

2. S/390-Ein-/Ausgabe-Organisation

2.1 Ein-/Ausgabe-Performance

Das Leistungsverhalten einer Rechnerarchitektur wird in der Regel von drei Komponenten bestimmt:

- Verarbeitungsleistung der CPU
- Größe des Hauptspeichers und Effektivität der Hauptspeicher-Ansteuerung
- Ein-/Ausgabe-Organisation

Die Ein- und Ausgabe eines Rechners bildet die Schnittstelle nach außen, d.h. sie verknüpft periphere Geräte (Tastaturen, Bildschirme, Drucker usw.) mit der Zentraleinheit und implementiert die wichtige Komponente, die in der Regel unsichtbar ist. Durch spezifische Hardware-Einrichtungen (Direct Memory Access, Ein-/Ausgabe-Prozessor) wird die CPU mittels direkter Datenübertragung zwischen Hauptspeicher und Ein-/Ausgabe-Einheit entlastet. Voraussetzung dafür bildet eine optimale Ansteuerung der Plattenspeicher, die in der Speicherhierarchie einer Rechnerarchitektur die Ressource mit der größten Zugriffszeit darstellt.

2.2 Plattenspeicher-Ansteuerung

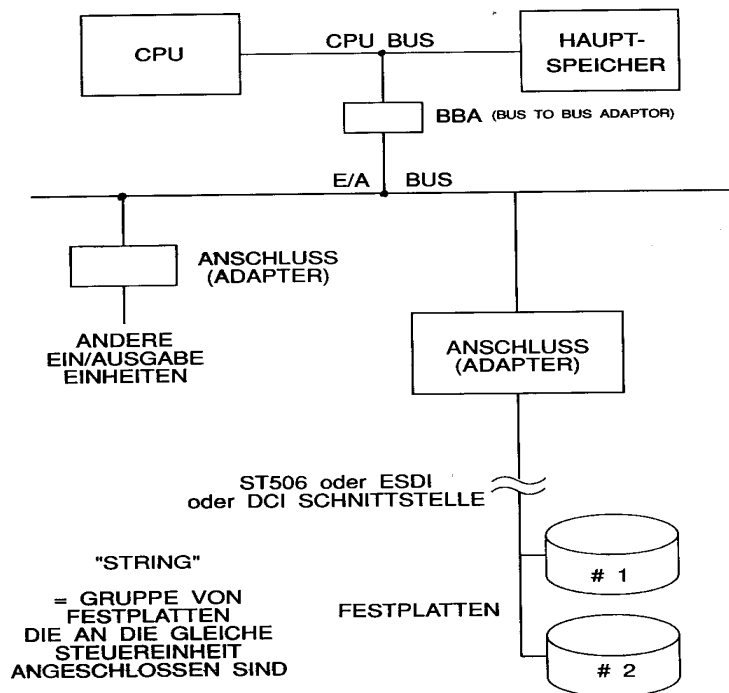
In Bezug auf die Anschlussmöglichkeiten der Plattenspeicher an den Ein-/Ausgabe-Bus bzw. über den CPU-Bus an die Zentraleinheit und den Hauptspeicher existieren zwei verschiedene Alternativen. Die erste verwendet einen Anschluss-Adapter, der über eine spezielle Schnittstelle ein oder mehrere Plattenspeicher mit dem Ein-/Ausgabe-Bus verbindet (Abbildung 1). Der Anschluss-Adapter kann aus einfacher passiver Logik bestehen (Floppy-Controller NEC PD 765 für PC's). Speziell für Plattenspeicher werden in dieser Funktionseinheit leistungsfähige RISC-Mikroprozessoren (E/A-Prozessor) zur Entlastung der CPU eingesetzt. Die Schnittstellen ST506 (Seagate Type 506) mit 5 MBit/s, ESDI (Enhanced System Device Interface) mit 10 MBit/s und DCI (Direct Controller Interface) mit 4.2 MByte/s sind inzwischen durch wesentlich leistungsfähigere ersetzt worden. Die Elektronik-Funktionen für die Plattenspeicher-Ansteuerung sind auf den Plattenspeicher selbst und auf den zugehörigen Anschluss-Adapter verteilt.

Zu den Aufgaben der Platten-Elektronik gehören:

- Umsetzen der analogen Lese-/Schreibsignale in Bit-Folgen
- Suchen des Spuranfang-Signals
- Steuerung des Zugriffmechanismus
- Selektieren des Lese-/Schreibkopfes

Die Elektronik des Anschluss-Adapters muss folgende Funktionen erfüllen:

- E/A-Befehle ausführen, z.B. SEEK, SEARCH, READ, WRITE (SEEK adressiert eine Spur, SEARCH sucht einen bestimmten Block einer Spur)
- Fehlerprüfung
- Fehlerkorrektur
- E/A-Befehlswiederholung
- Statusinformationen sammeln und an die CPU weiterleiten
- Unterbrechungssignale generieren und der CPU übergeben
- Eine von mehreren Festplatten selektieren
- Daten weiterleiten



PLATTENSPEICHERANSCHLUSS 1

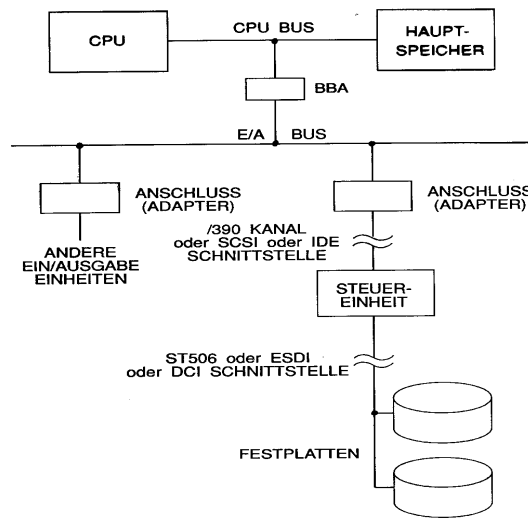
Abbildung 1: Plattenspeicheranschluss 1

In moderneren Rechnerarchitekturen wird der Anschluss-Adapter in zwei Funktionseinheiten aufgeteilt, d.h. von dem eigentlichen Anschluss-Adapter wird die Steuereinheit abgetrennt (Abbildung 2). Als Schnittstellen für diese Implementierung zwischen Adapter und Steuereinheit dienen neben der relativ einfachen IDE-Schnittstelle die /390 Kanal-Interface (OEMI, Original Equipment Manufacturer Interface)- und die SCSI (Small Computer System Interface)-Schnittstelle.

Die OEMI-Schnittstelle realisiert eine optimierte kanalgesteuerte Ein-/Ausgabe für IBM /390-Großrechner. Der Kanal (Anschluss-Adapter) und die Steuereinheit übernehmen gemeinsam die Rolle des Ein-/Ausgabe-Prozessors. Jede Steuereinheit kann unabhängig ein Ein-/Ausgabe-Steuerprogramm ausführen. Der IBM ES/9000 Großrechner gestattet den Anschluss von 256 Kanälen. Damit können 256 Plattenspeicher gleichzeitig Daten vom oder zum Hauptspeicher transportieren. Die OEMI- unterscheidet sich von der SCSI-Schnittstelle dadurch, dass die Länge des Kabels zwischen dem Anschluss-Adapter und der Steuereinheit bis zu 10 Meter betragen kann, während diese im Fall der SCSI-Schnittstelle auf 1-2 Meter begrenzt ist. Bei dem OEMI-Kabel wird bezüglich Abschirmung und Steckverbinder ein sehr hoher kostenintensiver Hardware-Aufwand betrieben.

Für die SCSI-Schnittstelle existiert inzwischen der SCSI III-Standard, der aus den Standards I und II hervorgegangen ist und sich wie auch im Fall der OEMI-Schnittstelle in Richtung einer seriellen Schnittstelle entwickelt. Beide Schnittstellen realisieren einen Datenbus von ursprünglich 8 Bit Breite. Moderne Implementierungen verfügen über 16 bzw. 32 Bit. Hinzu kommen noch eine Reihe von Steuersignalen, die sowohl uni- als auch bidirektionale Funktionen bilden. Die Terminologie für verschiedenen SCSI-Bus-Modi ist etwas verwirrend. So unterscheidet man zwei synchrone (5,10 MHz) Modi und einen asynchronen Modus mit unterschiedlichen Datenraten (Fast SCSI für 32 Bit: 40 MByte/s). Weiterhin arbeitet man zur Zeit mit einem Ultra SCSI-Verfahren, das eine Datenübertragungsrate bei 8 Bit und 16 Bit von 20 MByte/s bzw. 40 MByte/s gestattet. Die Anzahl der SCSI-Bus-Leitungen beträgt abhängig davon, ob es sich beim Anschluss um ein externes oder internes Gerät handelt, 25 bzw. 50. Beim Anschluss eines internen Gerätes wechseln sich Signal- und Masseleitungen ab, so dass zu den insgesamt 25 Signalleitungen (9 Daten-, 6 Masse-, 9 Steuer-Leitungen, 1 Leerleitung) noch weitere 25 Masseleitungen hinzukommen. Der SCSI-Bus befindet sich zu jeder Zeit in einem von 4 verschiedenen Zuständen. In der Bus Free-Phase kann sich einer von insgesamt 8 Teilnehmern darum bemühen, die Verfügungsgewalt über den Bus zu erhalten. Für den Fall, dass sich mehrere Teilnehmer darum bewerben, sorgt die Arbitrations-Phase über eine Prioritätensteuerung für eine Auswahl. In der Selektions-Phase

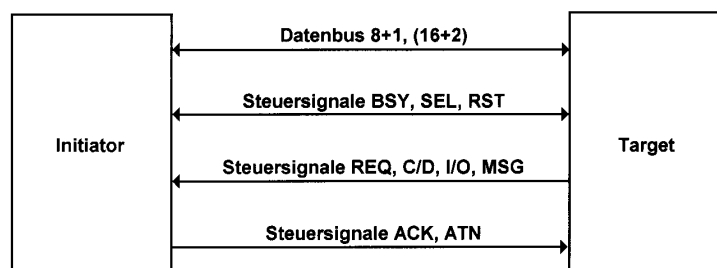
erhält ein Bewerber die Verfügungsgewalt über den SCSI-Bus und während der Information-Transfer-Phase erfolgt die eigentliche Datenübertragung zwischen zwei Partnern über den Bus.



PLATTENSPEICHERANSCHLUSS 2

Abbildung 2: Plattenspeicheranschluss 2

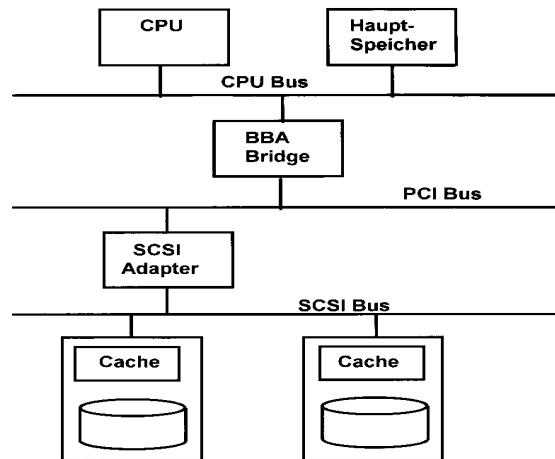
Die Kommunikation zwischen Initiator und Target beim SCSI-Bus zeigt die Abbildung 3. Um das Leistungsverhalten eines Plattenspeichers zu erhöhen, wird neben der Elektronik für die Steuereinheit ein Cash auf der Platte untergebracht. Die Performance von Servern hängt aber weniger von der CPU-Leistung als vielmehr von dem Ein-/Ausgabeverhalten ab.



SCSI BUS

Abbildung 3: Daten- und Steuerleitungen beim SCSI-Bus

In modernen Rechner-Architekturen erfolgt die Kommunikation zwischen CPU bzw. Hauptspeicher und Plattenspeicher über eine dreistufige Busstruktur (Abbildung 4). Neben einem leistungsfähigen CPU- und PCI-Bus ist der SCSI-Controller in der Lage, mehrere SCSI-Platten so zu steuern, dass er z.B. ein Lese-Kommando an die Platten-Elektronik einer SCSI-Platte1 schickt (Connect). Wenn diese noch nicht bereit ist, die Daten zu senden, dann wird sie vom SCSI-Bus logisch getrennt (Disconnect) und die Platte2 mit dem Bus verbunden. In der Zwischenzeit werden die Daten der Platte1 in ihren Cache gelesen. Wenn Platte2 auch nicht in der Lage ist, die gewünschten Daten auf den SCSI-Bus zu legen, dann wird die Platte1 wieder mit dem SCSI-Bus verbunden (Reconnect). Anschließend erfolgt die Übertragung der Daten vom Platten1-Cache zum Hauptspeicher.



Disconnect/Reconnect beim SCSI Controller

1. SCSI Controller gibt Lese Kommando an Platten Elektronik weiter
2. Platten Elektronik gibt SCSI Bus wieder frei
3. Lesen der Daten in den Cache
4. Übertragung Cache-Hauptspeicher

Abbildung 4: 3-stufige Busstruktur

Der Unterschied zwischen einem Unix- und einem /390-Rechner bezüglich der Steuereinheit besteht darin, dass prinzipiell in der /390-Architektur die Steuereinheit als eine separate Einheit implementiert ist. Unix- und /390-Rechner können mehrere Steuereinheiten enthalten. In der /390-Architektur heißt ein solcher Adapter "Kanal" (Channel). Mehrere parallele Kanäle (maximal 256) sind zu einem sogenannten Kanal-Subsystem zusammengefasst und machen die enorme Ein-/Ausgabe-Leistung der /390-Architekturen aus. Die P/390-Rechner gehören zu den kleinen /390-Architekturen, die ihre Plattenspeicher über eine echte SCSI- und nicht über die /390-Kanal-Schnittstelle anschließen. In diesem Fall werden die Kosten für ein solches /390-System entsprechend reduziert. Eine typische S/390-Ein-/Ausgabe-Konfiguration ist in der Abbildung 5 dargestellt. Jede Ein-/Ausgabe-Einheit wird über die spezifische Steuereinheit und dem zugehörigen Channel Path mit dem Channel-Subsystem mit dem Systembus bzw. CPU und Hauptspeicher verbunden.

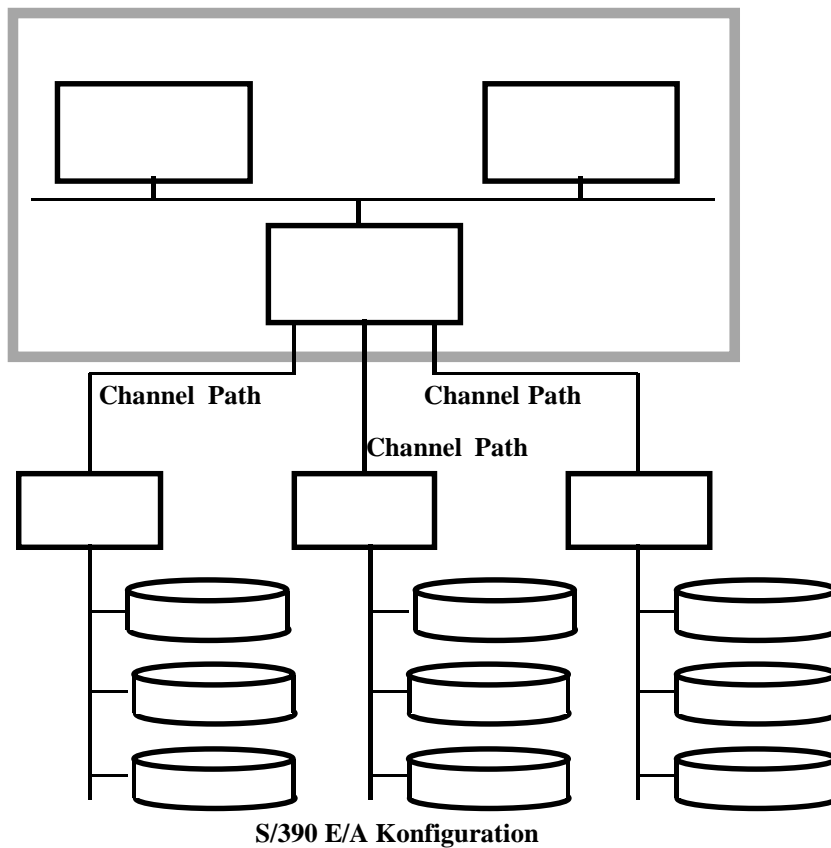


Abbildung 5: S/390-Ein-/Ausgabe-Konfiguration

Als Channel Path wird sowohl die physikalische als auch die logische Verbindung zwischen einem Prozessor bzw. seinem Channel Subsystem und einer Control Unit bezeichnet.

Es existieren zwei Channel Path-Typen:

- Das (ältere) "Parallel I/O Interface" überbrückt Distanzen bis zu 130 m. Sie hat Ähnlichkeit mit dem SCSI-Interface und wird mit einem parallelen Kupferkabel implementiert.
- Das "Serial I/O Interface" verwendet Glasfaser, und erlaubt Datenraten von 18 ("Escon") oder 100 ("Ficon") MByte/s. Es können Entfernungen bis zu 43 km überbrückt werden.

Eine Control Unit kann an mehrere Prozessoren angeschlossen und ein Prozessor über mehrere Channel Path mit einer Control Unit verbunden werden. Die Kopplung eines Ein-/Ausgabe-Gerätes mit mehr als einer Control Unit ist möglich.

2.3 Ein-/Ausgabe-Befehle

Ein- und Ausgabe-Befehle können in drei unterschiedliche Gruppen gegliedert werden:

- Befehle zur Steuerung von E/A-Geräten (Control), z.B. Band zurückspulen, Drucker Zeilenvorschub
- Befehle zum Abfragen des Zustandes eines E/A-Gerätes (Sense), z.B. Positionieren des Plattenspeicher-Zugriffsarms
- Befehle zur Datenübertragung, Read: E/A-Daten → Hauptspeicher, Write: Hauptspeicher-Daten → E/A-Gerät

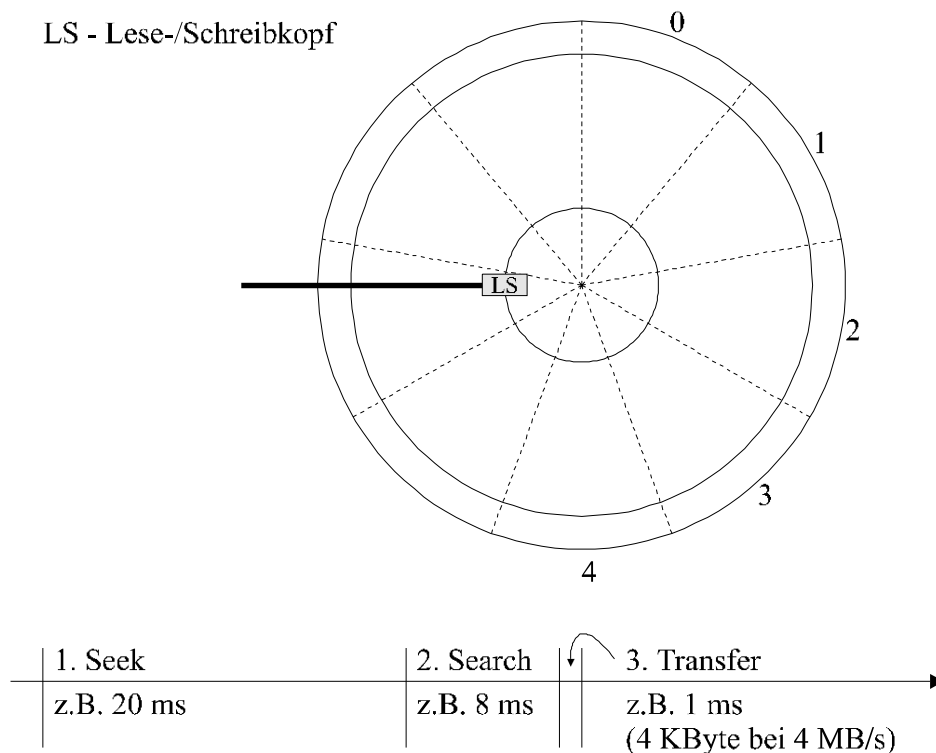
Für die eigentliche Daten-Ein-/Ausgabe ist es notwendig, dass jeder Ein-/Ausgabe-Adapter und jede Steuereinheit über eine bestimmte Anzahl von Registern verfügt. Zu diesen gehören ein Datenregister zur Ablage von Daten zwischen dem Ein-/Ausgabegerät und der Zentraleinheit bzw. dem Hauptspeicher sowie ein

Adressenregister, das den Hauptspeicher adressiert. Weiterhin gehören Steuer- und Statusregister dazu, wobei letztere von der CPU im Fehlerfall abgefragt werden. Da es sich im Normalfall um mehr als eine Ein-/Ausgabeeinheit handelt, die diese Register bei Bedarf benutzen wollen, wird die Register-Adressierung einheitlich in einem Ein-/Ausgabe-Adressraum zusammengefaßt. Dieser Adressraum liegt entweder innerhalb oder außerhalb des Hauptspeicher-Adressraums. Der Ein-/Ausgabe-Adressraum ist nicht durch einen Silizium-Speicher implementiert, sondern es handelt sich dabei um fest verdrahtete Adressen, die zu den Registern in den Adaptoren führen. Ein Beispiel für ein einfaches Plattenspeicher-Steuerprogramm (Kanalprogramm, ESA /390) wird durch folgenden Code dargestellt:

SEEK	Richtige Spur (Zylinder) finden
SEARCH	Richtigen Datensatz auf der Spur finden
TIC	(Transfer In Channel) nochmals versuchen
READ oder WRITE	Daten übertragen

Die diesem Programm entsprechenden Aktionen sind in der Abbildung 6 gezeigt.

LS - Lese-/Schreibkopf



1. LS Auf gesuchte Spur bringen
2. Warten, bis gesuchter Sektor am LS ankommt
3. Daten übertragen

Abbildung 6: Plattenspeicher Lese-/Schreiboperation

2.4 Arten der Ein-/Ausgabe

Die Zentraleinheit einer Rechnerarchitektur ist u. a. für den Datentransport zwischen Haupt- und Plattenspeicher verantwortlich. Es stellt sich dabei die Frage, auf welche Art und Weise die Zentraleinheit sicherstellt, dass dieser Datentransport zwischen Platten- und Hauptspeicher bzw. CPU problemlos und in möglichst kurzer Zeit durchgeführt werden kann.

Es existieren drei unterschiedliche Arten der Ein-/Ausgabe:

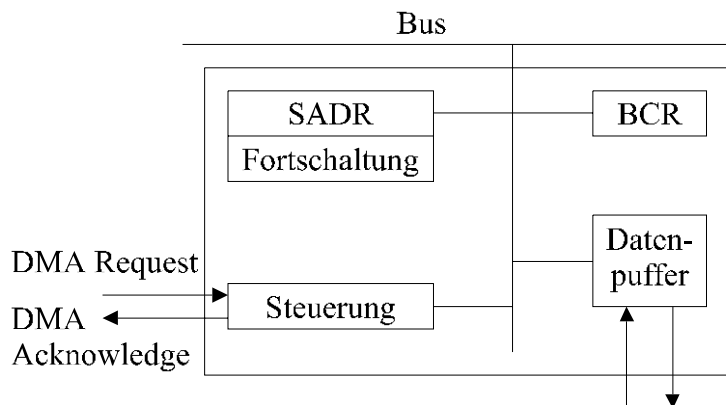
- Programmierte Ein-/Ausgabe
- Unterbrechungsgesteuerte Ein-/Ausgabe
- Kanalgesteuerte Ein-/Ausgabe

Während bei der 'programmierten Ein-/Ausgabe' die gesamte Steuerung von der CPU ausgeführt werden muss, übernimmt im Fall der 'unterbrechungsgesteuerten' und der 'kanalgesteuerten Unterbrechung' ein E/A-Prozessor diese Aufgabe.

Die erste Art der Ein-/Ausgabe besteht darin, einen Plattenspeicher mit Hilfe der 'programmierten Ein-/Ausgabe' anzusprechen. Dabei wird z.B. ein ganzer Sektor von 512 Byte von dem Plattenspeicher gelesen und in einem gleich großen Pufferspeicher der Steuereinheit abgelegt. Anschließend führt die CPU eine Routine aus, die Byte für Byte jeweils über einen E/A-Befehl von diesem Puffer in den Hauptspeicher oder umgekehrt schreibt. Es ist also ein E/A-Befehl notwendig, um 1 Byte, Halbwort oder Wort zu übertragen.

Ein weiteres Verfahren der Ein-/Ausgabe bildet die 'unterbrechungsgesteuerte Ein-/Ausgabe'. Bei diesem Verfahren wird durch den E/A-Befehl ein E/A-Gerät, ein Pufferbereich im Hauptspeicher und die Anzahl der zu übertragenden Bytes spezifiziert. Die Steuerlogik (Steuereinheit) des E/A-Gerätes nimmt die Übertragung vom oder zum Hauptspeicher selbständig vor und meldet den Abschluss der E/A-Operation an die CPU über eine E/A-Unterbrechung. Für die Übertragung eines Datenblocks beliebiger Länge wird somit nur ein E/A-Befehl benötigt. Dieses Verfahren wird allgemein als Direct Memory Access (DMA) bezeichnet. Die DMA-Steuerung wird von einem Mikroprozessor (DMA Controller) ausgeführt und hat folgende Aufgaben (Abbildung 7) zu erfüllen:

- Adressieren des Hauptspeichers durch Adressfortschaltung
- Adressieren der Geräteschnittstelle
- Steuerung der Buszugriffe für Lesen oder Schreiben
- Zählen der übertragenen Bytes
- Rückmelden an die CPU



SADR - Speicheradressregister
BCR - Bytezähler

Abbildung 7: Aufgaben der DMA-Steuerung

Die DMA-Einheit konkurriert mit der CPU bezüglich des Hauptspeichers bzw. des Bus-Zugriffs. Dieses Problem wird mit Hilfe der drei unterschiedlichen Betriebs-Modi a) - c) gelöst.

- Der erste Modus wird als 'Transparentes DMA' bezeichnet. Das bedeutet, dass der DMA-Controller darauf wartet, dass die CPU keine Zugriffe zum Hauptspeicher durchführt. Wenn dieser Fall eintritt, dann können diese freien Bus-Zyklen vom DMA-Controller genutzt werden. Daraus ergibt sich, dass der DMA-Controller der CPU untergeordnet ist.
- Der Modus des 'Cycle Stealing' reserviert mittels einer Steuerlogik eine bestimmte Anzahl von Bus-Zyklen für den DMA-Controller. Falls die CPU gerade diese Zyklen in Anspruch nehmen will, muss sie warten.

- c) Der 'Blocktransfer' (Burst Modus) teilt dem DMA-Controller für die Übertragung eines ganzen Datenblocks (evtl. mehrere Blöcke durch Verkettung) den Bus zu, und die CPU hat während dieser Zeit keine Möglichkeit, den Bus zu benutzen.

Welcher Modus konkret implementiert wird, hängt vor allem von den Anwendungen ab. Durch das 'Transparente DMA' wird der Bus optimal ausgelastet, während die Übertragungsraten minimal sind. Dagegen garantiert der 'Blocktransfer' höchste Übertragungsraten und ist deshalb z.B. für Datenbank-Anwendungen besonders geeignet.

Die dritte mögliche Art der Ein-/Ausgabe stellt einen Sonderfall der 'unterbrechungsgesteuerten Ein-/Ausgabe' dar und wird mit 'kanalgesteuerte Ein-/Ausgabe' bezeichnet. Letztere setzt auch eine programmierte (intelligente) Steuereinheit voraus. Diese Steuereinheit (E/A-Prozessor) führt eine Folge von E/A-Befehlen (Kanalprogramm) aus, um komplexe E/A-Operationen durchzuführen. Die Startadresse des Kanalprogramms wird der E/A-Steuereinheit jeweils zu Beginn der E/A-Operation übergeben. Dafür ist ein Befehl der Zentraleinheit erforderlich (START I/O). Anschließend führt die Steuereinheit eine Folge von E/A-Befehlen durch und meldet den Abschluss der E/A-Operation an die Zentraleinheit über eine E/A-Unterbrechung.

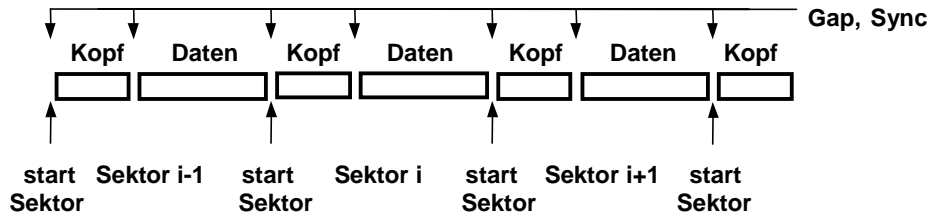
2.5 Datenspeicherung

Um das Zugriffsverhalten von Plattenspeichern so günstig wie möglich zu gestalten, wird bei einer Bewegung des Zugriffarms nicht nur ein Kopf, sondern die Gesamtheit aller Lese-/Schreibköpfe auf eine andere Spur gesetzt. Die übereinander liegenden Spuren eines Plattenspeichers bilden einen sogenannten Zylinder. Von einem Mechanismus, der elektronisch von einem Lese-/Schreibkopf auf einen anderen umschaltet, wird das Selektieren der richtigen Spur übernommen. Dieser Umschalt-Vorgang erfolgt in wesentlich kürzerer Zeit als der Spur-zu-Spur-Wechsel, da keine Bewegung des Armes notwendig ist.

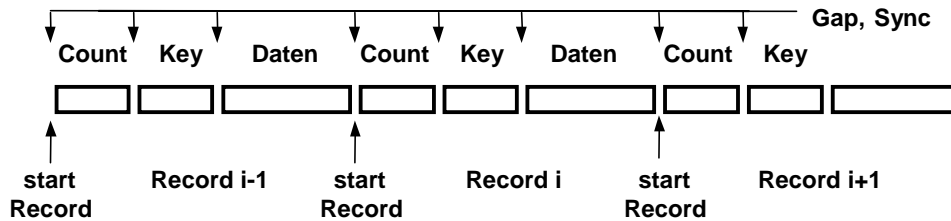
Jede Spur enthält eine bestimmte Anzahl von Sektoren bzw. Blöcken. Die Blockgröße variiert in Abhängigkeit von dem benutzten Betriebssystem. Die Betriebssysteme OS/390, OS/2, Windows 95 und Windows NT arbeiten mit Blockgrößen von 512 Byte, Unix-Betriebssysteme verwenden Blockgrößen von 1024 Byte. Die Sektorgröße ist dagegen konstant (512 Byte). Beim Zugriff auf einen spezifischen Block entsteht eine Zeitverzögerung, die sich aus zwei Komponenten zusammensetzt: Die Verzögerung zum Auffinden der richtigen Spur und die Verzögerung infolge des Suchvorgangs des richtigen Blocks innerhalb dieser Spur. Um die Zeit, die für diesen Vorgang erforderlich ist, zu minimieren, sind hohe Umdrehungszahlen der Plattenspeicher wünschenswert. Bei 3600 Umdrehungen/min für eine 14"-Platte erreicht man auf dem äußeren Rand Bahngeschwindigkeiten (Bahngeschwindigkeit = Winkelgeschwindigkeit * Bahnradius) von annähernd Schallgeschwindigkeit, die störende Turbulenzen des Luftstromes über der Plattenoberfläche hervorrufen. Eine Alternative zur Erhöhung der Platten-Umdrehungszahl bildet die Verringerung des Plattendurchmessers. Damit kann bei konstanter Bahngeschwindigkeit die Umdrehungszahl erhöht werden. Moderne 3.5"-Platten bewegen sich mit 3600-7200 Umdrehungen/min. Der Nachteil, der sich bei großen Umdrehungen bemerkbar macht, besteht in einem relativ hohen Geräuschpegel.

Plattenspeicher setzen sich aus mehreren übereinanderliegenden Aluminium-Scheiben zusammen (in der Regel 10), auf deren Ober- und Unterseite eine ferromagnetische Schicht aufgedampft ist. Aus Sicherheitsgründen werden die obere Fläche der obersten Scheibe und die untere Fläche der untersten Scheibe nicht benutzt. Trotz dieser Maßnahme können die übrigen Plattenoberflächen (insgesamt 18 bei 10 Scheiben) Bereiche enthalten, die nicht gelesen und beschrieben werden können. Um dieses Problem zu lösen, benutzen die Plattenspeicher spezielle Fehlerkorrekturmechanismen. Die Steuereinheit veranlasst z.B. im Fehlerfall, dass ganze Blöcke bzw. Sektoren als fehlerhaft markiert und damit nicht verfügbar sind. Da es sich oft um transiente Fehler handelt, übernimmt die Steuereinheit der /390-Architektur die Aufgabe, dass fehlerhafte Abschnitte wiederholt gelesen oder geschrieben werden (Befehlswiederholung). In diesem Zusammenhang steht auch das sogenannte Platten-"Discrabbing". Darunter versteht man das Fehler-Management der Steuereinheit, wenn letztere beschäftigungslos ist, d.h. es erfolgen nach einem bestimmten Algorithmus Plattenzugriffe. Falls es dabei zu Fehlersituationen kommen sollte, werden die Datenbereiche auf der Platte deaktiviert und auf andere, intakte und nichtbeschriebene Speicherplätze kopiert (Sektorrelocation).

Die Daten auf der Plattenoberfläche werden in zwei unterschiedlichen Formaten abgespeichert: Fixed-Block (FB)- und das Count-Key (CK)-Datenformat. Ihr logischer Aufbau ist in der Abbildung 8 dargestellt.



Fixed Block Record Format



Count, Key, Data (CKD) Format

Abbildung 8: Plattenspeicher-Datenformate

Die Daten im Fixed Block-Format sind in Blöcken identischer Grösse (512 oder 1024 Bytes) untergebracht. Zu jedem eigentlichen Datenfeld gehört noch ein Kopf, der Informationen über den spezifischen Block (Block-, Spur-Nummer etc.) enthält. Neben dem FB-Format ist noch das ältere CKD-Format in Gebrauch. Letzteres ist besonders durch die Felder Count und Key vor dem Datenfeld gekennzeichnet. Da die Datenfelder von einem Block zum nächsten eine sehr unterschiedliche Länge besitzen können und man in der Vergangenheit besonders darauf achtete, keinen Plattenspeicher-Platz zu verschenken, wurde das Count-Feld benutzt, um die Länge des Datenfeldes anzugeben. Trotz der inzwischen veränderten Situation bezüglich der Grösse von Plattenspeichern hat sich das CKD-Format bis heute gehalten, da riesige Datenmengen in diesem Format abgespeichert sind.

Ein Dateisystem verbirgt die Eigenschaften des physikalischen Datenträgers weitestgehend vor dem Anwender. Unix und NT behandeln Dateien als strukturlose Zeichenketten, die mittels Namen identifiziert werden. Dafür dienen Dateiverzeichnisse, die auch wie Dateien aussehen und als solche behandelt werden können. Während Dateien grundsätzlich sequentiell gelesen werden, sind Direktzugriffe in der Regel nur mit Hilfe von Funktionen, die gezielt auf bestimmte Zeichen auf-, vor- oder zurücksetzen, programmierbar.

Im Betriebssystem OS/390 ist das Dateisystem identisch mit dem Inhalt des physikalischen Datenträgers. Bei der Formatierung einer Datei auf dem Plattenspeicher wird die Datei-Struktur festgelegt. Man unterscheidet Formatierungen für Dateien mit direktem (DAM), sequentiellem (SAM) und index-sequentiellem (VSAM) Zugriff. Dateizugriffe unter OS/390 benutzen anstatt des Directory's komplexe Control-Blöcke. Letztere bilden die Datenbasis für eine spezifische Menge von Betriebssystemfunktionen.

Signifikante Unterschiede zwischen den Betriebssystemen OS/390 einerseits und Linux bzw. NT andererseits bestehen auch in der Art und Weise der Dateiverwaltung.

In den Betriebssystemen Linux und NT erfolgt die physikalische Zuordnung der Datei auf dem Plattenspeicher automatisch durch die Angabe der gewünschten Partition (C:, D:, E: usw.). Das Betriebssystem legt fest, wo die Datei innerhalb der ausgewählten Partition abgelegt wird. Die Nachteile dieser dynamischen Speicherplatzverwaltung ergeben sich aus der Fragmentierung und einer nicht-optimalen Speicherplatzzuordnung, die daraus resultiert, dass die komplexe Zugriffsmöglichkeit der Lese-/Schreibköpfe nicht implementiert ist.

Im OS/390-Betriebssystem müssen Dateien bezüglich der Größe und des physikalischen Adressbereichs manuell eingerichtet werden. Dazu gehören u.a. Festlegungen über den Namen der Platte, auf der die Datei angelegt werden soll (Volume Serial) und über die Größe des Datensets (Space Units). Letztere kann z.B. durch die Anzahl der Blöcke, Tracks oder Cylinder angegeben werden. Der Speicherplatz für eine Datei sollte Reserven für ein späteres Wachstum enthalten (Secondary Extends). Weitere Angaben über die Anzahl der Directory-Blöcke, das Record-Format und -Länge sowie der Block-Größe und des Datei-Typs (Library, Partitioned Dataset, Hierarchical File System) sind notwendig. Nachdem diese Vereinbarungen über die Datei getroffen wurden, kann die Datei zugewiesen werden (allocate).

2.6 S/390-Channel-Architektur

Im Unterschied zu anderen Hardware-Architekturen integriert die S/390 eine Funktionseinheit in Form des I/O-Subsystems, das alle Zugriffsmethoden zur Hardware implementiert. Der Zugriffsmechanismus stellt die gesamte Funktionalität, die jeder einzelne Gerätetreiber liefern müsste, zur Verfügung. Die komplexen Funktionen des I/O-Subsystems unterstützen die Gerätetreiber bei der Entscheidung zwischen unterschiedlichen Bustypen, zwischen Polling- und Interrupt-Verarbeitung, shared und non-shared Interrupt-Verarbeitung, DMA und Port I/O (PIO) usw. Da die S/390-Hardware-Plattform über eine große Anzahl unterschiedlicher peripherer Geräte (Plattenspeicher, Bandgeräte, Drucker und Kommunikations-Controller) verfügt, kann auf diese durch eine gut definierte Methode zugegriffen werden. Die Realisierung erfolgt einheitlich über I/O-Interrupts.

Channel-Subsystem (Unterpunkt zu S/390-Channel-Architektur)

Das Channel-Subsystem steuert den Datenfluss zwischen dem Hauptspeicher und den Ein-/Ausgabe-Einheiten. Sämtliche Ein-/Ausgabe-Operationen laufen parallel zu den Arbeiten der CPU ab. Dabei benötigt das Channel-Subsystem Hauptspeicherzyklen nur für den Transfer von Daten zwischen Hauptspeicher und Ein-/Ausgabegerät.

Ein-/Ausgabe-Operationen werden eingeleitet, indem die CPU Ein-/Ausgabe-Maschinenbefehle ausführt. In diesen Befehlen ist der einer Ein-/Ausgabe-Einheit zugeordnete Subchannel angegeben. Z.B. bewirkt der CPU-Befehl "Start Subchannel", dass vom Channel Subsystem selbständig und unabhängig von der CPU Ein-/Ausgabe-Operationen in der Form von "Channel Command Words" (CCW's) ausgeführt werden. In dieser Zeit ist die CPU in der Lage, andere Arbeiten auszuführen.

Subchannel

Ein Subchannel stellt einen bestimmten Speicherbereich dar. Letzterer enthält u.a. die CCW-Adresse sowie Channel Path Identifier. Eine Ein-/Ausgabe-Einheit ist in der Regel ein physikalisch identifizierbares Ein-/Ausgabe-Gerät. Prinzipiell können mehrere Einheiten einem Ein-/Ausgabe-Gerät zugeordnet werden (z.B. zwei Zugriffsarme eines Plattenspeichers implementieren 2 Ein-/Ausgabe-Einheiten mit zwei Subchannel-Adressen). Jeder Subchannel wird durch eine 16Bit-Adresse identifiziert, d.h. ein System kann bis zu 65536 unterschiedliche Ein-/Ausgabe-Einheiten adressieren. Typische System-Installationen benutzen einige hundert bis tausend Einheiten.

Die Steuereinheiten in einer S/390-Architektur werden über Channels an das Channel-Subsystem angeschlossen. Grundsätzlich könne mehrere Channels mit der gleichen Steuereinheit verbunden werden.

Channel Path

Das Channel-Subsystem kommuniziert mit den Ein-/Ausgabe-Einheiten über Verbindungen zwischen dem Channel-Subsystem und den Steuereinheiten. Ein Channel Path bildet die logische Verbindung zwischen Channel-Subsystem und Steuereinheit im Gegensatz zur physikalischen Verbindung in Form des Channels. Einerseits ist es möglich, Steuereinheiten über mehr als einen Channel Path an das Channel-Subsystem anzuschließen, andererseits können Ein-/Ausgabe-Einheiten an mehr als eine Steuereinheit angeschlossen werden. Einzelne Ein-/Ausgabe-Einheiten sind in der Lage, mit dem Channel-Subsystem über maximal acht verschiedene Channel Path's zu kommunizieren. Eine Ein-/Ausgabe-Operation kann z.B. erfolgen, indem sie über einen spezifischen Channel Path initiiert wird, während der anschließende Datentransfer über einen anderen Channel Path erfolgt. Das Channel-Subsystem übernimmt die Verwaltung der mehrfachen Channel Path's.

S/390-Parallel-Sysplex

2.7 Cluster-Architektur

Grundsätzlich besteht ein System (Prozessor, Node) aus mehreren CPUs, die auf einen gemeinsamen Hauptspeicher zugreifen (SMP, Symmetric Multiprocessor). Im Basisfall befindet sich nur eine Kopie (Instanz) des Betriebssystems im Hauptspeicher. In der Abbildung 9 ist ein solches System mit 4 CPUs dargestellt.

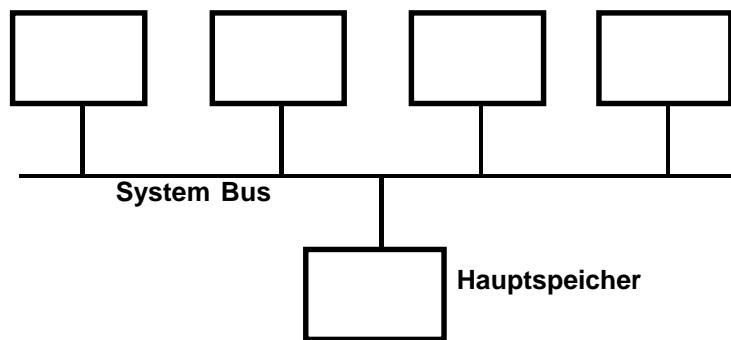


Abbildung 9: System aus 4 CPUs

In einem Cluster werden mehrere Systeme über ein Hochgeschwindigkeitsnetzwerk miteinander verbunden. Das Netzwerk kann als Bus oder (häufiger) als Crossbar-Switch implementiert sein (Abbildung 10).

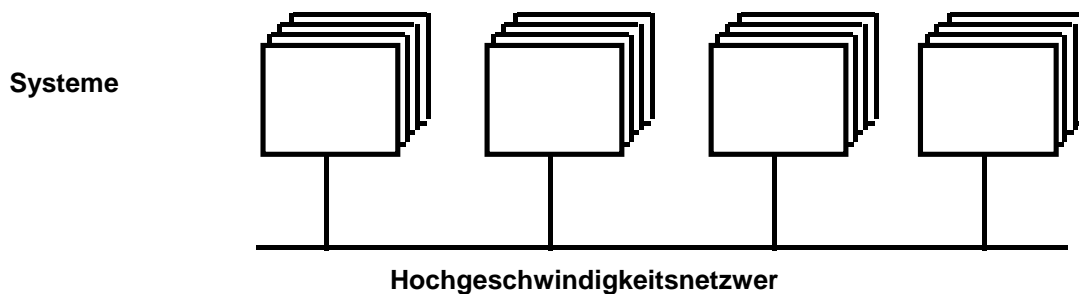


Abbildung 10: Cluster aus 4 Systemen

2.8 Cluster-Typen

Es wird generell zwischen drei unterschiedlichen Typen von Clustern unterschieden: High-Availability Cluster, Parallel Cluster, Single System Image Cluster.

Der High-Availability-Typ stellt ein fehlerfreies Cluster dar, das sich durch eine höhere Verfügbarkeit als einzelne Systeme auszeichnet. Ein spezieller Prozess ("Heartbeat"-Monitoring) überträgt bei einem Fehler auf dem primären Server die Verarbeitung auf einen Backup-Server. Im Normalfall ist der Backup-Server in dem Cluster beschäftigungslos, d.h. er ist jederzeit bereit, im Fehlerfall des primären Servers dessen Arbeit zu übernehmen. Beispiele für High-Availability-Cluster stellen die meisten der augenblicklich verfügbaren Unix-Cluster, der Microsoft Cluster-Server (Wolfpack) sowie der IBM RS/6000-Server mit Hilfe des AIX High Availability Cluster Multiprocessing Package (HACMP) dar.

Mehrere Computer-Hersteller bieten inzwischen den zweiten Cluster-Typ an. Die Parallel Cluster liefern neben der Skalierbarkeit erhöhte Verfügbarkeit durch parallele Middleware und schnellere Verbindungen zwischen den einzelnen Systemen. In diesen Clustern kann die Arbeit auf unterschiedliche Server so aufgeteilt werden, dass sie parallel verarbeitet wird. Dabei verrichtet jeder Server seine ihm übertragene Arbeit und die Ergebnisse werden anschließend zusammengefasst. In den meisten Implementierungen erfolgt zwischen den Servern eine Daten-

Partitionierung und bei Data Request's werden die angeforderten Daten zwischen den Servern ausgetauscht. Die Parallel Cluster verfügen aber wie die SMPs auch über Nachteile. Aus Effizienzgründen müssen Workload und Datenbanken unter den Servern partitioniert werden. Zusätzlich machen sich Eingriffe des Programmierers notwendig. Letztere betreffen insbesondere Speicher-Aufteilung, Leistungs-Feinabstimmung sowie Verfügbarkeit. Falls die Daten nicht sehr sorgfältig über das Cluster verteilt werden, kann ein Server mit Arbeit überladen werden, während ein anderer Server innerhalb des Systems nicht genügend zu tun hat. Dieses unausgewogene Workload führt dazu, dass die Antwort-Zeit des Clusters nicht akzeptabel ist.

Die effektivste Cluster-Form bildet der dritte Typ. Im Single-System Image Cluster erscheinen alle Server zur Client-Applikation als ein einheitliches System. In dieser "Shared Data" Methode hat jeder Server Zugriff auf alle Daten und irgendeine Transaktion kann auf einem beliebigen Server des Systems ablaufen. Die Workload-Balance stellt eine gleichmäßige Arbeitsteilung sicher. Nutzer können eine Anwendung auch dann weiter verwenden, wenn ein Server off-line ist, d.h. die Gesamtleistung aller Server des Clusters wird für eine Applikation annähernd linear skalierbar.

S/390 Enterprise Server können als Stand-alone Systeme, Loosely coupled Systeme oder als Parallel Sysplex Single System Image mit bis zu 32 Systemen arbeiten. Das S/390-Clustering weist Architektur-Merkmale auf, die bei Unix- und NT-Clustern noch nicht verfügbar sind. Die Abbildung 11 zeigt eine S/390 Großsystem-Konfiguration als Single System Image Cluster.

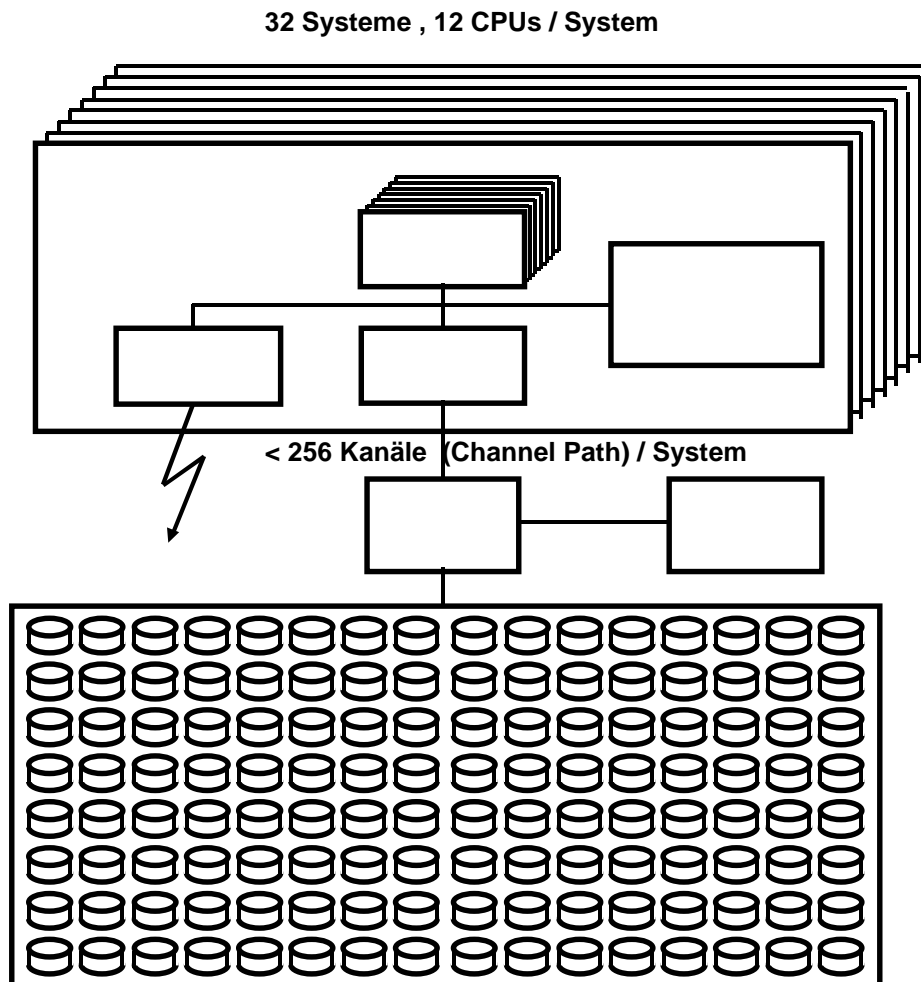


Abbildung 11: S/390 Großsystem-Konfiguration

2.9 Cluster-Architektur-Modelle

Die Klassifizierung von parallelen Systemen kann vorgenommen werden, indem sie mit den folgenden Architektur-Modellen verglichen werden. Es existieren drei verschiedene Cluster-Architektur-Modelle: Shared-Nothing (Data-Partitioning)-Modell, Shared-Disk (Shared-Data)-Modell, Shared-Everything-Modell.

Im Shared-Nothing-Modell besitzt jedes System einen Teil der Datenmenge, und dieser Teil kann nur von dem zugehörigen System gelesen und geschrieben werden. Durch die Daten-Partitionierung ist jedes System in der Lage, auf seine Daten über den Cache zuzugreifen ohne Rücksicht auf Kohärenzkontrolle mit Hilfe entsprechender Cross-System-Kommunikations-Mechanismen. Ein derartiges Modell liefert eine ausgezeichnete Skalierbarkeit. Die Schwierigkeit besteht darin, dass die Verarbeitungs-Kapazität eines Cluster-Knotens mit der vorgesehenen Workload-Zugriffsrates zu den eigenen Daten abgestimmt werden muss. Echtzeit-Workload-Schwankungen können die Prozessor-Ressourcen über- oder unterfordern. Letzteres führt zu einer Leistungsverschlechterung des Clusters

Das Shared-Disk-Modell sieht vor, dass jedes System auf die vollständige Datenmenge, die über die Plattenspeicher des Gesamtsystems verteilt ist, zugreifen kann. Der Vorteil dieses Cluster-Modells liegt in der Möglichkeit, das Workload dynamisch über alle Knoten des Clusters zu verteilen. Als Hauptnachteil ergibt sich aber eine schlechte Skalierbarkeit dieses Modells. In Shared-Data-Konfigurationen werden verteilte Lock-Management-Protokolle zur Steuerung der parallelen Prozesse im Cluster einschließlich des Message Passing zwischen den Systemen benutzt. Das Lock-Management ist notwendig, um zu garantieren, dass nur ein System die Shared-Data-Menge zu einer bestimmten Zeit exklusiv modifiziert. Eine globale Cache-Kohärenz-Steuerung sorgt für identische Daten in lokalen Caches für jedes System.

Außer den Plattenspeichern kann auch der Hauptspeicher mit in das Shared-Modell einbezogen werden. Dieses Verfahren wird bei einem SMP angewendet und bildet das Shared-Everything-Modell. Ein SMP selber implementiert kein Cluster-System, er kann aber als Block-Baustein für einen Cluster-Knoten dienen. Shared-Everything-Architekturen haben Effizienz-Vorteile bei einer relativ kleinen Prozessor-Anzahl. Die Skalierbarkeit nimmt aber mit zunehmender Prozessor-Zahl ab.

2.10S/390 Parallel Sysplex System-Modell

Das System-Modell des S/390 Parallel Sysplex besteht aus den Prozessor-Knoten, Shared Storage Devices, Netzwerk-Kontrollern und den Kern-Cluster-Technologie-Komponenten (Abbildung 12). Letztere umfassen Sysplex-Timer, Switch (ESCON Director) und Coupling Facility (CF).

Parallel Sysplex unterstützt bis zu 32 Prozessor-Knoten. Jeder dieser Knoten stellt einen SMP dar und enthält maximal 16 Prozessoren. Die Knoten müssen nicht homogen sein, d.h. ein Knoten kann aus S/390-CMOS- und ein anderer aus ES/9000-Bipolar-Prozessoren aufgebaut sein. Jeder der Prozessoren ist in der Lage, auf die Daten der Shared Storage Devices zuzugreifen (Shared Data (Disk)-Modell).

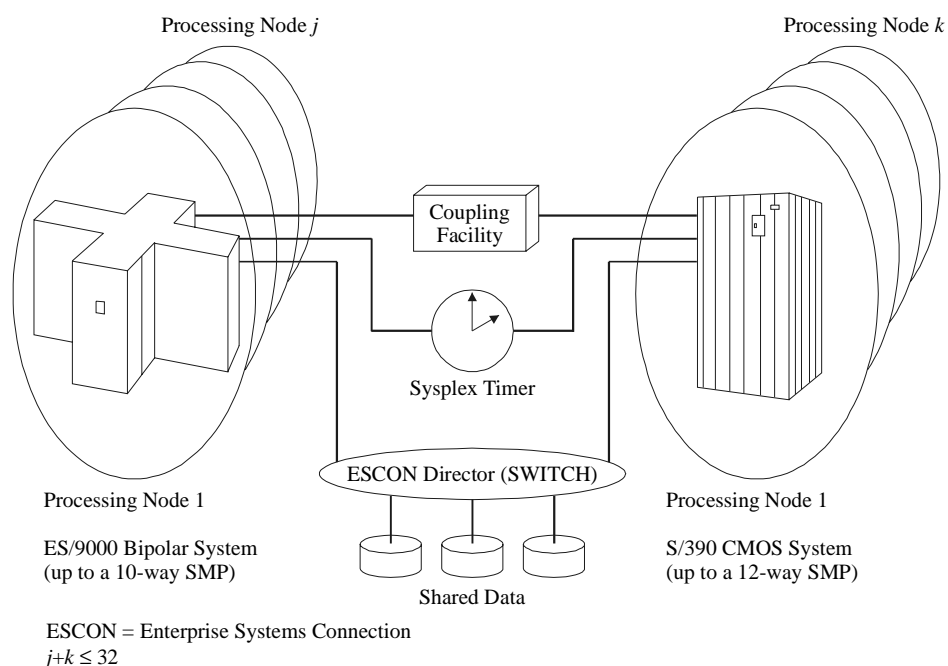


Abbildung 12: Parallel Sysplex System-Modell

Die I/O-Architektur benutzt das von IBM entwickelte ESCON (Enterprise Systems CONnection) Channel-Konzept. Das Channel-Subsystem der ES/9000-Architektur (Abbildung 13) bildet die Grundlage für den ESCON-Channel. Dieses integriert Off-Load-Prozessoren (IOPs), Channels und die sogenannte Staging-Hardware. Die IOPs sind verantwortlich für die Kommunikation zwischen den zentralen Prozessoren und den Work Queues des Channel Subsystems. Die Channels selber führen das Channel-Programm aus: Sie initialisieren das Channel-Programm, führen die Datenübertragung aus und liefern die abschließenden Status-Informationen an die IOPs. Die Staging-Hardware stellt die Kommunikations-Pfade zwischen den IOPs, den Channels und dem Rest des Systems zur Verfügung. Jeder Channel ist über seinen eigenen Channel Attachment Bus mit der Staging-Hardware verbunden. Sämtliche Kommunikationen mit anderen System-Komponenten erfolgen über diesen Bus.

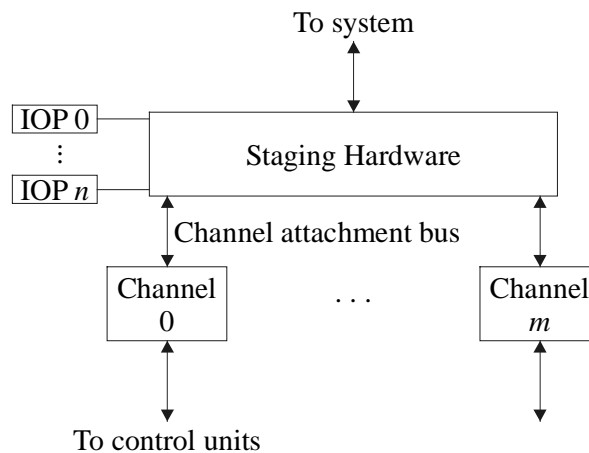


Abbildung 13: Struktur des ES/9000 Channel Subsystems

Von CF werden drei Verhaltensmodelle in Form von Cluster-Protokollen realisiert:

- Lock-Modell: Es unterstützt feingranulares, globales Locking für hohe Performance und Signalisierung von konkurrenten Ressource-Zugriffen.
- Cache-Modell: Liefert globale Kohärenz-Steuerung für verteilte lokale Prozessor-Caches und Shared Data Cache.
- Queue-Modell: Implementiert einen umfangreichen Satz von Queuing-Konstrukten für die Verteilung von Workloads und zur Realisierung von Message Passing sowie Sharing der Status Information.

CF setzt sich physisch aus Hardware und speziellem Mikrocode (Control Code) zusammen. Die Kopplung mit anderen S/390-Prozessoren erfolgt mittels des OS/390- oder MVS-Betriebssystems über High-Speed Coupling Links. Diese Links benutzen spezielle Protokolle für den Transport der Kommandos zum und vom CF. Die Hardware der Links besteht aus Glasfaser-Kanälen mit einer Übertragungsrate von 18 Mbyte/s (ESCON) und 100 MByte/s (FICON). CF-Speicher-Ressourcen können dynamisch partitioniert und einer der CF-Strukturen (Lock-, Cache-, Queue-Modell) zugewiesen werden. Innerhalb derselben CF sind mehrere CF-Strukturen desselben oder unterschiedlichen Typs möglich.

Der ESCON Director bildet die Kerneinheit der ESCON-Architektur. Er implementiert eine Switched Point-To-Point-Topologie für S/390-I/O-Channels und Control Units. Bis zu 60 Channels und Control Units können durch den Director dynamisch und nichtblockierend (Crossbar Switch) über ihre Ports miteinander verschaltet werden. Entfernungen von bis zu 3 km für optische Übertragungen sind möglich, d.h. ein ESCON Director erlaubt Channel-To-Channel- oder Channel-To-Control Unit-Distanzen bis zu 6 km, bei zwei Director-Einheiten bis zu 9 km. Diese zulässigen Entfernungen erhöhen sich bei der Implementierung des Laser-Link-Produkts ESCON XDF (Extended Distance Feature) auf 20, 40 bzw. 60 km. Die Abbildung 14 zeigt eine typische System-Konfiguration mittels ESCON Director.

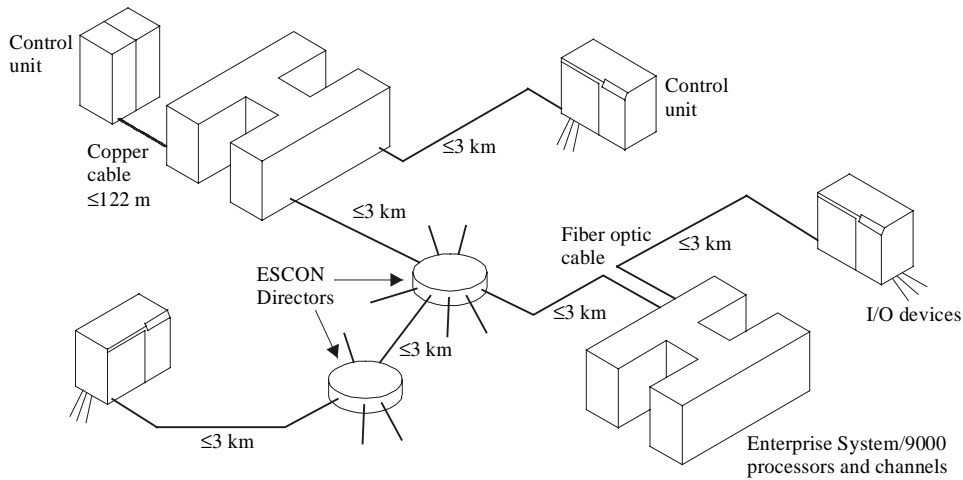


Abbildung 14: System-Verbindung über ESCON Director

Die Anforderungen an ein gekoppeltes Rechnersystem bezüglich Genauigkeit und Konsistenz des Taktes in den einzelnen Teilsystemen werden durch einen Sysplex Timer erfüllt. Letzterer stellt eine externe Zeit-Referenz (ETR, External Time Reference) dar. Der Timer generiert den synchronen Time Of Day (TOD)-Clock für alle Systemknoten bzw. -prozessoren. Von der ETR-Architektur (Abbildung 15) werden 3 Signale (oscillator signal, on-time signal, data signal) für die Clock-Synchronisation erzeugt und an die Central Processing Complexes (CPCs) gesendet. Der Sysplex Timer kann umgekehrt auch Informationen aus einer externen Quelle erhalten (z.B. Time Code Receiver). Im Falle eines Sysplex Timer-Ausfalls arbeiten die Knoten weiter und verwenden ETR aus einem lokalen Modul. Beim Wiederanschluss des Timers besteht je nach ETR-Parameter die Möglichkeit der Resynchronisation.

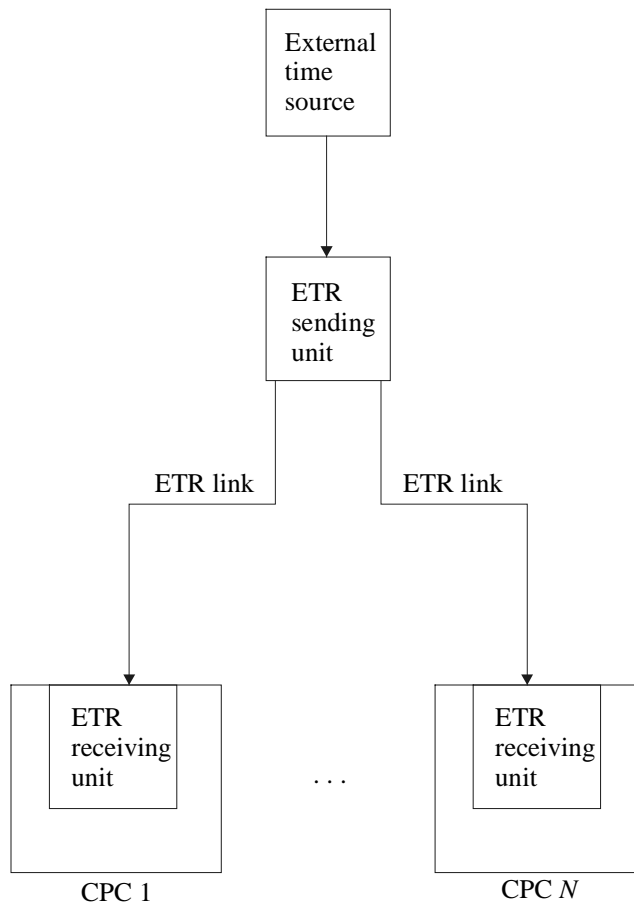


Abbildung 15: ETR-Netzwerk