

**KERNFORSCHUNGSZENTRUM
KARLSRUHE**

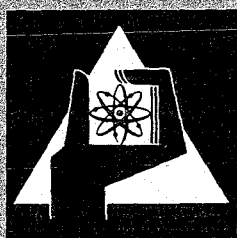
Januar 1973

KFK 1531

Institut für Datenverarbeitung in der Technik

**Eingriffsorganisation für schnelle Aufgabenwechsel
in einem Realzeitsystem**

G. Hepke, H. Herbstreith



**GESELLSCHAFT
FÜR
KERNFORSCHUNG M.B.H.**

KARLSRUHE

Als Manuskript vervielfältigt

Für diesen Bericht behalten wir uns alle Rechte vor

**GESELLSCHAFT FÜR KERNFORSCHUNG M. B. H.
KARLSRUHE**

KERNFORSCHUNGSZENTRUM KARLSRUHE

Oktober 1972

KFK 1531

Institut für Datenverarbeitung in der Technik

Eingriffsorganisation für schnelle Aufgabenwechsel
in einem Realzeitsystem

G. Hepke

H. Herbstreith

Gesellschaft für Kernforschung m.b.H., Karlsruhe

Kurzfassung

Alle Arbeiten in einem DV-System können in weitgehend unabhängige kooperierende Prozesse gegliedert werden, die auf unabhängigen Pseudoprozessoren ablaufen. Eine der wesentlichen Aufgaben des Betriebssystemkerns (Nukleus) ist es, die physikalischen Prozessoren und ihre Kommunikationswege auf das geforderte Netz von Pseudoprozessoren abzubilden.

Ein Teil dieser Abbildungsfunktion des Nukleus wird von der Eingriffsorganisation erfüllt, die sich speziell mit der Eingliederung der Eingriffsprozessoren in das System befaßt, d.h. der Prozessoren, deren wesentlichstes Kommunikationsmittel der Eingriff (Interrupt, Programmunterbrechung) ist.

Die Eingriffsorganisation eines Realzeitsystems muß neben guter Auslastung und variabler Konfigurierbarkeit der Prozessoren besonders die schnelle und flexibel programmierbare Reaktion auf Eingriffe ermöglichen.

Dieser Bericht erläutert das allgemeine Konzept einer Eingriffsorganisation für Realzeitsysteme am Beispiel einer speziellen Implementation mit entsprechend pragmatischen Vereinfachungen.

Abstract

All activities in a data processing system may be structured in nearly independent cooperating processes running on independent pseudoprocessors.

One of the essential tasks of the nucleus of the operating system is to project the physical processors and their communication lines into the network of pseudoprocessors.

One part of this nucleus function is done by the interrupt organisation which has the specific purpose of incorporating into the system the interrupt processors, that means the processors with interrupts as substantial equipment for communication.

Besides good loading and variable configuration of the processors the interrupt organisation of a realtime system must ensure the rapid and flexibly programmable reaction to interrupts.

This report describes the general concept of an interrupt organisation for realtime systems taking the example of a specific implementation with correspondingly pragmatical simplifications.

Einführung

Die Entwicklung, Handhabung und interne Organisation komplexer DV-Systeme sind ohne deren modularen Aufbau nicht mehr denkbar.

Die Gliederung des Systems in Baueinheiten erlaubt die weitgehend unabhängige Entwicklung, Implementation und Wartung der Einheiten. Modular kann das System in seiner Hard- und Softwarekonfiguration an spezielle Anforderungen angepaßt werden. Die Modularisierung der Arbeitsabwicklung im System in weitgehend unabhängige Prozesse ermöglicht flexible Ablaufstrategien zur Optimierung des Systemverhaltens. Auch komplexe Vorgänge bleiben transparent und dadurch manipulierbar.

Diese Vorteile machen die Modularisierung komplexer Systeme unumgänglich; entstehende Nachteile müssen in Kauf genommen und, soweit möglich, ausgeglichen werden:

- Die Modularisierung des Systems in Hardware-Baueinheiten, wie Rechnerkern, Kanäle, Peripheriegeräte überschneidet sich mit der 'Modularisierung' der Ablauforganisation in Prozesse, so daß z.B. im Ablauf eines Prozesses Schnittstellen zwischen verschiedenen Baueinheiten überwunden werden müssen.
- Der modulare Aufbau erfordert zusätzlichen Aufwand an Organisation, wenn die erwünschte statische und dynamische Flexibilität erreicht werden soll. Dies gilt bezüglich der variablen Konfigurierbarkeit der Hard- und Softwaremodule wie auch des Ablaufs der Prozesse, deren vernünftige Kooperation sicherzustellen ist.

Diese nachteiligen Folgen der Modularisierung können in Zukunft voraussichtlich durch die Mikroprogrammierbarkeit der Baueinheiten relativiert werden, derzeit erfordern sie erhöhten Softwareaufwand.

Dieser Bericht beschreibt das Konzept einer Eingriffsorganisation, das diesen Aufwand gering genug hält, um auch auf einem einfachen Rechner realisierbar zu sein, und das außerdem den speziellen Anforderungen eines Realzeitsystems gerecht wird. Das Konzept wurde weitgehend im Laborautomatisierungssystem CALAS70 realisiert, dessen innerer Aufbau in (1) beschrieben ist. Das Realzeitsystem

RESY71 der Firma AEG-Telefunken (2) wurde in Anlehnung an CALAS70 entwickelt.

Systementwurf auf Basis von Pseudoprozessoren

Alle Aufgaben, die das System bewältigen muß, werden von unabhängigen kooperierenden Prozessen ausgeführt. Diese laufen auf Pseudoprozessoren ab, die ihnen vom Betriebssystemkern (Nukleus) zur Verfügung gestellt werden. Der Begriff Pseudoprozessor deutet an, daß die physikalischen Prozessoren des Systems vom Nukleus einer Abbildung unterworfen werden.

Die Fähigkeiten der Pseudoprozessoren (z.B. Befehlsvorrat) müssen den Anforderungen der Prozesse entsprechen; soll also jeder Prozeß auf jedem Pseudoprozessor lauffähig sein, müssen die Pseudoprozessoren gleich sein. Insbesondere muß jeder über den gleichen Befehlsvorrat zur Kooperation und Kommunikation der Prozesse verfügen.

Die Pseudoprozessoren sollen hinsichtlich der Prozesse streng sequentiell arbeiten; Überholvorgänge sind nicht zugelassen. Untereinander arbeiten die Pseudoprozessoren parallel bzw. quasiparallel, so daß der Grad der Parallelität bzw. Quasiparallelität des Systems der Zahl der Pseudoprozessoren entspricht.

Die Prozesse selbst sind sequentielle Prozesse, deren richtiger Ablauf allein von der Einhaltung der vorgeschriebenen Befehlsfolge abhängt. Da sie zu jedem im System definierbaren Zeitpunkt einen definierten Zustand haben, können sie beliebig unterbrochen und weitergeführt werden.

Abbildung von Prozessoren auf Pseudoprozessoren

Zur Realisierung der Pseudoprozessoren benötigt der Nukleus die physikalischen Prozessoren des Systems, worunter die physikalisch parallel und weitgehend unabhängig arbeitsfähigen Einheiten des Systems zu verstehen sind: Rechnerkerne, Kanäle, Peripheriegeräte.

Diese Prozessoren sind vom beschriebenen Ideal der Pseudoprozessoren weit entfernt. Sie sind bereits von ihrem technischen Aufbau für spezielle Arbeiten eingerichtet, und die Kommunikationswege untereinander sind unterschiedlich und in Anzahl und Breite eingeschränkt; Bild 1 erläutert diesen Unterschied zwischen dem Netz der Pseudoprozessoren und dem der physikalischen Prozessoren.

Die Realisierung identischer Pseudoprozessoren durch die Abbildung aller existierenden realen Prozessoren in jedem einzelnen Pseudoprozessor würde eine beliebige Verteilung der Prozesse auf das entstehende symmetrische Netz gestatten. Da die sequentiellen Prozesse der Pseudoprozessoren die Prozessoren aber nicht parallel beanspruchen können, führt diese Konstruktion zu einer geringen Auslastung der Prozessoren.

Will man also die physikalische Parallelität der Prozessoren im System der Pseudoprozessoren ausnützen, müssen einzelne Prozessoren in einzelne Pseudoprozessoren abgebildet werden. Um ein Höchstmaß an Parallelität und damit an Prozessorausnutzung zu erreichen, würde die Abbildung im Maßstab 1:1 genügen; um aber Überholvorgänge von Prozessen an einem Prozessor zu ermöglichen und so das System dynamisch flexibel zu machen, muß dieser Prozessor auf soviele Pseudoprozessoren abgebildet werden, wie Überholungen geschachtelt möglich sein sollen. Diese Prozessorvergabe an Pseudoprozessoren ist eine wesentliche Funktion des Nukleus.

Voraussetzung für die Abbildbarkeit eines Prozessors auf mehrere Pseudoprozessoren ist, daß dieser Prozessor sequentiell im Sinne der sequentiellen Prozesse der Pseudoprozessoren arbeitet, denn nur dann kann er in der Bearbeitung eines Prozesses unterbrochen werden, ohne dessen korrekten Ablauf zu beeinträchtigen. Diese Voraussetzung ist i. allg. nur für die Rechnerkerne erfüllt, die im folgenden als 'Standardprozessoren' bezeichnet werden sollen. Dagegen arbeiten andere Prozessoren nicht sequentiell, ein Datentransport auf einem Kanal zum Beispiel läßt sich nicht beliebig unterbrechen und fortführen.

Prozessen auf solchen 'Eingriffsprozessoren' - diese Bezeichnung wird weiter unten klar - kann man also keine Überholbarkeit zubilligen, d.h. man bildet Eingriffsprozessoren 1:1 auf Eingriffspseudoprozessoren ab.

Kommunikation zwischen Standard- und Eingriffspseudoprozessoren

Für Kooperation und Kommunikation der Prozesse sollen alle Pseudoprozessoren über den gleichen Befehlsvorrat verfügen. Bild 2 zeigt einen einfachen Befehlsvorrat, der für viele Anwendungen eines Realzeitsystems genügt, sowie die Wirkungsrichtung der Befehle.

Mit den Befehlen CALL und WAKE UP wird von einem Prozeß ein anderer Prozeß gestartet bzw. fortgestartet, mit SLEEP und EXIT beendet sich ein Prozeß vorläufig bzw. endgültig. CALL und EXIT bzw. SLEEP und WAKE UP ergänzen sich jeweils und klammern die Dauer der Existenz bzw. der Ruhe eines Prozesses ein.

Dieser Befehlsvorrat muß bezüglich der Eingriffspseudoprozessoren eingeschränkt werden. Da die auf ihnen ablaufenden Eingriffsprozesse ununterbrechbar sind, können sich diese Prozesse auch nicht selbst unterbrechen. Damit wird der Befehl SLEEP für einen Eingriffspseudoprozessor unausführbar und der Befehl WAKEUP auf einen Eingriffsprozeß unnötig.

Unter Berücksichtigung dieser Ausnahmen stehen allen Pseudoprozessoren die gleichen Befehle zu Kooperation und Kommunikation zur Verfügung. Ihre Bereitstellung ist neben der Prozessorvergabe eine weitere wesentliche Aufgabe der Abbildungsfunktion des Nukleus.

Abbildung von Prozessoren auf Eingriffspseudoprozessoren

In Bild 3 sind zwei Ablaufdiagramme dargestellt, die die Maßnahmen zur Realisierung von Eingriffspseudoprozessoren erläutern helfen sollen. Bild 3a zeigt das Zusammenspiel zweier Standardpseudoprozessoren in einem System mit einem Standardprozessor, wie es am häufigsten vorkommt: Entsprechend den Kooperationswünschen der Prozesse und der Vergabestrategie des Nukleus wird der Prozessor zwischen Nukleus und den Standardpseudoprozessoren hin- und hergeschaltet.

Ein Beispiel des komplizierteren Zusammenspiels zwischen Standard- und Eingriffspseudoprozessoren zeigt Bild 3b: Der vom Standardpseudoprozessor aus abgesetzte Auftrag an den Eingriffspseudoprozessor

setzt diesen in Tätigkeit, indem der Nukleus den notwendigen Eingriffsprozessor aussucht und startet. Danach aktiviert der Nukleus wiederum den Standardpseudoprozessor. Nach dessen SLEEP vergibt der Nukleus den Standardprozessor an einen anderen Standardpseudoprozessor nächster Dringlichkeit. Für den Ablauf des Beispiels ist erst wieder der Zeitpunkt der Beendigung der Aktivität des Eingriffsprozessors wichtig. Durch einen Eingriff - daher der Name des Prozessors - in die Aktivität des Standardprozessors wird über den Nukleus die Fortführung des Eingriffsprozesses auf dem Standardprozessor erzwungen. Der Nukleus muß den Eingriffsprozeß vom Eingriffsprozessor auf den Standardprozessor verlagern und realisiert mit dieser Funktion den entsprechenden Eingriffspseudoprozessor.

Die dargestellte Ablauforganisation ist den Anforderungen eines Realzeitsystems angepaßt, für das naturgemäß dynamische Flexibilität und schnelle Reaktion auf Eingriffe wichtiger sind als hohe Auslastung aller Prozessoren, allgemein aller Betriebsmittel: Entsprechend dem Realzeitanspruch der Eingriffe wird ein Eingriffsprozeß unter Umgehung der Vergabefunktion des Nukleus für den Standardprozessor unbedingt fortgesetzt. Dies erhöht die Reaktionsgeschwindigkeit auf Eingriffe. Die Entscheidung, ob die weitere Reaktion auf den Eingriff derart unter höchster Dringlichkeit ablaufen muß und damit auch andere Eingriffe verzögern darf, liegt beim Eingriffsprozeß selbst. Er kann die notwendigen Arbeiten sofort selbst verrichten oder mittels CALL einen Standardprozeß aktivieren, dessen Ablauf dann der Vergabefunktion des Nukleus unterworfen ist.

Diese Alternative kann nicht ein für alle Mal im Eingriffspseudoprozessor entschieden werden, da in Realzeitsystemen die Entscheidung vom aktuellen Prozeß abhängt, d.h. auftragsspezifisch sein kann. Der Ablauf des Eingriffsprozesses nach dem Eingriff ist daher vom auftraggebenden Standardprozeß steuerbar, der das gewünschte Eingriffsprogramm als Auftragsparameter angeben kann.

Schnittstelle Standard-/Eingriffsprozessor

Für den Ablauf von Eingriffsprozessen sind, wie Bild 3b verdeutlicht, die beiden Arbeitsphasen des Nukleus typisch, in denen der Eingriffsprozeß vom Standard- auf den Eingriffsprozessor bzw. umgekehrt verlagert wird.

Diese Schnittstellenfunktion im Nukleus stellt sicher, daß Änderungen der Hardwarekonfiguration bzw. der entsprechenden Kommunikationswege zwischen Standard- und Eingriffsprozessoren sich nicht auf die Kommunikation zwischen Eingriffs- und Standardpseudoprocessoren auswirken. Außerdem können hier notwendige Funktionen des Eingriffswerkes, die von dessen Hardware nicht bereitgestellt werden, durch Software ergänzt werden. Der Nukleus erzeugt so ein Pseudoeingriffswerk, dessen Eingriffssignale den Anforderungen der Eingriffspseudoprocessoren entsprechen.

Implementierung

Für die spezielle Implementierung des vorgestellten Konzepts im System CALAS70 wurde angenommen, daß zu einer Zeit für jeden Eingriffspseudoprocessor jeweils nur ein korrespondierender Standardpseudoprocessor existiert, dessen Prozesse Aufträge an den Eingriffspseudoprocessor absetzen können.

Diese Einschränkung vereinfacht die Ablaufsteuerung und die Auftragsverwaltung der Eingriffspseudoprocessoren: Die Aktivierung der Eingriffsprozessoren kann aus dem Nukleus in den korrespondierenden Standardpseudoprocessor verlegt werden, die Ablaufsteuerung braucht dann nur Standardpseudoprocessoren zu berücksichtigen. Gesonderte Auftragswarteschlangen für die Eingriffspseudoprocessoren werden unnötig.

Diese pragmatische Modifizierung beeinträchtigt die dynamische Flexibilität dieses speziellen Systems nicht und verbessert dessen Zeit- und Arbeitsspeicherverhalten. Aus dem gleichen Grunde wurden statt CALL und WAKEUP für die Eingriffsprozesse die gesonderten Befehle CALLIT und WAKEIT eingeführt (IT für Interrupt = Eingriff).

Eingriffswerk und Pseudoeingriffswerk

Das Eingriffswerk des Rechners TR86 belegt entsprechend den von ihm direkt unterscheidbaren Eingriffssignalen eine Anzahl von festgelegten Kernspeicherplätzen. Ein Eingriff erzwingt die Unterbrechung des Programmlaufs des arbeitenden Standardprozesses und den Beginn einer Programmbearbeitung an dem zugehörigen Kernspeicherplatz; die Stelle der Programmunterbrechung wird in einem weiteren Kernspeicherplatz registriert.

An diese Eingriffspunkte sind die Eingriffsprozessoren in unterschiedlicher Weise angeschlossen: Beispielsweise liegen die vier unabhängigen Kanäle an einem einzigen Eingriffspunkt, andere Eingriffsprozessoren belegen mehrere Eingriffspunkte.

Von dem Pseudoeingriffswerk ist zu fordern, daß es jedem Eingriffspseudoprocessor einen eigenen Eingriffspunkt zur Verfügung stellt. Zu diesem Zweck bildet die entsprechende Nukleusfunktion den Eingriff auf einen Pseudoeingriff ab, der vom externen Teil des Eingriffsprozesses zu dessen internem Teil führt.

Um Konfigurationsänderungen leicht zu ermöglichen, werden diese Pseudoeingriffe über logische Anschlüsse geführt, die die Verbindung zwischen Eingriffs- und Eingriffspseudoprocessor herstellen; die Zuordnungen sind variabel.

Die Bilder 4 und 5 sollen die Initialisierung eines Eingriffsprozesses und dessen Ablauf erläutern.

Ankettung eines Auftrags an einen Eingriffspseudoprocessor

(Bild 4)

Bild 4 beschreibt die Verkettungen eines Auftrags an einen Eingriffspseudoprocessor.

Charakteristisch für die Auftragsbeschreibung an einen Eingriffspseudoprocessor ist der Zeiger auf ein Eingriffsprogramm, das im Zuge des Eingriffsprozesses bearbeitet werden soll. Es besteht im einfachsten Fall nur aus einem Weckruf (WAKEIT) an den auftragge-

benden Standardprozeß oder aus einem neuen Auftrag (CALL) an einen entsprechenden Standardprozeß.

Der in der Auftragsbeschreibung angegebene Eingriffspseudoprocessor ermöglicht über die beiden Zuordnungen der Anschlußbeschreibungsliste die Verknüpfung des Auftrags mit dem entsprechenden Eingriffsprozessor. Die Initialisierung des Prozessors wird sofort nach Durchführung des CALLIT vom auftraggebenden Standardprozeß veranlaßt.

Ablauf eines Eingriffs (Bild 5)

Ist der externe Teil eines Eingriffsprozesses auf den Eingriffsprozessor beendet, sendet dieser sein Signal an das Eingriffswerk und verursacht eine Unterbrechung des arbeitenden Standardprozesses.

Die Eingriffsfunktion des Nukleus stellt den Status des unterbrochenen Prozesses sicher und ermittelt die Nummer des betroffenen logischen Anschlusses. Über die Zuordnung von Anschluß zu Eingriffsprozessor in der Anschlußbeschreibungsliste wird der zugehörige Auftrag und das Eingriffsprogramm zur Fortsetzung des Eingriffsprozesses gefunden.

Schlußbemerkung

Das System, dessen Eingriffsbehandlung beschrieben wurde, entspricht in seinem Aufbau dem Schichtenmodell: Über der Basisschicht der Hardware liegt die Schicht des Nukleus und darüber die der kooperierenden Prozesse.

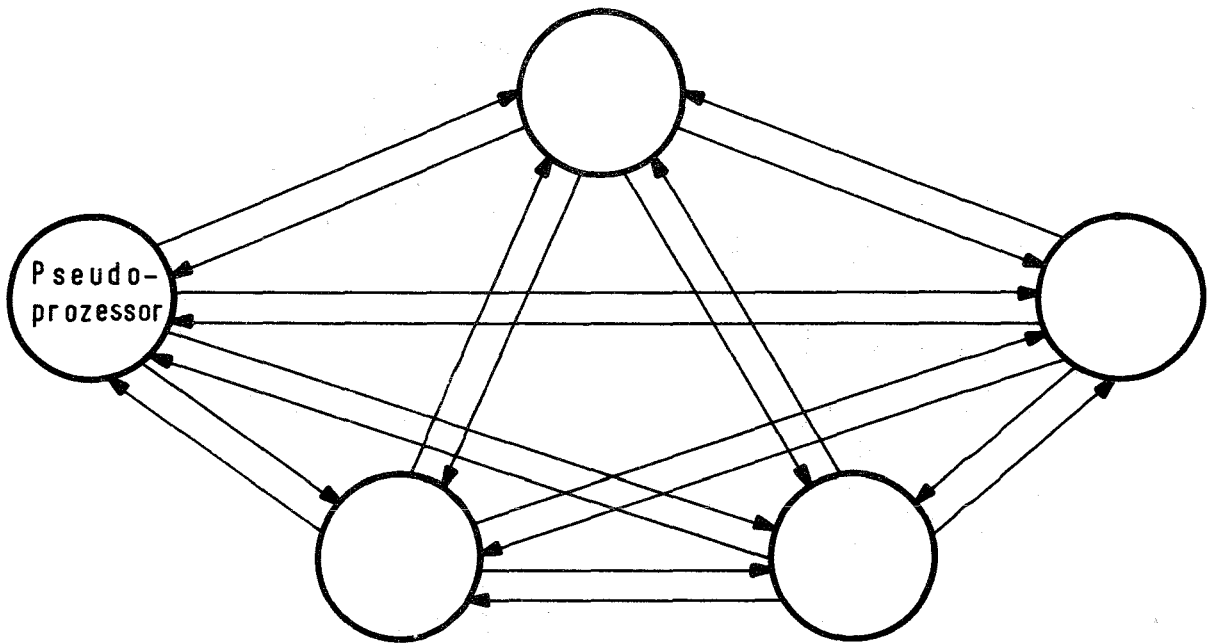
Die Anforderungen an ein Realzeitsystem und dessen klares Aufgabenprofil erfordern und ermöglichen einen Systemaufbau aus wenigen Schichten mit einfacher Struktur innerhalb der Schichten.

Dies läßt eine Weiterentwicklung gerade dieser Systeme möglich erscheinen, die Funktionen des Nukleus in einer Schicht mikroprogrammierbarer Hardware, z.B. einem 'Koordinationswerk' zu realisieren, das die Koordinierung aller Prozesse auf allen Prozessoren des Systems integriert.

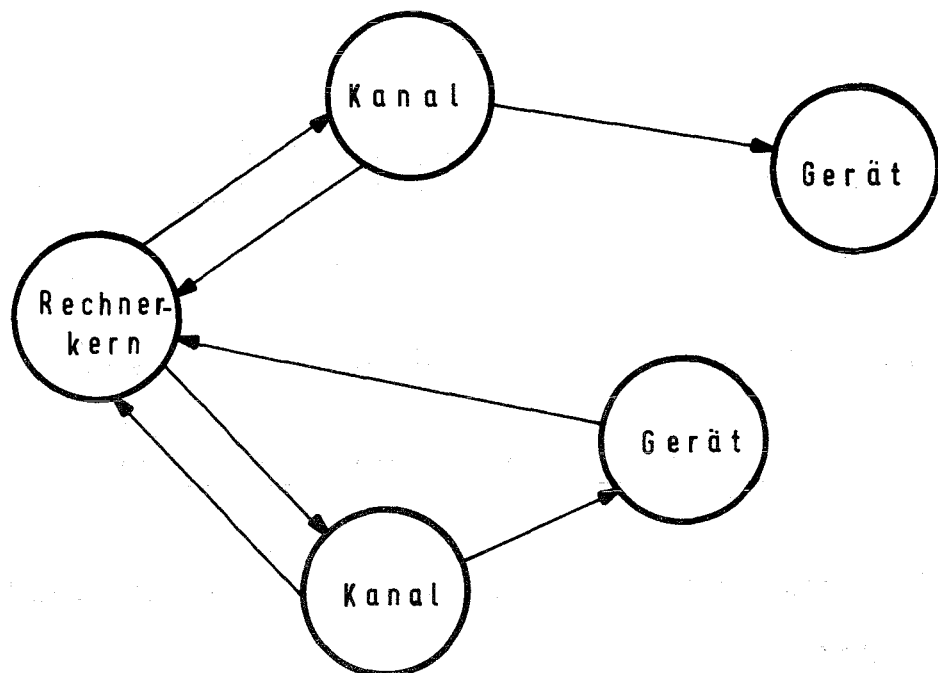
Literaturverzeichnis

- (1) Herbstreith, H., Hepke, G.
Ablaufsteuerung für ein Realzeitsystem mit
einfacher Hardwarestruktur
Bericht KFK 1530, August 1972

- (2) AEG-Telefunken, Fachbereich Informationstechnik
Realzeit-Betriebssystem RESY 71
TR86 Programmbibliothek

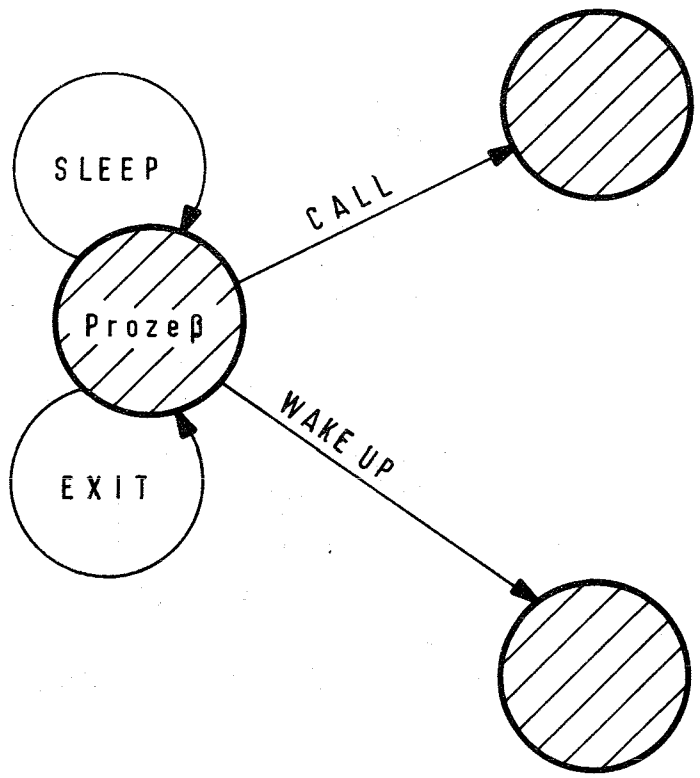


a) Kommunikationswege zwischen 5 idealen Pseudoprozessoren

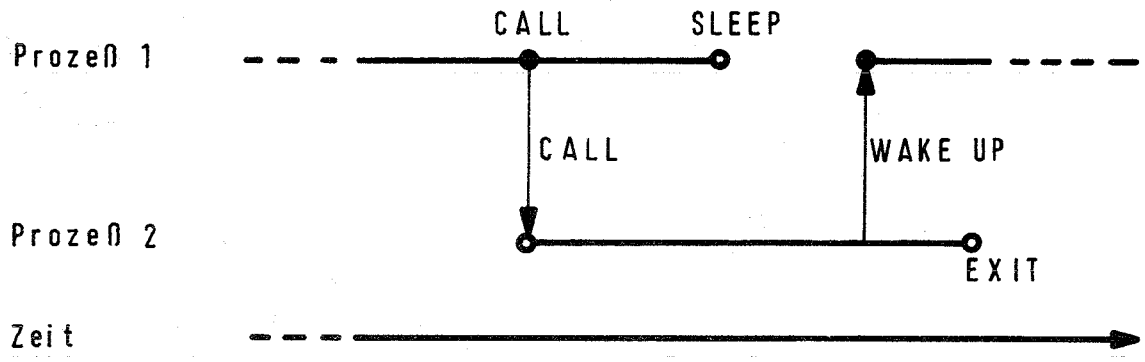


b) Kommunikationswege zwischen 5 realen Prozessoren

Bild 1: Gegenüberstellung der Kommunikationswege zwischen idealen Pseudoprozessoren und zwischen realen Prozessoren

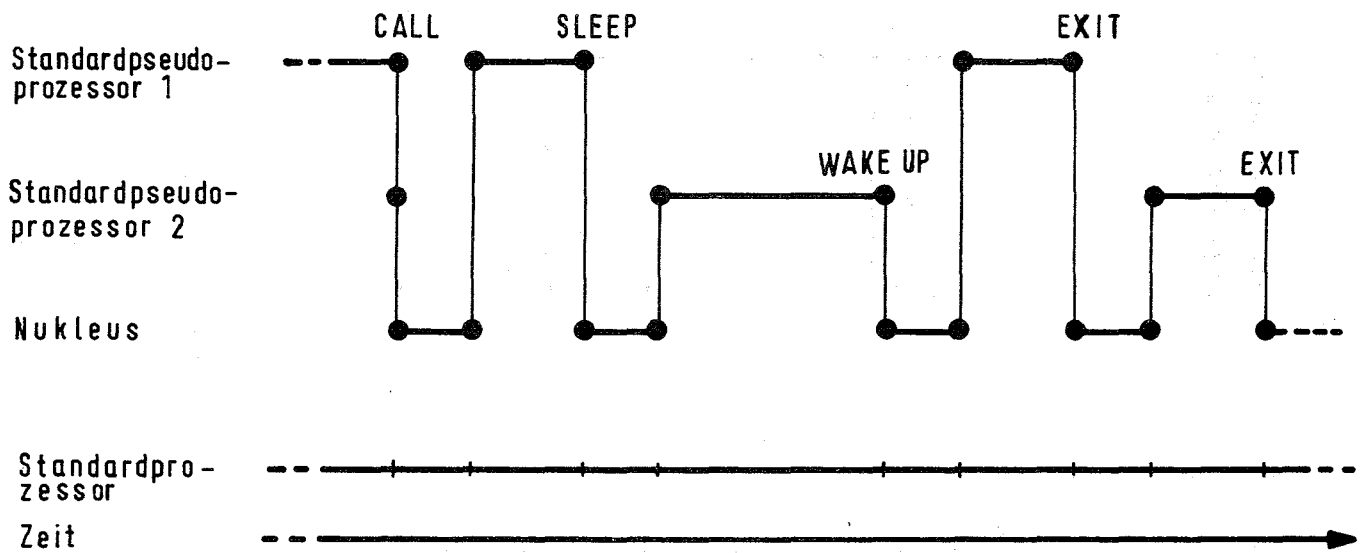


a) Befehle und ihr Wirkungsziel

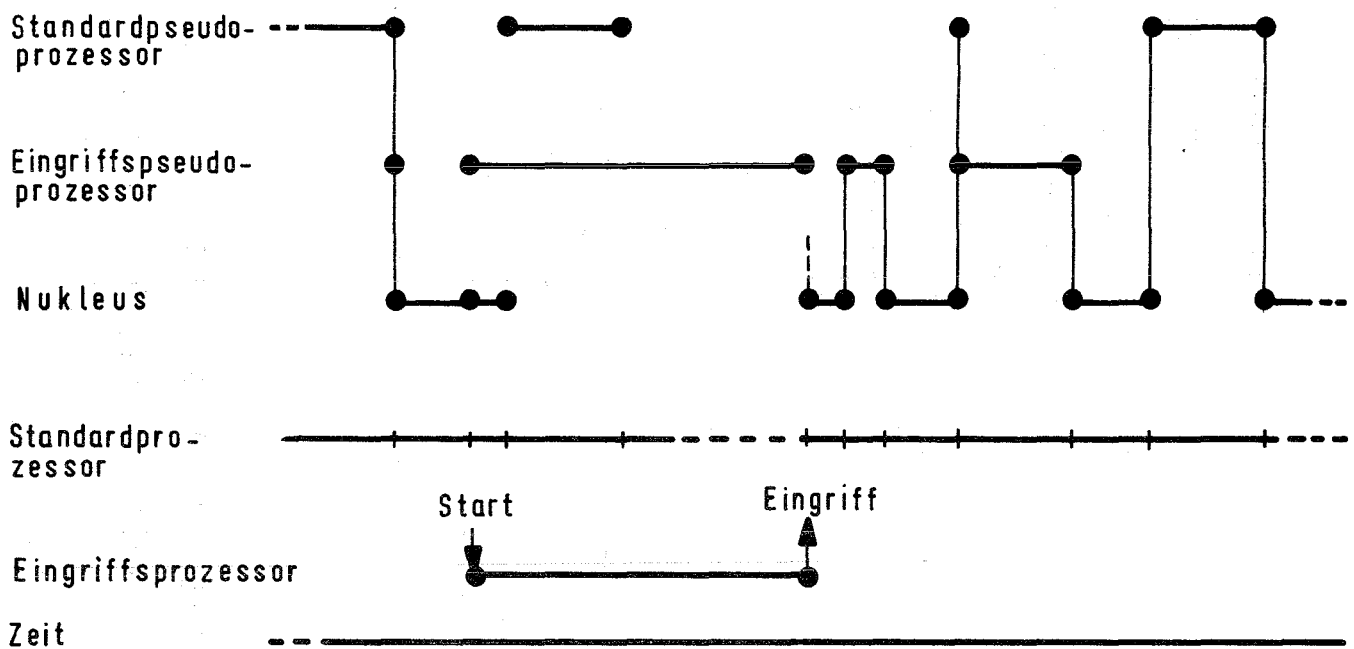


b) Typisches Ablaufdiagramm zweier kooperierender Prozesse

Bild 2: Einfacher Befehlsvorrat von Pseudoprozessoren, zu Kooperativ von Prozessen



a) Prozesse auf Standardpseudoprocessoren



b) Prozesse auf Standard- und Eingriffspseudoprocessoren

Bild 3: Ablaufdiagramme des Zusammenspiels von Prozessen auf Pseudoprocessoren

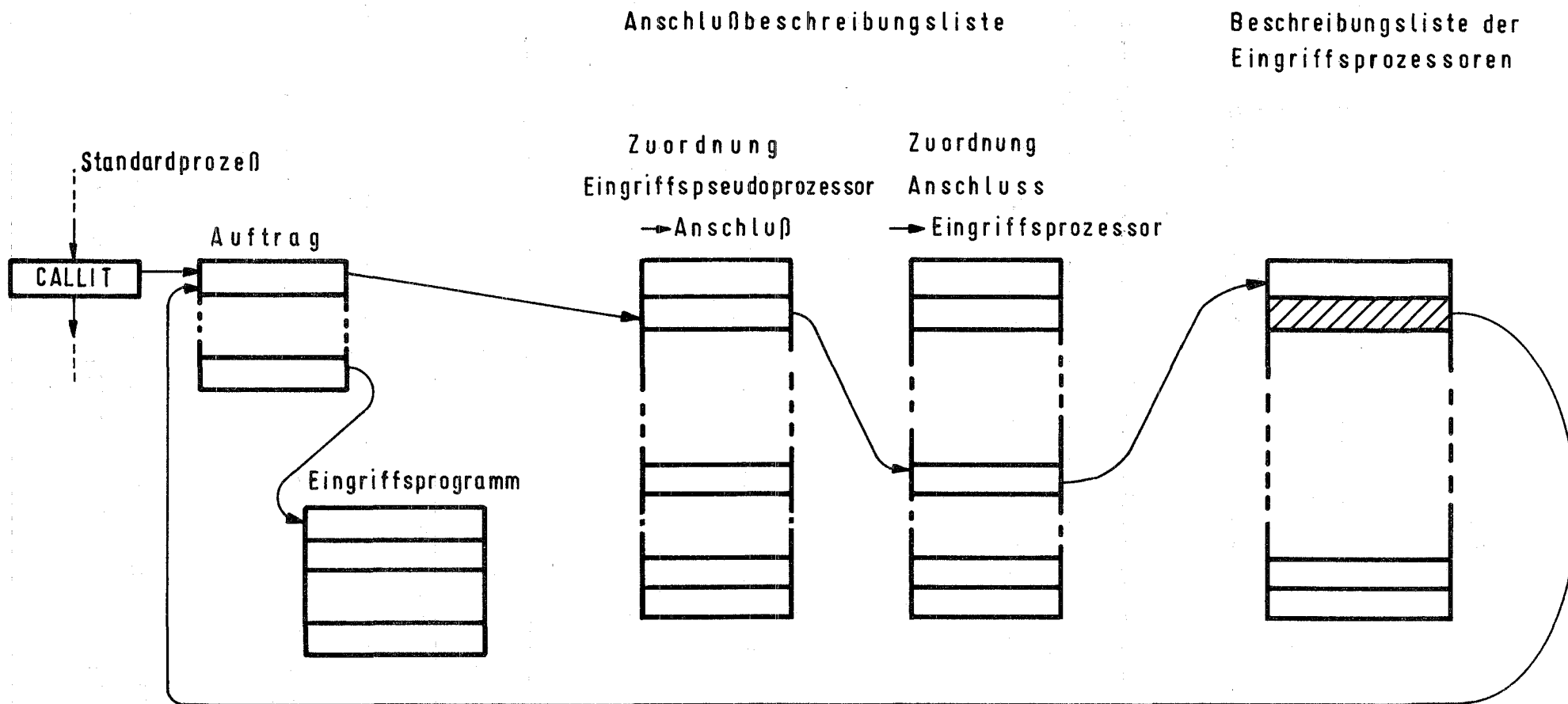


Bild 4: Ankettung eines Auftrags für einen Eingriffspseudoprocessor an den entsprechenden Eingriffsprozessor

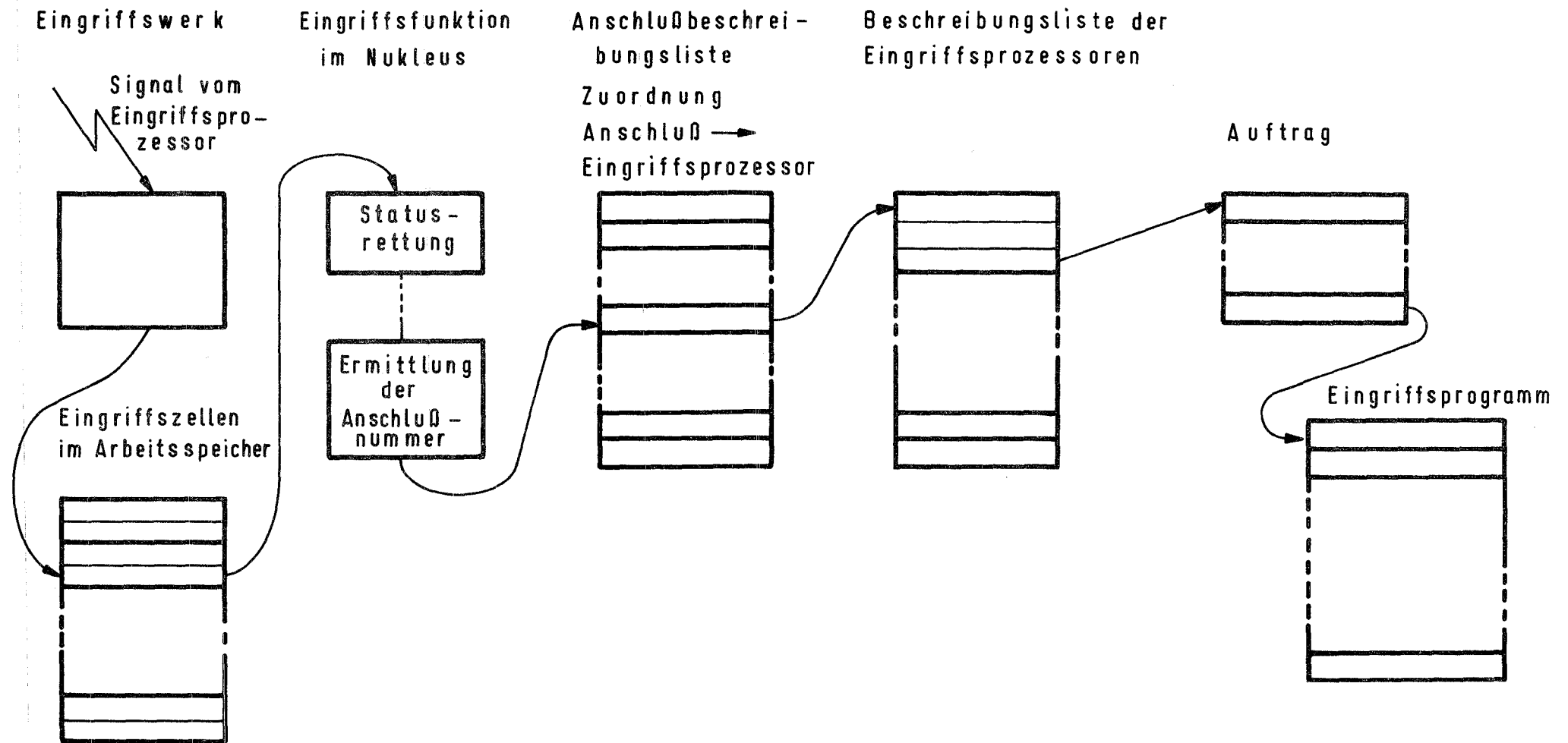


Bild 5 Schnittstelle beim Übergang eines Eingriffsprozesses vom Eingriffsprozessor auf den Standardprozessor

