

Eigenschaften einer modernen Ein-/Ausgabe Architektur

Helge Lehmann, Wilhelm G. Spruth
Version 16.0, inout16, 12.10.2002

Eigenschaften einer modernen Ein-/Ausgabe Architektur

Helge Lehmann, Wilhelm G. Spruth

1. Einführung

Das Leistungsverhalten moderner Großrechner wird im wesentlichen durch drei Faktoren bestimmt: CPU-Geschwindigkeit, Hauptspeichergroße und Ein-/Ausgabeleistung.

Für eine Reihe von Anwendungen ist (fast) nur die CPU Leistung entscheidend. Darunter fallen beispielsweise die Wettervorhersage und Klimamodelle. Viele Anwendungen verlangen einen ausreichend großen Hauptspeicher, um Leistungseinbußen, z.B. durch eine exponentiell ansteigende Seitenwechselrate, zu vermeiden. Die Anzahl der gleichzeitig aktiven Prozesse wird ebenfalls durch die Größe des Hauptspeichers vorgegeben.

Für die allermeisten Transaktionsverarbeitungs- und Datenbank-Anwendungen ist die Güte des Ein-/Ausgabe Subsystems der leistungsbestimmende Faktor. Eine deutliche Mehrheit der heute im Einsatz befindlichen Großrechner bearbeitet Aufgaben dieser Art. Es wird allgemein anerkannt, dass die Rechner der S/390 und zSeries Architektur an dieser Stelle über eine überlegende Technologie verfügen. Weniger bekannt sind jedoch die Elemente, die eine moderne Ein-/Ausgabe Technologie ausmachen.

Dies ist ein Themenkreis, der in modernen Lehrbüchern häufig vernachlässigt wird. Beispielsweise widmet das Standardwerk über Computer-Architektur von Hennesy/Patterson von 1100 Seiten lediglich etwa 50 Seiten der Ein-/Ausgabe [HEN]. Dies steht im Gegensatz zur Struktur von Großrechnern, in denen die Kosten für die Ein-/Ausgabe typischerweise um den Faktor 10 höher liegen als für den Prozessor Complex ([PFI]).

Die vorliegende Veröffentlichung beabsichtigt, moderne Ein-/Ausgabe Einrichtungen am Beispiel der S/390 und zSeries Architektur zu erläutern.

2. S/390 und zSeries Architektur

Die S/390 und heutige zSeries Architektur ist historisch-technologisch in Jahrzehnten gewachsen. Die 1964 von Amdahl, Blaauw und Brooks [AMD] eingeführte S/360 Architektur stellte seinerzeit einen Meilenstein in der Entwicklung des Computers dar [WI1, WI2]. Zum damaligen Zeitpunkt waren die Unterschiede in den Rechner Architekturen der einzelnen Hersteller sehr viel größer als dies heute der Fall ist. Viele Eigenschaften, die heute als selbstverständlich gelten, entstanden mit der Einführung der S/360 Architektur.

Diese historischen Wurzeln führen zu der weit verbreiteten Meinung, dass die S/390 bzw. zSeries Hard- und Software-Technologie veraltet sei und über kurz oder lang aussterben würde. In Wirklichkeit wurde die Architektur (und Technologie) kontinuierlich weiter entwickelt und verbessert, wobei gleichzeitig eine sehr gute Rückwärtskompatibilität für existierende Anwendungen erreicht wurde.

Die führende Marktposition der S/390 und zSeries Rechner im Marktsegment der großen kommerziellen Server ist vor allem auf Hardware- und Software-Technologie-Eigenschaften zurückzuführen, über die andere Rechner (noch) nicht verfügen. Auch in der Vergangenheit war S/390 gegenüber den Mitbewerbern technologisch immer um Einiges voraus. Beispiele für führende technologische Eigenschaften sind [HER]:

- Hardware Technologie [HAR],
- Ein-/Ausgabe Architektur,
- Virtualisierung mit Hilfe der Virtual Machine (z/VM) Einrichtung,

- Logische Partitionierung (LPAR) mit Hilfe der PR/SM Einrichtung [SPR],
- z/OS Goal-orientierter Workload-Manager [WW1],
- Clustering, Sysplex [JRD], [ISJ],
- Skalierung mit Hilfe der Coupling Facility [RAH].

IBM bezeichnet seine Hardware als zSeries (früher S/390) und das am meisten eingesetzte Betriebssystem als z/OS (früher OS/390). Die zSeries Systeme mit z/OS weisen gegenüber S/390 Systemen mit OS/390 eine zusätzliche 64 Bit-Unterstützung auf. Die wichtigste zSeries Implementierung wird als z900 bezeichnet; eine kleinere z800 wird seit 2002 ausgeliefert. Im Rahmen dieses Aufsatzes verwenden wir die Bezeichnungen zSeries und z/OS, schließen damit aber ausdrücklich S/390 und OS/390 mit ein.

In dem vorliegenden Beitrag werden die technologisch führenden Ein-/Ausgabe Eigenschaften des zSeries Systems beschrieben. Hierzu zählen folgende Hardware Einrichtungen:

- ein fortschrittliches Cache-Coherence Network erlaubt allen CPUs eines symmetrischen Multiprozessors einen schnellen Zugriff über den jeweils lokalen L1 Cache auf einen gemeinsam genutzten (shared) L2 Cache, und über diesen mittels eines *Main Storage Controllers* (MSC) auf den Hauptspeicher,
- eine hohe Bandbreite zum L2 Cache und zum Hauptspeicher,
- eine große Anzahl von leistungsfähigen *I/O Hubs*, deren Ports zum *1st order SAN* (*System Area Network*) oder zu Ein-/Ausgabe Adaptern führen,
- eine große Anzahl von verschiedenen Ein-/Ausgabe Adaptern, die ihrerseits den Zugang zu verschiedenen
 - > *Communication Networks* wie LANs, WANs, oder zu
 - > *2nd order SANs* (*Storage Area Networks*) wie ESCON, FICON oder Fibre Channel Fabrics ermöglichen,
- Fehlererkennung und -korrektur Einrichtungen auf allen Hardware- und Software-Ebenen, die damit einen Beitrag zu der erreichten Ausfallsicherheit von 99,999% einer zSeries Installation leisten.

Weitere Eigenschaften der zSeries Ein-/Ausgabe Architektur sind:

- das Offloading von Ein-/Ausgabe Operationen weg von der (oder den) CPU(s) hin zu einem (oder mehreren) *System Assist Prozessoren* (SAPs), die im Wesentlichen das zSeries *Channel Subsystem* (CSS) beherbergen,
- neuartige Einrichtungen wie z.B. die vollautomatische Behandlung von Belegt-Zuständen an beliebiger Stelle der Ein-/Ausgabe Topologie oder das *Channel Subsystem Priority Queuing* (CSSPQ),
- die Weiterentwicklung von SSCH/CCW/Command/Status basierenden Ein-/Ausgabe Protokollen (wie CKD und ECKD) hin zu Request Queue/Control Block basierenden Eigenschaften (wie QDIO/Gigabit-Ethernet und QDIO/SCSI)
- weitgehende Einrichtungen zur Verbesserung der Ein-/Ausgabe Konfigurations-Flexibilität, die sowohl statische als auch umfangreiche dynamische Änderungen ermöglichen.

3. Ein-/Ausgabe Topologie

Im Bild 1 ist die Struktur des z900 Central Electronic Complex (CEC) und sein Anschluss an externe Netze wie Communication Networks (WANs, LANs) und Storage Area Networks dargestellt. Ein gemeinsam genutzter (Shared) L2 Cache verbindet bis zu 20 Prozessoren (16 CPU, 3 SAP, 1 Reserve Prozessor) , 4 *Memory Bus Adapter* (MBA) Chips für den Anschluss von Ein-/Ausgabe Adapterkarten und mehreren Hauptspeicherkarten. Die MBA Chips stellen die zSeries Implementierung eines generischen I/O Hubs dar. Sie erfüllen eine ähnliche Aufgabe wie die Southbridge in einem PC, allerdings mit deutlich höherer Funktionalität. Der Shared L2, die Prozessoren und die MBA Chips befinden sich auf einem einzigen *Multi Chip Module* (MCM).

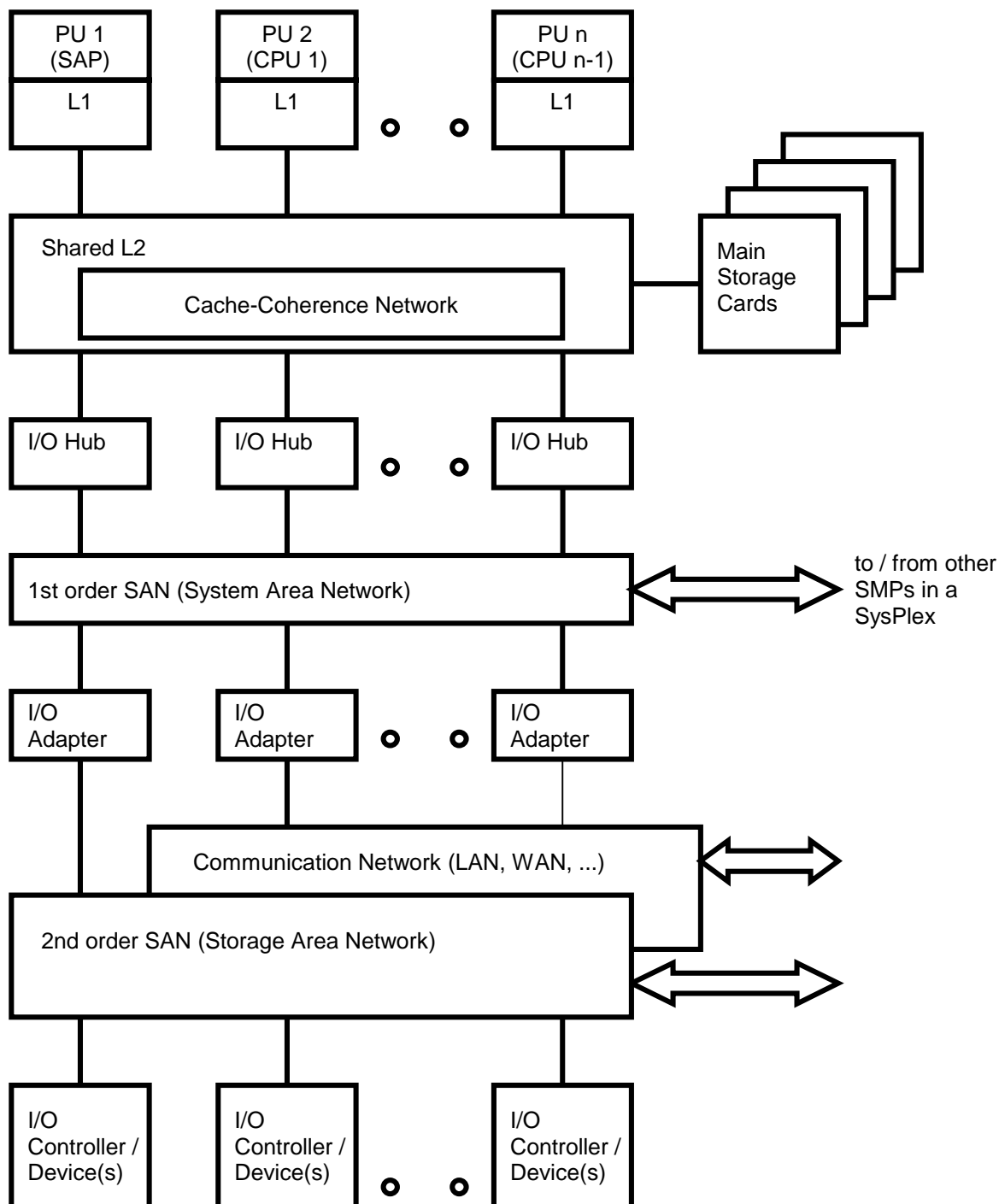


Abb 1 : zSeries SMP Ein-/Ausgabe Konfiguration

Alle Verbindungen sind Punkt zu Punkt und auf Grund der hohen Packungsdichte des MCMs sehr kurz, so dass die Busse mit bis zu 450 MHz betrieben werden können. Dies führt zu sehr guten Latency- und Bandbreiten Werten. So besitzt beispielsweise jede CPU und jeder SAP eine Bandbreite zum L2 Cache von 14.5 GByte/sec mit einer Zugriffszeit von 20 ns. Der L2 Cache stellt für alle Prozessoren eine gesamte Bandbreite von 290 GByte/sec zur Verfügung. Die Hauptspeicher Bandbreite beträgt 29 GByte/sec. Jedes MBA Chip schliesst 6 full duplex Byte-serielle Busse an, die mit einer 2ns double Data Rate Clock betrieben werden. Dies ergibt für jeden Bus eine Ein-/Ausgabe Bandbreite von 1 GByte/s in jeder Richtung. Nach Abzug des Protokoll-Overheads beträgt die effektive Datenbandbreite 750 MByte/sec in jeder Richtung. Die Bandbreite für alle I/O Hub Chips beträgt somit theoretisch 36 Gbyte/s, hiervon sind 29 GByte/s in der Praxis realisierbar. Die Busse

verwenden das STI Protokoll [HOK], welches eine ähnliche Rolle wie der PCI Bus in einem PC übernimmt, allerdings mit wesentlich höherer Datenrate und Funktionalität. Da der STI-Bus differenzielle Treiber und Receiver benutzt, kann die Kabellänge bis zu 15m betragen.

Das 1st order System Area Network (Abb. 2) besteht aus Switches, die ebenfalls mit STI-Bussen verbunden sind. Hier beträgt die Datenrate jedoch 500 oder 333 MHz. Jeder Switch hat 4 STI full-duplex Bus Ausgänge. Jeder dieser Ausgänge geht zu einer I/O Card, z.B. einer Common I/O Card oder einer ISC Card. Von den maximal 96 Ausgängen sind 84 nutzbar für I/O Cards (z.B. FICON, Gigabit Ethernet und andere). Die I/O Cards sind in Ein-/Ausgabe Rahmen (I/O Cages) untergebracht. Bis zu 3 I/O Cages mit jeweils bis zu 28 Ein-/Ausgabe Adapter Karten können an das System angeschlossen werden.

Solch ein System kann nun auf verschiedene Art und Weise in einen als Parallel Sysplex bezeichneten Rechnerverbund (Cluster) integriert werden [RAH]. Zum einen können mit Hilfe des *Integrated Cluster Buses (ICB)*, der das Clustering Protokoll auf der Basis eines STI-Kabel implementiert, mehrere z900 Systeme von MBA zu MBA direkt miteinander verbunden werden. Die maximale Entfernung beträgt hierbei 10 m. Zum anderen werden für räumlich verteilte Cluster (*Geographically Dispersed Parallel Sysplex [GRE]*) bis zu 20 km lange optische Verbindungen benutzt, wofür eine spezielle Karte, die *Inter System Channel (ISC)* Karte, benötigt wird.

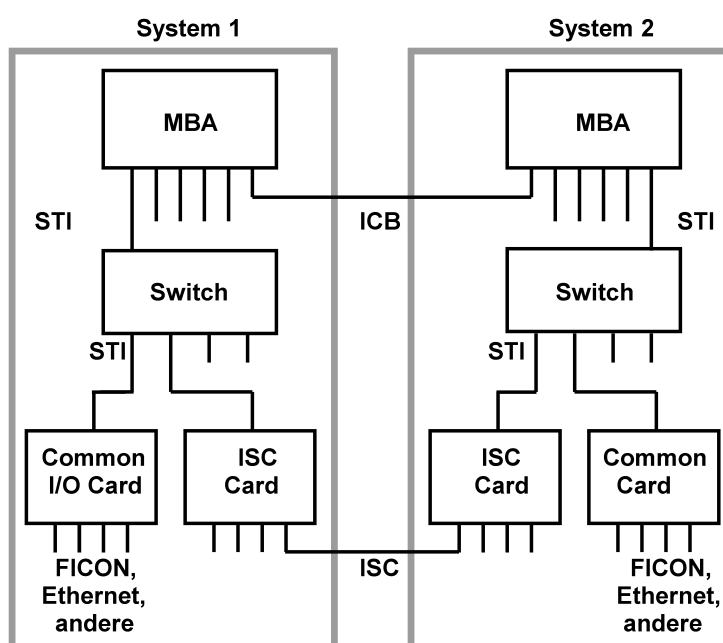


Abb. 2 : System Area Network

4. S/390-z900 Ein-/Ausgabe Architektur

4.1. Traditionelle S/390 Ein-/Ausgabe Architektur

Die traditionelle S/390 Ein-/Ausgabe Architektur wird durch Kanäle (Channels), Subchannels und Kanalprogramme (Channel Programs) charakterisiert.

Der Kanal ist ein leistungsfähiger Ein-/Ausgabe Adapter, der speziell in Hinblick auf die Anforderungen eines zSeries Ein-/Ausgabe Subsystems hinsichtlich Leitungsfähigkeit (performance), aber insbesondere auch RAS (**R**eliability, **A**vailability, **S**erviceability) Anforderungen entwickelt wurde. Die wichtigsten Ein-/Ausgabe Adapter der z900 und z800 Systeme basieren auf einer *Common I/O Card* (s. Abb. 2), die in Abschnitt 6 näher beschreiben wird. Ein Subchannel ist ein logisches Abbild eines Ein-/Ausgabe Gerätes, das über eine Steuereinheit (Control Unit) an den Kanal angeschlossen ist.

Sowohl diese Architektur als auch die SCSI Architektur gehen auf den 1964 entwickelten Selector Kanal der S/360 Architektur zurück. Der SCSI Bus und das serielle SCSI Interface haben eine ähnliche Funktion wie der S/390 Kanal, wenn auch mit geringerer Funktionalität.

Die Verbindung zwischen Kanal und Steuereinheit, ursprünglich basierend auf einem elektrischen Kabel mit Busstruktur ("parallel channel"), hat sich zu einem optischen Netzwerk ("FICON Channel") weiterentwickelt, das Entfernungen bis zu 100 km zwischen Rechnersystem und Steuereinheit erlaubt, bei Datenübertragungsraten bis zu 2 Gbit/s. FICON ist ein Mitglied der Fibre Channel Protokoll Familie. Es unterscheidet sich von anderen Mitgliedern der Familie dadurch, dass es in der obersten Schicht (FC-4) an Stelle eines Audio, Video, 802.2 oder seriellen SCSI Protokolls ein FICON Protokoll definiert, das die Ein-/Ausgabe Architektur der zSeries unterstützt. FICON ist die Voraussetzung für zahlreiche Einrichtungen, über die z.B. SCSI in der parallelen oder der Fibre Channel seriellen Ausführung nicht verfügt. Neben dem zSeries spezifischen FICON Protokoll unterstützen die neuesten zSeries System aber auch das SCSI Protokoll über Fibre Channel (SCSI FCP).

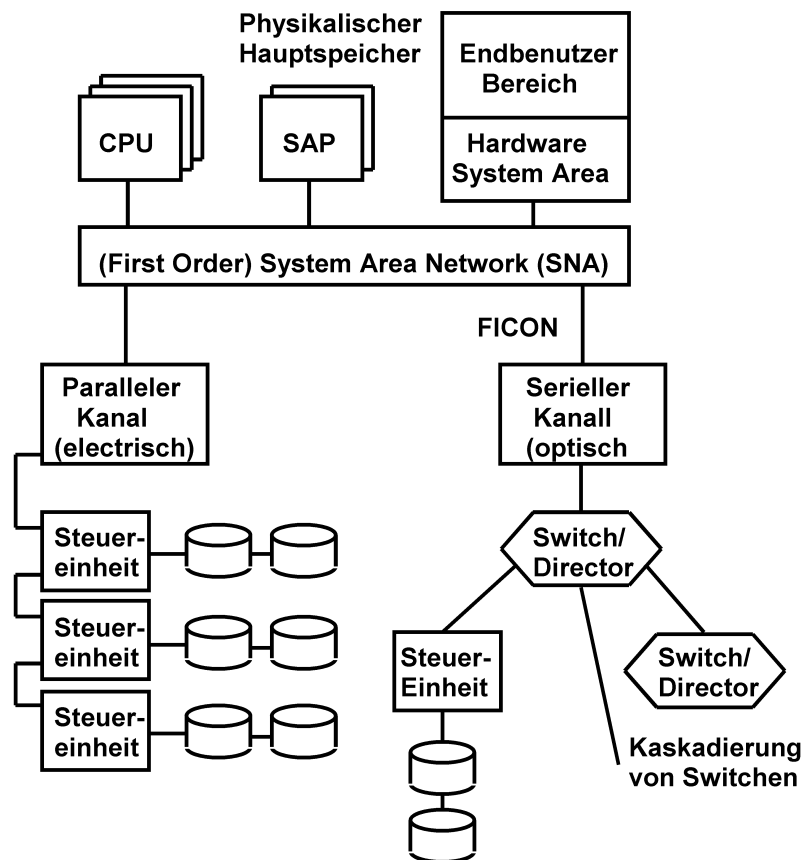


Abb. 3 : Parallele und serielle Kanäle der S/390 bzw. zSeries

In einem Fibre Channel Netzwerk, basierend auf dem FICON oder SCSI Protokoll, können über Fibre Channel Switches und Directors eine Vielzahl von Servern und Steuereinheiten bzw. Endgeräten in einem Storage Area Network (2nd order SAN) miteinander verknüpft werden. Wir bezeichnen als Director einen Switch mit erweiterten Funktionen, z.B Redundanz oder Recovery Einrichtungen. Storage Area Netzwerke im zSeries Umfeld verwenden in der Regel Directors an Stelle von einfachen Switches.

Eine beliebig komplexe Ein-/Ausgabe Operation wird durch ein Kanalprogramm (Channel Program) definiert. Das Kanalprogramm besteht aus einzelnen Befehlen, die als Channel Command Words (CCWs) bezeichnet werden. Es stehen Mechanismen wie z. B. Bedingungsabfragen und bedingte Sprünge zur Verfügung. Jede Ein-/Ausgabe Operation besteht aus den folgenden Schritten:

- Die Ausführung des Kanalprogramms wird durch einen speziellen Ein-/Ausgabe-Maschinenbefehl der CPU, den *Start Subchannel* (SSCH) Befehl, angestoßen.

- Die Ausführung des Kanalprogramms erfolgt in Kooperation zwischen dem *Channel Subsystem* (siehe unten), das auf dem *System Assist Processor* (SAP) läuft, dem Kanal, und einer Steuereinheit.
- Die Beendigung eines Kanalprogramms wird durch eine Ein-/Ausgabe Unterbrechung, zusammen mit entsprechender Statuspräsentation, an die CPU gemeldet.
- Eine besondere Eigenschaft der zSeries Architektur besteht darin, dass der Zugriff auf einen Kanal wie auf die daran angeschlossenen Steuereinheiten und Endgeräte gleichzeitig von mehreren Betriebssystemen erfolgen kann, die parallel auf der oder den zSeries CPU(s) ausgeführt werden. Channel Subsystem und Kanal stellen dabei sicher, dass diese Ein-/Ausgabe Operationen sich nicht gegenseitig stören.

4.2 Queued Direct I/O

Während die eigentliche Ausführung eines Kanalprogramms durch einen SAP sowie den Kanal erfolgt und damit parallel zur Bearbeitung von zSeries Programmen durch die CPU abgewickelt wird, belastet sowohl das Starten eines Kanalprogramms als auch dessen Beendigung die CPU (letztere wird durch eine Unterbrechung signalisiert). Hier bringen die „Queued Direct I/O“ Protokolle der zSeries, die von der Common I/O Card unterstützt werden, eine deutliche Verbesserung der Leistungsfähigkeit eines Systems, da sie sowohl die CPU entlasten, als auch die zum Starten bzw. für die Beendigung einer Ein-/Ausgabe Operation benötigte Zeit reduzieren. Damit werden sowohl die Zugriffszeiten (access times) als auch der Durchsatz, gemessen in Ein-/Ausgabe Operationen pro Sekunde, verbessert.

Die Queued Direct I/O Protokolle der zSeries basieren auf (theoretisch) endlos laufenden Kanalprogrammen (Abb. 4). Nachdem eine Ein-/Ausgabe Operation mit einem SSCH Maschinenbefehl gestartet wurde, werden nachfolgende Ein-/Ausgabe Operationen vom CPU Programm (in der Regel auf Betriebssystemebene) in eine Output oder Request Queue gepackt, die der Kanal (implementiert auf Basis der Common I/O Card) ausliest und bearbeitet. Solange diese Queue nicht leer wird, braucht der Kanal hierfür keinen weiteren Anstoß. Lediglich wenn das CPU Programm einen Request in eine leere Queue ablegt, muss der Prozessor des Kanals aufgeweckt werden, um die Bearbeitung der Queue weiterzuführen.

Umgekehrt legt der Kanal eingehende Nachrichten oder Fertigmeldungen in eine Input oder Completion Queue, die vom CPU Programm sequentiell ausgelesen wird. Auch hier ist ein Unterbrechungssignal nur dann erforderlich, wenn eine solche Nachricht bzw. Fertigmeldung in eine leere Queue abgelegt wird.

4.3 Andere Entwicklungen

Das zSeries *Fibre Channel Protocol for SCSI* (FCP) ermöglicht den Zugriff auf Fibre Channel SCSI Plattenspeicher. Einzelheiten sind in [ALD] beschrieben.

Manche Rechnerarchitekturen setzen ebenfalls auf Queuing Mechanismen basierende Ein-/Ausgabe Protokolle ein. Allerdings bieten diese in der Regel nicht die Schutzmechanismen, die die RAS Anforderungen der zSeries zwingend vorschreiben. Sie erlauben auch keine unabhängige und geschützte Kommunikation mit verschiedenen Betriebssystemen, die auf derselben Hardware-Plattform laufen. Beides wird bei den Queued Direct I/O Protokollen der zSeries garantiert. Insbesondere kommt hier die Firewall Funktion, die in der STI-PCI Bridge (siehe unten) implementiert ist, zum Tragen.

Ein wichtige Neuentwicklung auf dem Gebiet der Ein-/Ausgabe Protokolle stellt der *InfiniBand* (IB) Standard dar, einer gemeinschaftlichen Entwicklung der Firmen Compaq, Dell, Hewlett-Packard, IBM, Intel, Microsoft und Sun Microsystems. InfiniBand ist als eine leistungsfähiges Ein-/Ausgabe Architektur für zukünftige Rechnersysteme geplant; insbesondere sind für Standard Ein-/Ausgabe Operationen keine Interaktion mit dem I/O Supervisor des Betriebssystems mehr erforderlich.

InfiniBand benutzt ein ein- oder mehradriges optisches oder elektrisches Kommunikationsnetz, das in der optischen Ausführung maximale Entfernungen und Datenraten (pro Ader) in der gleichen Größenordnung wie FICON zulässt. Das InfiniBand Protokoll verbindet queuing-basierende Ein-/Ausgabe Mechanismen, definiert einen geschützten Datentransfer zwischen dem Hauptspeicher des Rechners und der Ein-/Ausgabe Welt, basierend auf einem Firewall-Konzept und unterstützt mehrere unabhängige, parallel laufende Betriebssysteme auf einem Rechnersystem. Es greift damit einige der wichtigsten Eigenschaften der heutigen zSeries Ein-/Ausgabe-Systeme auf.

Die zSeries Architektur, einschließlich der Ein-/Ausgabe Architektur, ist in einem offiziellen Architektur Handbuch beschrieben [PRI] .

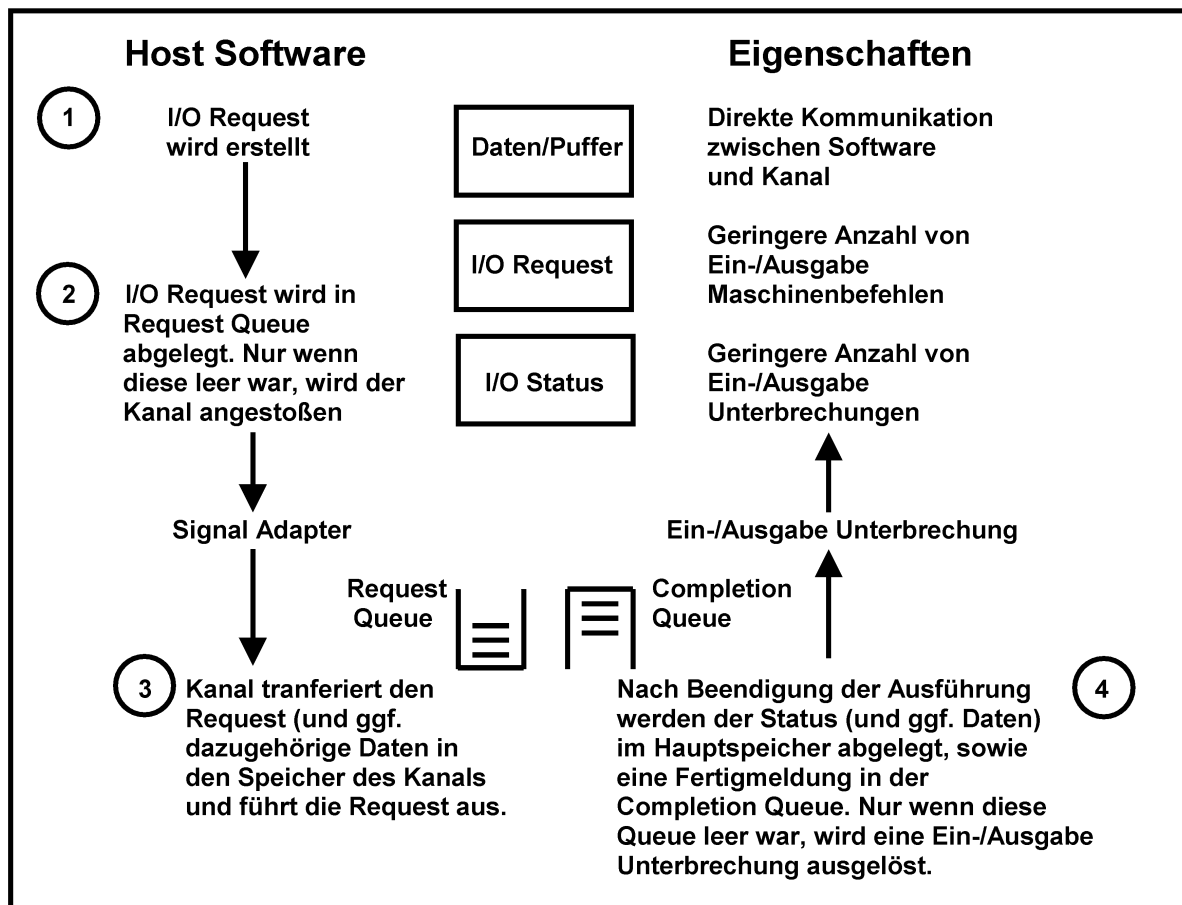


Abb. 4 : Queued Direct I/O Channel Ein-/Ausgabe Schnittstelle (z.B. Fibre Channel)

5. Channel Subsystem

5.1 Definition

Der größte Unterschied zwischen der heutigen z/Architektur und anderen Rechner-Architekturen besteht im Ein-/Ausgabe Subsystem. Sein zentraler Bestandteil ist das *Channel Subsystem* (CSS), das ursprünglich 1977 eingeführt und seitdem kontinuierlich verbessert wurde. Während vorher eine starke Verflechtung von Betriebssystem Komponenten und der I/O Welt existierte, wurde mit der Einführung des Channel Subsystems der Versuch unternommen, das Ein-/Ausgabe Subsystem konsequent abzukapseln, von den Betriebssystem Komponenten zu entkoppeln und es mit einer definierten Schnittstelle zu versehen. In heutiger Terminologie könnte man hier von einem objekt orientierten Ansatz sprechen, der den Architekten mit der Definition des Channel Subsystem bereits vor 25 Jahren gelungen ist. Unter dem Begriff Channel Subsystem verstehen wir dabei die Summe aller physikalischen und logischen Ressourcen und Einrichtungen, die notwendig sind, um den Informationsaustausch zwischen dem Hauptspeicher und den angeschlossenen Ein-/Ausgabe Geräten zu veranlassen und zu kontrollieren. Bei Systemen der zSeries gehören zu den physikalischen Ressourcen:

- dedizierte Prozessoren, die erwähnten *System Assist Prozessoren* (SAPs)

- ein gesonderter, dem Betriebssystem und den Anwendungen nicht zugänglicher Speicherbereich, die *Hardware System Area (HSA)*
- die I/O Hubs
- und die Ein-/Ausgabe Adapter.

Die logischen Ressourcen bestehen im wesentlichen aus:

- dem Microcode (bei der zSeries Architektur als *Licensed Internal Code (LIC)* bezeichnet, [SPR]), der z.T. auf den CPUs, vor allem aber auf den SAPs und in den Prozessoren der Ein-/Ausgabe Adapter ausgeführt wird
- sowie den zur Durchführung und Kontrolle der Ein-/Ausgabe Operationen notwendigen Datenstrukturen. Diese befinden sich zum größten Teil in der HSA, aber auch in lokalem Speicher der Ein-/Ausgabe Adapter.

Das Channel Subsystem (CSS) verwaltet bis zu 256 Kanäle (Channels) und bis zu 65.536 angeschlossene Ein-/Ausgabe Geräte (devices) für bis zu 15 Betriebssysteme, die gleichzeitig auf einem zSeries Rechner laufen können (Virtualization, LPAR, PR/SM) . Damit werden diese Betriebssysteme von immer wiederkehrenden Arbeiten entlastet ("offloading") und können überlappend zu der laufenden Ein-/Ausgabe Operation andere Maschinenbefehle ausführen. Die Architektur unterstützt bis zu 8 parallel zueinander verlaufende Datenpfade zu einem Ein-/Ausgabe Gerät ("multipathing"). Ein solcher Datenpfad kann im Allgemeinen folgende Zwischenstationen beinhalten:

- Kanal (Channel)
- Switch
- Steuereinheit (Control Unit)
- Ein-/Ausgabe Gerät (Device)

Das CSS entscheidet, welcher Pfad genommen wird ("path finding"). Insbesondere werden nur funktionsfähige Pfade ausgewählt und möglichst solche, die frei sind, d.h. nicht gerade mit anderen Aufgaben betreut sind. Tritt trotzdem der Fall ein, dass der ausgewählte Kanal, der Switch, die Steuereinheit oder das Gerät den anstehenden Auftrag nicht sofort erledigen können ("busy condition"), so übernimmt das CSS die Aufgabe, diese Ein-/Ausgabe Operation erneut zu starten ("busy handling"). Da das CSS sehr kurze Kommunikationspfade zu den Kanälen besitzt, ist diese zentrale Verwaltung der Ein-/Ausgabe Ressourcen sehr effektiv.

Diese Zusammenhänge sind in Abb. 5 wiedergegeben

5.2 Behandlung von Belegt-Zuständen (“Busy Handling”)

In der oben gezeigten Ein-/Ausgabe-Topologie mit Channel Subsystem, Kanalpfaden, Switches, Steuereinheiten und Ein-/Ausgabe Geräten kann es beim Starten einer Ein-/Ausgabe Operation an allen Stellen zu meist temporären Belegt-Zuständen kommen:

1. Die erste Hürde ist das Finden eines freien Kanalpfades aus einer Auswahl von bis zu 8 Pfaden zu dem angesprochenen Ein-/Ausgabe Gerät. Sollten alle hierfür vorgesehenen Pfade belegt sein, wird der Auftrag im CSS gehalten und nach kurzer Zeit erneut versucht zu starten.
2. Die zweite Hürde stellen die Switches auf dem Weg zur Steuereinheit dar, die ein “belegt” auf dem für diese Verbindung vorgesehenen abgehenden Pfad(en) entdecken können. Auch in diesem Fall wird der Auftrag im CSS gehalten und nach kurzer Zeit erneut versucht zu starten.
3. Die dritte Hürde stellt die Steuereinheit dar, die u.U. wegen Überlastung diesen Auftrag zu diesem Zeitpunkt nicht annehmen kann. Auch in diesem Fall wird der Auftrag im CSS gehalten und sofort wieder versucht zu starten, sobald die Steuereinheit ein “no longer busy” signalisiert.
4. Die vierte Hürde stellt das Gerät selbst dar, das zufälligerweise zu diesem Zeitpunkt eine Ein-/Ausgabe Operation im Auftrag eines anderen Systems ausführen könnte. Auch in diesem Fall wird der Auftrag im CSS gehalten und sofort wieder versucht zu starten, sobald das Gerät ein “no longer busy” signalisiert.

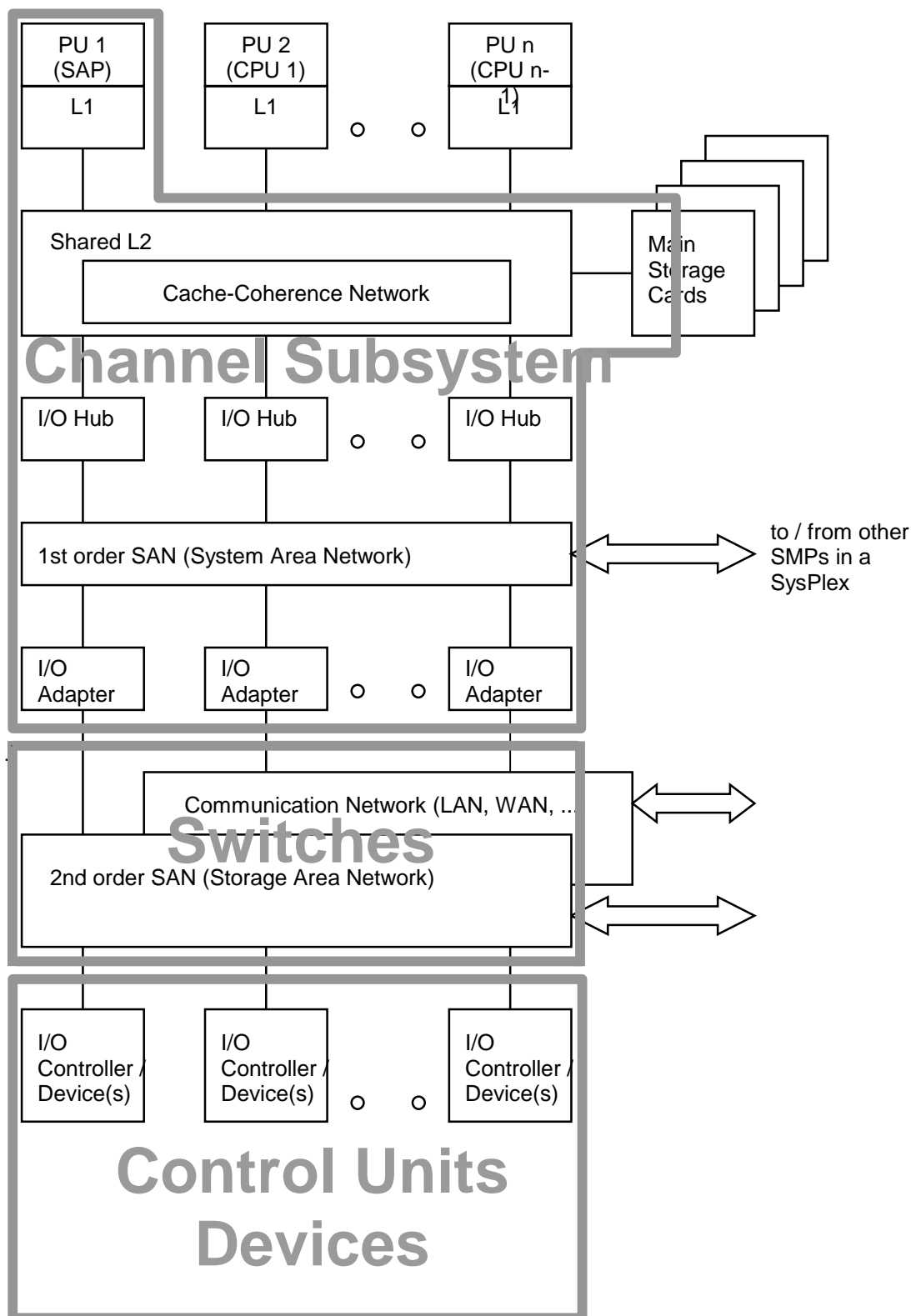


Abb. 5 : Zusammenspiel des Channel Subsystems, der 2nd Order SAN Switches und der Control Units

Alle diese "Hürden" sollten bei einer sorgfältig ausgelegten Ein-/Ausgabe-Topologie eher selten auftreten. Trotzdem ist mit dem hier aufgezeigten Mechanismus ein Konzept vorhanden, das vollautomatisch und ohne Einbuße an CPU-Leistung alle potenziellen Belegt-Zustände behandelt, ohne dass sich das Betriebssystem oder die Anwendungen darum kümmern müssen.

5.3 Asynchronität der Ein-/Ausgabe Maschinenbefehle

Eine wesentliche Fähigkeit des Channel Subsystems ist die Entlastung des Betriebssystems durch überlappende Ausführung von Ein-/Ausgabe Maschinenbefehlen mit anderen, nicht Ein-/Ausgabe bezogenen Maschinenbefehlen. Hierzu werden die Ein-/Ausgabe Maschinenbefehle asynchron ausgeführt. Das bedeutet, dass nach einer kurzen formalen Prüfung des Ein-/Ausgabe Auftrages der Maschinenbefehl aus Sicht des Betriebssystems abgeschlossen und zu Ende ist (der Condition Code wird gesetzt), der Auftrag inhaltlich aber erst danach bearbeitet wird. Die Effektivität dieses Verfahrens wird auf einer z900 noch dadurch unterstützt, dass diese asynchrone Abarbeitung des Ein-/Ausgabe Auftrages auf einem eigens dafür reservierten Prozessor (SAP) durchgeführt wird.

5.4 Channel Subsystem Priority Queuing

Neben dieser soeben beschriebenen Entlastung der Betriebssysteme durch das asynchrone Ausführen von Ein-/Ausgabe Operationen auf einem eigenen Prozessor leistet das Channel Subsystem weitere eigene Beiträge zur Steigerung des Durchsatzes im Ein-/Ausgabe Bereich. Einer dieser Beiträge läuft unter dem Stichwort "Priority Queuing".

In einem hoch ausgelasteten System mit vielen Prozessoren laufen in der Regel viele verschiedene Anwendungen mit stark unterschiedlichen Anforderungen an das Antwortzeitverhalten der von ihnen angestoßenen Ein-/Ausgabe Operationen. Ein extremer Fall sind Online-Datenbank-Abfragen mit nur wenigen Ein-/Ausgabe Operationen und einer "sub-second response time"-Anforderung. Ein anderer extremer Fall sind Batch-Programme, die im Hintergrund laufen.

Mit dem Channel Subsystem Priority Queuing (CSPPQ) wird in Verbindung mit dem z/OS Work Load Manager (WLM) und dem Dynamic Channel Path Management (DCM) jeder Ein-/Ausgabe Operation eine individuelle Priorität (von 0 bis 15) mitgegeben. Dieser Prioritätswert wird vom CSS beim Starten von Ein-/Ausgabe Operationen beachtet, wobei die Operationen mit einem höheren Prioritätswert Vorrang haben. Der Algorithmus ist so ausgelegt, dass auch Ein-/Ausgabe Operationen mit niedrigen Prioritäten nicht unendlich lange warten müssen. Die im Einzelfall in einem System verwendeten Prioritätswerte werden durch den System-Programmierer festgelegt. Die Verbesserung des Antwortzeitverhaltens von Ein-/Ausgabe Operationen mit hoher Priorität ist deutlich messbar, und kann z.B. für die Erfüllung von Service Level Agreements wichtig sein.

Die hier beschriebene Einrichtung ist Bestandteil einer als Intelligent Resource Directors (IRD) bezeichneten Komponente des z900 Rechners und des z/OS Betriebssystems.

6. Datensicherheit - und Zuverlässigkeit

Die meisten Marktforschungsunternehmen attestieren z900 und z/OS eine überlegene Zuverlässigkeit- und Verfügbarkeit im Vergleich zu anderen Rechnern. Eine Verfügbarkeit von 99,999 % wird an dieser Stelle häufig genannt. Dies bedeutet nicht mehr als 5 Minuten Ausfallzeiten während eines Jahres für geplante und ungeplante Unterbrechungen des Rechnerbetriebs. Für viele Unternehmen sind derartig gute Verfügbarkeitswerte wichtig.

Eine große Anzahl unterschiedlicher Hardware- und Softwareeigenschaften trägt zu der guten Verfügbarkeit bei. Auch das Ein-/Ausgabe Subsystem leistet hierzu einen Beitrag. Zwei besonders wichtige Maßnahmen bestehen darin zu verhindern, dass

- falsche oder fehlerhafte Daten vom/zum Hauptspeicher transferiert werden, und
- im Falle eines Adressierungsfehlers DMA Daten in die falsche Adresse im Hauptspeicher geschrieben werden.

6.1 Ein-/Ausgabe in PCs und Workstations

Die Welt der Ein-/Ausgabe-Adapter wird heute von PCI-Adapter Karten dominiert, unabhängig von der Art der Ein-/Ausgabe (wie z.B. Speicherzugriff, lokale oder nicht-lokale Kommunikation) und des verwendeten Übertragungsprotokolls (wie z.B. SCSI, IDE/ATA, Fibre Channel für Plattenspeicher oder Ethernet, ATM usw. für Kommunikationsanschlüsse).

Bei PCs und Workstations wird die Operation der PCI Adapter von dem Betriebssystem oder einer Applikation initiiert, die auf dem zentralen Prozessor des Rechners ausgeführt wird. Dazu greift das Programm über *Memory Mapped I/O (MMIO)* direkt auf Register und Datenbereiche in dem Adapter zu, unter Verwendung von Adressen im Adressbereich des PCI-Busses. Die zu übertragenden Daten werden dann meist vom PCI Adapter per *Direct Memory Access (DMA)* vom Hauptspeicher des Rechners zu einem Puffer im PCI Adapter bzw. umgekehrt übertragen.

Um an einem größeren Server mehrere PCI Busse zu betreiben und zu diesem Zwecke auch etwas größere Entfernungen zwischen dem oder den Prozessoren und den PCI Bussen überbrücken zu können, implementieren viele Server eine proprietäre Verbindung zwischen Prozessor und PCI Bussen. Häufig wird auch hier die Operation des PCI Adapters direkt durch *Memory Mapped I/O* vom Prozessor aus gesteuert (auch als *load/store I/O mode* bezeichnet). Dieser Zugriff wird lediglich über die proprietäre Verbindung durchgeschleust, transparent sowohl für das ausführende Programm als auch für den PCI Adapter. Entsprechendes gilt auch für den DMA-getriebenen Datentransfer.

Der PCI Bus in seiner ursprünglichen Form hat nur sehr limitierte Schutzmechanismen, um die Integrität der übertragenen Speicheradressen und Daten zu garantieren. Dieses Problem wurde erst mit späteren Weiterentwicklungen, wie dem PCI-X Bus und dem PCI Express, an dessen Spezifikation derzeit gearbeitet wird, verbessert.

Weiterhin sitzen auf den meisten PCI Adaptern selbst Prozessoren und andere Komponenten unterschiedlichster Ausprägung, die wenig oder gar keine Prüfmechanismen zur Erkennung interner Fehler besitzen. Da in vielen Fällen der PCI Adapter selbst einen DMA Transfer in den Hauptspeicher des Systems durchführt, kann er dabei fehlerhafte Daten transferieren, oder aber im Falle eines Adressierungsfehlers Daten an die falsche Adresse im Hauptspeicher schreiben.

6.2 zSeries Ein-/Ausgabe

Die Kanäle, die eine zentrale Rolle im zSeries Ein-/Ausgabe-System spielen, wurden ursprünglich aus diskreten Bauelementen unter Berücksichtigung der besonderen Anforderungen der IBM Großsysteme hinsichtlich Datensicherheit, Zuverlässigkeit und Verfügbarkeit entwickelt. Heute werden für die Entwicklung der Kanäle vielfach die gleichen Standardkomponenten verwendet, die auch in PCs eingesetzt werden, da diese aufgrund der grossen Stückzahlen ein sehr gutes Preis-Leistungsverhältnis bieten. So spielen PCI Adapter auch in zSeries Rechnern eine wesentliche Rolle.

Dabei wird jedoch der zSeries CEC-Bereich wesentlich stärker von dem PCI-Bereich mit seinen Standardkomponenten abgeschirmt, als dies auf den meisten anderen Rechnersystemen üblich ist. Insbesondere führt der PCI Adapter auf der zSeries keinen direkten DMA Transfer in den zSeries Hauptspeicher durch. Weiterhin werden spezielle Maßnahmen zur Erkennung und Korrektur von Fehlern ergriffen.

Eine wichtige Komponente des Ein-/Ausgabe-Subsystem der zSeries Systeme ist die *Common I/O Card*. Diese Karte bildet die Basis sowohl für den FICON und Fibre Channel Kanal, als auch für Ethernet, ATM und andere schnelle Kommunikationsprotokolle. Die *Common I/O Card* stellt ein PCI bzw. PCI-X Interface zur Verfügung. An dieses kann entweder eine Standard PCI Karte angeschlossen werden, oder aber Komponenten mit integriertem PCI Interface.

Abb. 6 zeigt die *Common I/O Card* in ihrer Verwendung als FICON bzw. Fibre Channel Karte.

Die *Common I/O Card* besteht aus einem PowerPC Prozessor, Memory Controller, lokalem Speicher für Programmcode und Daten, sowie einer STI-PCI-Bridge. Dazu kommt eine protokoll-spezifische Interface Komponente, in diesem Falle ein Interface Controller für Fibre Channel.

Sowohl der PowerPC als auch der Memory Controller sind doppelt vorhanden, jeweils als sogenannter Master und Checker. Alle Operationen der Master-Komponenten werden parallel von den Checkern durchgeführt. Eine Funktionseinheit in der STI-PCI-Bridge vergleicht die Ergebnisse. Bei Nicht-Übereinstimmung wird die Operation abgebrochen. Damit werden insbesondere Adressierungsfehler vermieden, die sonst zu einem Zugriff auf falsche Bereiche im zSeries Hauptspeicher führen würden.

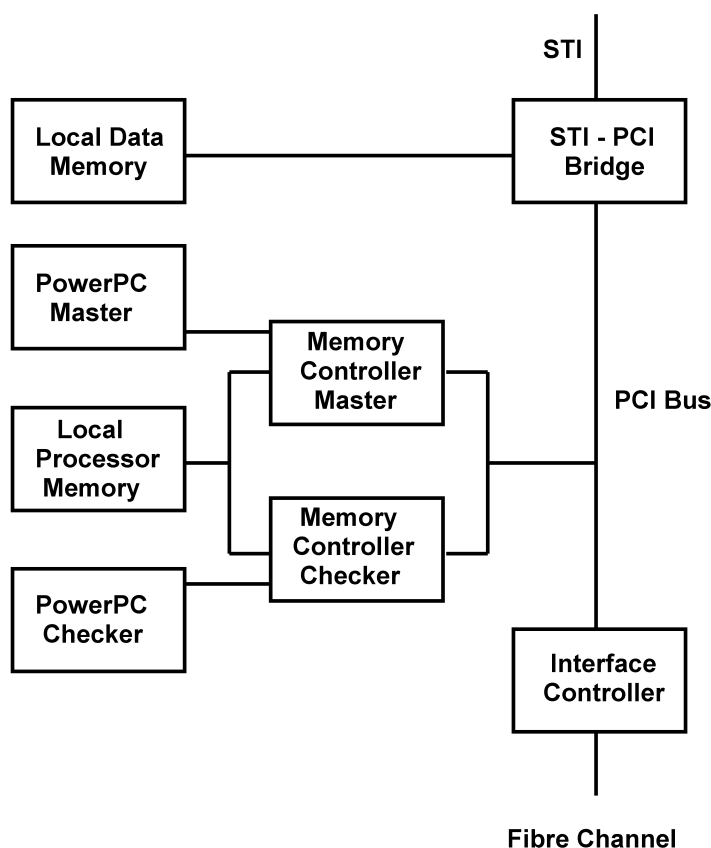


Abb. 6 : zSeries Fibre Channel Kanal, basierend auf der Common I/O Card

Der Transfer von Daten zwischen dem Hauptspeicher der zSeries und dem Ein-/Ausgabemedium (wie z.B. dem Fibre Channel) erfolgt jeweils mit Zwischenpufferung im Speicher der Common I/O Card. Der Transfer zwischen zSeries Hauptspeicher und Puffer in der Common I/O Card wird per DMA von der STI-PCI-Bridge durchgeführt, während der PCI Adapter den DMA Transfer zwischen Puffer in der Common I/O Card und dem externen Übertragungsmedium (jeweils in beide Richtungen) durchführt. Damit ist sichergestellt, dass niemals der PCI Adapter, sondern nur die STI-PCI-Bridge auf den Hauptspeicher der zSeries sowohl lesend als auch schreibend zugreifen kann. Durch interne Schutz- und Prüfmechanismen in der STI-PCI-Bridge werden sowohl Datenübertragungs- als auch Adressierungsfehler ausgeschlossen.

Die *STI-PCI Bridge* spielt somit eine gewisse *Firewall-Rolle*. Der gesamte systemseitige Bereich, also vom zSeries Prozessor bis zur *System Interface Bridge*, ist durch eine Vielzahl von Mechanismen gegen Fehler jeglicher Art geschützt. Wenn Daten über die *System Interface Bridge* diesen Bereich verlassen, um extern gespeichert zu werden, wird ein Cyclic Redundancy Check (CRC) generiert, der mit den Daten zu der externen Control Unit geschickt wird, so dass die Integrität der Daten verifiziert werden kann. Umgekehrt enthalten alle Daten, die von einer Control Unit empfangen werden, einen solchen CRC Wert, der von der STI-PCI-Bridge überprüft wird. Damit ist sichergestellt, dass Übertragungsfehler an einer beliebigen Stelle auf dem Weg von der *System Interface Bridge* zur externen Control Unit oder zurück erkannt werden.

Der CRC Mechanismus garantiert, dass die gelesenen Daten korrekt und konsistent sind. Er garantiert jedoch nicht, dass es auch die richtigen Daten sind. Es könnte ja sein, dass durch einen Adressierungsfehler in der Steuereinheit oder der Elektronik der Platteneinheit selbst Daten von der falsche Stelle auf der Platte gelesen werden. Wenn dabei ein komplettes Datenpaket gelesen wird, hat dieses einen gültigen CRC, so dass dieser Fehler unerkannt bleiben würde. Deshalb wird auf der Platte mit jedem Datenpaket und seinem CRC auch die Adresse diesen Datenpakets auf der Platte abgelegt; dieses wird dann beim Lesen der Daten wieder überprüft, so dass auch dieser Adressierungsfehler abgedeckt wird.

7. Partitionierung

Die Flexibilität der Ein-/Ausgabe Architektur eines zSeries Servers wird mit Hilfe eines Virtualisierungskonzeptes deutlich erhöht.

PR/SM (Processor Resource/System Manager) ist eine weitere LIC-Funktion (Microcode). Sie ist ein Bestandteil der z/Architektur und in den meisten zSeries Rechnern implementiert. Hiermit wird die Partitionierung eines physikalischen Rechners in mehrere logische Rechner ermöglicht, die Logical Partition (LPAR) genannt werden (Abb. 7). Jeder logische Rechner hat sein eigenes Betriebssystem, seinen eigenen unabhängigen realen Hauptspeicherbereich und seine eigenen Kanäle sowie Ein-/Ausgabe Geräte. PR/SM dispatched die einzelnen Betriebssysteme auf den verschiedenen CPUs eines SMP. Jedes Betriebssystem kann nur die Ressourcen benutzen, die für die logische Partition, in der es läuft, definiert sind. Eine gemeinsame Nutzung von Krypto-Koprozessoren und Ein-/Ausgabe Adaptern und Geräten durch mehrere LPARs ist möglich

Das z/VM-Betriebssystem bietet ebenfalls die Möglichkeit des Betriebes von Gastbetriebssystemen. Historisch sind PR/SM und der z/VM-Kernel aus der gleichen Code-Basis entstanden. S/390-Rechner anderer Hersteller verfügen über mit PR/SM vergleichbare Einrichtungen; Beispiele sind Hitachis MLPF oder Amdahls Multiple Domain Facility. IBM hat damit begonnen, auch andere ihrer Rechner-Plattformen mit einer PR/SM-ähnlichen Funktionalität auszurüsten. Für die Intel-Architektur bietet die Firma VMware (www.vmware.com) ein Softwareprodukt mit nicht ganz vergleichbaren Fähigkeiten an.

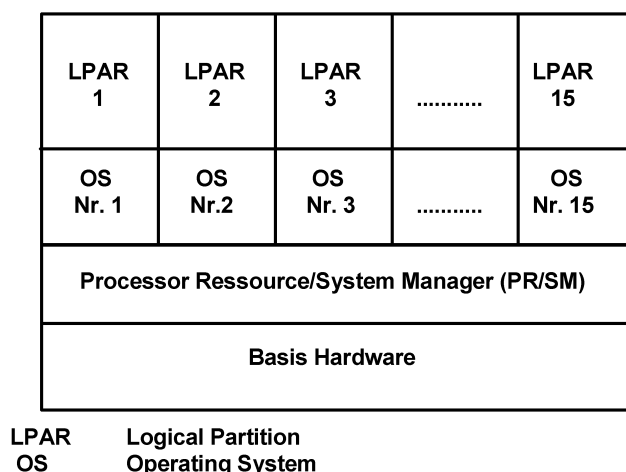


Abb. 7 : PR/SM und LPAR

PR/SM LPARs haben entsprechend einer Zertifizierung die gleichen Sicherheitseigenschaften wie räumlich getrennte Rechner (E4 Zertifizierung).

Die einfachste Stufe der Konfigurations-Flexibilität stellen die rekonfigurierbaren Kanalpfade dar. Definiert man einen Kanalpfad als rekonfigurierbar, so kann er, ohne andere Ein-/Ausgabe Aktivitäten zu stören oder zu unterbrechen, von einer logischen Partition abgegeben und einer anderen logischen Partition zugeordnet werden. Noch größere Flexibilität erreicht man durch die Eigenschaft, dass Kanäle von verschiedenen Partitionen gemeinsam genutzt werden können. Dies bedeutet, dass ein einziger physikalischer Kanal virtualisiert wird, so dass jede Partition, die auf diesen Kanal zugreifen darf, meint, einen eigenständigen Kanalpfad zu dem angegebenen Switch oder der entsprechenden Steuereinheit zu besitzen. Somit laufen bis zu 15 logische Kanalverbindungen über eine einzige physikalische Strecke. Die entsprechende zSeries Einrichtung wird als MIF (Multiple Image Facility) bezeichnet.

Man kann eine Logische Partition (LPAR) so definieren, dass sie folgende Einheiten enthält:

- Ein oder mehrere CPUs (oder aber auch nur Bruchteile einer im *Time Slice*-Verfahren genutzten CPU, capping)
- Central Storage (Hauptspeicher),
- Expanded Storage (optional)
- Kanäle

Es ist außerdem möglich, eine LPAR als Coupling Facility zu definieren [RAH]. LPARs besitzen folgende Eigenschaften:

- Es können maximal 15 LPARs definiert werden.
- Der reale Hauptspeicher für jede LPAR ist isoliert, ebenso der expanded Storage.
- Über die dynamische Speicher-Rekonfiguration kann eine LPAR realen Hauptspeicherplatz abgeben oder erhalten.
- Alle Kanäle können als rekonfigurierbar definiert werden. Kanäle werden den LPARs zugewiesen. Es ist möglich, rekonfigurierbare Kanäle zwischen den LPARs zu bewegen. Die Kanäle können zwischen den LPARs durch Kommandos bewegt werden, ohne das Betriebssystem zu unterbrechen. Die entsprechende Einrichtung wird als *Dynamic Channel Path-Management* bezeichnet. Hierzu können Tasks benutzt werden, die in der z/OS Management Console verfügbar sind.
- MIF (Multiple Image Facility) erlaubt es, Kanäle von 2 oder mehr LPARs zur selben Zeit zu nutzen.
- LPARs können per Definition so viele CPUs besitzen, wie installiert sind. Es ist möglich, CPUs den LPARs zuzuordnen oder diese gemeinsam (shared) zu benutzen. CPUs, die man ausschließlich einem LPAR zuordnet, sind nicht verfügbar für andere aktive LPARs. Die Ressourcen von gemeinsam benutzten CPUs werden aktiven LPARs zugewiesen, wie sie benötigt werden. Man kann CPU-Ressourcen begrenzen, wenn es erforderlich ist.

Der in der z/Architektur implementierte *Intelligent Resource Director (IRD)* ist eine Erweiterung des PR/SM und LPAR Konzeptes. IRD übernimmt die Optimierung der Prozessor- und Kanal-Ressourcen über die verschiedenen Logical Partitions (LPARs). Die drei wichtigsten Funktionen des IRD sind:

- *LPAR CPU-Management*,
- *Dynamic Channel Path-Management* und
- *Channel Subsystem Priority Queuing*.

Die IRD-Funktionen werden vor allem von der Work Load Manager-Komponente des z/OS-Betriebssystems genutzt.

8. Danksagung

Der vorliegende Beitrag entstand aus einem Gemeinschaftsprojekt der beiden Verfasser mit mehreren Mitarbeitern der IBM Laboratorien in Böblingen. Besonders hervorzuheben sind die Beiträge von Herrn Gerhard Banzhaf, Herrn Jürgen Märgner sowie Herrn Thomas Schlipf. Die genannten Herren, zusammen mit Herrn Helge Lehmann, waren maßgeblich an der Entwicklung des z900 Ein-/Ausgabe Systems mit beteiligt.

9. Literatur

[ALD] I. Adlung, G. Banzhaf, W. Eckert, G. Kuch, S. Mueller, and C. Raisch: FCP for the IBM eServer zSeries systems. Access to distributed storage. IBM Journal of Research and Development, Vol. 46, No. 4/5, July/September 2002.

[AMD] G. Amdahl, G. Blaauw, F. Brooks: *Architecture of the IBM System/360*. IBM J. Res. Devel. 8 (2), 87-101 (1964).

[GRE] T. A. Gregg, R. K. Erickson: Coupling I/O channels for the IBM eServer z900. IBM Journal of Research and Development, Vol. 46, No. 4/5, July/September 2002.

[HAR] H. Harrer et al. : First- and second-level packaging for the IBM eServer z900. IBM Journal of Research and Development, Vol. 46, No. 4/5, July/September 2002.

[HEN] J. Hennessy, D. Patterson: *Computer Architecture: A Quantitative Approach*. Third Edition, Morgan-Kaufman, 2002.

[HER] P. Herrmann, U. Keschull, W. G. Spruth: *Einführung in z/OS und OS/390*. Oldenbourg, 2002. ISBN 3-486-27214-4.

[HOK] J. M. Hoke et al : Self-timed interface of the input/output subsystem of the IBM eServer z900. IBM Journal of Research and Development, Vol. 46, No. 4/5, July/September 2002.

[ISJ] Sonderheft des IBM Systems Journal zum Thema Sysplex, Vol. 36, No.2, 1997.

[JRD] Sonderheft des IBM Journal of Research and Development zum Thema Sysplex, Vol. 36, No.4, 1992.

[PFI] G. F. Pfister : *In Search of Clusters, Seite 469-470*. Prentice Hall 1998, ISBN: 0138997098.

[PRI] *z/Architecture Principles of Operation*. IBM Form No. SA22-7832-00.

[RAH] E. Rahm, W. G. Spruth: *Sysplex Cluster-Technologien für Hochleistungsdatenbanken*. Datenbank Spektrum, Heft 3, 2002, S. 16.

[SPR] W. G. Spruth: *The Design of a Microprocessor*. Springer, 1988. ISBN 3-540-51395-7.

[WI1] T.A. Wise: *I.B.M.'s \$5,000,000,000 Gamble*. Fortune Sept. 1966, p. 118. Eine Kopie ist unter <http://jedi.informatik.uni-leipzig.de> zu finden.

[WI2] T.A. Wise: *The Rocky Road to the Market Place*. Fortune Oct. 1966, p. 138. Eine Kopie ist unter <http://jedi.informatik.uni-leipzig.de> zu finden.

[WW1] <http://www-1.ibm.com/servers/eserver/zseries/zos/wlm/documents/velocity/velocity.html>. Eine Kopie ist unter <http://jedi.informatik.uni-leipzig.de> zu finden.