

Interner Bericht
DESY F58 81/01
Mai 1981

S t r u k t u r - u n d E n t w u r f s a s p e k t e
e i n e s 8 M B y t e - B i l d s e q u e n z s p e i c h e r s

von

G.C. Nicolae und M. de Rijke

Eigentum der Property of	DESY	Bibliothek library
Zugang: Accessions:	9. JULI 1981	
Leihfrist: Loan period:	7	Tage days

DESY behält sich alle Rechte für den Fall der Schutzrechtserteilung und für die wirtschaftliche Verwertung der in diesem Bericht enthaltenen Informationen vor.

DESY reserves all rights for commercial use of information included in this report, especially in case of apply for or grant of patents.

"DIE VERANTWORTUNG FÜR DEN INHALT
DIESES INTERNEN BERICHTES LIEGT
AUSSCHLIESSLICH BEIM VERFASSER."

Inhalt

1. Einleitung
2. Struktur des Bildsequenzspeichers
3. Entwurfsaspekte der Systemkomponenten
 - 3.1. RTA-Bus Interface
 - 3.2. Speichercontroller
 - 3.3. Speichermodul
4. Zusammenfassung
5. Literatur

1. Einleitung

Bei der digitalen Verarbeitung von Videobildserien kommt der digitalen Speicherung einer gesamten Videobildserie in Direkt-Zugriffsspeichern eine immer größer werdende Bedeutung zu. Die Gründe hierfür sind die folgenden:

- Die Bildsequenzen können in Echtzeit ohne die Verwendung von Analogspeichern schnell und einfach digitisiert werden.
- Die Schnelligkeit von Bildverarbeitungsalgorithmen kann durch den direkten Zugriff zum räumlichen und zeitlichen Kontext eines Bildelementes entscheidend erhöht werden.
- Verarbeitete Bildsequenzen können in beliebiger Geschwindigkeit und Folge wiedergegeben werden.

Im Rahmen eines Projektes zur Analyse von angiographischen Bildsequenzen (1) wurde für das Bildverarbeitungssystem CA-1 ein Bildsequenzspeicher mit einer Kapazität von 8 M pixels und einer Bandbreite von 100 nsek/pixel entwickelt. Die erforderliche Speicherkapazität für die digitale Speicherung einer gesamten Videobildserie einerseits, und die notwendige Bandbreite dieses Speichers für die Darstellung eines Videobildes andererseits haben lange Zeit die Realisierung eines solchen Bildsequenzspeichers als besonders aufwendig erscheinen lassen. Technologische Fortschritte, die in den letzten Jahren auf dem Gebiet der Halbleiterspeicher erzielt worden sind, haben die technischen Voraussetzungen gebracht, solche Bildsequenzspeicher kostengünstig zu realisieren.

Diese Beschreibung befaßt sich mit den relevanten Struktur- und Entwurfsaspekten eines 8 MByte Bildsequenzspeichers, der als Erweiterung der Speicherkapazität des vorhandenen Digital-Video-Systems (DVS) implementiert worden ist (2). Die Architektur des DVS-Systems ist in Bild 1 gezeigt. Der Sequenzspeicher ist direkt an dem Real-Time-Asynchronous Bus (RTA-Bus) angeschlossen und ersetzt den vorhandenen Bildspeicher, so daß die Funktionsweise der verschiedenen Prozessoren im System weitgehend unver-

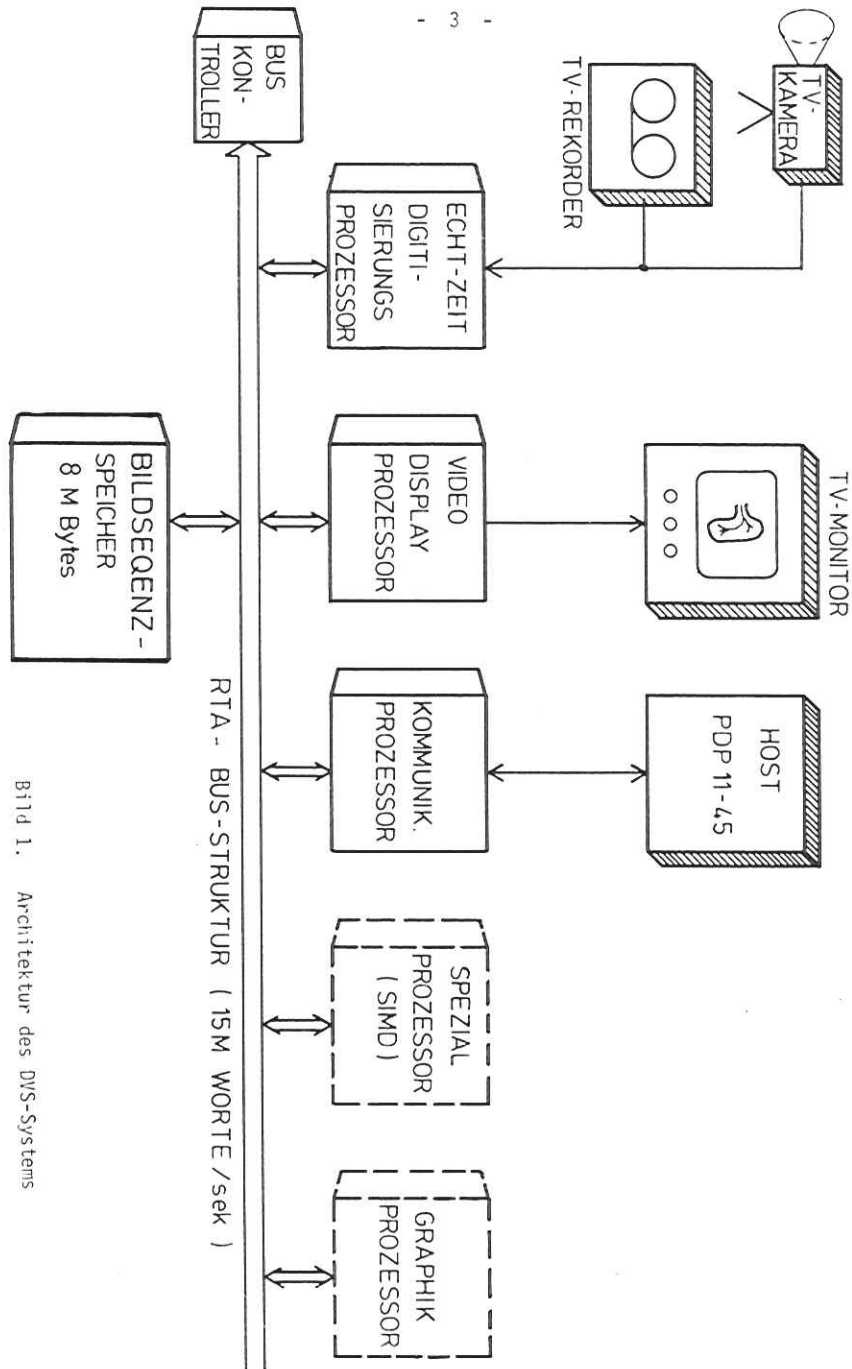


Bild 1. Architektur des DVS-Systems

ändert bleibt. Mit dem Bildsequenzspeicher kann man eine Videobildserie von maximal 128 Bildern mit jeweils 256^2 Bildpunkten in Echtzeit speichern und direkt adressieren.

2. Struktur des Bildsequenzspeichers

Bild 2 zeigt die globale Struktur des Bildsequenzspeichers. Der Bildsequenzspeicher besteht aus einem Speicherkontroller und aus Speichermodulen. Mit Hilfe des Speicherkontrollers erscheint der Bildsequenzspeicher dem Anwender als ein Stapel von Videobildern - hier auch Seiten genannt - deren Größe und Anzahl beliebig programmierbar sind. Dies bedeutet, daß der Anwender unabhängig von der Seitengröße und der Länge der Bildserie die vorhandene Speicherkapazität optimal ausnutzen kann. Beschränkend gilt nur die Bedingung

$$\text{Seitengröße} \times \text{Sequenzlänge} \leq \text{Speicherkapazität (1)}$$

Diese Bedingung verbietet lediglich die Überschreitung der vorhandenen Speicherkapazität. Der Bildsequenzspeicher hat also für den Anwender eine virtuelle Speicherorganisation, die entsprechend dem betrachteten Bildpunktquader programmierbar ist. Somit kann der Bildsequenzspeicher für die jeweilige Anwendung angepaßt werden. Die physikalische Speicherorganisation, die durch die physikalische Anordnung der Speichermodule bestimmt wird, ist jedoch unveränderlich. Der Bildsequenzspeicher ist physikalisch in Blöcken von 1 Megawort und jeweils 64 Bits organisiert. Die Wortlänge von 64 Bits wurde gewählt, um die Bandbreite des Speichers zu verbessern. Die Abbildung des anwenderspezifischen virtuellen Adreßraumes auf den festen physikalischen Adreßraum wird in Echtzeit durch den Speicherkontroller realisiert.

Der Bildsequenzspeicher ist über ein RTA-Bus-Interface am DVS angeschlossen. Das RTA-Bus-Interface leitet unter Berücksichtigung des auf dem RTA-Bus geltenden Protokoll die Transferaufträge verschiedener Prozessoren an den Speicherkontroller weiter.

Eine für den Entwurf des Bildsequenzspeichers wesentliche Forderung bestand darin, daß die für die Echtzeit-Digitalisierung und für die Dar-

stellung der Videobilder erforderlichen Zugriffsraten eingehalten werden müssen. Dies bedeutet, daß auf dem RTA-Bus Transferraten von 5 M Wörter/sek stattfinden können. Dafür war es notwendig, den Ablauf des Datentransfers im Bildsequenzspeicher in einer Pipelinestruktur zu organisieren.

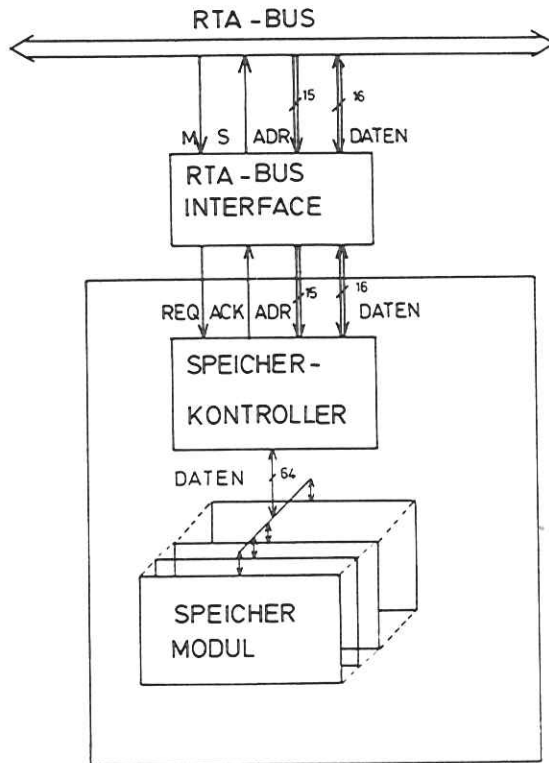


Bild 2. Struktur des Bildsequenzspeichers

3. Entwurfsaspekte der Systemkomponenten

Die nachfolgende Beschreibung erläutert die Struktur und die damit zusammenhängenden Entwurfsaspekte der verschiedenen Komponenten des Bildsequenzspeichers.

3.1. RTA-Bus-Interface

Das RTA-Bus-Interface ist als Bindeglied zwischen dem RTA-Bus und dem Speichercontroller zu betrachten. Es kommuniziert einerseits mit dem RTA-Bus und andererseits mit dem Speichercontroller. Auf dem RTA-Bus gilt für die Kontrolleleitungen M und S ein asynchrones Transferprotokoll, dessen detaillierte Beschreibung in (3) nachgelesen werden kann. Für die Kommunikation mit dem Speichercontroller über die Leitungen REQ, ACK wurde sinnvollerweise auch ein zeitfreies Handshake-Protokoll

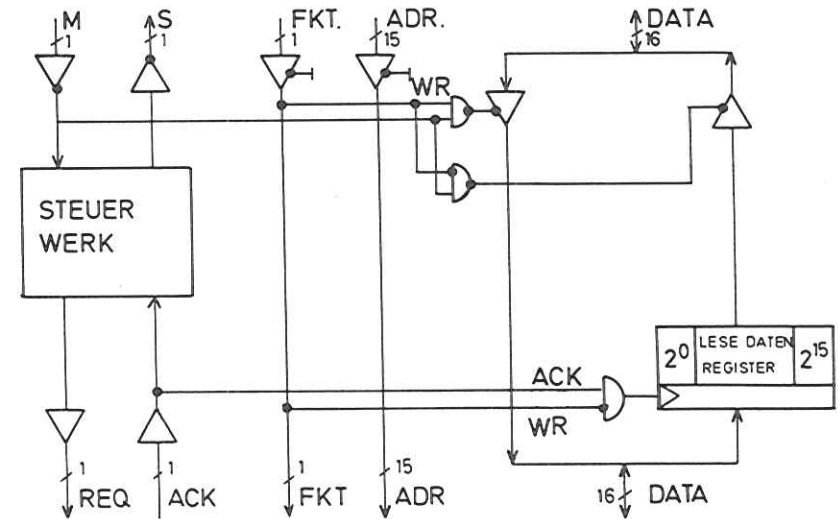


Bild 3. Struktur des RTA-Interfaces

gewählt. Das RTA-Bus-Interface konnte somit als asynchrones Schaltwerk realisiert werden. Die Struktur des RTA-Bus-Interfaces ist in Bild 3 dargestellt. Das Operationswerk besteht hauptsächlich aus verschiedenen Datentreibern bzw. -empfängern für die Kontroll-, Adreß-, Funktions- und Datenleitungen an dem RTA-Bus. Diese Treiber werden operationell gesteuert. Für den Datenweg in Leserichtung ist für die Datenpufferung ein Pipelineregister vorgesehen, das eine erste Stufe der vorher erwähnten Pipelinestruktur darstellt.

Der Aktionsablauf des in Bild 3 gezeigten Steuerwerks ist mit dem Petrinetz in Bild 4 beschrieben. Es zeigt, daß nach der erfolgreichen Auftragsvermittlung an den Speicherkontroller (ACK↑) die Kommunikation mit dem RTA-Bus einerseits und mit dem Speicherkontroller andererseits parallel ablaufen. Damit werden die Zugriffszeiten zum Bildsequenzspeicher reduziert. Das Petrinetz im Bild 4 kann wie im Bild 5 gezeigt in ein äquivalentes Petrinetz transformiert werden, bei dem die stark verkoppelten Zustandsgraphen nur über Leseplätze zusammenhängen. Aus diesem Petrinetz mit zeitfreien und elementaren Kommunikationskomponenten können die Automatengraphen in Bild 6 abgeleitet werden. Die Implementierung solcher asynchroner Automaten ist bekannt und braucht daher hier nicht weiter diskutiert zu werden.

Das wesentliche Ergebnis beim Entwurf des RTA-Bus-Interfaces ist die Tatsache, daß durch die asynchrone und parallele Modellierung des Schaltwerkes die Verzögerungszeiten für die Kommunikation zwischen dem RTA-Bus und Speicherkontroller auf 30 nsek minimiert werden konnten.

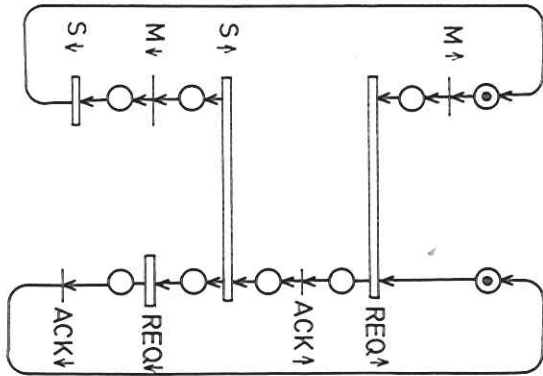


Bild 4. Petrinetz des Steuerwerkes im RTA-Bus-Interface

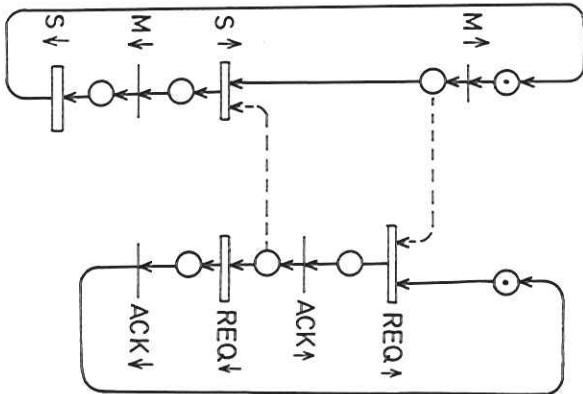


Bild 5. Äquivalentes Petrinetz mit Leseplatz-Kommunikation

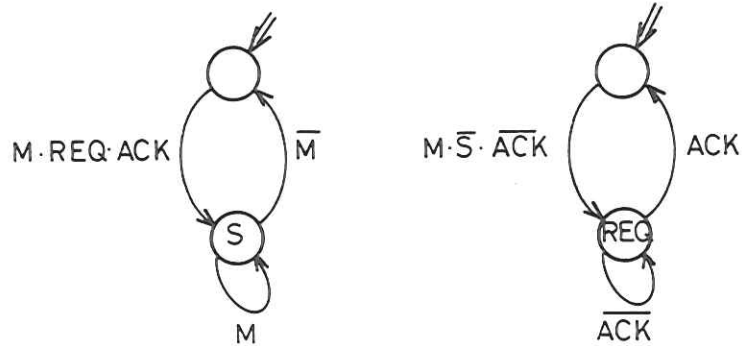


Bild 6. Automatengraph des RTA-Bus Interfaces

3.2. Speichercontroller

Der Speichercontroller hat die Aufgabe, den virtuellen Adreßraum des vom Anwender gedachten Bildpunktquaders auf den physikalischen Adreßraum der Speichermodule abzubilden. Diese komplexe Aufgabestellung läßt die Realisierung des Speichercontrollers nur in Form eines synchronen Schaltwerkes zu.

Die informationelle Struktur des Speichercontrollers ist in Bild 7 dargestellt. Das Schaltwerk setzt sich aus einem Steuer- und aus einem Operationswerk zusammen. Um die notwendige Geschwindigkeit für die Abbildung der Adreßräume zu erreichen, wurden die Operationen des Speichercontrollers durch die Parallelisierung des Operationswerkes realisiert. Daher ist das implementierte Operationswerk relativ umfangreich geworden.

Das Bild 7 zeigt, daß die physikalische Speicheradresse durch Addition aus der Basisadresse und aus der aktuellen virtuellen Adresse gebildet wird. Die Basisadresse, die den Anfang der gewünschten Seite bestimmt, wurde vorher per Programm in das Basisadressregister geladen. Für die Lese- und Schreibrichtung sind getrennte Adreßregister vorgesehen, so daß bidirektionale Datentransferaufträge, eventuell von verschiedenen Prozessoren am RTA-Bus, gleichzeitig und ohne störende Wechselwirkungen hinsichtlich der Datenrate geführt werden können.

Für die Daten selbst wurden konsequenterweise auch getrennte Wege für das Lesen und Schreiben implementiert. In jeder Transferrichtung werden die Daten in getrennte Pipelineregister zwischengespeichert. Diese Pipelineregister können Datenvektoren von 64 Bits speichern, wobei beim Lesen ganze Wörter von 64 Bits, beim Schreiben jedoch nur Wörter von 16 Bits gepuffert werden. Dies entspricht der Tatsache, daß beim Lesen die Daten vom Speicher in 64 Bits Wörter angeboten werden. Für die Schreibrichtung gibt es ein zusätzliches Statusregister, das die Gültigkeit der vier 16-Bits-Wortsegmente des Schreibpipelineregisters entsprechend der Vorgeschichte der Schreibaufträge protokolliert. Somit können bei einem schreibenden Speicherzugriff nur diejenigen 16 Bits-Wortsegmente tatsächlich geschrieben werden, die entsprechend dem Statusregister gültig sind. Bei jedem Transferauftrag werden die Daten zunächst in die Pipelineregister gepuffert. Aufgrund der Reihenfolge der spezifizierten virtuellen Adressen können die Daten direkt aus dem Lesepipelineregister gelesen bzw. in das Schreibpipelineregister geschrieben werden. Dadurch können die Anzahl der reellen und zeitraubenden Speicherzugriffe erheblich verringert und somit die Bandbreite des Bildsequenzspeichers beachtlich erhöht werden. Der maximale Wirkungsgrad dieser Organisationsform der Pipelineregister wird erreicht, wenn die Reihenfolge der virtuellen Adressen die Schrittweite ± 1 hat.

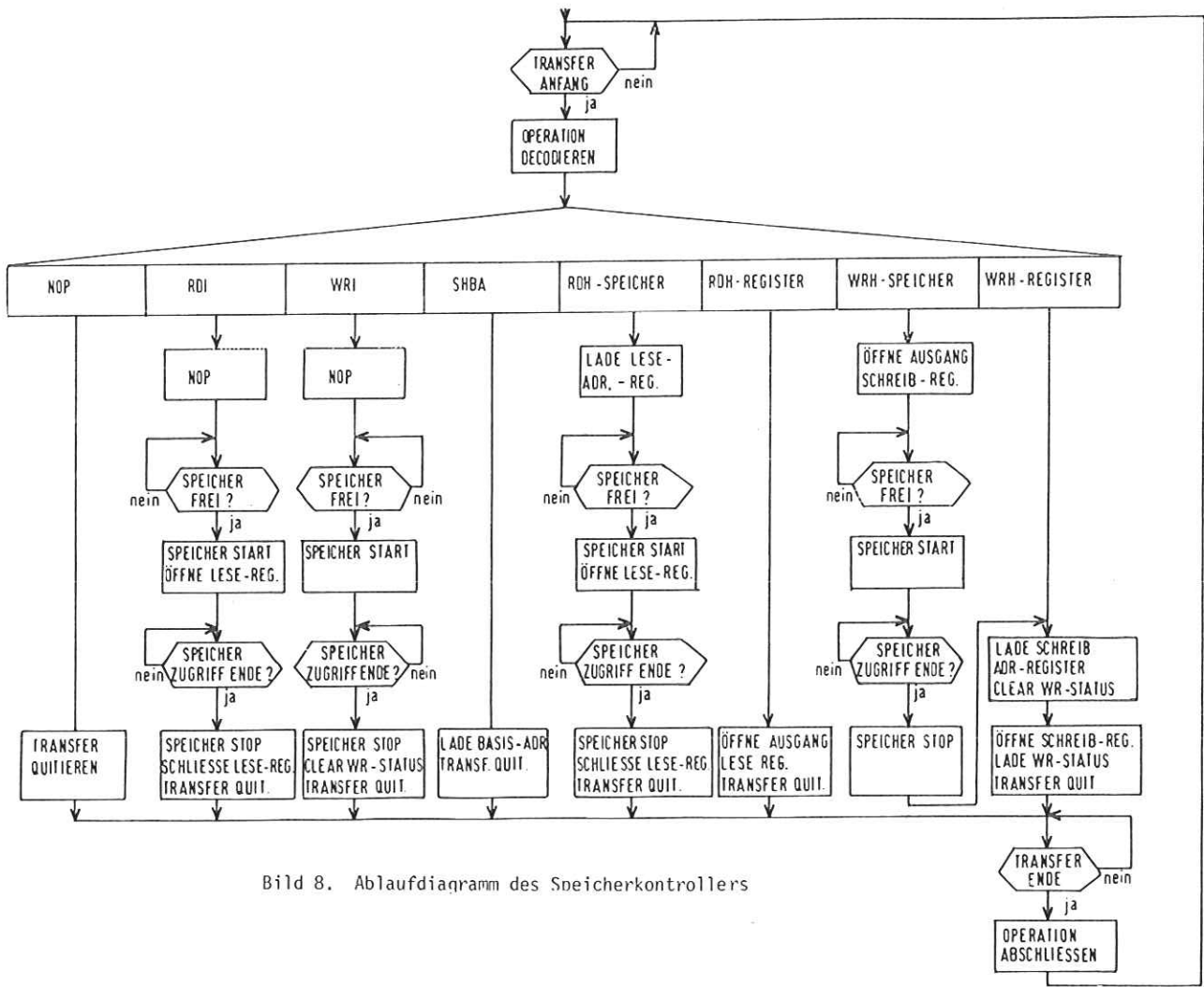


Bild 8. Ablaufdiagramm des Speichercontrollers

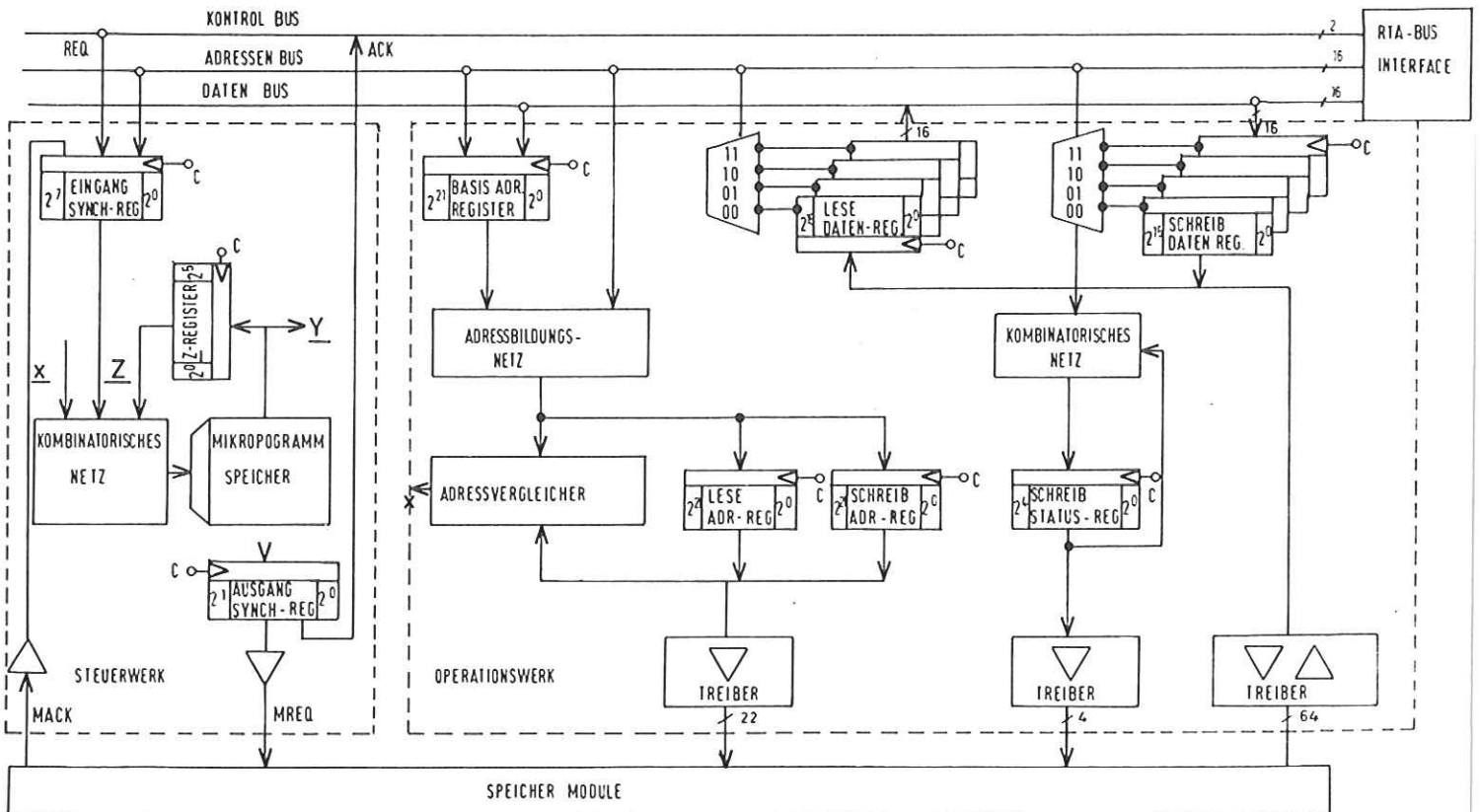


Bild 7. Struktur des Speichercontrollers

In diesem Fall muß für eine Gruppe von vier Transferaufträgen nur ein einziges Mal zum Speicher zugegriffen werden. Dies ist genau die Situation, die im Falle der zeitkritischen Prozessoren wie der Echtzeit-Digitisierer oder der Video Display Prozessor (2) auftritt.

Das Steuerwerk ist als Mikroprogrammwerk realisiert. Um eine kurze Reaktionszeit des Speichercontrollers zu gewährleisten, wurde das Steuerwerk automatentheoretisch betrachtet als ein Mealy-Automat implementiert. Das Ablaufdiagramm in Bild 8 stellt den internen Aktionsablauf des Speichercontrollers dar. Es zeigt, daß der Speichercontroller zwei Klassen von Anweisungen hat.

1. Anweisungen für die Festlegung der virtuellen Speicherorganisation

- SHBA: = SET Basisadreibregister. Die SHBA-Anweisung bewirkt das Laden des 22 Bits-langen Basisadreibregisters. Die Bildung der physikalischen Speicheradresse findet operationell statt. Es muß bemerkt werden, daß das Adreibbildungsnetz nur Additionen ohne Überträge durchführen kann und dadurch die Seitengröße auf Zweierpotenzen beschränkt werden muß. Die Basisadresse hat somit zwei Abschnitte: einen rechten Abschnitt, der nur "0" beinhalten darf und die Seitengröße bestimmt sowie einen linken Abschnitt, der sowohl "0" als auch "1" beinhalten darf und die Seitennummer bestimmt.

2. Anweisungen für den Speicherzugriff

- RDH: = READ virtuelle Adresse. Die RDH-Anweisung bewirkt das Lesen der spezifizierten virtuellen Adresse. Ob dadurch ein Pipelineregister - oder ein Speichertransfer stattfindet, hängt von der Vorgeschichte der Transferaufträge ab und ist für den Anwender irrelevant. Die Leseadresse wird dabei "modulo Seitengröße" interpretiert.

- WRH: = Write virtuelle Adresse. Die WRH-Anweisung bewirkt das Schreiben der spezifizierten virtuellen Adresse. Auch hier kann der Anwender zwischen einem Pipelineregister - und einem Speichertransfer nicht unterscheiden. Die Schreibadresse wird hier auch "modulo Seitengröße" interpretiert

- WRI: = WRITE IMMEDIATE. Die WRI-Anweisung bewirkt das Schreiben der sich im Schreibpipelineregister eventuell noch befindlichen Daten in den Speicher. Es handelt sich hier um einen Register-Speicher-Transfer. Die WRI-Anweisung wird vornehmlich als Abschlußanweisung am Ende des Schreibens eines Datenblocks verwendet.

RDI: = READ IMMEDIATE. Die RDI-Anweisung bewirkt das Lesen der Daten, die sich im Speicher unter der jeweils gültigen Adresse des Leseadreibregisters befinden, vom Speicher in das Lesepipelineregister. Es handelt sich hier um ein Speicher-Register Transfer. Die RDI-Anweisung wird benutzt, wenn "READ after WRITE"-Operationen stattfinden müssen.

Das Operationswerk und das Steuerwerk bilden hinsichtlich des Aktionsablaufes eine weitere Pipelinestruktur zusammen. Dies bedeutet, daß während der Ausführung eines Mikroprogrammschrittes die Interpretation des folgenden Mikroprogrammschrittes stattfindet. Die Periode eines solchen Mikroprogrammschrittes beträgt bei dem implementierten Speichercontroller 70 nsek. Der Mikroprogramm Speicher enthält zusätzlich zum eigentlichen Mikroprogramm auch ein Selbsttest-Mikroprogramm, so daß eine genaue Diagnose des Speichercontrollers ohne Meßaufwand möglich ist.

3.3. Speichermodul

Der Bildsequenzspeicher ist physikalisch in 8 Blöcke von jeweils 128 kWörter x 64 Bits organisiert. Jeder Speicherblock beinhaltet 8 Speichermodule von jeweils 128 kWörter x 8 Bits, die parallel angeordnet sind. Die Zykluszeit eines Speichermoduls beträgt 400 nsek. Aus Kompatibilitätsgründen mit der alten Speicherorganisation des DVS (siehe (2)) wurde als Ersatz für den sogenannten Farbspeicher noch ein zusätzlicher Speicherblock eingeführt. Dieser Speicherblock kann sowohl über den RTA-Bus als auch parallel mit einem anderen Speicherblock über einen dedizierten Datenweg gelesen werden. Diese Strukturbesonderheit hat hier jedoch keine weiter diskutierbare Bedeutung. Der Bildse-

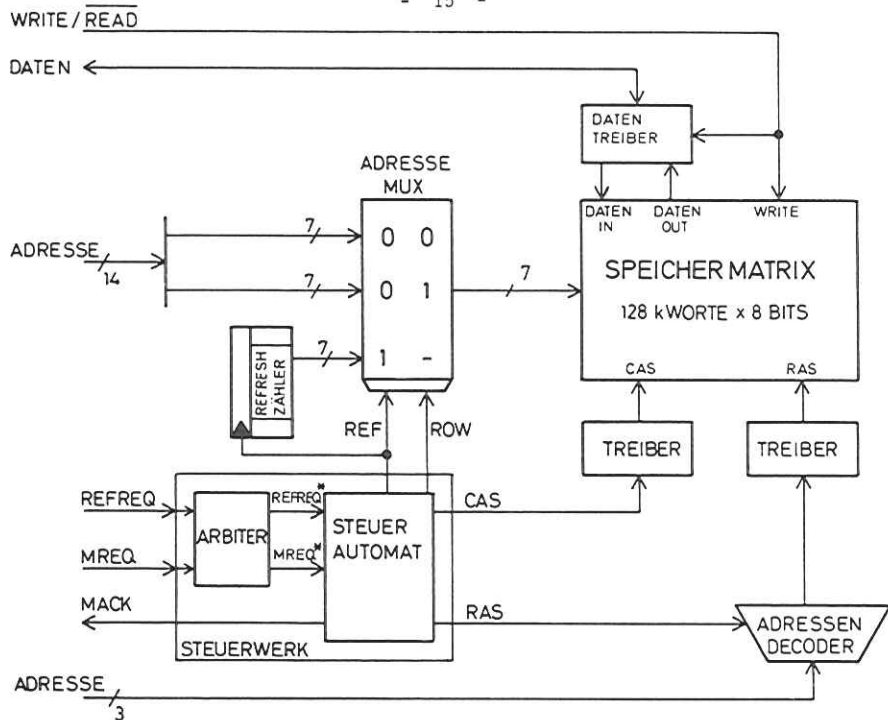


Bild 9. Struktur des Speichermoduls

quenzspeicher setzt sich somit aus insgesamt 72 Speichermodulen zusammen. Die Speichermodule können für die Realisierung verschiedener physikalischer Konfigurationen des Bildspeichers sowohl in Adressen- als auch in Datenrichtung kaskadiert werden.

Die Struktur des realisierten Speichermoduls ist in Bild 9 veranschaulicht. Für die Schnittstelle zum Controller wurde eine einfache parallele Handshake-Schnittstelle (MREQ, MACK) gewählt, so daß der Speichermodul einfach anzusteuern ist. Die Speichermatrix enthält 64 dynamische Speicherbausteine mit einer Speicherkapazität von jeweils 16k Wörtern x 1 Bit. Da diese Bausteine nur im Adreß-Multiplex bezüglich der Taktsignale "RAS" und "CAS" adressiert werden können (5), muß man sowohl die Taktsignale selbst als auch den Adreß-Multiplex mit Hilfe eines vorgeschalteten Steuerwerkes erzeugen. Der Zustandsgraph in Bild 10 beschreibt das realisierte Verhalten des Steuerwerkes, wobei die angegebenen Verzögerungszeiten T1 bis T6 durch die dynamischen Speicherbausteine technisch bedingt sind. Der wegen der dynamischen Speicherbausteine notwendige Refresh (REFREQ) der Speichermatrix kann sowohl synchron als auch asynchron in Bezug auf den Speicherzyklus (MREQ) stattfinden. Im letzteren Fall ist die Entscheidung zwischen einem Speicherzyklus und einem Refreshzyklus dem vorgeschalteten Arbitrer überlassen, dessen Ausgangssignale MREQ⁺ und REFREQ⁺ sich gegenseitig ausschließen. Der physikalisch realisierte Speichermodul ist im Bild 11 gezeigt.

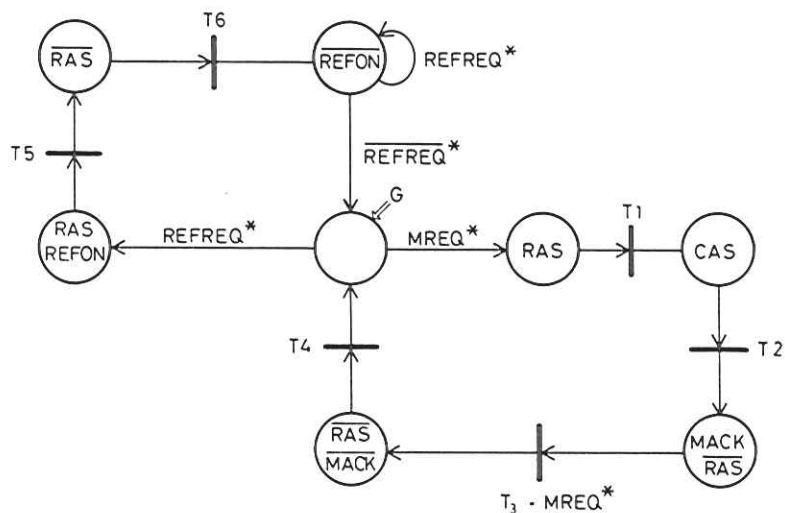


Bild 10. Zustandsgraph für das Steuerwerk des Speichermoduls

4. Zusammenfassung

Der hier beschriebene Bildsequenzspeicher hat sich im Zusammenhang mit der digitalen Verarbeitung von Videobildserien bewährt. Der Programmierer kann sich hauptsächlich auf die Verarbeitungsalgorithmen konzentrieren, ohne den physikalischen Standort seiner Bilder näher betrachten zu müssen. Die Rekonfigurierbarkeit des Bildsequenzspeichers ermöglicht darüber hinaus eine einfache Anpassung an das gewünschte Bildformat und gleichzeitig die maximale Ausnutzung der vorhandenen Speicherkapazität.

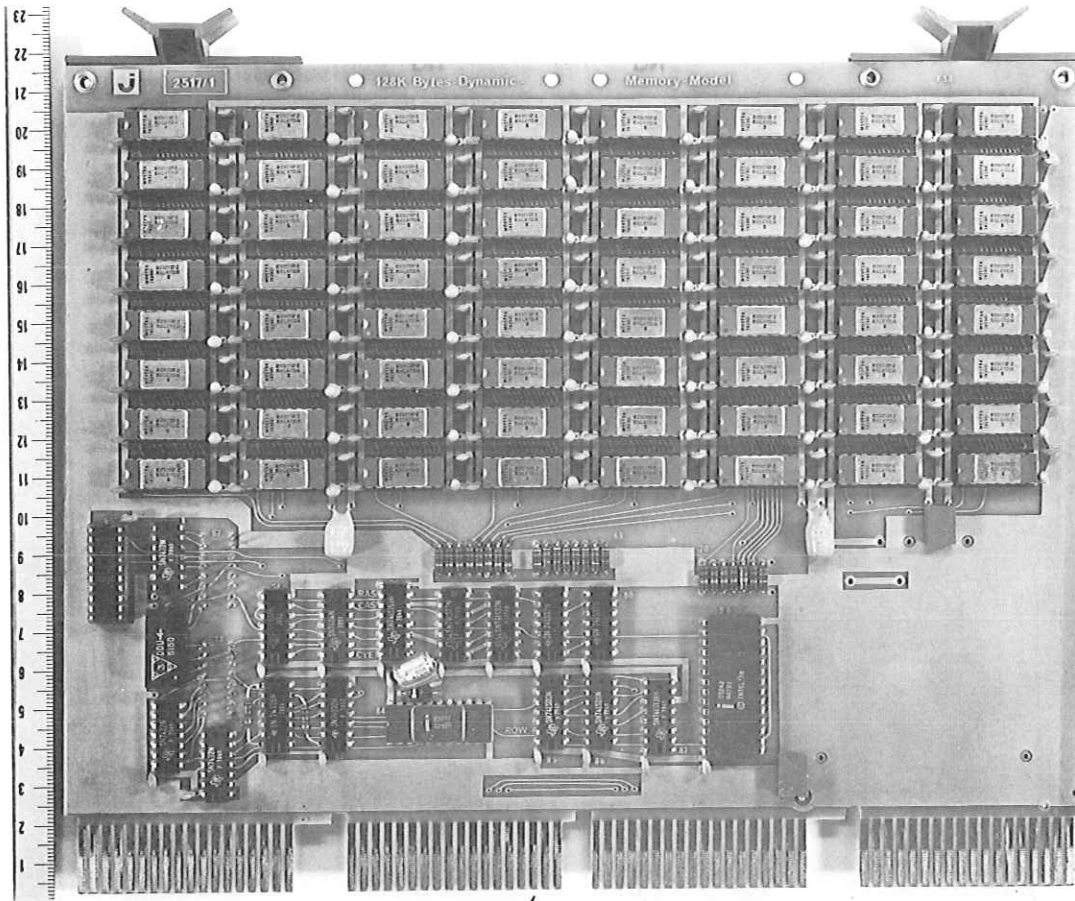


Bild 11 Speichermodulplatine

Der Entwurf des Bildsequenzspeichers für den Anschluß an das vorhandene DVS-System hat gezeigt, daß die Erweiterung des Adreßraumes und des Datenvektors nicht ohne Strukturkompromisse, und die damit verbundenen Leistungsverluste, durchgeführt werden kann. In diesem Zusammenhang sollten zwei Aspekte erwähnt werden :

- Die Erweiterung des Datenvektors von 16 Bits auf 64 Bits und des Adreßvektors von 15 Bits auf 22 Bits in der Speicherebene hat zu erheblichen Übersprechproblemen geführt, die elektrisch - und nicht schaltwerktechnisch - bedingt sind. Nur mit Hilfe besonderer Maßnahmen, die sich sowohl in die Gestaltung der Layouts als auch in den Abschluß der als Flachkabel realisierten Signalwege niederschlagen, konnte eine sichere und störungsfreie Funktion erreicht werden.

- Die einzig mögliche Realisierungsform des Speicherkontrollers als synchrones Schaltwerk in einer asynchronen Umwelt - RTA-Bus und Speicher sind asynchrone Gebilde - hat zu Durchsatzverluste durch die unvermeidlichen Synchronisationszeiten geführt. Diese Synchronisationszeiten bedeuten, daß einerseits die Durchsatzrate auf dem RTA-Bus verringert wird und andererseits, daß die Speicherzykluszeiten verlängert werden.

Diese Aspekte, die für den Entwurf und für die Implementierung des Bildsequenzspeichers einschränkend waren, haben jedoch für die praktische Anwendung der Computer-Angiographie zu keinen nennenswerten Einschränkungen geführt.

5. Literatur

- (1) Höhne, K.H.; Böhm, M.; Nicolae, G.C.: "The Processing of x-Ray Image Sequences", DESY DV-79/01
March 1979
- (2) Nicolae, G.C.; Höhne, K.H.: "Digital Video System for Real-Time Processing of Image Series", DESY DV - 78/02
April 1978
- (3) Nicolae, G.C.; Wendt, S.: "Design of a Microprogrammed Video Display Processor for Real-Time Image Processing",
Euromicro Symposium on Microprocessing and Microprogramming,
Göteborg, August 1979
- (4) Petri, C.A.: "Kommunikation mit den Automaten"
Dissertation, Darmstadt, 1962
- (5) Fa. Mostek, Datenblatt MK 4116, 1976

