

Mehrrechner-PEARL für Mehrrechner- und Mehrprozessor-Systeme

von Dr. H. Steusloff, Karlsruhe

Zusammenfassung

Mehrrechner- und Mehrprozessorsysteme setzen sich in allen Bereichen der Systemautomatisierung und damit auch im Wehrbereich zunehmend durch. Dabei ist besonders im Wehrbereich eine breite Palette unterschiedlicher Anwendungsumgebungen zu berücksichtigen: Wehrmaterial wird zukünftig ebenso mit Mehrrechner- oder Mehrprozessorsystemen (embedded systems) automatisiert, wie Leit- und Führungssysteme. Die Nutzung der Eigenschaften von PEARL für die Programmierung auch solcher Systeme läßt eine Erweiterung dieser Sprache zur Beschreibung von Strukturen notwendig erscheinen. Zwei erprobte Ansätze für MEHRRECHNER-PEARL (Dornier, Friedrichshafen und IITB, Karlsruhe) werden vorgestellt und diskutiert.

Schlüsselwörter: Mehrrechnersysteme, Programmiersprachen, verteilte Systeme

Summary

Multicomputer and multiprocessor systems show increasing importance in all areas of system automation, military systems included. Especially in the defence area a large variety of application environments has to be considered: embedded systems will be automated by multicomputer systems as well as guidance and control systems. To utilize the properties of PEARL in programming these systems, an extension of this language is necessary towards the description of structures. Two approved developments of a MULTICOMPUTER-PEARL (Dornier, Friedrichshafen and IITB, Karlsruhe) are presented and discussed.

Keywords: Multicomputer system, programming languages, distributed systems

Zukunft zunehmend eingesetzt werden müssen. Solche Mehrrechnersysteme sind durch höhere Programmiersprachen zu programmieren, um die heute vorliegenden Anforderungen bezüglich Wirtschaftlichkeit und Zuverlässigkeit der Programmsysteme zu erfüllen. Die derzeit hierfür besonders geeignete Echtzeitprogrammiersprache ist PEARL (Process and Experiment Automation Realtime Language [1]). Das Modulkonzept und der Systembeschreibungsteil dieser Sprache erleichtert ihren Einsatz für Mehrrechnersysteme mehr als andere derzeit verfügbare Sprachen.

Es fehlen andererseits auch bei PEARL nach DIN 66253 gewisse Sprachmittel, die den Entwurf und die Programmierung von Mehrrechnersystemen wirkungsvoll unterstützen, wie solche zur Systemstrukturbeschreibung oder zur Beschreibung und Steuerung von Konfiguration und Rekonfiguration der Anwendungsprogramme, auch in Abhängigkeit vom Betriebszustand des Rechnersystems. Ebenso ist die Prozeßkommunikation und die Synchronisation von Programmen in verschiedenen Prozessoren zu beachten.

Bei Ergänzungen von PEARL nach DIN 66253 zu Mehrrechner-PEARL ist zu berücksichtigen, daß die Hardware von Mehrrechnersystemen nach

- der Art der Prozessoren
- dem Kommunikationsprinzip und seiner Realisierung sowie nach
- den Methoden zur Erhöhung der Fehler-toleranz

zu unterscheiden ist. Für die Nutzung dieser je nach Einsatzfall unterschiedlich ausgeprägten Merkmale von Mehrrechnersystemen müssen Sprachmittel zur Verfügung gestellt werden, die dem Anwendungsprogrammierer ein Verbleiben in der höheren Sprache erlauben.

2. EINSATZUMGEBUNGEN FÜR MEHRRECHNER-PEARL

Im militärischen Bereich besteht die Notwendigkeit, Mehrrechnersysteme als Automatisierungskomponenten innerhalb von Gesamtsystemen, z.B. für Avionik und Waffensysteme einzusetzen (embedded systems). Hier-

1. PROBLEMEINFÜHRUNG

Bedingt durch Forderungen nach erhöhter Signalverarbeitungsleistung sowie äußerster Zuverlässigkeit und Verfügbarkeit von rechnergestützten Systemen im Verteidigungsbereich, werden dort Mehrrechnersysteme in

bei ist wesentlich eine

- hohe Speicher- und Laufzeiteffizienz der Programme,
- Verringerung der Fehlermöglichkeiten bei der Programmierung und
- gute Dokumentation sowie wirtschaftliche Erstellung und Wartung der Programme.

Um komplizierte Kommunikationsverfahren zu vermeiden, wird oft eine möglichst lose Kopplung der einzelnen Prozessoren untereinander angestrebt.

Die Struktur des Gesamtsystems und damit auch seines Automatisierungssystems wird während des Betriebes nicht geändert, d.h. die Flexibilitätsanforderungen sind diesbezüglich gering. Notwendige Fehlertoleranzmaßnahmen liegen damit für die Lebensdauer des Systems fest und können sowohl durch statische Redundanz der Hardware (m aus n-Systeme) erreicht werden, als auch vom Entwickler je nach den Systemanforderungen in anderer Weise fixiert werden.

Eine andere Einsatzumgebung von MEHRRECHNER-PEARL ist gegeben bei der Verwendung von Mehrrechnersystemen zur Automatisierung technischer Anlagen, deren Struktur und Eigenschaften während des Betriebes auf einfache Weise an neue Anforderungen angepaßt werden müssen (open-ended design). Dies können z.B. sowohl militärische Command, Control and Communication Systems als auch industrielle Anlagen sein. Die Systemkonfiguration und die Fehlertoleranzmaßnahmen müssen dabei leicht, ggf. on-line und automatisch anpaßbar sein. Eine vorübergehende Leistungsdegradation wird aus Wirtschaftlichkeitsüberlegungen zugelassen und deshalb dynamische Redundanz vorgesehen. Damit ist die Kommunikation zwischen den Prozessoren komplizierter. Speicherplatzeffizienz hat eine geringere Bedeutung; auf die Laufzeiteffizienz wird dennoch hoher Wert gelegt.

Die Struktur und Eigenschaften während des Betriebes auf einfache Weise an neue Anforderungen angepaßt werden müssen (open-ended design). Dies können z.B. sowohl militärische Command, Control and Communication Systems als auch industrielle Anlagen sein. Die Systemkonfiguration und die Fehlertoleranzmaßnahmen müssen dabei leicht, ggf. on-line und automatisch anpaßbar sein. Eine vorübergehende Leistungsdegradation wird aus Wirtschaftlichkeitsüberlegungen zugelassen und deshalb dynamische Redundanz vorgesehen. Damit ist die Kommunikation zwischen den Prozessoren komplizierter. Speicherplatzeffizienz hat eine geringere Bedeutung; auf die Laufzeiteffizienz wird dennoch hoher Wert gelegt.

3. MEHRRECHNER-PEARL-SYSTEME

Aus der geschilderten Situation entstanden auf der Grundlage von PEARL nach DIN 66253 zwei MEHRRECHNER-PEARL-Systeme für die Einsatzerfahrungen vorliegen und die im folgenden charakterisiert seien:

3.1 MEHRRECHNER-PEARL bei Dornier

Das MEHRRECHNER-PEARL-System von Dornier ist in den Dokumenten [2], [3] und [4] beschrieben. Es handelt sich um ein Programmiersystem, das ursprünglich für Avioniksysteme konzipiert wurde und daher die erste, in Abschnitt 2 geschilderte Einsatzumgebung vorfin-

det. Insbesondere ist das zugrundeliegende geräte-technische Kommunikationssystem ein paralleler Bus bei gleichzeitig loser Kopplung der Prozessoren.

Die Erweiterung zu MEHRRECHNER-PEARL besteht in dem Attribut "GLOBAL NET" (Gegenstück: "IN MODULE n") für alle netzglobalen Programmobjekte (Variable, SEMAphore, TASKs u.a.m.). Hierdurch werden Compiler und Binder veranlaßt, diese Objekte in ein spezielles Kommunikationsverfahren einzubetten und im Bedarfsfall die entsprechend geänderten Operationen auf solche netzglobalen Objekte zu generieren. Das PEARL-Ablaufsystem besteht aus einem Prozeßumschalter als Kern (ca. 300 Worte) und Bibliotheksprozeduren, die bei Bedarf vom Binder eingefügt werden, wenn ein PEARL-MODULE gebunden wird. Dadurch wird das Betriebssystem jeweils anwendungsspezifisch generiert. Die Kommunikation wickelt ein als Bibliotheksprozedur verfügbares Treiberprogramm ab.

Das Programmierzugungssystem ist ablauffähig auf den Rechnern PDP-11/70, VAX 750, AEG 80-20 und SIEMENS 7.760. Es erzeugt Code für die Zielsysteme MUDAS, AEG 80-20 und INTEL 8086. Ein Testsystem, lauffähig auf dem Zielsystem MUDAS, erlaubt über die Compilerausgabe (hexadezimale Adressen) das Setzen von Haltepunkten sowie Lesen und Schreiben von Speicherzellen, ebenfalls hexadezimal. Für das Zielsystem AEG 80-20 ist ein in PEARL realisiertes Testsystem auf Quellsprachebene vorgesehen.

3.2 MEHRRECHNER-PEARL bei IITB

Der Sprachumfang des IITB-Compilers [6] erfüllt Basic-PEARL nach DIN 66253 vollständig und bleibt bei den darüber hinausgehenden Sprachelementen innerhalb von Full-PEARL [5]. Diese Sprachelemente sind u.a. TYPE, der Datentyp REF, die Dateibehandlung und vollständige formatierte Standard-Ein-/Ausgabe, der CHARACTER-Datentyp sowie die Feldinitialisierung.

Die Erweiterungen zu MEHRRECHNER-PEARL sind in [7] konzipiert und ihr Realisierungsweg in [8] beschrieben. Es handelt sich um den strukturbeschreibenden STATIONS-Teil, den LOAD-Teil für die Programm-Konfiguration und -Rekonfiguration sowie eine Erweiterung des SYSTEM-Teils für alternative Datenwege und Ersatzwerte; im eigentlichen PEARL nach DIN 66253 wurde nichts geändert.

Aufgrund der hohen Flexibilitätsanforderungen ist jeder Rechner mit einem vollständigen PEARL-Betriebssystem für alle Tasking-Funktionen und die (Uhr-) Zeitverwaltung sowie mit einem Netzwerk-Betriebssy-

stem für die Kommunikation ausgerüstet. Zur Erhöhung der Fehlertoleranz können MODULEs dynamisch verlagert werden. Dies erfolgt durch einen dynamischen Lader, der durch Betriebs- und Fehlerzustände gesteuert wird.

Das Programmerzeugungssystem ist ablauffähig auf SIEMENS 310, 300R30 und 7.760 sowie ab 1982 auf INTEL-Entwicklungssystemen Serie III; es erzeugt Code für SIEMENS 310 und 300R30, das RDC-System des IITB sowie INTEL 8086. Ein in PEARL geschriebenes Testsystem auf Quellsprachebene für die Zielmaschinen (auch Remote-Testing für Teilrechner) ist in Arbeit. Ein Ablaufsimulator auf dem Time-Sharing-System SIEMENS 7.760 ist vorgesehen.

4. REALISIERUNG VON MEHRRECHNER-PEARL

Im folgenden sind die in Abschnitt 3 erwähnten Realisierungen von MEHRRECHNER-PEARL genauer beschrieben. Diese Realisierungen bauen grundsätzlich auf einigen sie besonders begünstigenden PEARL-Eigenschaften auf:

- Strukturierung von PEARL-Modulen in inhaltlich unterschiedliche Teile: SYSTEM-Teil und PROBLEM-Teil
- Kommunikation zwischen PEARL-Modulen über GLOBALE Objekte
- Explizite Sprachmittel zur Reaktion auf Fehlerzustände: ON-Reaktion.

Damit ist PEARL vom Entwurf her bereits "MEHRRECHNER-fähig".

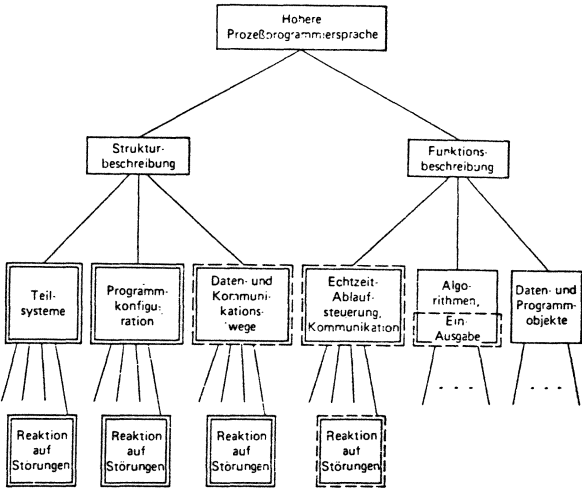


Bild 1: Grundstruktur einer Prozeßprogrammiersprache für verteilte Automatisierungssysteme.

4.1 Grundstruktur einer Prozeßprogrammiersprache für verteilte Systeme

Um die in den folgenden Abschnitten beschriebenen Erweiterungen von PEARL für den Einsatz in MEHRRECHNER-Systemen einordnen zu können, ist in Bild 1 die Grundstruktur einer Prozeßprogrammiersprache für verteilte Systeme dargestellt. Es fällt auf, daß neben den in üblichen Programmiersprachen bereits vorhandenen Sprachelementen für die Beschreibung von Funktionen ein weiterer Sprachenteil für die Beschreibung von Strukturen vorgesehen ist. Die doppelt umrahmten Funktionen sind - besonders im Hinblick auf den Einsatz für die Programmierung von Automatisierungssystemen - nur bei wenigen modernen Programmiersprachen, zum Teil nur bei PEARL verfügbar; diese Funktionen sind daher gestrichelt eingerahmt. Die mit durchgezogenen Strichen doppelt eingerahmten Funktionen sind heute in keiner Programmiersprache vorhanden; sie sind daher in MEHRRECHNER-PEARL zu ergänzen.

4.2 Die MEHRRECHNER-PEARL-Realisierung von DORNIER

Die in Literatur [2], [3] und [4] beschriebene Realisierung des MEHRRECHNER-PEARL-Systems der Firma DORNIER bezieht sich ausschließlich auf die Kommunikation in einem verteilten Mehrprozessor-System. Für diese Kommunikation kann der Programmierer sogenannte netzglobale Größen definieren (Bild 2), indem er hinter das GLOBAL-Attribut das Zusatz-Attribut NET fügt. Der Compiler wird dadurch veranlaßt, den Aufruf einer speziellen Netz-Kommunikationsprozedur abzusetzen. Diese Kommunikationsprozedur teilt Veränderungen netzglobaler Größen im gesamten Netz mit. Eine Ausnahme bilden TASKs; sie werden ebenfalls mit dem Attribut GLOBAL NET deklariert, bei ihrer Spezifikation wird jedoch explizit angegeben, in welchem Modul die TASK deklariert wurde.

```
MEHRRECHNER - PEARL für "EMBEDDED SYSTEMS" (DORNIER)
-----
Kommunikation über netzglobale Größen :
■ Allgemeine Objekte
  DCL F (10) FIXED (15) GLOBAL NET ;
  SPC VALUE FLOAT (31) GLOBAL NET ;

■ SEMAphore
  DCL A SEMA GLOBAL NET ;
  SPC A SEMA GLOBAL NET ;

■ TASKs
  T : TASK PRIO 3 GLOBAL NET ;
  SPC T TASK IN MODULE 11 ;

Keine Initialisierung von netzglobalen Größen !
Keine netzglobalen Prozedurparameter mit IDENT
J e d e Wertänderung netzglob.Obj. systemweit
```

Bild 2: MEHRRECHNER-PEARL-Ansatz der Firma Dornier.

Da das MEHRRECHNER-PEARL-System von DORNIER eine lose Kopplung der Prozesse voraussetzt, ist die Verwendung netzglobaler Größen als Prozedurparameter mit dem Attribut IDENT nicht vorgesehen. Auch können netzglo- bale Größen nicht initialisiert werden, da sie in mehreren Kopien im Kommunikationssystem existieren und daher bei Systemanlauf eine dynamische Initiali- sierung vorgesehen werden müßte.

Die in Bild 2 gezeigten Sprachmittel für die Kommuni- kation über netzglobale Größen erweitern den PEARL- Standard nach DIN 66253.

4.3 MEHRRECHNER-PEARL-Realisierung des IITB

Die MEHRRECHNER-PEARL-Realisierung des IITB berück- sichtigt alle in Bild 1 doppelt umrahmten Komponen- ten einer Prozeßprogrammiersprache für verteilte Sys- teme. Die Spracherweiterungen sind so gestaltet, daß PEARL nach DIN 66253 nicht verändert wird. Die Erweiterungen betreffen insbesondere die Teilsystem- Beschreibung in einem eigenständigen Sprachenteil, die Beschreibung der Programmkonfiguration und -Re- konfiguration in einem zusätzlichen sogenannten La- de-Teil innerhalb von PEARL-Modulen sowie eine Er- weiterung des PEARL-SYSTEM-Teils um die dynamische Zuordnung von Prozeßdaten-Ersatzwegen und -Ersatz- werten bei Störungen. Dieser Ladeteil sowie die Er- weiterung des PEARL-SYSTEM-Teils sei im folgenden kurz beschrieben.

4.3.1 Programmkonfiguration und -Rekonfiguration

Sprachelemente zur Beschreibung einer Programmkonfi- guration und -Rekonfiguration in Abhängigkeit vom Betriebszustand des Automatisierungssystems seien Ladeanweisungen (Bild 3) genannt. Sie bestehen aus der Angabe eines Teilsystems, aus einer sogenannten Ladepriorität, aus einer Betriebszustandsbedingung und aus optionalen Zusätzen. Die Ladepriorität gibt die Wichtigkeit eines Programmmoduls relativ zu ande- ren Moduln innerhalb eines Teilsystems an (Funktions- priorität), sie ist zu unterscheiden von der Ablauf- priorität einzelner Teilprogramme. Der Betriebszu- stand INITIAL bezeichnet den Urlade- und Normalzu- stand, während die in Klammern eingeschlossenen lo- gischen Ausdrücke nicht normale Betriebszustände mit Hilfe von Statuszeichnern aus der Teilsystembeschrei- bung beschreiben. Die optionale Angabe einer Start- nummer für den Initialzustand beschreibt die Urstart- Reihenfolge eines Programmsystems. Das optionale At- tribut RES[IDENT] bewirkt ein Urladen von Program- men in Ersatz-Teilsysteme, um dort auch bei Ausfall der Programmladeeinrichtung verfügbar zu sein oder bei zeitkritischen Programm-Rekonfigurationsfällen

den Ausfall von Verarbeitungsfunktionen zu verkürzen.

```
MODULE (MOD1);
LOAD;
TO STA1 LDPRIO 5 INITIAL STARTNO 1;
TO STA3 LDPRIO 5 ON (STA1PR AND NOT STA3PR) RES;
TO STA2 LDPRIO 5 ON (STA1PR AND STA3PR);
:
SYSTEM;
:      /* DESCRIPTION OF DATAWAYS, ADAPTATION OF
:      DATA ITEMS, REPLACEMENT DATA          */
:
PROBLEM;
:      /* ALGORITHMS, PROGRAM FLOW CONTROL, I/O */
:
MODEND;
```

Bild 3: MEHRRECHNER-PEARL-Ansatz des IITB: Modul- Ladeteil.

4.3.2 Prozeßdaten-Ersatzwege und -Ersatzwerte

Die Sprachelemente zur Beschreibung von Prozeßdaten- Ersatzwegen und -Ersatzwerten bei Störungen sind in einer Datenwegbeschreibung (PEARL-SYSTEM-Teil) ange- ordnet, um die Beschreibung von Datentransport und Datengewinnung von den Verarbeitungsalgorithmen zu trennen (Bild 4).

```
SYSTEM;
:
:
TEMP09 : -> ANIN560 * 5
/*      -> ANIN571 * 5,
CORR: TEMP09=TEMP09/3-16
-> ANIN572 * 5,
CORR : CALL ADJUST (TEMP09,2)
-> REP : TEMP09 = 1300,
PLAUS: (HI=1500,LO=1000,DELTA=10)*/;
HEAT09 : (- ANOUT560 * 15 ..... ;
:
:
```

Bild 4: MEHRRECHNER-PEARL-Ansatz des IITB: Prozeß- daten-Ersatzwege und -Ersatzwerte.

Der Name TEMP einer Prozeßdatenquelle möge eine ein- zulesende Temperatur bezeichnen, deren Meßfühler standardmäßig mit dem Gerät ANIN560, Kanal 8, einem Analogeingang, verbunden sei. Alternative Verbin- dungen sollen bestehen zu den Geräten ANIN571 und ANIN572, gegebenenfalls auch von zusätzlichen (re- dundanten) Meßfühlern her. Bei diesen Alternativen können Korrekturalgorithmen zur Anpassung der einge- lesenen Werte an unterschiedliche Geräte- und Daten- wegeigenschaften angegeben werden. Die letzte Alter- native bezeichnet einen Ersatzwert, in derselben Zei- le stehen auch die Prüfkriterien für Plausibilitäts-

prüfungen der gewonnenen Meßwerte. Die Bearbeitung einer solchen Datenwegbeschreibung erfolgt in der Reihenfolge ihrer Notation, abhängig vom Betriebszustand der Datenwege und dem Ergebnis der Plausibilitätskontrolle. Sind alle Datenwege gestört oder liefern sie nicht plausible Werte, so gilt der Ersatzwert als Meßgröße. Damit steht für die mit Prozeß-Ein/Ausgabedaten korrespondierenden Variablen der algorithmischen Programmteile (PEARL-PROBLEM-Teil) immer ein gültiger Wert zur Verfügung, ohne die Verarbeitungsalgorithmen mit Reaktionen auf Störungen zu belasten.

5. ANFORDERUNGEN AN MEHRRECHNER-PEARL IM VERTEIDIGUNGSBEREICH

Die vorstehend beschriebenen Realisierungen von MEHRRECHNER-PEARL sind erprobte und eingesetzte Entwicklungen für die in Abschnitt 2 dargestellten Einsatzumgebungen. Da diese Einsatzumgebungen jeweils Extreme darstellen, ist

ein funktional abgestufter Satz von Ergänzungen zu PEARL für den Einsatz in MEHRRECHNER-Systemen unterschiedlichen Typs

anzustreben. Ein solcher Satz von Ergänzungen, aufgebaut auf die Ergebnisse und Erfahrungen mit den bisher vorhandenen und noch durchzuführenden Implementationen und Anwendungen, sollte durch die Normung von MEHRRECHNER-PEARL in seiner Einsatzfähigkeit unterstützt werden.

Ebenso wichtig ist aber auch die Unterstützung durch angepaßte Programmierumgebungen. Hier sind vordringlich Testsysteme auf Hochsprachenebene unter Berücksichtigung der speziellen Probleme von MEHRRECHNER-Systemen erforderlich. Insbesondere ist das Testen von Programmen in verteilten Prozessorstationen von einer zentralen Teststation aus zu ermöglichen. Solche Testsysteme, die auch in der Betriebsphase von MEHRRECHNER-PEARL-Programmsystemen für Wartungszwecke von hoher Wichtigkeit sind, nutzen das Kommunikationsmedium des verteilten Rechnersystems; sie müs-

sen darüberhinaus Zugriff zu den lokalen Betriebssystemen der verteilten Mikroprozessorstationen haben. Damit liegt es nahe, neben den üblichen Programmtestfunktionen auch Statusinformationen des verteilten Rechnersystems am Testplatz verfügbar zu machen und damit die Wartung zu unterstützen.

6. Literatur

- [1] DIN 66253: Programmiersprache PEARL
Deutsches Institut für Normung (DIN), DIN 66253, Teil 1 (1979) und Teil 2 (1980).
- [2] Lang, W.: Sprachbeschreibung DS-PEARL Subset.
Dornier System GmbH, Friedrichshafen, Nov. 1979.
- [3] Blank, J. et. al.: Bericht zum Projekt "MULTI-COMPUTER PEARL". Dornier System GmbH, Friedrichshafen, November 1979.
- [4] Graf, H.; Lewandowski, Fr.: Bericht zum Projekt "Testsystem AEG 80/20 MIL". Dornier System GmbH, Friedrichshafen, 1982.
- [5] Kappatsch, A. et. al.: Full PEARL Language Description. Kernforschungszentrum Karlsruhe (FRG). Report KFK PDV 130 (1977) und DIN 66253: Programmiersprache PEARL, Full PEARL. Entwurf zum Teil 2 von DIN 66253, NI-5.8/4-80, NI-5.8.1/1.80, Deutsches Institut für Normung (DIN).
- [6] Werum, W.; Windauer, A.: PEARL, Beschreibung mit Anwendungsbeispielen. Vieweg (1978).
- [7] Steusloff, H. U.: Zur Programmierung von räumlich verteilten, dezentralen Prozeßrechnersystemen. Dissertation. Fakultät für Informatik der Universität (TH) Karlsruhe (1977).
- [8] Heger, D.; Steusloff, H.U.; Syrbe, M.: Echtzeit-rechnersystem mit verteilten Mikroprozessoren. Bundesministerium für Forschung und Technologie, Forschungsbericht DV 79-01 Datenverarbeitung (1979).