

INTERNSCHRIFT Nr. 36

THEMA: Ein Vorschlag zur Verbesserung der Verwaltung und
zur rationellen Ausnutzung der EA-Kachel

VERFASSER: Ramsperger/Sapper

DATUM: 14.1.1970

FORM DER ABFASSUNG

☒ ENTWURF
☐ AUSARBEITUNG
☐ ENDFORM

SACHLICHE VERBINDLICHKEIT

☐ ALLGEMEINE INFORMATION
☐ DISKUSSIONSGRUNDLAGE
☒ ERARBEITETER VORSCHLAG
☐ VERBINDLICHE MITTEILUNG
☐ VERALTET

ÄNDERUNGSZUSTAND

BEZUG AUF BISHERIGE INTERNSCHRIFTEN

Vorkenntnisse aus: IS 28, Hardware Modelle, 4. Bericht
Erweiterung von:
Ersatz für:

BEZUG AUF KÜNFTIGE INTERNSCHRIFTEN

Vorkenntnisse zu:
Erweiterung in:
Ersetzt durch:

ANDERWEITIGE LITERATUR

Beschreibung des Kanalbewachers im BS1 und BS3;
private Aufzeichnungen, bei Herrn Ramsperger einzusehen.

Ein Vorschlag zur Verbesserung von Verwaltung
und Belegung der EA-Kachel

Inhalt:

0. Vorbemerkungen

1. Bisher bekannte Methoden der Speicherzuteilung
2. Ein Vorschlag zur verschnitttfreien Belegung

0. Vorbemerkungen

Wie bereits in Internschrift 28 beschrieben, arbeitet das EA-Befehlswerk (im folgenden kurz als EABW bezeichnet) mit speziellen EA-Werks-Befehlen (kurz EAWB), die nur in der "EA"-Kachel 1 abgelegt sein können; die Zellen mit den Adressen 0 bis 62 sind dabei für Kanalbefehlszellen (KBZ), die Zellen 64 bis 126 für die Eingriffszuordnung vorgesehen.

Die Befehle für einen EA-Vorgang sind durch Fortstartadressen (FA) verknüpft (s.u.).

Eine Aufgabe des Kanalbewachers (KB) ist es, die ihm mit einem Auftrag vorbereiteten EA-Befehle in die für das EABW gültige Form aufzubereiten und an freie Stellen in der EA-Kachel abzu-
legen.

Fortstartadressen sind keine absoluten Adressen, sondern beziehen sich auf den Anfang der EA-Kachel; da sie immer Ganzwerte bezeichnen, ist das letzte Bit für die Adressierung irrelevant - es ergänzt den Operationscode des EAWB. (IS 28, Seiten 6 und 21a).

Zwischen der Absolutadresse n eines EAWB und dessen von der Hardware verarbeiteten Adresse n_{rel} gilt die Beziehung:

$$n_{rel} = n/2 - 2048 \text{ bzw. } n = 2048 + 2 \cdot n_{rel}.$$

Sei m der betrachtete EA-Werksbefehl, $m = \langle \text{EAWB} \rangle$, so enthalten dessen Bits 38 bis 47 die auf den nächsten Befehl verweisende Hardware-FA:

$$FA(m) := \langle \text{EAWB} \rangle_{38, 47}$$

Die tatsächliche Speicheradresse des nächsten EAWB ist

$$\tilde{FA}(m) := 2048 + 2 \cdot FA(m)$$

Für eine EA-Kette m_1, m_2, \dots, m_r gilt:

$$\tilde{FA}(m_i) = m_{i+1} \quad \text{für } 1 \leq i < r.$$

Man kann für alle EA-Ketten einen gemeinsamen Stopbefehl einführen:

$$\tilde{FA}(m_r) = \langle \text{Standard-Stopbefehl} \rangle$$

Eine EA-Kette kann man starten:

direkt (bei freiem Kanal und Gerät)

etwa durch Eintrag eines "leeren" (wegen Abschnittslänge Null nicht transportierenden) Fortstart-Befehls in die KBZ mit $FA(KBZ) = m_{1rel}$ und anschließendem Y-Befehl,

indirekt

durch Anhängen an eine EA-Kette $n_1 \dots n_s$ für den gleichen Kanal vermöge $FA(n_s) := m_{1rel}$.

Ein Teil der Ganzworte in der EA-Kachel ist also durch EA-Ketten laufender oder noch nicht laufender EA-Vorgänge belegt.

Die EA-Kachelverwaltung muß für einen neuen aufzubereitenden EA-Auftrag schnell und ausreichend Platz zur Verfügung stellen, daher sollen die Dauer der Belegung durch eine EA-Kette und eventueller Verschnitt minimal sein.

1. Bisher bekannte Methoden

Wir beschreiben zunächst in der Praxis erprobte Methoden der EA-Kachelverwaltung.

Das inzwischen historische "Doppel-TR4-System" brauchte auf eine effektive Ausnützung der EA-Kachel keine Rücksicht zu nehmen, da es eine dem TR-4-Verteiler entsprechende Befehlsfolge enthält, die ihre eigenen Verteilerlisten führt. Kettenbildungen in der EA-Kachel kommen deswegen allenfalls vor, wenn beim EA-Transport Kachelgrenzen überschritten werden.

Bei den Betriebssystemen BS1[†] und BS3 ist die EA-Kachel in 128 Bereiche zu je 8 Ganzworten aufgeteilt. Dafür existierte eine globale Belegungsliste in 3 Ganzworten. Jedem der Achterbereiche ist genau ein Bit in einem dieser Ganzworte zugeordnet:

$$\left. \begin{array}{l} \langle 1.GW \rangle_i = 8 - \text{Ber. } i \\ \langle 2.GW \rangle_i = 8 - \text{Ber. } 48 + i \\ \langle 3.GW \rangle_i = 8 - \text{Ber. } 96 + i \end{array} \right\} \langle \dots \rangle_i = \text{das } i\text{te Bit von } \langle \dots \rangle$$

Dabei gilt die Festsetzung:

$$\begin{array}{c} \text{frei} \hat{=} L \\ \text{8-Ber.} \left\{ \begin{array}{l} \text{frei} \hat{=} L \\ \text{belegt} \hat{=} 0 \end{array} \right. \text{im zugehörigen Belegungsbit} \end{array}$$

Wenn alle 8-Ber. frei sind, hat also die globale Belegungsliste das Aussehen:

KGZ/EGZ			
00000000	LL...		L
LL...			L
LL...		L	00.. 0

nicht vorhandene
8-Ber.

[†] inzwischen bei AEG-Telefunken nicht weiter verfolgt

Bei Belegung eines 8-Ber. wird das erste Belegungsbit mit L gesucht und daraus die Adresse des 8-Ber. berechnet. Das gefundene Bit wird auf 0 gesetzt.

Ist kein 8-Ber. mehr frei, so wird gewartet, bis einer frei geworden ist.

Für jeden beim KB befindlichen aufbereiteten Auftrag werden in einem dem Auftrag zugeordneten Element Informationen aufgehoben, die für den EA-Verkehr dieses Auftrags von Bedeutung sind. Dazu gehört insbesondere, welche 8-Ber. der Auftrag belegt. Das geschieht mit einer lokalen Belegungsliste, die genauso aufgebaut ist wie die globale, nur mit dem Unterschied, daß hier für die relevanten Bereiche gilt: frei $\hat{=}$ 0 und L $\hat{=}$ belegt. Ein neu belegter 8-Ber. wird auch hier entsprechend vermerkt.

Ein aufbereiteter EAWB wird in den nächsten freien Platz im 8-Ber. abgelegt. Ist dieser voll, wird ein neuer aufgefördert.

Die Freigabe der 8-Ber. eines Auftrags geschieht durch Addition im Speicher

$\langle k. \text{ GW globale Liste} \rangle := \langle k. \text{ GW globale Liste} \rangle + \langle k. \text{ GW lokale Liste} \rangle$
und

$\langle k. \text{ GW lokale Liste} \rangle := 0$

für $1 \leq k \leq 3$.

Die starre Einteilung in 8-Ber. hat wohl folgende Gründe:

1. 8 ist Zweierpotenz
2. bei 4-Ber. zu hoher Verwaltungsaufwand
3. bei 16-Ber. (oder größeren) zu viel Verschnitt.

Beim KB können sich an Aufträgen befinden

im BS1: 2/Gerät	> 141 GW *)	} Bedarf für Listen der EA-Kachel- belegung
im BS3: 8/Kanal	> 387 GW	

*) bei 23 EA-Geräten

Der Verschnitt ist abhängig von der Art der Aufträge und der Häufigkeit ihres Auftretens.

Um etwa sequentielle Information des Umfanges $1K$ von der Trommel in eine Kachel zu holen, benötigt man nach den Konventionen aus Abschnitt 0 eine EA-Kette von 2 EAWB; dadurch würde aber ein ganzer 8-Ber. als belegt gelten.

Es lassen sich durchaus Fälle starken EA-Verkehrs denken, in denen EA-Aufträge warten müssen, obwohl in Form von unnötigen Speicherverschnitt genügend Platz für ihre Aufbereitung vorhanden wäre. Insbesondere dann, wenn die zukünftige EA-Belastung unbekannt ist, sollte man eine möglichst verschnittfreie Belegung anstreben.

2. Eine verschnittfreie Belegung der EA-Kachel

Die nachfolgend beschriebene Methode nutzt wesentlich den Zwei-Adress-Mechanismus der EAWB aus.

Die Belegung der EA-Kachel ist verschnittfrei.

Die Aufbereitung der EAWB kann auch dann beginnen, wenn nur ein EAWB-Platz frei ist (während der Aufbereitungszeit eines EAWB wird vielleicht schon neuer Platz frei).

Dynamisch und statisch ist diese Methode weniger aufwendig als die von BS1 und BS3 benutzte.

Ähnlich wie bei der EA-Kette werden auch die freien GW $f_1 \dots f_k$ der EA-Kachel zu einer Frei-Kette verknüpft, die der Freilliste in der Speicherorganisation der Listenprogrammierung entspricht.

$$\tilde{FA}(f_i) = f_{i+1} \quad \text{für } 1 \leq i < k \quad (\text{so}),$$

$$FA(f_k) = 0 \quad (\text{anders}).$$

Es wird ein Zeiger z eingeführt:

$$\langle z \rangle = \begin{cases} 0 & \text{kein EAWB-Platz frei} \\ f_{1\text{rel}} & \text{sonst} \end{cases}$$

Ein EAWB-Platz kann belegt werden, wenn $\langle z \rangle \neq 0$ ist. Nach der Belegung wird z umgestellt: $\langle z \rangle := \text{FA}(f_1^{\text{alt}})$.

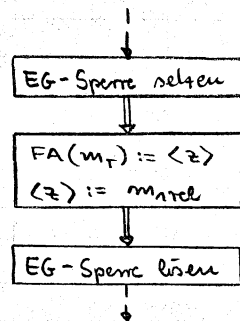
Der Platz einer EA-Kette wird freigegeben durch Vorsetzen vor die bisherige Frei-Kette vermöge FA-Verknüpfung.

Im einzelnen können Belegung und Freigabe so aussehen:

7.7

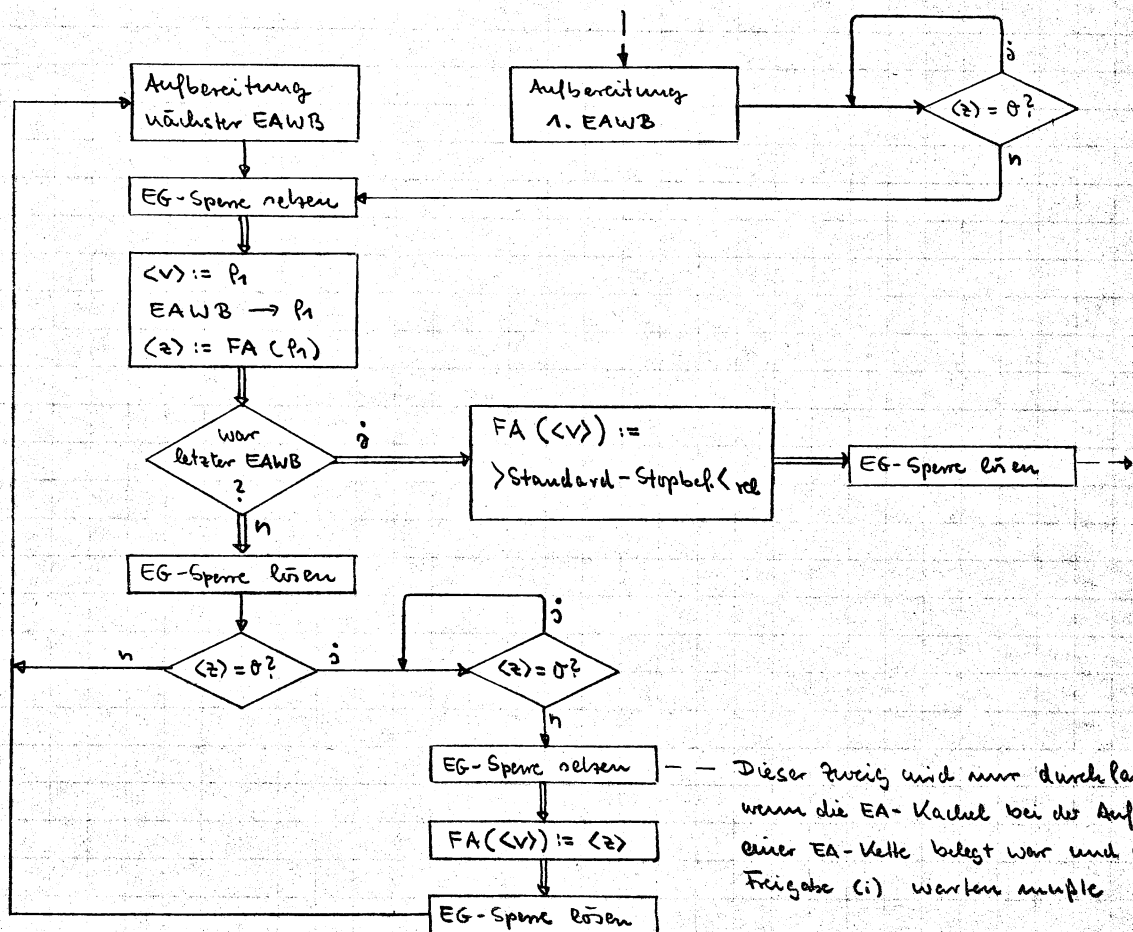
(i) Freigabe des Platzes einer EA-Kette $m_1 \dots m_r$:

(dem Aufgebenden muss m_1, m_r bekannt sein (ist es auch aus anderen Gründen))



Bei Vorhandensein einer Frei-Kette ($\langle z \rangle \neq \emptyset$) wird diese vorne um die aufzugebende EA-Kette verlängert, anderenfalls wird die EA-Kette zur Frei-Kette

(ii) Belegung eines EAWB-Platzes:



Dieser Zweig wird nur durchlaufen, wenn die EA-Kachel bei der Aufbereitung einer EA-Kette belegt war und auf Freigabe (i) warten musste