenen Maskenbedingungen ausgeführt:



- T — Speichere ein Byte in den Ziel-Bereich, wenn das aus der QUELLE gelesene Daten-Byte 'TAGGED' (gekennzeichnet) ist.
- NT — Speichere ein Byte in den ZIEL-Bereich, wenn das aus der QUELLE gelesene Daten-Byte 'not tagged' (nicht gekennzeichnet) ist.
- SOTC — Speichere ein Byte im ZIEL-Bereich, wenn QUELLE Daten-Byte = SOTC (siehe Anmerkung).
- NSOTC — Speichere ein Byte im ZIEL-Bereich, wenn QUELLE Daten-Byte ≠ SOTC (siehe Anmerkung).

Anmerkung:

Ist Bit 0 oder 1 des TRANSPORTIERE Befehls-Byte gesetzt, dann sind die Ende-Befehle für die QUELLE- oder ZIEL-Kanäle auf [(SOTL) □ (SOTH)] bzw. [(DETL) □ (DETH)] beschränkt. In SOTH muß Bit 7 auf 0 gesetzt sein.

Maskenbedingungen bestimmen also, in welcher Weise Daten aus der QUELLE das ZIEL beeinflussen sollen. Nur QUELLE-Daten, die alle Bedingungen erfüllen, werden in den ZIEL-Bereich übertragen.

Transportiere Bytestring

Kommando-Byte: 8X (Hex)

- X = 0 Transportiere unbedingt
- X = 8 Transportiere nur 'tagged' Daten
- X = 4 Transportiere nur 'not tagged' Daten
- X = 2 Transportiere nur Daten = SOTC
- X = 1 Transportiere nur Daten + SOTC

Transportiere einen Bytestring beliebiger Länge vom QUELLE-Bereich zum ZIEL-Bereich des Speichers. Die Befehle werden gemäß den im Befehls-Byte spezifizierten Maskenbedingungen ausgeführt. Abb. 6E-1 zeigt den Ablauf eines Transport-Befehls.

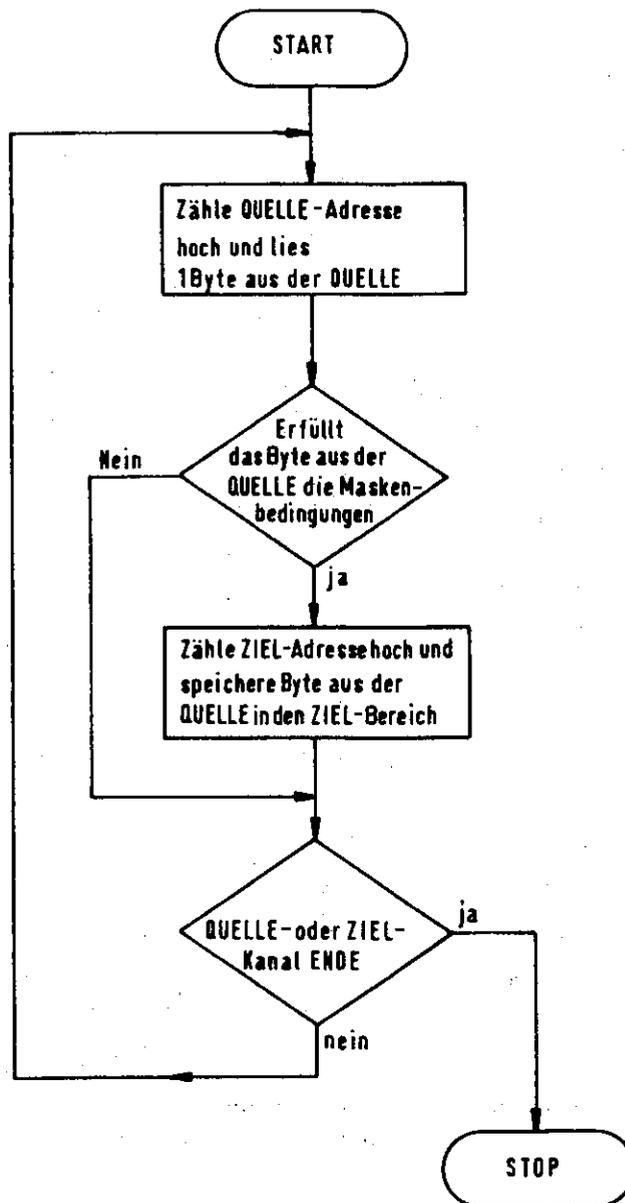


Abb. 6E-1: Transportiere Bytestring — Vereinfachtes Ablaufdiagramm

Schreibe Code

Kommando-Byte: 4X (Hex)

- X = 0 Schreibe WCODE unbedingt in den ZIEL-Bereich
- X = 8 Schreibe WCODE in den ZIEL-Bereich, wenn das QUELLE Daten-Byte 'tagged' ist
- X = 4 Schreibe WCODE in den ZIEL-Bereich, wenn das QUELLE Daten-Byte 'not tagged' ist
- X = 2 Schreibe WCODE in den ZIEL-Bereich, wenn QUELLE Daten-Byte = SOTC
- X = 1 Schreibe WCODE in den Ziel-Bereich, wenn QUELLE Daten-Byte ≠ SOTC

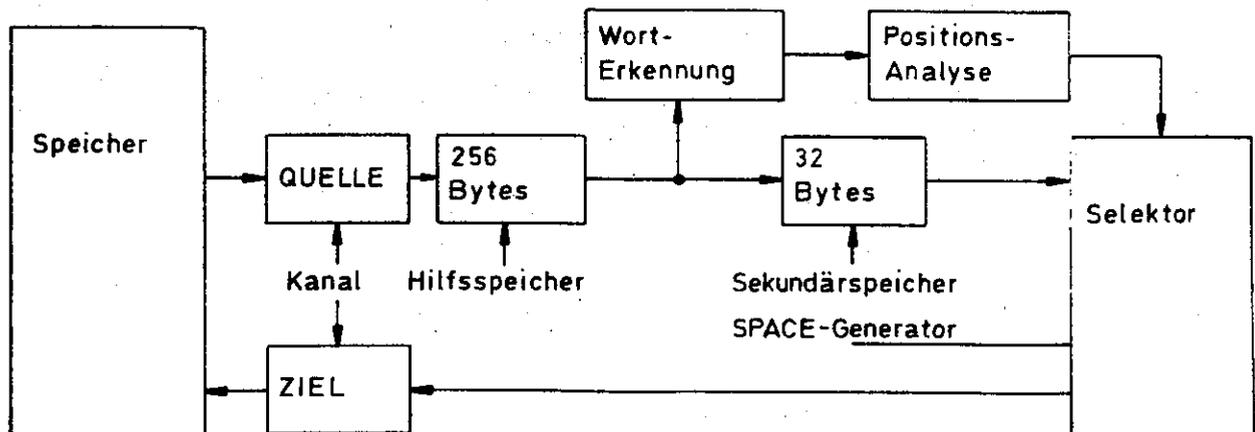
Bringt den über OFL gelieferten WCODE in den ZIEL-Bereich des Speichers. Der 'Schreibe Code' wird gemäß den Maskenbedingungen im Befehls-Byte behandelt, die QUELLE-Daten dienen dabei nur als Masken-Information. Bei der Befehlsausführung wird die ZIEL-Adresse für jedes aus der QUELLE gelesene Byte erhöht, d. h. die Fortzählung der beiden Adressen ist gleichzeitig. Daten im ZIEL-Bereich, welche wegen der Maskenbedingungen nicht überschrieben werden, bleiben unverändert. Abb. 6E-2 zeigt den Ablauf eines Schreib-Befehls.

6E.1.2.3 WORD WRAPAROUND Befehl

Kommando: COM3

Kommando-Byte: siehe unten

Die Befehle dieser Gruppe transportieren Daten aus dem QUELLE-Bereich in den ZIEL-Bereich des Speichers, wobei 1 Hilfsspeicher und 1 Sekundärspeicher verwendet werden:



Transportiere Daten

Kommando-Byte: 80 (Hex)

Transportiert unbedingt Daten aus dem QUELLE-Bereich in den Ziel-Bereich des Speichers. Das Bytestring-Steuerwerk mit seinem 16-Byte-Puffer liest einen String aus der QUELLE in den 256-Byte-Hilfsspeicher ein (damit steht ein Puffer mit einer Länge von 272 Bytes zur Verfügung). Von da gehen die Daten in Gruppen von 16 Bytes in den ZIEL-Bereich des Speichers. Die Puffer werden im gleichen Maße aufgefüllt wie die Speicherung im ZIEL abläuft.

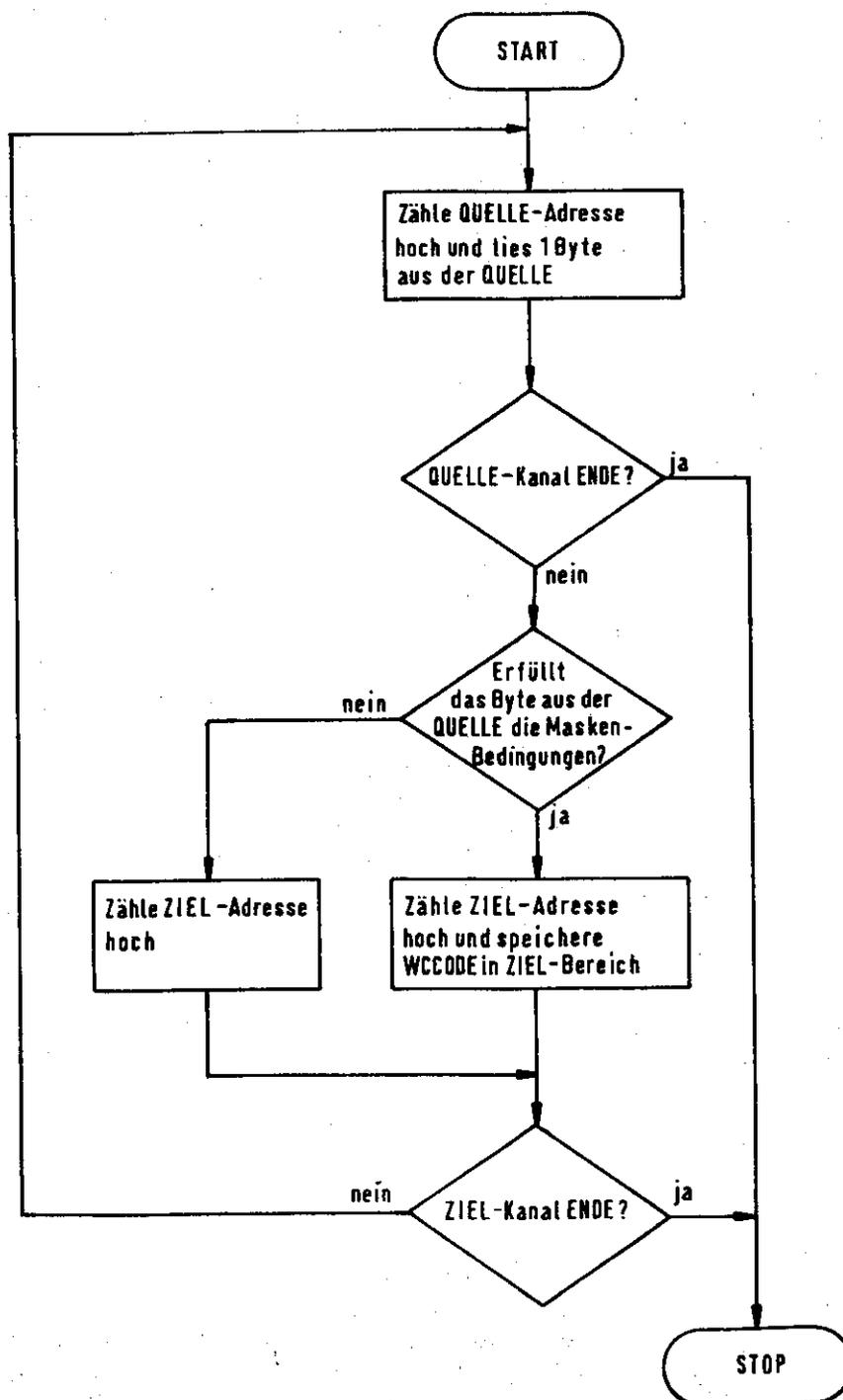


Abb. 6E-2: Schreibe Code — vereinfachtes Ablaufdiagramm

Wird während der Ausführung des Befehls das Zeichen QUELLE ENDE erkannt, wird nichts mehr aus der QUELLE gelesen, sondern es wird nur noch der Inhalt des Puffers zwecks Speicherung zum ZIEL weitergegeben. Wenn nicht ein Zeichen ZIEL ENDE erkannt wird, gelangen alle aus der QUELLE gelesenen Daten in den ZIEL-Speicher. Sobald dieser Vorgang beendet ist, wird Status-Bit 5 gesetzt und eine Unterbrechungs-Anforderung übermittelt.

Wird jedoch ein Zeichen ZIEL ENDE erkannt, so wird die Abhandlung des Befehls vorübergehend angehalten und eine Unterbrechungs-Anforderung übermittelt. Es finden vorerst keine weiteren Übertragungen statt und die etwa noch in den Puffern gehaltenen Daten bleiben unverändert. Neue ZIEL-Adressen können ausgegeben werden. Wird ein weiterer Befehl ,Transportiere Daten' ausgegeben während Status-Bit 4 gesetzt ist, so wird der vorher angehaltenen Vorgang mit der neuen ZIEL-Adresse weiterbehandelt (die QUELLE-Adresse ist wie vorher). Ist vorher Bit 5 gesetzt worden, dann wird nichts aus der QUELLE ausgelesen. Neue ,Transportiere Daten' Befehle sollten jedoch nur nach einem Befehl DVCL gegeben werden. Abbildung 6E-3 zeigt den Befehlsablauf.

Transportiere Daten mit Wraparound

Kommando-Byte: Spalten-Adresse 0-79 (dezimal)

Dieser Befehl transportiert Daten von der QUELLE zum ZIEL, ohne daß Wörter auf dem Bildschirm verstümmelt werden. Zu Beginn des Vorgangs wird der 256-Byte-Hilfsspeicher aus dem QUELLE-Kanal gefüllt. Dann wird ein Wort in den 32-Byte-Sekundärspeicher übertragen und von da über den ZIEL-Kanal an den Speicher weitergegeben. Abbildung 6E-4 zeigt den Befehlsablauf. Ein neuer Befehl ,Transportiere Daten mit Wraparound' sollte nur nach einem Befehl DVCL gegeben werden (siehe auch Anmerkung 5).

Die Behandlung dieses Befehls unterliegt folgenden Regeln:

- Ein Wort ist ein durch SPACE (ASCII 20) oder NULL (ASCII 00) abgeschlossener Bytestring. Dieses einzelne SPACE Zeichen wird zusammen mit den anderen Zeichen im Wortpuffer gespeichert, so daß ein Wort entsteht. Führende SPACE oder NULL Zeichen werden vernachlässigt und nicht in den Wortpuffer gebracht. Wird jedoch ein Befehl ausgegeben, dessen QUELLE auf SPACE oder NULL verweist, dann wird dieses erste SPACE wie ein Wort behandelt. Außerdem können 'tagged' (gekennzeichnete) SPACE (ASCII A0) und NULL (ASCII 80) als Wortbegrenzer verwendet werden.
- Der ZIEL-Kanal speichert das Wort (einschließlich seines nachfolgenden SPACE) nur, wenn es in die Zeile auf dem Bildschirm paßt. Die Länge der Zeile wird durch ein OFL-Kommando bestimmt. Ist das Wort zu lang für die Zeile, dann füllt das Bytestring-Steuerwerk die Zeile mit SPACE auf und bringt das Wort in die nächste Zeile. Ist es länger als 32 Bytes oder länger als die nächste Zeile, dann werden die ersten 32 Bytes des Wortes in die nächsten 32 Speicherzellen des ZIEL-Kanals geschrieben, auch wenn dabei die Bildschirmzeile oder ein Wort verstümmelt werden.
- Zur Ausführung eines Befehls gehört die Speicherung in der ersten Bildschirmzeile. Dieser Speichervorgang wird durch die ,Zeichenmarke' (Character Locator) gesteuert, dessen Wert durch das Kommando-Byte bestimmt wird. Das Bytestring-Steuerwerk nimmt an, daß in der ersten Zeile die Anzahl der für die Speicherung von Wörtern verfügbaren Spalten $Y - X$ ist (Y = spezialisierte Länge der Bildschirmzeile, X = Character Locator).
- Der Befehl wird nur ausgeführt, wenn jedes Wort von einem SPACE-Zeichen gefolgt ist. Wagenrücklauf, Ende der Zeile etc. sind keine Wortbegrenzer im Sinne dieses Befehls.
- Tritt während der Befehlsabhandlung das Zeichen QUELLE ENDE auf, so wird die Abhandlung vorübergehend angehalten, und Bit 5 des Status-Byte wird gesetzt. Die Daten in den Puffern bleiben unverändert. Sobald dann ein neuer QUELLE-Befehl mit ,Transportiere Daten mit Wraparound' auftritt, wird die Befehlsabhandlung unter Benutzung des neuen QUELLE-Befehls und des alten ZIEL-Befehls fortgesetzt.

Tritt während der Befehlsabhandlung das Zeichen ZIEL ENDE auf, so wird die Abhandlung vorübergehend angehalten, und Bit 4 des Status-Byte wird gesetzt. Sobald dann neue ZIEL-Befehle mit nachfolgendem ,Transportiere Daten mit Wraparound' auftreten, wird die Befehlsabhandlung unter Benutzung des neuen ZIEL-Befehls, aber mit den alten QUELLE-Befehlen, fortgesetzt.

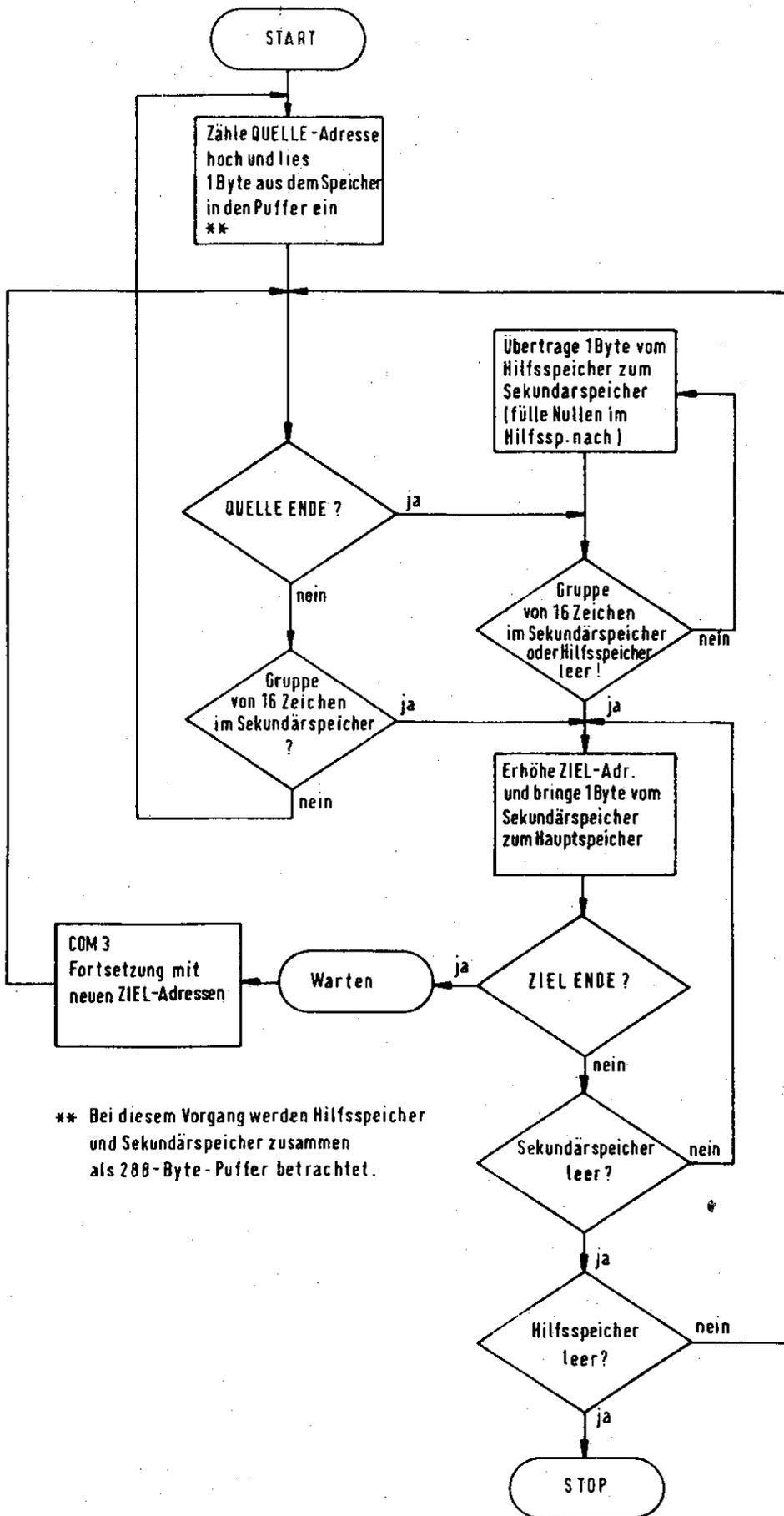


Abb. 6E-3: Transportiere Daten — vereinfachtes Ablaufdiagramm

Diese Eigenschaften sind besonders nützlich, wenn das Bytestring-Steuerwerk in Verbindung mit Speicher- oder Daten-Grenzbedingungen benutzt wird, z. B. bei Ende des Speichers.

Wird die Grenze erreicht, so wird das System neue Befehle ausgeben, um die Befehlsabhandlung dann bei einer neuen Speicherzelle fortzusetzen.

- Durch Festsetzen einer Zeilenlänge (mittels Befehl OFL) auf 80 oder länger kann die Funktion 'Wraparound' blockiert werden (es kommt also lediglich zu einem gepufferten 'Transportiere'-Vorgang).

6E.2 Unterbrechungssteuerung

— Vorrang-Ebene 3 —

Der Ausgabe-Zyklus wurde beendet, d. h. eine oder beide ENDE-Bedingungen wurden erfüllt. Identisch mit IFL Status-Bit 7 (NOT BUSY Flag).

6E.3 Taktsteuerung

Die Synchronisierung basiert auf der Kombination der für jeden Befehl auszuführenden QUELLE und ZIEL Zugriffsoperation

QUELLE- oder ZIEL-Zugriff	=	15 μ s / Byte
Automatischer Puffer-Transfer	=	0,63 μ s / Byte.

```

0 1 2 3 4 - - - - - 77 78 79
A B C C D - - - - - A - -
NEW - L I N E

```

Bildschirmzeilen vor Ausführung

löschen und Text ein Zeichen nach links schieben
(Delete Character)

```

0 1 2 3          77 78 79
A B C D - - - - - A - -
NEW - L I N E

```

Bildschirmzeilen nach Delete Character 'C' unter Verwendung des Befehls 'Transportiere Daten mit Wraparound'

Zeilenlänge = 80
Character Locator = 3

```

0 1 2 3          77 78 79
A B C D - - - - - A - N
EW - L I N E

```

Bildschirmzeilen nach Delete Character 'C' unter Verwendung des Befehls 'Transportiere Daten'

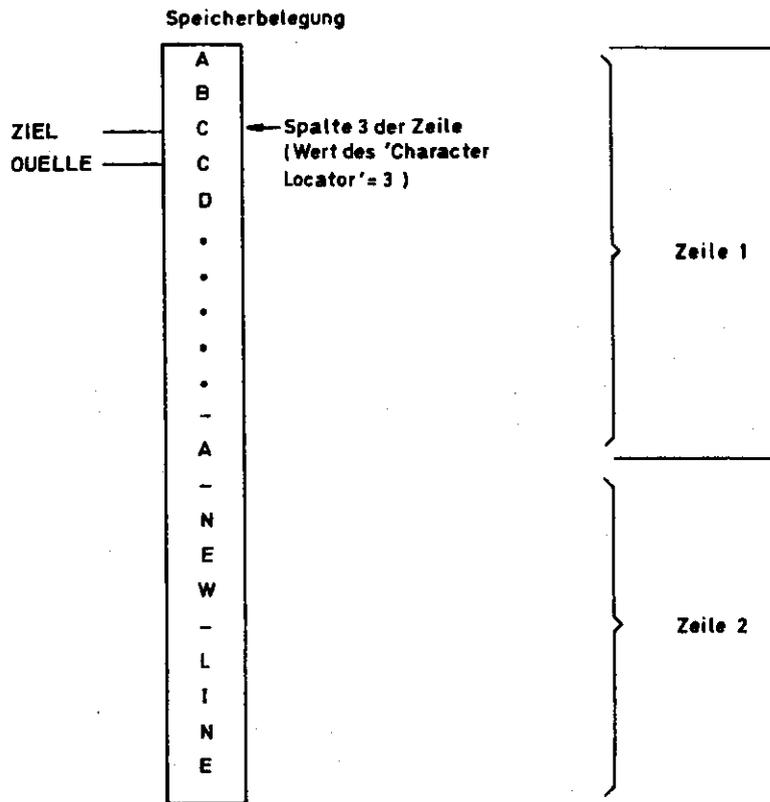


Abb. 6E-5: Beispiel für Textverarbeitung (Text Editing)