

**VP2000 – Gigaflops im  
7.500-Bereich**

**STARG-Tagung**

**Konstanz**

**6. März 1986**

Joachim Backes  
Regionales Hochschulrechenzentrum der  
Universität Kaiserslautern

Erwin-Schrödinger-Straße  
D-6750 Kaiserslautern

# Inhaltsverzeichnis

1. Einleitung: Zielsetzung des Projektes VP2000.....	3
2. Benutzerschnittstellen.....	6
3. Die Verbundkomponente.....	8
3.1 Möglichkeiten der Kopplung von MSP und VSP.....	8
3.2 Funktionen von JES/MAS im MSP.....	8
3.3 Funktionen von JES/MAS im BS2000.....	9
3.4 Dateitransfer zwischen BS2000 und VSP.....	10
3.5 Grunddienste der Kopplung BS2000 $\Leftrightarrow$ VSP.....	11
4. Die Portierung des FORTRAN77-Systems.....	13
4.1 Komponenten des FORTRAN77-Systems im MSP.....	13
4.2 Komponenten des FORTRAN77-Systems im BS2000.....	13
4.3 Vorgehensweise bei der Umstellung.....	15
4.3.1 Technik.....	15
4.3.2 Aufwand.....	16
4.3.3 Reihenfolge.....	17

## 1. Einleitung: Zielsetzung des Projektes VP2000

Die Vektorprozessoren-Familie VP (VP50/VP100/VP200/VP400) mit ihrem Betriebssystem VSP kann derzeit nur in Verbindung mit einem oder mehreren Rechnern des Siemens-Systems 7.8xx unter MSP oder anderen IBM- bzw. IBM-kompatiblen Systemen unter MVS benutzt werden. Ziel dieses Projektes ist es, anstelle dieser Rechner eine oder mehrere Anlagen des Siemens-Systems 7.5xx unter BS2000 verwenden zu können und damit dem BS2000-Benutzer die Leistungen eines Vektorprozessors zur Verfügung zu stellen.

Das Betriebssystem des Vektorprozessors (VSP) unterscheidet sich vom Betriebssystem MSP in zwei grundlegenden Punkten. Es enthält einerseits Erweiterungen gegenüber dem MSP sowohl zur Unterstützung der Vektorprozessor-Hardware und des VP-Arbeitsspeichers wie auch zur Auswahl des Prozessors je Job-Step (VJCS). Andererseits wurde es gegenüber dem MSP auf jene Funktionen reduziert, die notwendig sind, um VP-Jobs bei optimaler Ausnutzung der Rechner-Ressourcen ausführen zu können. Daher sind Funktionen, die zur Durchführung eines interaktiven Betriebs notwendig sind, nicht vorhanden.

Eine effektive Nutzung des VSP ist nur dann möglich, wenn ausschließlich Programme zum Ablauf gebracht werden, die die Spezialeinrichtung Vektor-Unit auch wirklich verwenden. Daher werden die Programmerstellung, Tests, Verwaltung, Kommunikation und Aufbereitung der Ergebnisse auf einem dem VP vorgeschalteten Front-End-Prozessor (FEP) mit System MSP durchgeführt. Für die Kopplung beider Systeme über gemeinsame Platte und/oder Kanalkopplung stehen die derzeit im System 7.8xx üblichen Konzepte des lokalen Verbunds (JES/MAS, JES/E und NJE) zur Verfügung.

Man hat damit ein Gesamtsystem-Modell "Aufbereitungsmaschine – Produktionsmaschine" realisiert in einem bislang homogenen Verbund zweier Systeme, der eine Arbeitsteilung teils erzwingt, teils dem Benutzer nahelegt.

Für die Integration des Vektorprozessors in eine BS2000-Konfiguration kommen zwei Möglichkeiten in Betracht: Die Hereinnahme der reinen VP-Hardware oder die Verwendung dieser Hardware unter ihrem Betriebssystem VSP. Die erste Alternative scheidet deswegen aus, weil eine Erweiterung des BS2000 zum Betrieb eines VP auf der Basis einer XA-fähigen BS2000-Version erfolgen müsste. So bietet sich die Lösung an, einen Verbund zwischen BS2000 und VSP zu schaffen, in dem das BS2000 die Rolle der Aufbereitungsmaschine für das Vektorprozessor-System übernimmt.

Einen wesentlichen Unterschied zum Modell bei Fujitsu darf man jedoch nicht übersehen: Bei Fujitsu ist der Verbund homogen, beim BS2000 dagegen heterogen. Dies bedeutet insbesondere, dass für die VSP-Auftragsschnittstelle, für die Interpretation von VSP-Meldungen und für den Zugriff auf Daten der VP-Platten nicht die originären Sprachmittel des BS2000 funktionieren. Im Unterschied zum MSP-Benutzer muss der den VP verwendende BS2000-Anwender mit zusätzlichen Schnittstellen eines Fremdsystems konfrontiert werden.

Aus dem bisher gesagtem schälen sich zwei beinahe unabhängige Aspekte für dieses Projekt heraus, die gleichzeitig auch dessen Schwerpunkte bilden:

- Die Kopplung des Vektorprozessors VP unter VSP an einen Rechner des Siemens-Systems 7.5xx unter BS2000 zu einem heterogenen Verbund (Realisierung durch das RHRK) und
- die Portierung des gesamten FORTRAN77-Systems vom MSP in das BS2000 (Compiler, Utilities für Test und Tuning) einschließlich des vektorisierenden FORTRAN77-Compilers, des FORTRAN-Laufzeitesystemes sowie der FORTRAN-Unterprogramm-Bibliothek SSLII/VP, einer Anpassung der MSP-Bibliothek SSLII im Hinblick auf die optimale Nutzung der Vektorleistung (das FORTRAN77-System stellt im FEP das einzige Interface zur Erlangung von Vektorleistung dar); Realisierung durch Siemens AG.

Die Implementierung der Schnittstellen erfolgt unter Beachtung folgender Paradigmen:

- Das BS2000 muss sich dem VSP gegenüber wie ein MSP verhalten, damit keine Eingriffe in das VSP notwendig werden. Dadurch werden zusätzliche Komponenten im BS2000 für die Bedienung der Kopplungs-Hardware und zur Unterstützung von Jobtransfer und Dateizugriff notwendig.
- Für den Standardfall der VP-Verwendung (vektorisiertes Programm zum Produktionslauf an das VSP übergeben und auf Ergebnisse warten) soll der Benutzer möglichst ohne zusätzliches VSP-Verständnis wie im BS2000 verfahren können.
- Für den versierten Anwender, der sich in Einzelheiten dieses speziellen Rechnerverbundes einarbeitet, sollen die VSP-Leistungen möglichst uneingeschränkt zur Verfügung stehen.
- Alle neuen Benutzerschnittstellen sind Schnittstellen eines künftigen BS2000-Produktes und mit der BS2000-Kommandosprache abgestimmt, d.h. nach SDF-Regeln gebildet, mit SDF-A erstellt und der Siemens-internalen Qualitätskontrolle unterworfen.
- Die BS2000-VP-Benutzerschnittstelle soll so angelegt werden, dass außer FORTRAN auch andere Compilersysteme (z.B. PASCAL) einbettbar sind.
- Die neuen Funktionen im BS2000 sollen ausschließlich als nichtprivilegierte Software ablaufen.
- Die vom Compiler generierten Objektmoduln sollen im BS2000 in PLAM-Bibliotheken abgelegt werden.
- Das VP-System ist gegenüber dem BS2000 als "wertvolles" Betriebsmittel anzusehen. Deshalb werden möglichst viele Funktionen in das BS2000 verlagert.
- Zugriffsrechte zum VSP werden auch im BS2000 verwaltet.
- Die BS2000-seitige Administration des VSP wird so gestaltet, dass ein minimaler Entwicklungsaufwand erforderlich ist. Administrationsprogramme wie z.B. Rechenzeitstatistik laufen nach Möglichkeit im VSP direkt ab. Insbesondere muss das VSP immer ausreichend konfiguriert sein (z.B. mit Bandmaschinen), damit u.a. auch die Datensicherung unter VSP mit VSP-Software durchgeführt werden kann.
- Für die Wartung liegen zwei Systeme mit unterschiedlichen Wartungsschnittstellen vor. Es ist aber zu vermuten, dass in dem originären VP/VSP-Wartungskonzept die Existenz einer MSP-Kopplung für bestimmte Wartungsvorgänge vorausgesetzt

wird. Die technischen und organisatorischen Belange der VP/VSP-Wartbarkeit unter den hier geänderten Konfigurationsbedingungen bedürfen einer Klärung.

- Manuale zur Benutzung des Vektorprozessors mit dem BS2000 werden sowohl für Zielgruppe Programmierer als auch für die Zielgruppe Rechenzentrum/Administrator/ Operateur erstellt. Verweise aus diesen Handbüchern auf Sekundärliteratur der MSP-oder MVS-Serie sind unvermeidbar.

## 2. Benutzerschnittstellen

Zu den Projektzielen gehört es, dem BS2000-Benutzer das Betriebsmittel VP möglichst analog zu seiner gewohnten Umgebung zur Verfügung zu stellen. Grundvoraussetzung hierfür ist es, dass alle Leistungen auf Kommandoebene verfügbar sind. Die Definition dieser Kommandos erfolgt durchgängig auf der Basis der neuen Kommandosprache SDF, so dass Erklärungen und Hilfen dort sofort in komfortabler Weise integriert sind. Um nicht unnötige Beschränkungen entstehen zu lassen, die bei der Abbildung einer Steuersprache auf eine andere fast zwangsläufig entstehen, wurde darauf geachtet, dass auch für den erfahrenen VSP-Anwender der komplette Umfang der dortigen Steuersprache vom BS2000 aus verfügbar ist. Dies schließt jedoch nicht die Dialogschnittstelle des MSP ein, da das VSP seinerseits nur im Batch arbeitet.

Im einzelnen sind folgende Kommandos verfügbar:

COMPILE-FORTRAN-SOURCE	Für die Übersetzung eines FORTRAN-Programmes mit dem FORTRAN77- bzw. FORTRAN-77/VP-Compiler.
LINK-VECTOR-PROGRAM	Für das Binden eines VSP-Programmes.
ENTER-VECTOR-PROGRAM CALL-VECTOR-PROGRAM	Für die Ausführung eines fertig gebundenen Programms im VSP. Dabei bedeutet CALL, dass auf die Beendigung der Programmausführung gewartet wird, während bei ENTER nach der Kreation des VSP-Auftrages sofort weitergearbeitet werden kann.
ASSIGN-VECTOR-SYSDTA	Beschreibt die Eingabedaten von VSP-Programmen.
ASSIGN-VECTOR-SYSLST	Beschreibt die Ausgabedaten von VSP-Programmen einschließlich Druckerparametern, Zielrechner, Zeichensatz usf.
MODIFY-VECTOR-PROTECTION	Ändert das Passwort für den VSP-Zugang.
OPTIMIZE-VECTOR-FORTRAN-SOURCE	Ermöglicht das Optimieren von Programmen mittels FORTUNE und VECTUNE.
SHOW-VECTOR-JOB-STATUS	Liefert aktuelle Information über den Bearbeitungsstand eines VSP-Auftrages.
CANCEL-VECTOR-JOB	Erlaubt vom BS2000 aus den Abbruch eines VSP-Auftrages.
ENTER-VECTOR-JOB CALL-VECTOR-JOB	Ermöglichen die Ausführung von VSP-Aufträgen, die voll in der VSP-Steuersprache formuliert sind. Zum Unterschied von CALL und ENTER siehe oben.

SET-VECTOR-FILE-LINK	Beschreibt den Zusammenhang zwischen FORTRAN-Ein/Ausgabe-Einheit und VSP-Dateiname.
CREATE-VECTOR-FILE	Generiert die erforderlichen AMS-Kommandos für eine VSP-Datei und veranlasst ggf. deren Ausführung im VSP. Dabei steht AMS für die Utility KQCAMS, die im VSP Dateien in VSAM-Katalogen bearbeitet.
CONVERT-VECTOR-FILE	Generiert die erforderlichen AMS-Kommandos für das Kopieren einer normalen VSP-Datei in eine SAM-Datei und veranlasst ggf. deren Ausführung.
DELETE-VECTOR-FILE	Generiert die erforderlichen AMS-Kommandos für das Löschen einer VSP-Datei und veranlasst ggf. deren Ausführung.
LIST-VECTOR-CATALOG	Generiert die erforderlichen AMS-Kommandos für Katalogauszüge von VSP-Dateien und veranlasst ggf. deren Ausführung.
PRINT-VECTOR-FILE	Generiert die erforderlichen AMS-Kommandos für das Ausdrucken von VSP-Dateien und veranlasst ggf. deren Ausführung.
VERIFY-VECTOR-FILE	Generiert die erforderlichen AMS-Kommandos für die Überprüfung von VSP-Dateien und veranlasst ggf. deren Ausführung.
TRANSFER-VECTOR-FILE	Kopiert eine komplette BS2000-Datei in einen kompletten VSP-Data Set oder umgekehrt.

Der Benutzer soll in der Regel nicht merken, ob seine Aktivitäten den Start eines separaten Auftrags im VSP auslösen. Durch die Konstruktion der Kommandos ist es möglich, Funktionen zwischen den Rechnern entsprechend dem Implementierungsfortschritt zu verlagern, ohne dass dies an der Benutzeroberfläche zu Änderung der Bedienung führt, höchstens zu geänderten Meldungen.

### 3. Die Verbundkomponente

#### 3.1 Möglichkeiten der Kopplung von MSP und VSP

Die drei standardmäßig angebotenen Softwareprodukte für das Zusammenspiel des VP-Systems mit dem Front-End-Rechner sind

- NJE über CTC,
- JES/MAS,
- JES/E,

von denen je nach Einsatzfall eines auszuwählen ist. Es war von vornherein klar, dass aus Aufwandsgründen nicht alle drei Produkte für die Kopplung zum BS2000 angeboten werden können. Die Hauptgründe für die Wahl von JES/MAS (dem Äquivalent zu JES2 beim IBM-System MVS) und den Ausschluss der beiden anderen Möglichkeiten sind:

- JES/E hätte nur ein einziges BS2000 als Eingabesystem zum VP unterstützt.
- JES/E und NJE bedingen durch den CTC einen höheren Hardwareaufwand als JES/MAS.
- NJE belastet das VP-System dadurch, dass beim Datenaustausch alle Information durch den Rechner transportiert wird.
- JES/E hat einen deutlich höheren Implementierungsaufwand als JES/MAS.

Selbstverständlich haben die beiden ausgeschlossenen Varianten auch Vorteile, die aber nicht so stark sind, dass sie die Nachteile aufgewogen hätten.

#### 3.2 Funktionen von JES/MAS im MSP

Bei JES/MAS arbeiten bis zu 7 Rechner, die sogenannten Members, mit eigenen Betriebssystemen gemeinsam auf einem einzigen Spool. Der Gesamtkomplex der Members ist ein Knoten im Sinne von NJE. Die Leistungen von JES/MAS werden durch einzelne Prozessoren abgehandelt, so

- Auftragseingabe,
- Konvertierung der Steuersprache,
- Auftragsplanung,
- Ausführungskontrolle,
- Ausgabeplanung,
- Ergebnisausgabe,
- Auftragslöschung.

Die Kommunikation zwischen den einzelnen JES-Prozessoren erfolgt über gemeinsame Listen, der sogenannten Jobqueue, mit ihren Unterlisten. Zur Bearbeitung befindet sich die Jobqueue im Arbeitsspeicher; sie wird anschließend den anderen Members des Spoolsys-

tems durch Transport auf eine der gemeinsamen Platten (in die "Checkpointdatei") zur Verfügung gestellt.

In einem MSP werden üblicherweise sämtliche Komponenten von JES/MAS zum Einsatz kommen. Im Gegensatz hierzu werden auf einem VSP, das an einem solchen Spoolknoten beteiligt ist, nicht alle Funktionen zum Einsatz kommen. Da der VP ein wertvolles Betriebsmittel ist, das möglichst wenig im Skalarmodus arbeiten soll, wird man versuchen, möglichst viele der JES-Funktionen auf das als Frontend vorhandene MSP zu verlagern. Dies ist sogar zwingend für diejenigen Teile, die bestimmte Geräte bedienen, da der VP keinen Bytemultiplexkanal kennt und daher gar nicht in der Lage ist, Drucker, Leser usw. zu betreiben. In der Hauptsache werden am VSP die Prozessoren Auftragsplanung, Ausführungskontrolle und Ausgabeplanung ablaufen.

### 3.3 Funktionen von JES/MAS im BS2000

Das Ziel der Übertragung eines Teils der JES-Funktionen in das BS2000 ist es, das MSP als zwingend erforderliche Komponente eines VP-Systems zur Kommunikation mit der Umwelt entbehrlich zu machen. Dies bedeutet, dass aus Aufwandsgründen in einer ersten Stufe möglichst viele Komponenten in das VSP verlagert werden. Es entfällt auch die Möglichkeit eines gekoppelten MSP/VSP-Systems, zwischen einzelnen Jobsteps die Zentraleinheit zu wechseln, wie es in der quasi-homogenen MSP/VSP-Umgebung möglich ist. Diese Erweiterung, die auch bei Einsatz von MVS als Frontend-Betriebssystem nicht möglich ist, wurde ein völliges Redesign des BS2000 erfordern.

Funktionen, die nicht auf das VSP übertragbar sind, betreffen insbesondere die Ein- und Ausgabe von Aufträgen. Da der Vektorprozessor grundsätzlich keinen Bytemultiplexkanal bedient und auch keinen Dialogbetrieb unterstützt, müssen fast alle Aufträge im Frontendsystem eingegeben werden. Eine Ausnahme bilden diejenigen Aufträge, die aus anderen VSP-Jobs heraus über den Internal Reader initialisiert werden.

Da die restlichen Funktionen der Auftragsein- und -ausgabe vom BS2000 erbracht werden müssen, sind dort die nachfolgend beschriebenen Komponenten zu realisieren, da gemäß Projektvorgabe das VSP nicht geändert werden soll:

- Der Checkpointprozessor realisiert Zugriffe, Koordination und Verwaltung der für alle Members eines JES/MAS-Systems gemeinsamen Checkpointdatei auf einer hierfür exklusiv genutzten "Shared Disk". Hierbei hat das BS2000 nicht nur das Format auf der Platte genau einzuhalten, sondern auch den Zugriffs- und Reservierungsalgorithmus ebenso wie die Bearbeitungsstrategien der einzelnen Warteschlangen und das Restartverhalten einschließlich der Konsistenzprüfungen.
- Der Internal Reader realisiert benutzerseitig eine dem ENTER-JOB-Kommando entsprechende Leistung in Richtung VSP. Er hat nicht nur die Eingabewerte und VSP-Kommandos zu lesen und entsprechend formatiert auf der Spoolplatte abzuliegen, sondern auch die entsprechenden Einträge in der Jobqueue vorzunehmen. Außerdem sind JES/MAS-Anweisungen zu analysieren und zu verarbeiten.
- Der Outputwriter des BS2000 betreibt im Gegensatz zu seinem Äquivalent im MSP die Geräte nicht selbst, da keine entsprechende Schnittstelle zum Spool des BS2000 besteht. Vielmehr werden die Ergebnislisten im Pub-Space des Benutzers

abgelegt, und in Abhängigkeit von der Ausgabe Klasse wird eventuell noch ein Auftrag mit entsprechendem PRINT-Kommando gestartet. Der Outputwriter muss hierzu das Datenformat des JES-Spool-Systems bearbeiten können und auch die Joboutputelemente aus der Jobqueue interpretieren.

- Ein Kommandoprozessor muss die vom Konsoloperateur bzw. Benutzer kommenden JES-Kommandos analysieren, auswerten oder ggf. weiterleiten. Hier wird wegen des begrenzten Umfangs der JES-Komponenten im BS2000 sicherlich nicht das ganze Spektrum der MSP-Möglichkeiten zu realisieren sein. Entsprechende Abprüfungen mit modifizierten Meldungstexten sind vorgesehen.

Die beiden Prozessoren *Internal-Reader* und *Outputwriter* werden zur Leistungssteigerung als Subtasks in mehreren Ausprägungen parallel ablaufen können.

Ebenso ist Vorsorge getroffen, dass sowohl mehrere BS2000 als auch MSP und/oder VSP auf einem JES/MAS-Spool arbeiten können.

Es erscheint auch möglich, mit geringem Aufwand eine JES2-Kopplung zu MVS-Systemen zu realisieren. Genau wie bei der Zusammenarbeit mit MSP/VSP ist hier ein besonderes Augenmerk auf die Abstimmung der Systemversionen zu richten.

### 3.4 Dateitransfer zwischen BS2000 und VSP

Während die Betriebssysteme MSP und VSP physikalische Sätze auf den Plattenspeichern mit variabler Länge im CKD (Count-Key-Data)-Format schreiben und sich damit sehr ähnlich wie das Betriebssystem MVS der Fa. IBM verhalten, schreibt das BS2000 grundsätzlich physikalische Sätze mit der konstanten Länge von 2 KBytes im FBA (Fixed Block Architecture)-Format. Bereits hieraus ergibt sich, dass die Plattenspeicher - ganz unabhängig von Dateistruktur, Interncodes etc. - zwischen diesen beiden Welten total inkompatibel sind, obwohl auch in der IBM-Welt das FBA-Format bekannt ist und benutzt wird.

Aus diesem Grund ist es erforderlich, zwischen den Dateiwelten des BS2000 einerseits und des MSP/VSP andererseits eine Übergangsmöglichkeit zu schaffen. Hierbei ist ab der Systemversion V8.5 des BS2000 nicht mehr die Hardware das zentrale Problem, denn durch die Einführung des /370- Kanals in die BS2000-Welt sind hardwaremäßig gemeinsame Plattenspeicher in den beiden Systembereichen anschließbar. Für die darauf aufbauende Software sind sehr unterschiedliche Lösungsansätze denkbar:

1. Implementierung der MSP/VSP-Zugriffsmethoden im BS2000
2. Direkte Kopie von/zu BSZDDD-Dateien auf/von VSP-Platten
3. Benutzung der JES/MAS-Kopplung zum Filetransfer innerhalb des Auftragsverbunds

Die Methode (1) ist mit weitem Abstand mit dem größten Implementierungsaufwand verbunden und mit vertretbarem Aufwand und Terminvorgaben nicht realisierbar. Das genaue Gegenteil lässt sich von Ansatz (3) sagen. Hier könnten Teile des Produkts FTU (File Transfer Unit) verwendet werden. Für größere Dateien wird aber der dynamische Transportaufwand so groß, dass das teure Betriebsmittel VP über Gebühr in Anspruch genommen wird. Auch die Belastung der Plattenkanäle durch mehrfachen Transport der Information wird

die Systemleistung deutlich beeinflussen. Als Vorablösung soll diese Variante übergangsweise am Regionalen Hochschulrechenzentrum Kaiserslautern einen frühen Betrieb des VP am BS2000 ermöglichen.

Unter dem Titel COPY2000 soll im BS2000 als Realisierung der Möglichkeit (2) ein Dienst angeboten werden, der es erlaubt, DVS-Sätze des BS2000 zu lesen, in die entsprechende Form von CKD-Records des VSP-DMS zu bringen und diese dann direkt auf gemeinsam zugreifbare Platten zu schreiben, deren Eigentümer das VSP ist. Das gleiche gilt sinngemäß für die Gegenrichtung. Für einen effizienten Ablauf und vertretbaren Implementierungs- sowie Lernaufwand des Benutzers sind einige Einschränkungen gegenüber allen denkbaren Optionen zu machen:

- Es werden nur ganze Dateien transferiert;
- im Ziel bereits vorhandene Dateien werden überschrieben;
- die Dateiattribute in beiden Betriebssystemen müssen sinnvoll aufeinander abbildbar sein;
- es werden nur auf den untersten Protokollebenen Korrekturversuche nach Fehlern unternommen;
- die erforderlichen Berechtigungsabprüfungen, Zugriffskoordination etc. werden über einen Standardauftrag abgewickelt und nicht von COPY2000 selbst;
- Dateien können vom BS2000 aus nicht eingerichtet, verlängert oder verkürzt werden.

### 3.5 Grunddienste der Kopplung BS2000 ↔ VSP

Die Kommunikation zwischen den beiden Systemen erfolgt über Plattenspeicher, auf die von beiden Rechnern her zugegriffen werden kann (Shared Disks). Auf Seite des VSP werden diese Platten in üblicher Weise vom Data Management System (DMS) betrieben. Im BS2000 sind es keine Platten des Datenverwaltungssystems (DVS). Sie werden vielmehr von der BS2000-Software betrieben, die im Rahmen des Projekts VP2000 erstellt wird.

Die unterste Ebene der VP2000-Software wird von der *Abstract Device Access Method (ADAM)* realisiert, die Makroaufrufe in CCW-Ketten umsetzt und letztere zum Ablauf bringt. Die ADAM-Makros sind in ein zentrales Programm eingebunden, den Disk Transfer Modul (DTM), der alle gemeinsamen Plattenspeicher vom BS2000 her betreibt. Über diesem sind im Sinne einer Protokollhierarchie die beiden getrennten Funktionsbereiche JES und FC angesiedelt, die ihrerseits jeweils über Fast Intertask Communication (FITC) mit den Benutzerprozeduren verkehren, welche die VECTOR-Kommandos realisieren. Die beiden Funktionsbereiche sind für die Aufgaben Job Entry Subsystem (JES) und File Copy (FC) zuständig.

Die Benutzer stellen an die Komponenten JES und FC nur Aufträge zum Job- bzw. Datentransfer. Sie verkehren dazu mit einem zentralen JES- bzw. FC-Task. Diese Zentralkomponenten kreieren benutzerspezifische JES- bzw. FC-Subtasks, welche dann auf den betreffenden BS2000-Dateien satzweise und auf den VSP-Dateien (bzw. Spoolbereichen) blockweise arbeiten. Sie führen ihre Aufgaben parallel und unabhängig voneinander durch. Die

zentralen JES- und FC-Komponenten besitzen keine Schnittstellen zum DTM, sondern nur zu den sekundären JES- und FC-Komponenten, die ihrerseits DTMSchnittstellen haben. Dies hat Vorteile

- unter dem Schutz- und Sicherheitsaspekt, dass diejenigen Module, die Benutzerkontakt haben, keinen DTM-Kontakt haben,
- bei langfristigen Wartezuständen auf BS2000-Wechselgeräte,
- durch die geringere Zahl von Kontakten zwischen Benutzer und JES- bzw. FC-Komponente.

## 4. Die Portierung des FORTRAN77-Systems

### 4.1 Komponenten des FORTRAN77-Systems im MSP

Das FORTRAN-System besteht aus den folgenden Komponenten:

- FORTRAN77-Compiler zur Erzeugung von skalarem Objektcode und zugehörigem Laufzeitsystem,
- FORTRAN77/VP-Compiler zur Erzeugung von vektorisierbarem Objektcode und zugehörigem Laufzeitsystem,
- Interactive Vectorizer zur Verbesserung der Effizienz der vektorisierten Programme,
- FORTUNE zur dynamischen Analyse eines Programmablaufs,
- TESTFORT77 zur interaktiven Fehlersuche,
- DOCK/FORT77 zur interaktiven Fehlersuche im Full Screen Mode,
- SSLII (Scientific Subroutine Library, enthält mathematische Unterprogramme in skalarer Form),
- SSLII/VP (enthält mathematische Unterprogramme in vektorisierter Form).

Die Entwicklung eines FORTRAN-Programms wird im allgemeinen nach folgendem Schema durchgeführt: Zuerst wird das Programm mit dem FORTRAN77-Compiler übersetzt und nach fehlerfreier Übersetzung mit den Laufzeitmoduln zu einem ablauffähigen Programm gebunden. Bei einem Testlauf werden mit TESTFORT77 oder DOCK/FORT77 etwaige Ablauffehler analysiert und die entsprechenden Änderungen im Programm durchgeführt. Dann wird mit FORTUNE der Ablauf des Programms untersucht, und besonders zeitaufwändige Teile werden einer nochmaligen Inspektion unterzogen. Eventuell notwendige oder mögliche Verbesserungen werden durchgeführt und nochmals getestet.

Das nun fehlerfreie und für den skalaren Ablauf optimierte Programm wird anschließend mit dem FORTRAN77/VP Compiler übersetzt. Dabei wird der erzeugte Code vom Compiler vektorisiert, ohne dass Änderungen am Quellprogramm erforderlich sind (im Gegensatz zu manch anderen Herstellern, die die Vektorisierung eines Programms durch FORTRAN-Spracherweiterungen ermöglichen). Mit Hilfe des Interactive Vectorizer erhält man Information darüber, welche Leistungssteigerung von der Vektorisierung zu erwarten ist und an welchen Stellen im Programm der Compiler mittels zusätzlicher Information des Anwenders (z.B. zu erwartende TRUE/FALSE-Rate bei IF-Anweisungen u.ä.) noch weitere Verbesserungen im Code erreichen kann. Nach eventueller Eingabe dieser Benutzerinformation und erneuter Übersetzung wird das Programm mit den vektorisierten Laufzeitmoduln und, falls erforderlich, mit den für den VP optimierten mathematischen Routinen zu einem VP-ablauffähigen Programm gebunden.

### 4.2 Komponenten des FORTRAN77-Systems im BS2000

Bei den Anwendern des FORTRAN77-Systems im BS2000 muss grundsätzlich zwischen zwei Gruppen unterschieden werden: ein Teil der Anwender will existierende und getestete Programme im VP ablaufen lassen, der andere Teil will neue Programme entwickeln.

Im ersten Fall ist es erforderlich, dass die Quellen dahingehend überprüft werden, ob sie dem FORTRAN77-Standard entsprechen und keine für den BS2000-FOR1-Compiler spezifischen Spracherweiterungen enthalten. Dies kann durch einen Übersetzungslauf mit dem FORTRAN77-Skalarcompiler mit eingeschaltetem Standard-Checker erreicht werden. Eventuell betroffene Codeteile müssen dann umgeschrieben werden, damit eine fehlerfreie Übersetzung möglich ist.

Im zweiten Fall, d.h. bei der Neuentwicklung von FORTRAN-Programmen, soll der Anwender wie im MSP den portierten FORTRAN77-Compiler verwenden können. Einschränkend werden ihm jedoch einige MSP-spezifische Optionen wie z.B. die Adressierung von COM-MON-Daten oberhalb 16 MB nicht zur Verfügung stehen.

Grundsätzlich wäre es auch möglich, statt des skalaren FORTRAN77-Compilers im BS2000 den FOR1-Compiler zu verwenden, wogegen aber die Tatsache spricht, dass zwar beide Compiler den FORTRAN77-Standard unterstützen, darüberhinausgehende Erweiterungen jedoch gänzlich unterschiedlich sind und FOR1 derzeit keinen Standard-Check erlaubt. Daher kann es bei der Übersetzung mit dem FORTRAN77/VP-Compiler leicht zu Fehlern kommen mit Ursachen, die nur schwer erkennen lassen, dass unterschiedliche Spracherweiterungen dafür verantwortlich sind.

Die vom FORTRAN77-Compiler erzeugten Objektmoduln werden BS2000-seitig in PLAM-Bibliotheken abgelegt, wobei ihr Format in eine für den BS2000-Binder geeignete Form gebracht wird. Sind die so erzeugten Objektmoduln für einen Ablauf im BS2000 bestimmt, sind geringe Änderungen am Objektcode notwendig, die aus unterschiedlichen Schnittstellen in MSP und BS2000 resultieren.

Diese Objektmoduln können dann mit TSOSLNK unter Zuhilfenahme der ins BS2000 portierten FORTRAN77-Laufzeitbibliotheken zu einem ablauffähigen BS2000- oder VSP-Programm gebunden werden. Was die Testtools betrifft, so sind zwei Wege denkbar: Entweder müssen die Objektmoduln an das BS2000-Testsystem IDA/AID angepasst werden, oder die zum FORTRAN77-System gehörenden Testwerkzeuge TESTFORT77 und DOCK/FORT77 sind im BS2000 zu adaptieren.

Eine Analyse des Ablaufs der Programme erfolgt analog zum MSP im BS2000 über das Tuningwerkzeug FORTUNE, wozu auch die Bibliothek SSLII auch im BS2000 zur Verfügung gestellt wird.

Liegt das optimierte Programm vor, wird mit FORTRAN77/VP die Vektorisierung durchgeführt; weitere Verbesserungen können mit dem Tool "Interactive Vectorizer" vorgenommen werden.

Da wegen der Entlastung des VP das Binden der VP-Programme ebenfalls im BS2000 erfolgen muss, eine Portierung des Binders in das BS2000 aus lizenzrechtlichen Gründen aber entfällt, müssen auch die vom vektorisierenden Compiler erzeugten Objektmoduln in PLAM-Bibliotheken BS2000-Binder-gesiegt abgelegt werden. Gleiches gilt auch für die Bibliothek SSLII/VP. Nach dem Binden müssen schließlich die erzeugten Programme in VSP-

Lademodule umgewandelt werden. Offen ist noch das Problem, ob bei komplizierten Lademodulstrukturen ein Nachbinden im VSP erforderlich sein wird.

## 4.3 Vorgehensweise bei der Umstellung

### 4.3.1 Technik

Bei den Überlegungen, welche Adaptionstechnik bei der Portierung des FORTRAN-Systems gewählt werden soll, waren folgende Fakten maßgebend:

- Die beiden FORTRAN77-Compiler (skalar/vektorisierend), die Test- und Tuningtools sind in SPL/100 geschrieben, das FORTRAN77-Laufzeitsystem in Assembler und die SSLII-Bibliotheken in FORTRAN.
- Die gewählte Adaptionstechnik muss es erlauben, möglichst ohne zusätzlichen Aufwand mehrere Portierungen von neuen Versionen der einzelnen Komponenten durchführen zu können.
- Eine Portierung des SPL/100-Compilers ist aus vertraglichen Gründen nicht möglich.

Zuerst wurde die Arbeitsweise des SPL/100-Compilers untersucht: Er besteht aus zwei Hauptkomponenten, dem Makroprozessor und dem eigentlichen Prozessor. Der Output einer Übersetzung ist eine Assemblerquelle, die dem MSP-Assembler als Input dienen kann. Eine geeignete Optionensteuerung erlaubt es, sowohl auf den expandierten SPL/100-Code als auch auf den generierten Assemblercode zuzugreifen und diesen zu portieren.

Grundsätzlich besteht die Möglichkeit einer Portierung auf 3 verschiedenen Stufen (SPL/100-Code, Assemblercode und Objektcode; für das Laufzeitsystem nur die beiden letzten). Eine Untersuchung der einzelnen Vorgehensweisen führte zu folgendem Ergebnis:

- Eine Übernahme der SPL/100-Quellen ins BS2000 wurde entweder eine SPL/100-Compiler-Portierung erzwingen (vertraglich nicht möglich, s.o.), oder gravierende Erweiterungen des SPL4-Compilers erfordern. Auch dieser Weg kann ausgeschlossen werden, da die Unterschiede beider Sprachumfänge zu groß sind.
- Eine Übernahme der Assemblerquellen ins BS2000 erfordert darin einige Änderungen, um eine Verarbeitung durch den BS2000-Makroprozessor und -Assembler zu ermöglichen. Dieser Weg lässt sich mittels eines Preprozessors beim Transport automatisch durchführen. Da wegen der Forderung, mehrmalig ohne zusätzlichen Aufwand portieren zu können, möglichst wenig Änderungen an den Quellen selbst durchgeführt werden sollen, ist im BS2000 eine Makrobibliothek zu erstellen, die die verwendeten MSP-Makros auf BS2000-Schnittstellen abbildet. Nur bei direkten Zugriffen auf MSP-Systemtabellen (!! ) lassen sich Quelländerungen nicht vermeiden.
- Auch beim Portieren des Objektcodes ist ein Prozessor erforderlich, der die aus den unterschiedlichen Formaten resultierenden Änderungen durchführt. Zusätzlich ist eine MSP-Makrobibliothek zu erstellen, die MSP-Systemschnittstellen auf BS2000-Systemschnittstellen abbildet und während der Assemblierung im MSP zum Tragen kommt. Notwendige Quell-Änderungen (Zugriff auf Systemtabel-

len!) laufen dann im MSP ab.

Die Entscheidung über die Wahl der Portierungstechnik fiel zugunsten der zweiten diskutierten Möglichkeit, da hier der Großteil der Arbeiten im BS2000 ablaufen kann, wo ja auch alle Tests und eventuellen Neuentwicklungen (z.B. für den Dialogteil) durchzuführen sind.

### 4.3.2 Aufwand

Am Beispiel des skalaren FORTRAN77-Compilers, der einer genauen Durchsicht unterzogen wurde, ergeben sich die nachstehenden Werte als Grundlage für eine Aufwandsabschätzung.

Der SPL/100-Sourcecode des Compilers beinhaltet etwa 250.000 Quellzeilen. Die systemabhängigen Teile des Compilers (wie z.B. das Einlesen der Optionen, der Quelle oder der Includes) sind größtenteils in den beiden Controllern (Main- und Compilercontroller) zusammengefasst. Darin werden ca. 50 verschiedene Systemmakros verwendet, die auf BS2000-Systemschnittstellen abgebildet werden müssen. Der Prompter des Compilers ist voll auf die Funktionen von MSP/TSS abgestimmt und muss für den BS2000-Dialog neu implementiert werden.

Der vektorisierende FORTRAN77/VP-Compiler unterscheidet sich in den systemabhängigen Teilen nur wenig vom FORTRAN77-Compiler und verwendet zum Teil die gleichen Routinen.

Erstes Ziel ist es daher, einen im Batchbetrieb ablauffähigen Modellcompiler zu erstellen, mit dem die Einbettung in das BS2000, d.h. die Umstellung aller Systemschnittstellen, getestet werden kann. Erst dann werden die restlichen systemunabhängigen Teile des Compilers umgestellt und eingebunden.

Im Laufzeitsystem kommen etwa 60 verschiedene Systemmakros vor. Einige davon betreffen allerdings im BS2000 nicht verfügbare Systemkomponenten wie z.B. die Dateizugriffsmethode VSAM oder die sogenannte "High Speed I/O" und brauchen daher nicht umgesetzt zu werden.

Über die restlichen Komponenten des FORTRAN77-Systems können derzeit aufgrund mangelnder Unterlagen noch keine detaillierten Aussagen gemacht werden.

### 4.3.3 Reihenfolge

Beim Festlegen der Reihenfolge, in der die einzelnen Komponenten portiert werden, ist die folgende Gewichtung von Prioritäten maßgebend:

1. Anbieten der Vektorleistung im BS2000
2. Programmentwicklung und Tuning im BS2000
3. Tests und Diagnose im BS2000

Um die Vektorleistung im BS2000 zur Verfügung zu stellen, ist es notwendig, den FORTRAN77-Compiler in beiden Versionen - skalar und vektoruell - zu portieren. Wie die bisherigen Erfahrungen bei der Vektorisierung von FORTRAN-Programmen gezeigt haben, ist für eine effektive Nutzung der Vektor-Unit auch die Unterstützung durch den Interactive Vectorizer unbedingt erforderlich. Nach Umstellung dieser Komponenten des FORTRAN77-Systems hat der BS2000-Anwender die Möglichkeit, seine existierenden Programme im BS2000 zum Ablauf zu bringen.

Bei der Erstellung von neuen Programmen muss zuerst der fehlerfreie Ablauf in der skalaren Form gewährleistet sein. Um die mit dem FORTRAN77-Compiler übersetzten Programme im BS2000 zum Ablauf bringen zu können, muss das skalare Laufzeitsystem portiert werden, und an dem vom Compiler erzeugten Objektcode sind einige Änderungen durchzuführen, da das Objektformat von MSP und BS2000 nicht identisch ist. An dieser Stelle wird die Komponente FORTUNE adaptiert. Der Zeitpunkt der Umstellung ist aber noch nicht festlegbar, da ein FORTUNE-Redesign angekündigt.

Das weitere Vorgehen bezüglich der Testhilfen TESTFORT77 und DOCK/FORT77 ist bislang offen. Die Entscheidung über deren BS2000-Adaptierung oder darüber, ob eine Anpassung über die BS2000-Tools IDA/AID vorzunehmen ist, wird zu einem späteren Zeitpunkt getroffen.