Kapitel 2

Zur CPU parallel arbeitende Einheiten

2.1. Coprozessoren

Es sind dies zur CPU parallel arbeitende Geräte, die den Befehlsstrom abhören und die Rechnung übernehmen, wenn sie ihnen zugeordnete Befehle entdecken.

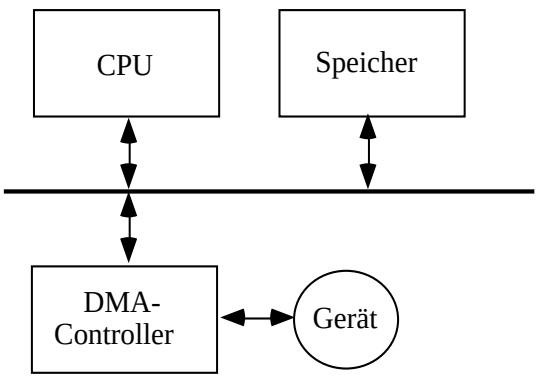
Beipiele: Gleitkommaeinheiten (floating point unit, FPU)

Sie erkennen GK-Befehle und stoßen die zugehörigen Befehlsfolgen an. Häufig arbeiten GK-Befehle auf einem eigenen Satz von GK-Registern. Da GK-Befehle i. allg. in mehreren Schritten abgearbeitet werden, fallen sie aus dem Raster üblicher RISC-Befehle heraus.

2.2. Direkter Speicherzugriff (direct memory access, DMA)

Die CPU initiert den blockweisen Transfer von Daten von und zu einem Gerät (Platte, Drucker, Frame-Grabber). Die CPU muß die Parameter für den Transfer übergeben:

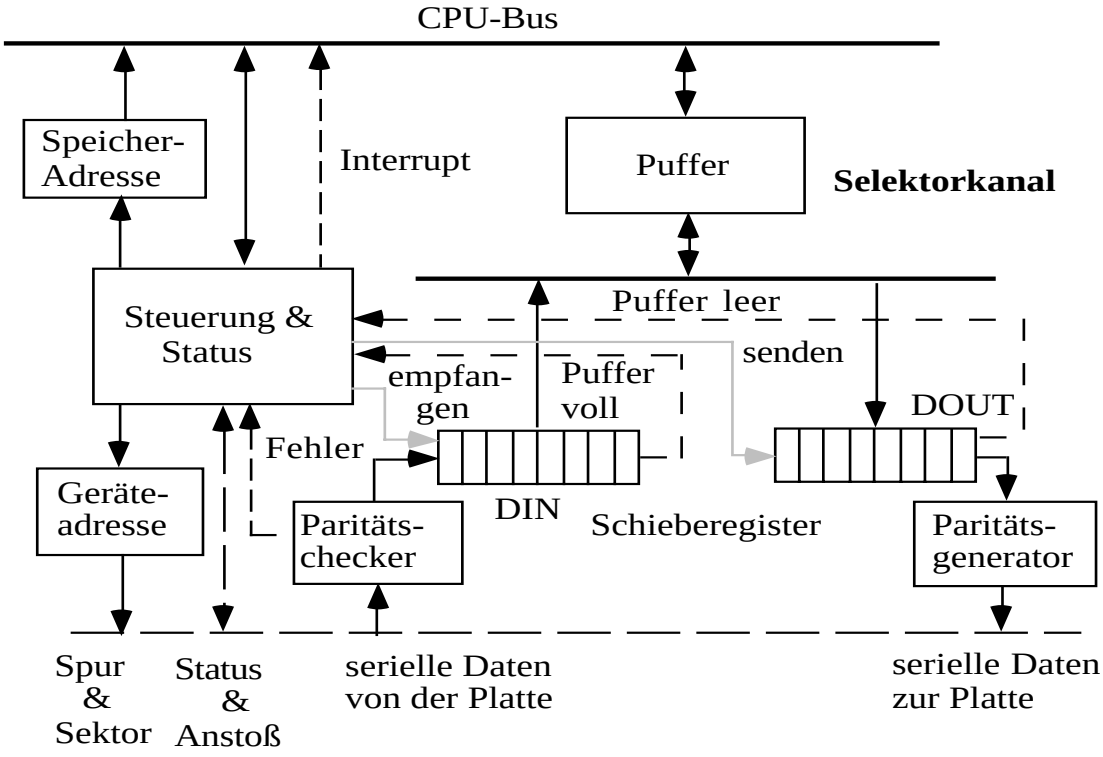
- Anfangsadresse im Speicher
- Blocklänge (Anzahl der zu übertragenden Bytes)
- beim Transfer auf eine Platte die Anfangsadresse (Spur und Sektor) auf der Platte
- die Übertragungsrichtung beim Ansprechen einer Platte (bei Drucker oder Frame-Grabber ist die Richtung vorgegeben).



Im DMA-Controller ist ein Statusregister, das von der CPU gelesen werden kann und Informationen über den Status des Geräts und den erfolgreichen/fehlerhaften Transfer enthält. Da die Zeit für die Übertragung eines Blocks (Plattensektor, Videobild) lang ist gegen einen Takt, meldet sich der DMA-Controller mit einem Interrupt zurück, wenn er fertig ist. Im Statuswort kann die CPU erkennen, ob der Transfer noch läuft oder beendet ist mit/ohne Fehler.

Ein interner Puffer gleicht Unterschiede in den Transfergeschwindigkeiten aus: die Platte gibt durch ihre Drehung die Zeiten vor, in denen Bytes gelesen/geschrieben werden und die Kamera liest die Bilder mit ihrer eigenen Taktfrequenz aus.

Für einen Platten-Controller (selector channel) sind einige Gerätefunktionen noch einmal separat dargestellt:



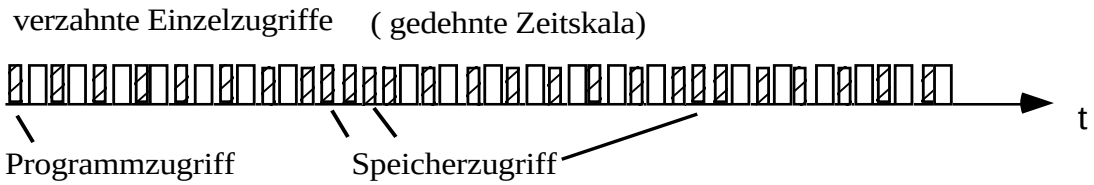
Nach einem Anstoß führt die DMA-Einheit selbständig aus: (hier Ausgabe zum Gerät)

- Busanforderung
- Adressierung des Speichers
- Datentransfer von Speicher mit Synchronisierung
- Zwischenpufferung
- Transfer zum Gerät (vom Gerät her gesteuert)
- Adressfortschaltung
- Rückmeldung via Interrupt bei Fehlern oder am Ende der Übertragung

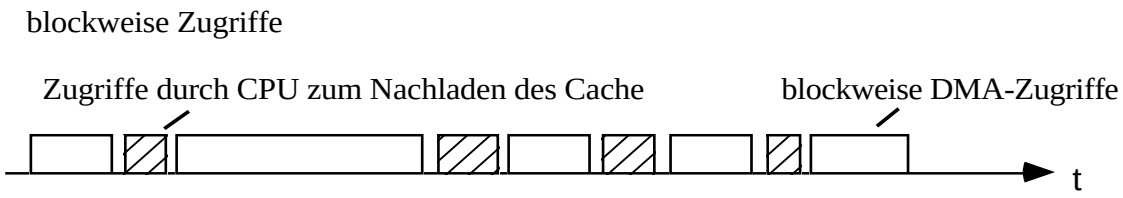
Geräte mit selbständigem Zugriff auf den Speicher können auch Multiplexerkanäle sein: viele langsame Geräte hängen an einem internen Puffer und werden blockweise in den Speicher übertragen. Man findet das bei großen Rechnern mit sehr vielen bitseriellen Ein-Ausgabegeräten (veraltet).

Der Zugriff vom DMA-Controller auf den Speicher tritt in Konkurrenz mit den Zugriffen der CPU beim Lesen von Befehlen und Lesen/Schreiben von Daten im Speicher.

Die CPU hat Vorrang beim Zugriff auf den Speicher, da die Daten in der DMA gepuffert werden. Man erreicht dann einen **verzahnten Zugriff** auf den Speicher: in den Zeitintervallen, in denen die CPU rechnet, kann die DMA auf den Speicher zugreifen.



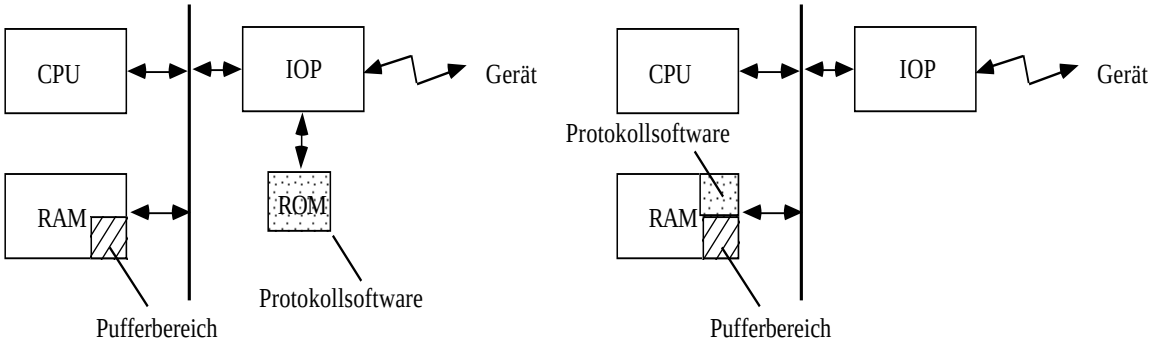
Hat die CPU einen Daten- und Programmcache, dann kann der Zugriff blockweise organisiert sein:
Solange die CPU eine Seite aus dem Speicher nachlädt oder zurückschreibt, wird der Buszugriff gesperrt. Sonst kann die DMA Daten am Stück lesen/schreiben.



2.3. Ein-Ausgabe-Prozessoren (input-output-processor, IOP)

Zur CPU arbeiten parallel Geräte, die permanent mit eigenen Programmen E/A-Vorgänge abwickeln. Das Programm eines IOP kann im Speicher oder in dem IOP selber liegen.

Die CPU gibt Pufferbereiche im Speicher vor, aus denen Daten ausgelesen bzw. in die eingeschrieben werden soll.



2.3.1. LAN-Controller

Dieser Ein-Ausgabeprozessor bedient ein lokales Netzwerk, z. B. ein Ethernet.

Die CPU gibt für das Versenden eines Paketes die Adresse vor und den Zeiger auf den Pufferbereich, in dem der Text als File steht. Der LAN-Controller wickelt dann selbständig das Protokoll ab: Verpacken des Textes in Paketen nach HDLC-Norm, Einfügen der LAN-Adresse, Absenden und Überwachen des Transfers.

Alles, was die CPU erfährt ist am Ende des Transfers ein Interrupt durch den LAN-Controller und sie kann in einem Statuswort Informationen über den Ablauf des Transfers einsehen.

Der Empfang eines Paketes wird durch den LAN-Controller organisiert: Er horcht die Leitung ab, empfängt Pakete, die für ihn bestimmt sind, packt sie aus und transferiert sie in einen von der CPU angegebenen Pufferbereich. Danach alertiert er über einen Interrupt die CPU, die in einem Statuswort Informationen über den Transfer nachlesen kann.

Beispiel: 82586 Local Communication Controller
Ein LAN-Controller für Intel 80x86-Rechner.
Der Controller hat sein Programm im Speicher.
Er verwaltet das Ethernet-Protokoll TCP/IP.
Er kennt eine Reihe von Aktions-Befehlen:

- NOP	:	tue nichts
- set up individual address	:	Laden der Adresse des 82586 als Ethernet-Partner
- configure	:	Laden von Operationsparametern mit Default-Wert "Ethernet"
- set up multicast address	:	Multicast Adressen laden als Ethernet-Partner
- transmit	:	Senden eines einzelnen Frames
- TDR	:	time-domain reflectometry Ausmessen der Reflexionen auf dem Koaxialkabel des Ethernet über die Laufzeit kurzer Impulse
- diagnose	:	ein Diagnoseprogramm ablaufen lassen
- dump	:	Umladen der internen Register des 82586 in den Speicher

Der Communication Controller interpretiert die Kommandos, die von der CPU im Speicher in einer Kommandoblockliste (command block list, CBL) abgelegt werden.

Daneben gibt es einen Systemkontrollblock (SCB), der die logische Kommandoschnittstelle zur CPU bildet, und einen Empfangsrahmenbereich (receive frame area, RFA), in dem ein empfangener Rahmen abgelegt wird.

Für zu sendende Daten muß ein Pufferbereich angegeben werden, aus dem der Controller die Daten holt.

2.3.2. Einfache IOP's

Im einfachsten Fall sind Ein-Ausgabeprozessoren zu einfachen Interfaces verkümmert: Zähler/Timer, serielle I/O, parallele I/O.

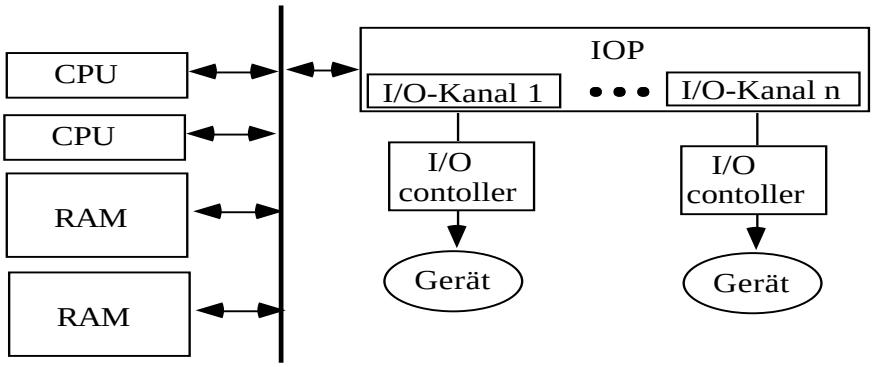
An der Grenze stehen Buscontroller für genormte Busse: z. B. für den GPIB-Bus (general purpose interface bus). Sie wickeln das Protokoll für die Datenübertragung auf dem GPIB-Bus ab und der Rechner kann sich aus Sicht des Busses verhalten als Controller oder Talker oder Listener und "versteht" die genormten Befehle auf dem Bus.

2.3.3. IOP's an großen Rechnern, Kanalwerke

An größeren Anlagen müssen ggf. viele (schnelle) Geräte versorgt werden.

Das wird zum Teil übernommen von E/A-Prozessoren mit mehreren E/A-Kanälen, an denen über I/O Controller die Geräte hängen.

Der IOP übernimmt die Abwicklung der Treiber und Protokolle für die Geräte und puffert Daten aus den Geräten zwischen.



2.4. Graphikprozessoren

2.4.1. Anforderungen

Es sollen Daten und Bilder auf einem Graphikbildschirm dargestellt werden.

Die Größe der Bildpunkte definiert die erreichbare Auflösung.

Vom Rechner her werden Bildpunkte (Pixel) generiert.

Das Auflösungsvermögen des Auges setzt den Maßstab:
in Leseentfernung (15 cm) ist die Auflösung nicht besser als 0.05 mm (50 μ); d. h.
2×10^{-4}.

Schon ein Bild mit 0.2 mm Auflösung (200 μ) wird als gut empfunden (0.2 mm Auflösung entspricht 5 Strichen/mm).

Drucker mit 600 dpi (dots per inch) entspricht einer Punktdichte von 23,6 Punkten/mm oder einer Auflösung von 0.042 mm.

Photographie erreicht 100 - 2000 Striche/mm.

Legt man eine Auflösung von 5 Strichen/mm zugrunde, dann sollte eine DIN A4-Seite von 210 x 297 mm auf einem Bildschirm mit 1050 x 1485 Pixeln darstellbar sein. Der Bequemlichkeit halber werden wir im Folgenden rechnen mit Bildgrößen von 1024 x 1536 = (2 x 512) x (3 x 512) Pixeln. Man erhält eine Bilddichte von 5 x 5 = 25 Pixel/mm^2 (Hologramm: Bilddichte 4 x 10^6 Pixel/mm^2 !)

Pro Pixel werden 1 - 32 Bit bereitgestellt:

1 Bit	:	s/w
8 Bit	:	1 aus 256 Graustufen
	oder :	4 Bit für eine aus 16 Farben, 4 Bit für eine aus 16 Helligkeitsstufen (logarithmische Skala)
	oder :	eine aus 256 Farben inklusive schwarz und weiß (8 x 8 Farbtafel)
16 Bit	:	eine aus 4096 Farben (12 Bit), eine von 16 Graustufen (4 Bit)
	oder :	eine aus 65.536 Farben inklusive schwarz und weiß
24 Bit	:	3 x 8 Bit für die Intensitäten der drei Grundfarben R (rot) G (grün) B (blau) ; (Farbpalette mit 16 Mio. Farben)
	oder :	3 x 8 Bit für Farbwert (Hue, H) und Sättigung (Saturation, S) und Helligkeit (Lightness, L)
32 Bit	:	3 x 8 Bit für Farbe, 8 Bit für Helligkeit/ Transparenz,/ z-Puffer

Um Flackerfreiheit zu sichern, soll das Bild auf dem Schirm mit 70 - 100 Hz dargestellt werden.

Rechnet man das Beispiel mit 100 Hz durch

$\Rightarrow$ 1024 x 1536 = 1.572.864 Pixel

100 Hz $\Rightarrow$ 157.286.400 Pixel pro Sekunde sind zum Bildschirm zu transportieren (alle 6.3 ns ein neues Pixel). Pro Pixel 24 Bit $\Rightarrow$ 470 MByte/s sind bereitzustellen.

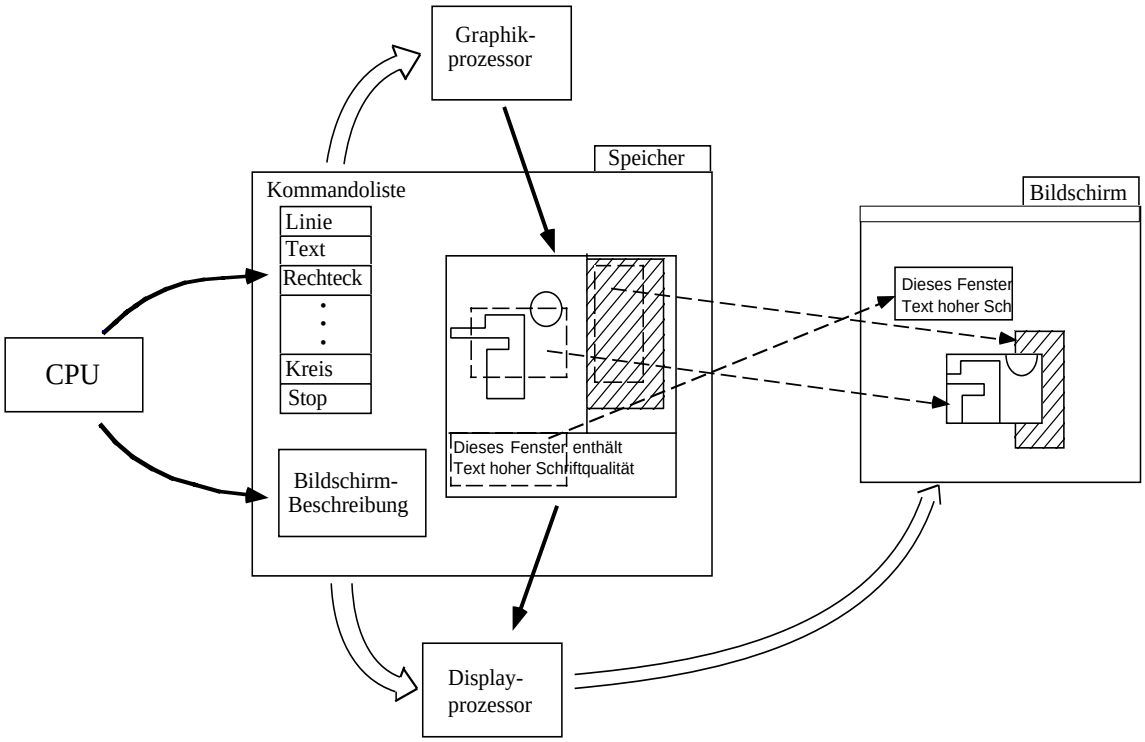
2.4.2. Generelle Struktur der graphischen Datenverarbeitung

Die CPU erzeugt zunächst für die darzustellenden Objekte eine Beschreibungsliste in Form von Kommandos zur Erzeugung graphischer Primitive.

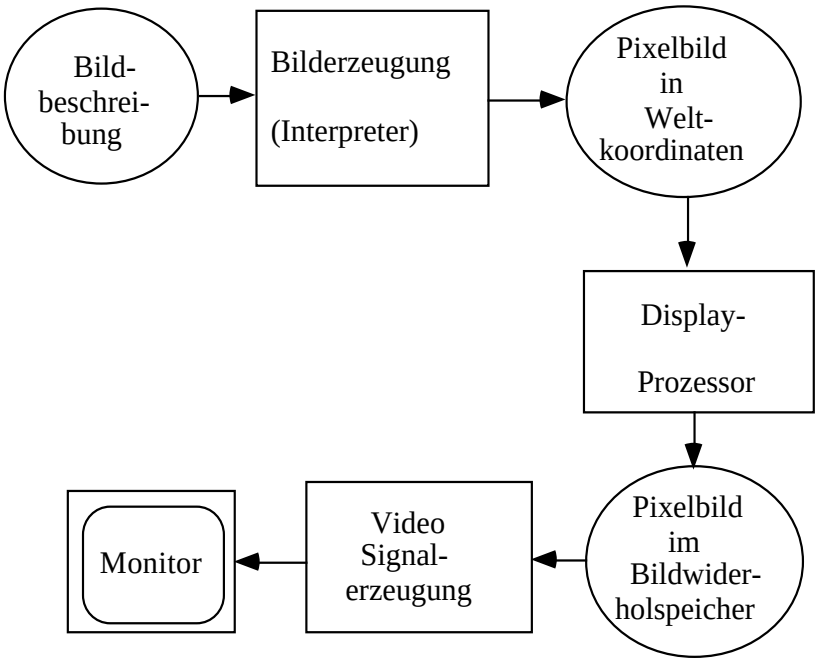
Aus dieser Kommandoliste wird dann ein Graphikprozessor eine Bildbeschreibung in Weltkoordinaten erzeugen, die schon pixelbasiert ist.

I. allg. gibt es viele verschiedene Bilder, die auf dem Bildschirm übereinandergelegt werden.

Das "Wie" wird beschrieben durch eine Bildschirmbeschreibung, aus der ein Displayprozessor das fertige Pixelbild im Bildwiederholspeicher erzeugt, das vom CRT-Controller ausgelesen und als RGB-Signale an den Bildschirm gegeben wird.

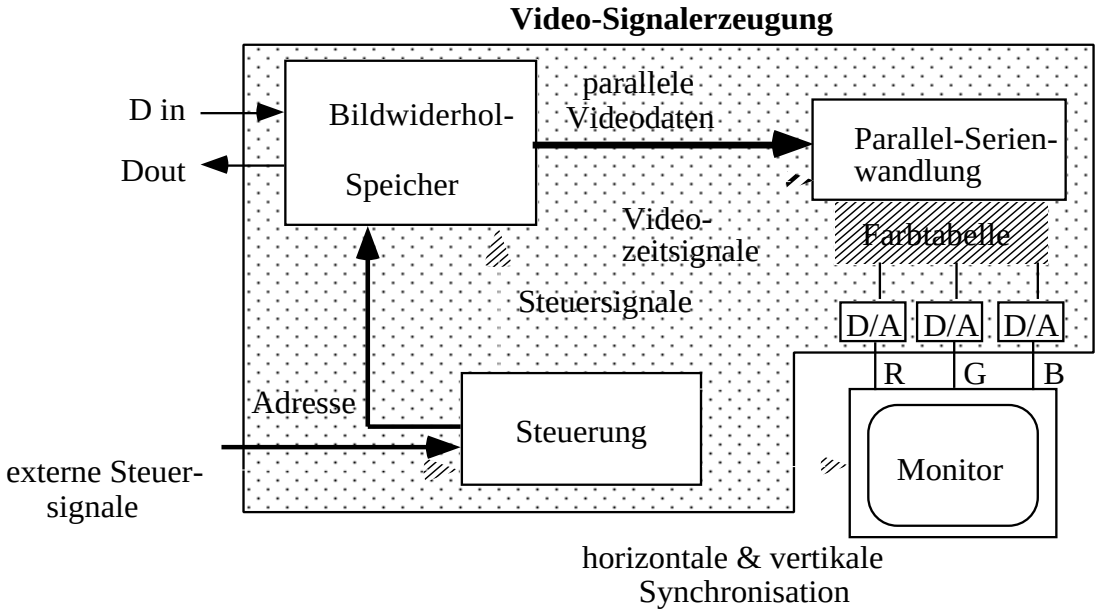


Im Sinne einer Darstellung mit aktiven Einheiten und passiven Daten ist die Bildverarbeitung darstellbar als folgedes Bild.



2.4.3. Auslesen des Bildwiederholspeichers

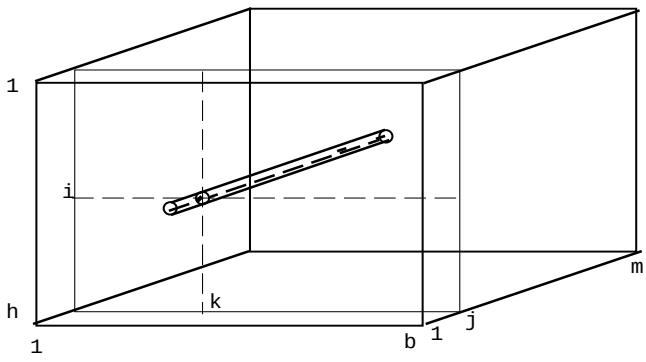
Wegen der hohen Ausleserate des Bildwiederholspeichers (frame buffer) von 470 MByte/s und weil dieser Vorgang nicht unterbrochen werden darf (Streifen auf dem Bildschirm) ist das Auslesen Sache eigener Hardware, des CRT-Controllers, der unabhängig von der CPU arbeitet.



Der Bildwiederholspeicher kann pixel- oder ebenenorientiert aufgebaut sein.

2.4.3.1. Pixel-Architektur

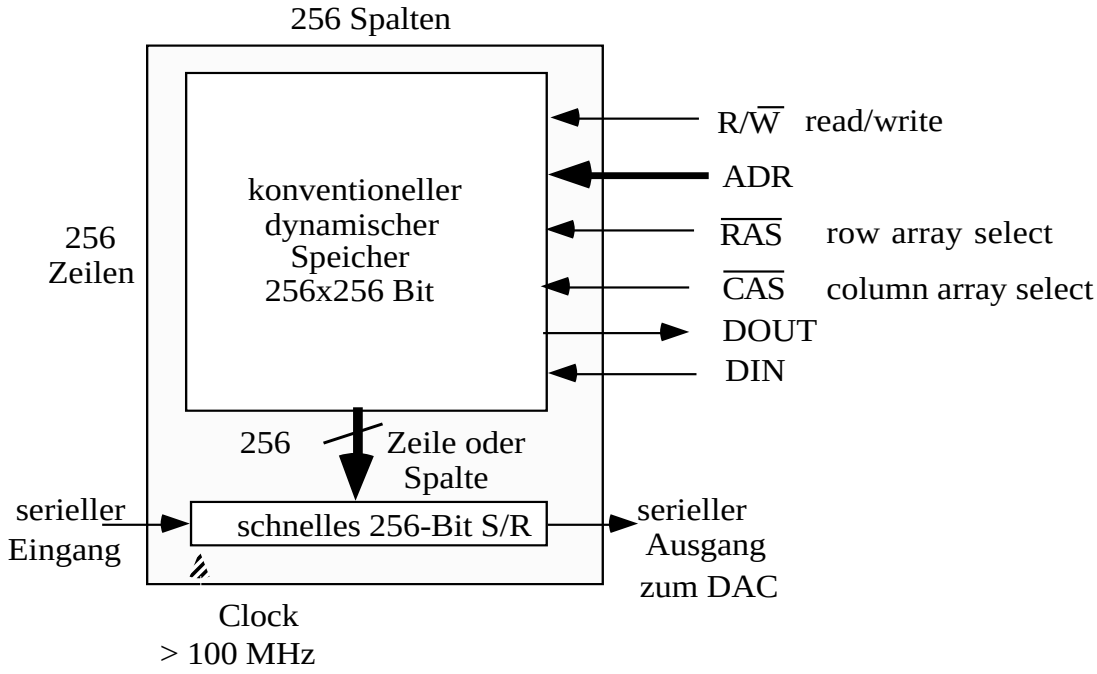
Alle Pixel bei der Bildkoordinate (i, k) werden parallel für alle Ebenen ausgelesen.



Pixel-Architektur

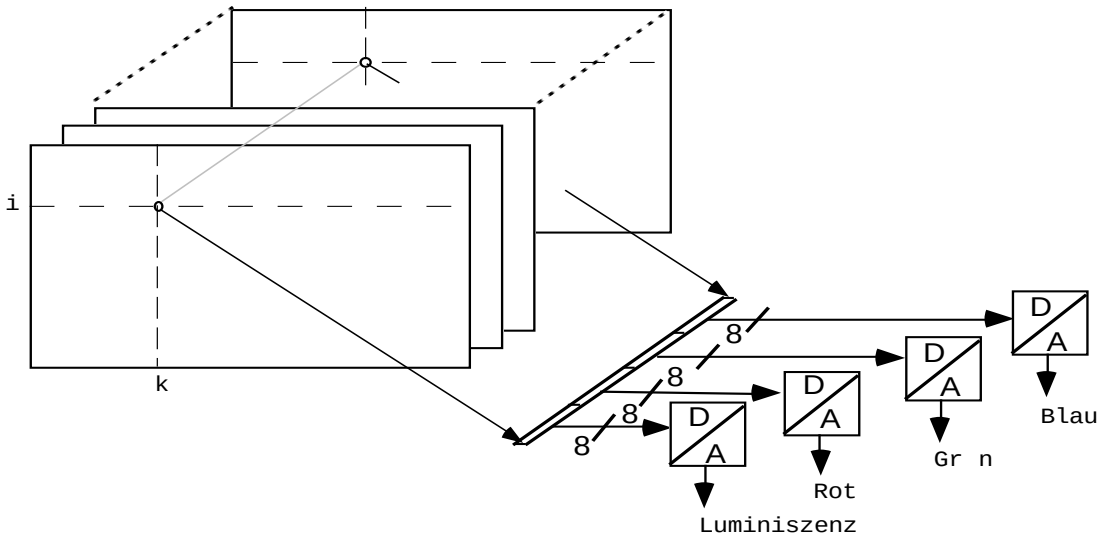
die Pixelwerte bei (i,k)
werden parallel
für alle Ebenen
j = 1,…,m
ausgelesen

Um den geforderten hohen Datentransfer zu erreichen verwendet man eigene Video-DRAM's.

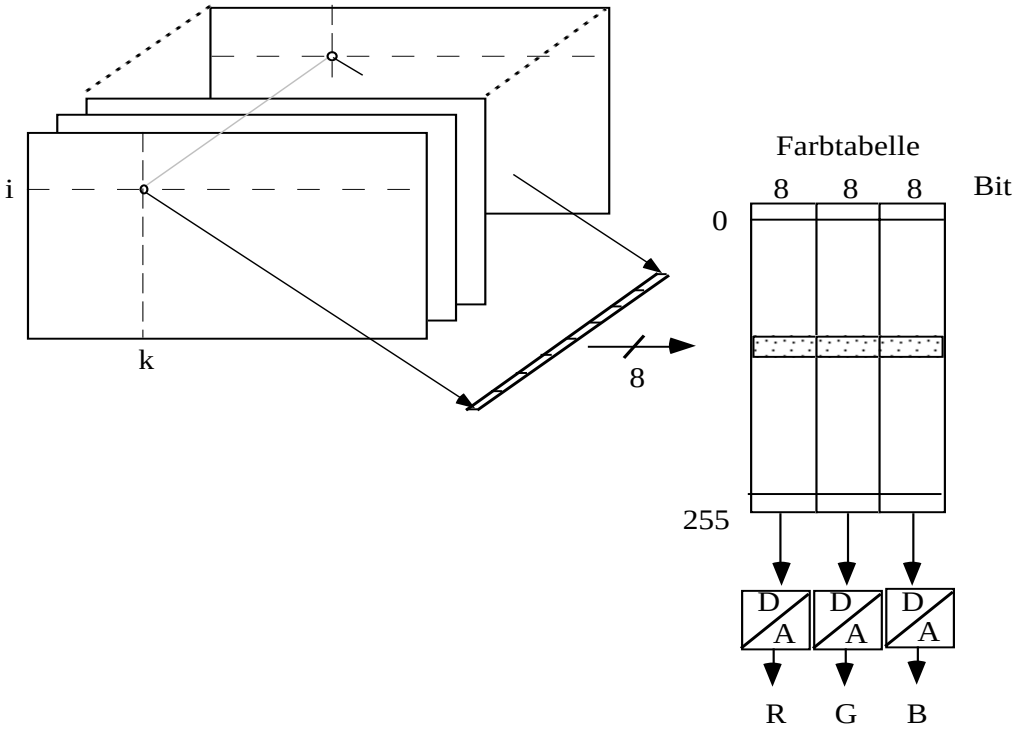


In diesen DRAM's werden alle Bit einer Zeile (z. B. 256 Bit) parallel in ein schnelles Schieberegister gelesen und seriell ausgetaktet. Das passiert für alle Ebenen gleichzeitig.

An den Eingängen der DAC's liegen dann die Pixelbitwerte an und werden gewandelt.

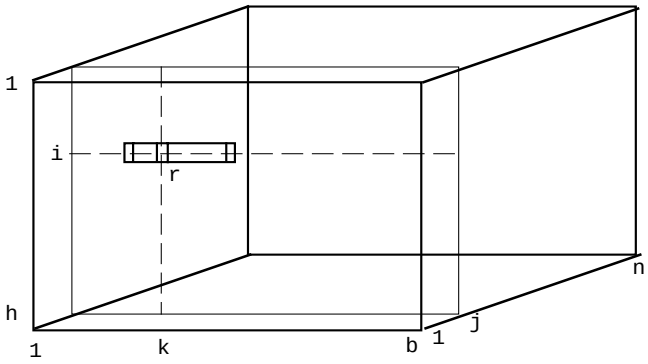


Bei einfachen Rechnern wird noch eine Farbtabelle zwischengeschaltet: 8 Bit aus dem Pixelspeicher wählen über die Tabelle eine von 256 Farben auf dem Monitor aus (inklusive schwarz und weiß).



2.4.3.2. Ebenen-Architektur

Um die Verwendung von Spezial-Speicherbausteinen zu umgehen kann man den Bildwiederholspeicher als Ebenenarchitektur aufbauen.



in der Ebene j steht der Pixelwert $P(i,k)_j$ an der Stelle r eines Worts. Beim Auslesen paralleler Zugriff auf n Werte in einer Ebene

Der Bildwiederholspeicher besteht aus n Speicherebenen mit normalen Speicherchips. Das Pixel (i, k) findet sich in Position r eines Wortes (i, n) in allen n Ebenen.

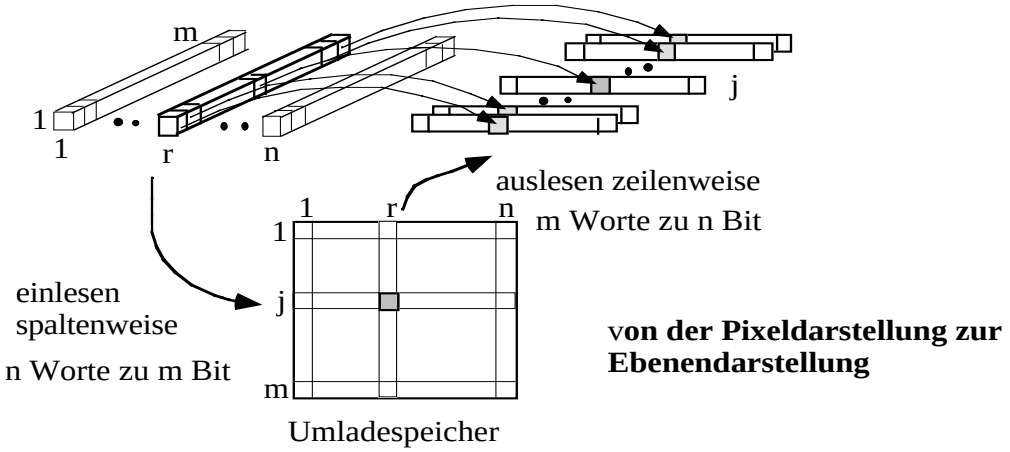
Ausgelesen wird wortweise (32 Bit) parallel in n Schieberegister von Wortlänge, und von dort aus seriell in die DAC's ausgetaktet.

In den Zeiträumen, in denen ein Wort aus dem Schieberegister ausgetaktet wird, kann der Speicher nachgeladen werden. Die Schieberegister entkoppeln die Bilddarstellung auf dem Bildschirm von Bildaufbau in Bildwiederholspeicher.

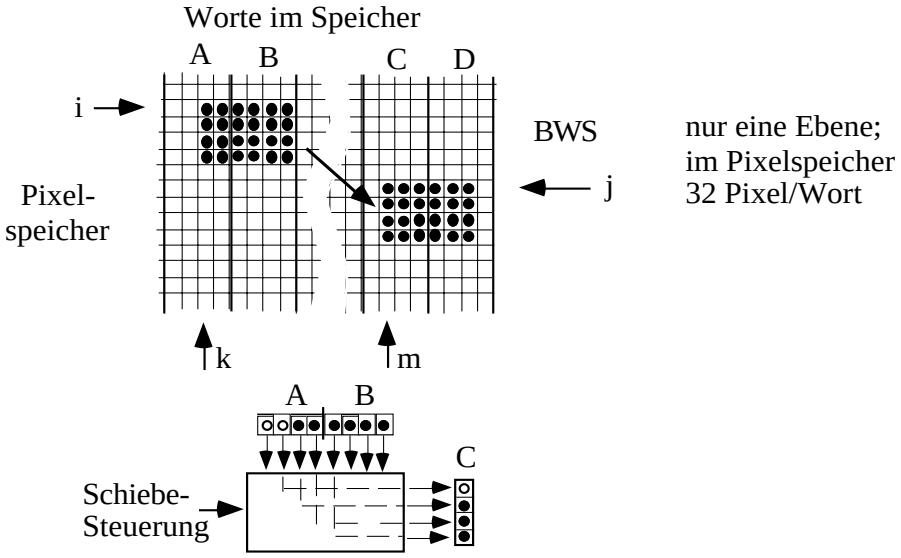
2.4.4. Display-Prozessor

Seine Aufgabe ist der Aufbau des darzustellenden Bildes im Bildwiederholspeicher. Dazu hat er das pixelorientierte Bild im Speicher und eine Bildbeschreibung.

Wenn die Darstellung in Weltkoordinaten (z. B. als Pixeldarstellung) nicht übereinstimmt mit der Darstellung im Bildwiederholspeicher (Ebenendarstellung) muß eine Umtransformation vorgenommen werden:

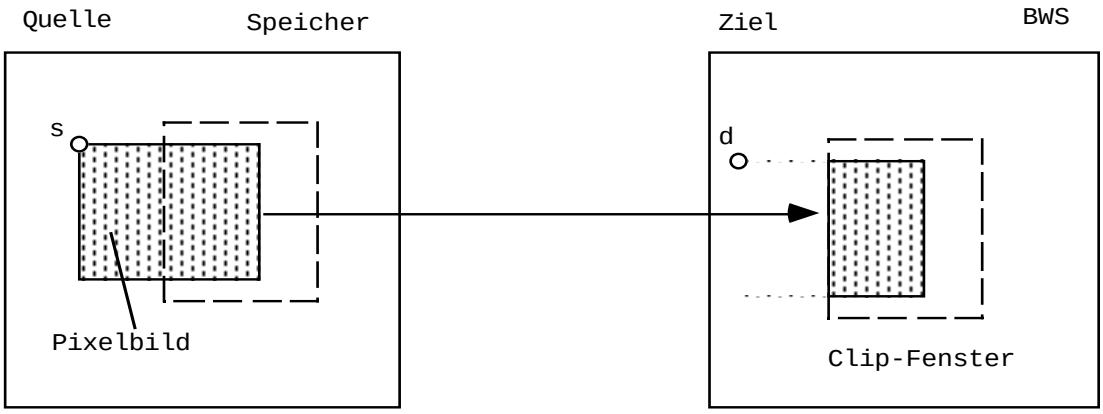


In der gleichen Weise ist eine Transformation nötig, wenn Bilder in Ebenendarstellung vorliegen, aber das Bild im BWSP an einer anderen Stelle liegt wie in der Pixeldarstellung in Weltkoordinaten.



Die Worte müssen ggf. über einen Barrelschifter geschickt werden.

Diese Forderung tritt auf beim Verschieben von Bildern. Eine weitere Aufgabe des Displayprozessors ist das Clipping: nur ein Teil des Bildes soll dargestellt werden.



2.4.5. 2-D Graphikunterstützung

Dies sind spezielle Prozessoren zur Umwandlung von 2-D-Bildbeschreibungen in Pixelbilder. Sie folgen einer Norm.

Graphisches Kernsystem (GKS)

ISO-Standard für 2-D-Graphik (mit Erweiterung auf 3-D).

Genormtes System der Erzeugung, Manipulation und Ausgabe zweidimensionaler Vektor- und Rasterbilder.

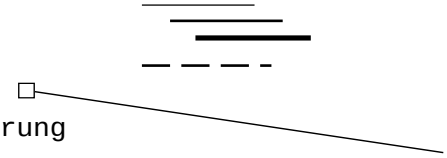
Sämtliche Funktionen zur Interaktion, Speicherung, Veränderung und Bildstrukturierung sind geräteunabhängig.

Es gibt

- Primitive zur Darstellung und Erfassung von graphischen Informationen.
  Die Zusammenfassung mehrerer Primitive zu einer Einheit (Segmentbildung) ist möglich.

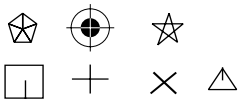
- Polygonzug: (polyline) geordnete Menge von Punkten, die einen nicht notwendig geschlossenen Linienzug beschreiben

  ⇒ Strichzeichnung mit Attributen wie
  Liniendicke
  Farbe
  Strichart
  Pickerkennung
  berlagerungssteuerung



- Markierungen: (polymarker) erzeugt zu einem Punkt eine zentrierte Marke

  Attribute:
  Markenform
  Farbe

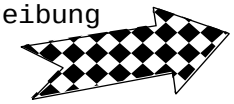


- Schrift: (text) Zeichenketten mit Attributen

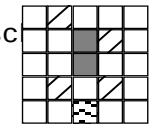
  → Fontbeschreibung, Pr zisionsangabe, Farbe

  → Gr §e, Abst nde, Orientierung, Zeilenabstand, Pickerk

- F llgebiet: (fill area) Einf rben eines geschlossenen Polygonzuges mit einem M Atribut ist die Musterbeschreibung



- Zellmatrix: (pixel array) Erzeugung eines Bildes auf einem Rasterbildsc Attribute: Farbe und Gr §e



- Verallgemeinertes Darstellungselement (generalized drawing primitive)

  Realisierung von selbstgeschriebenen Primitiven
  z. B. Kreisen, Ellipsen, Icons, ...

- Segmente

  Zusammengebaute Bilder mit einem Namen versehen.
  Können nur noch als Ganzes manipuliert werden.

  - öffnen, schließen, löschen
  - transformieren
  - sichtbar / unsichtbar machen
  - hervorheben
  - bei Überlappungen ordnen
  - in andere Segmente einfügen
  - identifizieren
  - umbenennen.

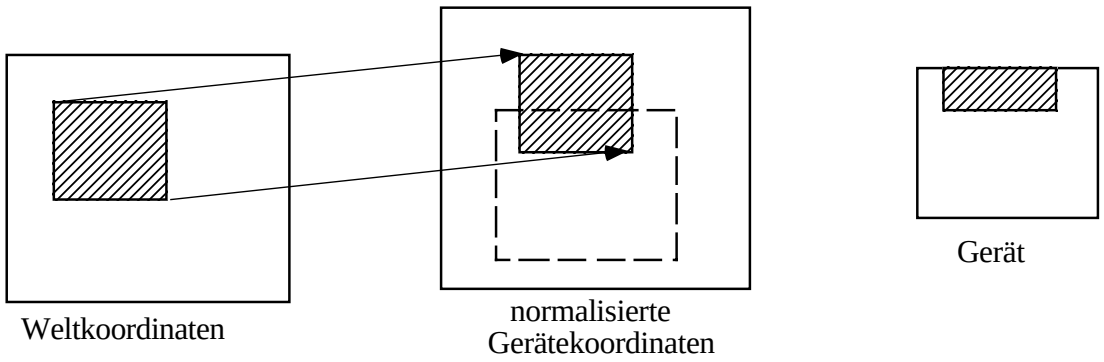
  Mehrere Primitive eines Segments können zu einer Gruppe zusammengefaßt und mit einem Namen versehen werden.

Koordinatensysteme und Transformationen

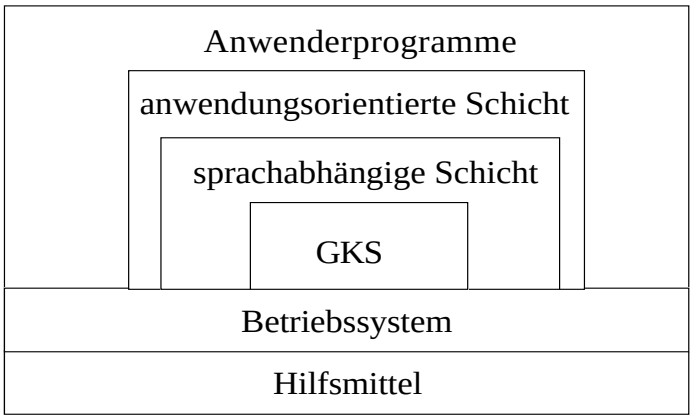
Im Anwenderprogramm: Weltkoordinaten

Abbildung auf ein normalisiertes Gerätekoordinatensystem von dort auf die einzelnen Gerätekoordinaten.

Ein Gerät bildet dabei ein Fenster aus dem Bereich der normalisierten Gerätekoordinaten ab.



Schichtenmodell des GKS:



Zur Ein- Ausgabe hin wird ein virtueller Arbeitsplatz (Workstation) beschrieben.

Der Benutzer macht seine Ausgaben (und Eingaben) unabhängig von der reellen Hardware und eine Umsetzfunktion paßt den virtuellen Arbeitsplatz an das reale E/A-Gerät an.

Eingabefunktionen:

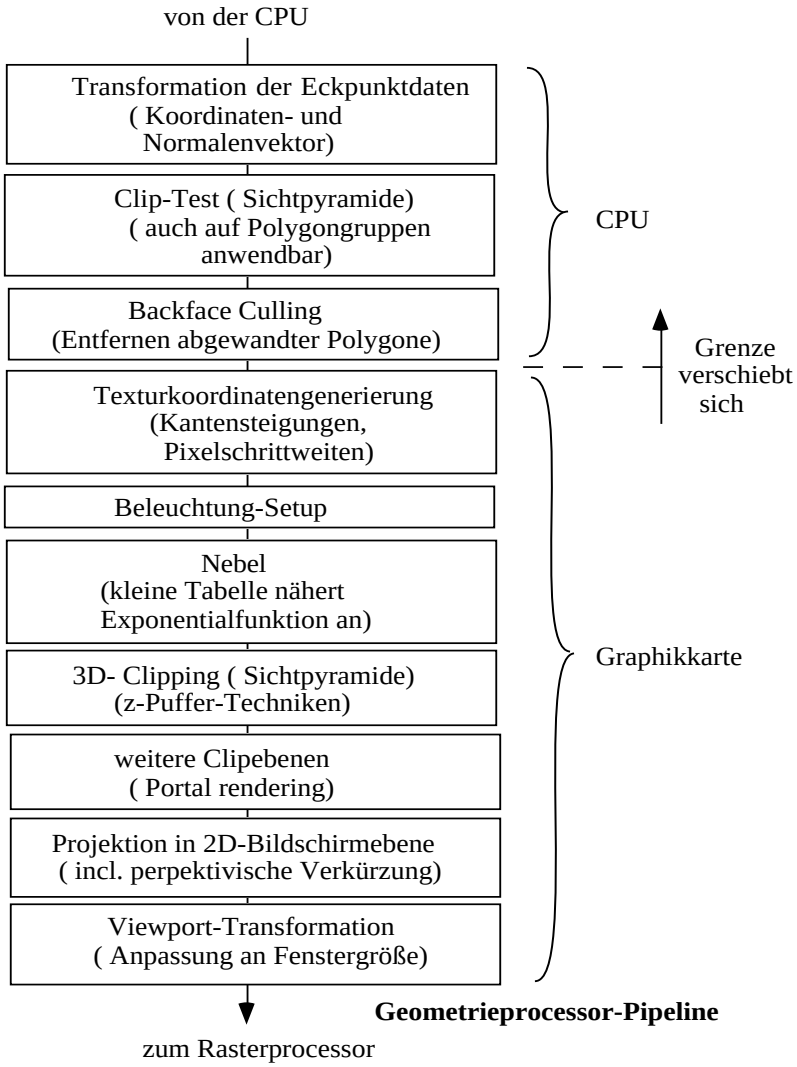
PICK:	übergibt die Kennzeichnung eines logischen Segments
LOCATOR:	überprüft ein Koordinatenpaar (in Anwenderkoordinaten)
VALUATOR:	überprüft eine real-Größe
STRING:	überprüft einen Textstring
CHOICE:	wählt eine Variante aus (ganze Zahl)

Betriebszustände eines logischen Eingabegerätes:

REQUEST (Anforderung):	warten, bis eine Eingabe erfolgt
SAMPLE (Abfrage):	momentaner Zustand wird zurückgegeben
EVENT (Ereignis):	Abfragen einer Eingabewarteschlange

2.4.6. 3-D-Graphikunterstützung

Der Ablauf der Erstellung eines 3-D-Bildes kann so dargestellt werden:



Ausgangspunkt ist hier eine Polygondarstellung der Oberfläche eines 3-D-Körpers. Ein Polygon (3- oder 4-Eck) wird durch seine Eckpunkte mit Koordinaten (x_i, y_i, z_i) und Farbe (rt_i, gr_i, bl_i), und durch die Flächennormale n und eine Angabe über die Transparenz der Oberfläche oder ihre Tektur beschrieben.

2.4.6.1. Koordinatentransformation

Jede Bewegung eines Objektes läßt sich beschreiben durch die Anwendung einer Transformation auf die Eckpunkte der Polygone und somit auf die Normalen.

Sei $\mathbf{x} = (x, y, z, w)$ die Koordinate eines Punktes (homogene Koordinaten), dann ist die Transformation $\mathbf{x}' = \mathbf{x} \cdot \mathbf{A}$, mit der Matrix $A = S \cdot R_z \cdot R_y \cdot R_x \cdot T$ für alle Punkte durchzuführen.

Die Matrix A setzt sich zusammen aus der Hintereinanderausführung einer Translation T, von Rotationen R_x, R_y, R_z und einer Skalierung S .

Koordinatentransformationen

Translation

$x' = x + Tx$

$y' = y + Ty$

$z' = z + Tz$

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ Tx & Ty & Tz & 1 \end{pmatrix}$$

in homogenen Koordinaten :

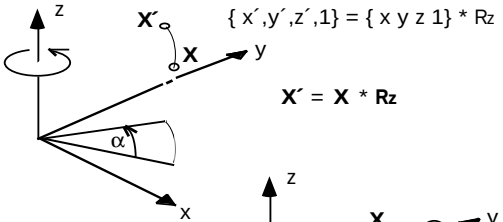
$\{ x', y', z', 1 \} = \{ x, y, z, 1 \} * T$

$\mathbf{X' = X * T}$

Zeilenvektor Matrix

Rotation um z- y- x -Achse mit Winkeln α, β, γ

$$\mathbf{Rz} = \begin{pmatrix} \cos\alpha & \sin\alpha & 0 & 0 \\ -\sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

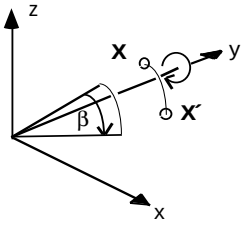


$\{ x', y', z', 1 \} = \{ x\ y\ z\ 1 \} * Rz$

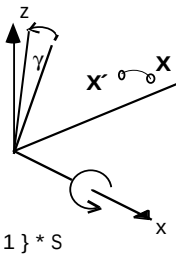
$\mathbf{X' = X * Rz}$

$$\mathbf{Ry} = \begin{pmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$\mathbf{X' = X * Ry}$



$$\mathbf{Rx} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\gamma & \sin\gamma & 0 \\ 0 & -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



$\mathbf{X' = X * Rx}$

Skalierung

$x' = x * Sx$

$y' = y * Sy$

$z' = z * Sz$

$\{ x', y', z', 1 \} = \{ x\ y\ z\ 1 \} * S$

$$\mathbf{S} = \begin{pmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$\mathbf{X' = X * S}$

Allgemein : $\mathbf{X' = X * S * Rz * Ry * Rx * T}$

! Matrixmultiplikation nicht kommutativ !

Die einmalige Berechnung der Matrix A erfordert 4 x 16 x 4 = 256 Operationen.

$\mathbf{x' = x \cdot A}$ $\qquad$ $\mathbf{A = S \cdot R_z \cdot R_y \cdot R_x \cdot T}$

$$[x'', y'', z'', w''] = [x, y, z, w] \cdot \begin{bmatrix} a11 & a12 & a13 & a14 \\ a21 & a22 & a23 & a24 \\ a31 & a32 & a33 & a34 \\ a41 & a42 & a43 & a44 \end{bmatrix}$$

$x' = a11\ x + a21\ y + a31\ z + a41\ w$
$y'' = a12\ x + a22\ y + a32\ z + a42\ w$
$z'' = a13\ x + a23\ y + a33\ z + a43\ w$
$w'' = a14\ x + a24\ y + a34\ z + a44\ w$

4 parallele Operationen

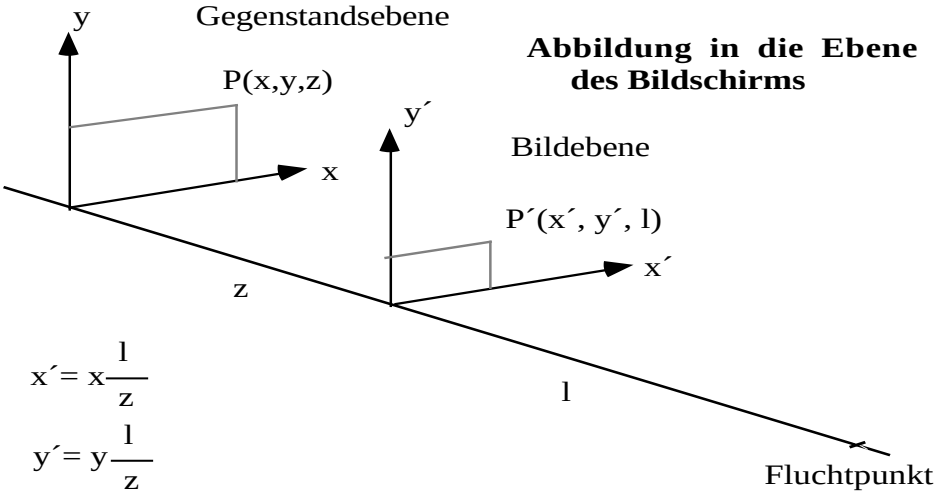
alle Punkte des K

Die Transformation der Koordinaten kann durch Hardware unterstützt werden, die die Operationen parallel für alle Komponenten eines Punktes durchführt.

2.4.6.2. Abbildung in die Ebene des Bildschirmes

Am Ort des Auges des Betrachters wird ein Fluchtpunkt angenommen und die Eckpunkte in der Bildebene berechnet. Zugleich wird dabei nachgeprüft, ob ein Objektpunkt überhaupt sichtbar ist, oder jenseits des Bildausschnitts liegt, oder seine Normale vom Betrachter fort zeigt.

Dabei wird zugleich auch schon die Verdeckung der Polygone aufgelöst (hidden line removal, hidden surface removal) unter Verwendung von z-Puffer-Techniken, die wiederum durch Hardware unterstützt werden können.



Die Punkte im Inneren eines Polygons werden auf Pixel mit einer durch den Pixelabstand gegebenen Auflösung projeziert.

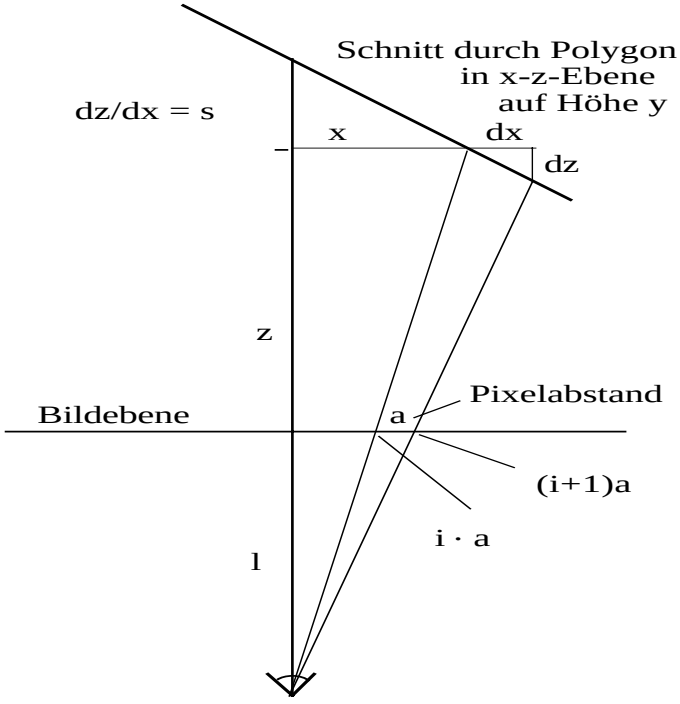


Abbildung der Punkte
im Inneren eines Polygons
auf die Bildebene

$(i+1)a/l = (x+dx)/(z+dz)$

$(i+1)a/l = (x+dx)/(z+sdx)$

$(z+sdx)(i+1)a/l = x + dx$

$$dx = \frac{z\cdot(i+1)\cdot a - x\cdot l}{l - s\cdot a\cdot(i+1)}$$

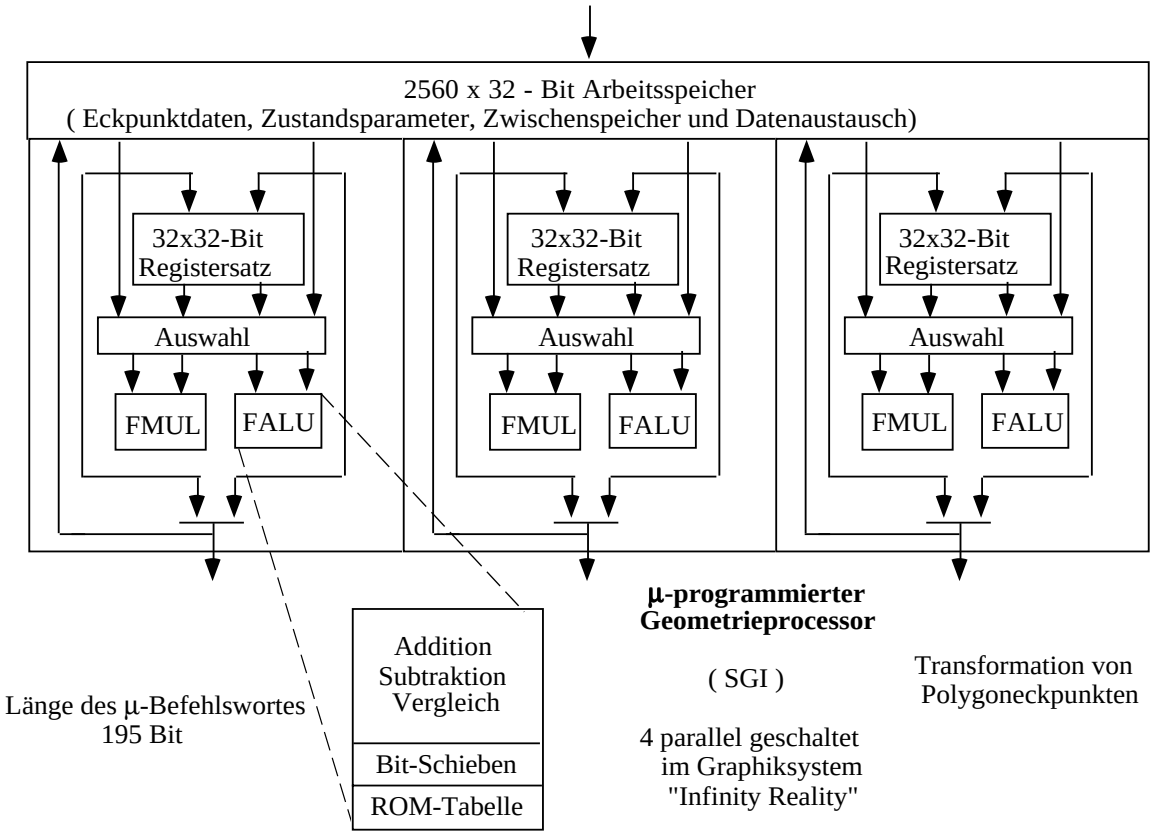
$dz = s\cdot dx$

Zugleich mit der Zuordnung der Objektpunkte zu Pixeln wird das Beleuchtungsmodell gerechnet.

2.4.6.3. Beleuchtungsberechnung

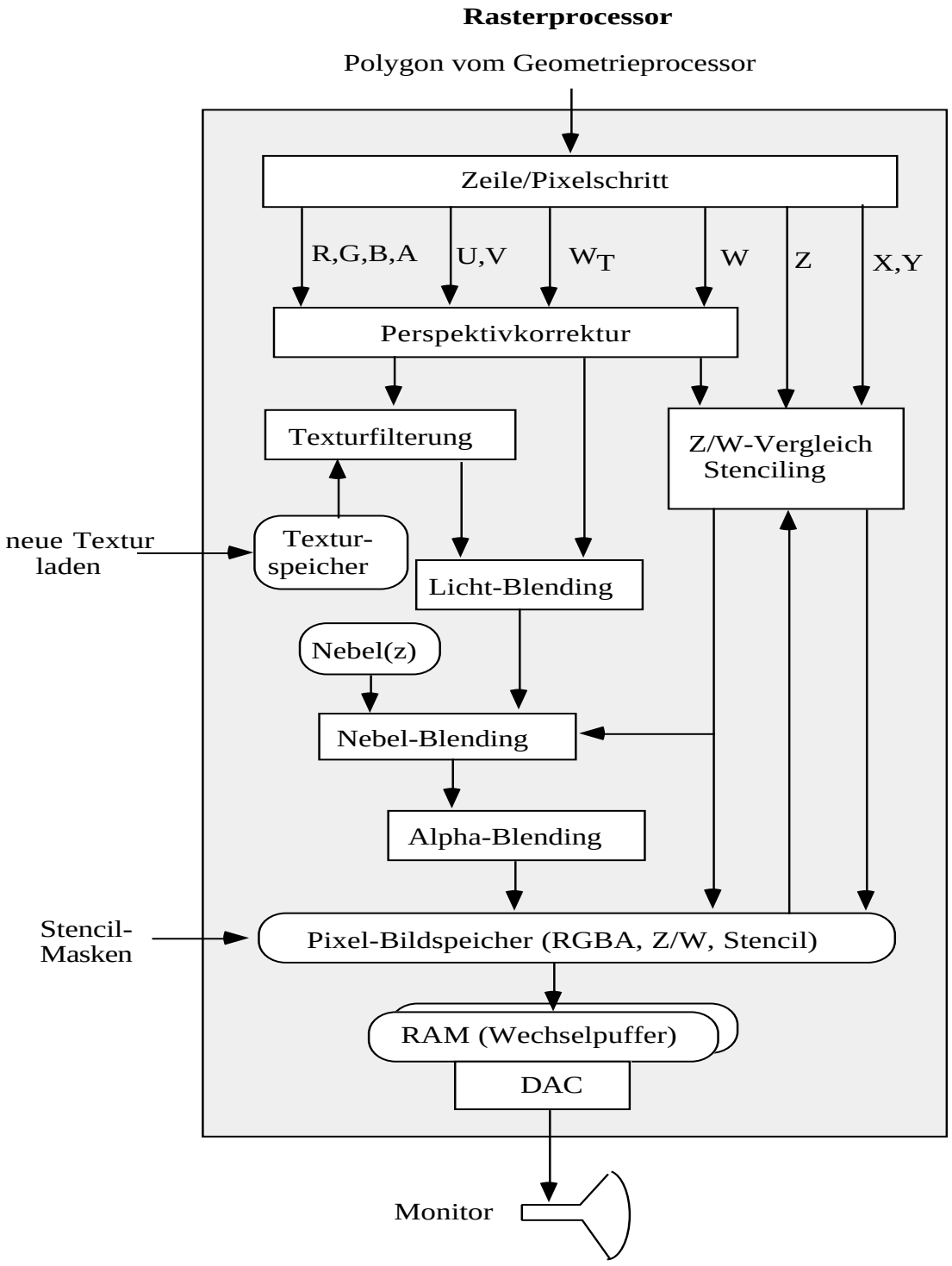
Aus Kenntnis der Flächennormalen und der Lichtquellen in der Scene kann der reflektierte Anteil des Lichtes gerechnet werden (Gouraud-Shading, Phong-Shading) bis hin zur Berechnung von Schattenwürfen und mehrfachen Reflexionen (Ray-Tracing). Auch diese Berechnungen sollten durch Hardware unterstützt werden.

Ein Beispiel ist der Graphikprozessor von SGI



mit drei parallelen Rechenwerken.

Der Displayprozessor wird bei 3-D-Graphiksystemen ersetzt durch einen Rasterprozessor.



Literatur: Byte, Mai 1998, pp. 42

J. Dünow, Polygon-Treibsätze, c't 1999, Heft 2, pp. 188 - 195

J. Dünow, Pixelpunkt, c't 1999, Heft 2, pp. 130 - 135