

3D-Echtzeit Grafikrechner mit UNIX 32-Bit Micro

Trotz des Erscheinens leistungsfähiger Grafikprozessoren bieten weiterhin nur komplexe Systemarchitekturen Lösungen für die 3D-Echtzeitsimulation. Bei vertretbarem Aufwand eignet sich dafür besonders das Konzept der Vektormaschine. Zusätzlich werden die Grafik-Arbeitsplätze immer häufiger mit einem hohen Maß an lokaler Intelligenz für die Applikations-Software ausgestattet. Für die Integration in einen Arbeitsplatz, wie das IMI555, bietet sich das auf dem Chip Satz der Serie 32000 basierende ICM-3216 an, da es in den Leistungsbereich einer VAX vorstößt.

Die Entwicklung neuer Workstations darf heute nicht nur unter dem Aspekt der Grafik geschehen. Vielmehr hat sich mit den UNIX Implementationen System V und Berkely 4.2 ein Standard auf fast allen intelligenten Grafikarbeitsplätzen etabliert. Schon in der Konzeptionsphase ist die Entscheidung zu treffen, ob ein neuer UNIX Microrechner zu entwickeln ist, oder eine Lösung vom Markt eingekauft und integriert werden kann. Um die Kosten niedrig zu halten und sich ganz auf die Entwicklung der Grafikmaschine konzentrieren zu können, entschied sich Interactive Machines für die Integration eines modernen Microrechnerkonzeptes. Besonders interessant erscheinen die Single-Board Rechner, die niedrigen Preis und hohe Leistung vereinigen. Im Gegensatz zu VME- oder MULTIBUS2-Micros benötigen sie keine Backplane und Buslogik - Speicherzugriffe können ohne Wartezyklen gefahren werden.

Die Wahl fiel auf das ICM-3216 von National Semiconductor, dessen Konzept neue und einzigartige Funktionen bietet.

Modular ohne Busplatine

Das ICM-3216 - ICM für Integrated Computer Module - vereinigt bis auf den Speicher alle Komponenten eines UNIX Micros auf einem Modul, der CPU-Platine. Speicher-Module werden über einen Speicherbus auf die Rückseite der CPU-Platine aufgesteckt, wobei die Verbindung zwischen den Platinen als 96-polige VG Leiste ausgeführt ist. Mit zwei Speicher-Modulen können bis zu 8 MByte Arbeitsspeicher bei 10MHz ohne Wartezyklus betrieben werden.

Kern des CPU-Moduls ist das CPU-Cluster, bestehend aus NS32016 CPU, NS32201 TCU (Timing Control Unit), NS32202 ICU (Interrupt Control Unit), NS32081 FPU (Floating Point Unit) und NS32082 MMU (Memory Management Unit). Bis zu fünf Peripheriegeräte wie Terminals, Drucker, Plotter, Joysticks, Trackballs oder Mäuse können über die vier RS232C und den Centronics Port angeschlossen werden. Das Small Computer System Interface (SCSI) ist Schnittstelle für Festplatten, 1/4" Streamer und Floppylaufwerke. Eine Z80A CPU und ein NCR 5385E SCSI Controller entlasten dabei den NS32016 Prozessor von der I/O Steuerung.

Um weitere Peripherie an das ICM-3216 anschließen zu können, steht zusätzlich zum lokalen Speicherbus ein synchroner 16 bit Bus, der Minibus auf, dem CPU-Modul zur Verfügung.

Bild 1: ICM-3216 Blockschaltbild

MiniBus

Das Interface zwischen CPU und MiniBus wird durch den MiniBus Interface Chip (MBIC), ein NSC 6224 Gate Array, unterstützt. Der MBIC setzt Lese- und Schreibzyklen der CPU in MiniBus Zyklen um, bedient Interrupts, generiert Wartezyklen zur Synchronisation der CPU mit dem MiniBus und stellt Steuersignale für lokalen Speicher zur Verfügung.

Bild 2: MiniBus Signale

Der Minibus unterstützt synchrone 16 Bit und Dual-Ported Speicherzugriffe bei einer Taktfrequenz von 8 MHz und asynchrone 8 Bit und 16 Bit I/O Operationen. Im synchronen Betrieb hat er beim Schreiben eine Übertragungsrate von 5.3 MByte/sec und beim Lesen von 4.0 Mbyte/sec. Bis zu 8 Bus Master können am MiniBus betrieben werden. Der Adressbereich des MiniBus umfaßt 4 MByte die sich wie folgt aufteilen:

00 0000 --> 3D FFFF

16 Bit Speicher Bereich (synchroner Betrieb)

3E 0000 --> 3E FFFF

8 Bit I/O Bereich (256 Ports)

Diese 64 kByte des Adressbereiches werden in asynchrone I/O Zyklen übersetzt. Dabei enthalten AD15 - AD08 die Portadresse und AD07 - AD00 die Daten. Da Portadresse und Daten parallel mit einem Transfer übertragen werden sind nur 256 Ports definiert.

3F 0000 --> 3F 7FFF

16 Bit I/O Bereich (16384 Ports)

Diese 32 KByte des Adressbereiches werden in asynchrone erweiterte I/O Zyklen übersetzt. Mit AD14 - AD00 wird der I/O Port für den 16 Bit Transfer adressiert.

3F 8000 --> 3F BFFF

physikalischer Adressbereich

Diese 16 KByte des Adressbereiches sind für die Adressierung von Modulen, die eine vollständige Implementation des MiniBus besitzen, reserviert.

3F C000 --> 3F FFFF

freier Adressbereich

In vielen der gegenwärtigen Microprozessorsysteme sind die letzten 16 Kbyte des Adressbereichs mit Interruptvektoren und Systemtabellen belegt. Da man bei der Definition davon ausging, daß der MiniBus auf die letzten 4 Mbyte eines Systems gelegt wird, sind die letzten 16 KByte des MiniBus nicht belegt.

Die 51 Signale des MiniBus sind wie der Speicher-Bus über 96 polige VG-Leisten ausgeführt. Ein Subset des Busses (46 Signale) ist auf die Reihen A und C des VG-Steckers gelegt, damit der Anschluß von Modulen über 64 poliges Flachbandkabel möglich ist. Dieses Design erspart eine Busplatine (die VG-Stecker und Leisten sind der eigentliche Bus) und kommt mit einem minimalen Aufwand an Steuerlogik aus. - im Gegensatz zum VME oder MULTIBUS2 -

Bild 3: Minimales 16 Bit CPU MiniBus Interface mit MBIC

Bild 4: Einfaches 8 Bit Slave Interface am MiniBus

Bei der Integration in den Grafikrechner wurden nur die synchronen Lese- und Schreiboperation und der asynchrone 8 Bit I/O Transfer des MiniBus implementiert. Das Interface zwischen Systembus und MiniBus wurde mit einer T-State Maschine realisiert, die zum Minibus hin einen Subset des MBIC und zum Systembus einen Subset eines erweiterten IEEE-696 Busprotokolls zur Verfügung stellt. Über den MiniBus sind immer nur 2 MByte Seiten des Systembus direkt ansprechbar. Um den gesamten Speicherbereich des Systembus von 16 MByte adressieren zu können, wird auf dem Interface ein Register als I/O-Port vom Minibus aus mit der Seitenadresse geladen. Aus den drei höchstwertigen Bit des Registers werden die Adresssignale AD21-AD23 für den Systembus generiert. Diese Implementation des MiniBus erlaubt es in den Display- und Refreshspeicher mit 3 MByte/sec zu schreiben und aus ihm mit 2,5 MByte/sec zu lesen.

Bild 5: synchroner Schreibzyklus auf dem MiniBus

Bild 6: synchroner Lesezyklus auf dem MiniBus

Bild 7: asynchroner 8 Bit I/O Zyklus auf dem MiniBus

Schnelle 3D Grafik

Der Grafikkrechner selber beinhaltet eine leistungsfähige Multiprozessorarchitektur mit Pipelining für die verschiedenen Aufgaben wie Darstellung, Refresh, Steuerung und Transformation.

Die Displayliste wird in einer PHIGS-ähnlichen Sprache und Struktur vom lokalen Mikroprozessor im 2 MByte großen Displaymemory aufgebaut und verwaltet. In die Displayliste werden die dreidimensionalen Beschreibungen der Objekte mit den dazugehörigen Transformationen in Fließkommaformat abgelegt. Für Transformationen und Bildmanipulationen stehen 3D-Befehle wie Rotation, Verschiebung, Perspektive, Mehrfachansicht, Maßstabsänderung, Clipping, Zoom, Depth Cueing, Sun Vector und Back-Face Hidden Line Rejection zur Verfügung.

Ein mikroprogrammierter bipolarer Bitslice-Prozessor, der Displayprozessor, arbeitet die Displayliste ab und legt die Bildvektoren im Refreshspeicher ab. Dabei steuert er zwei Spezialprozessoren, den Addierer/Subtrahierer und den Multiplizierer/Dividierer, die ebenfalls aus bipolaren Bitslices bestehen und mit einer Rechenleistung von 8 Mio. Fließkommaoperationen pro Sekunde die Geometriedaten der Objekte transformieren. Diese Kombination ermöglicht die Transformation von 200000 3D-Vektoren pro Sekunde und eine dynamische Darstellung von komplexen dreidimensionalen Objekten.

Der 256 KByte Refreshspeicher, in dem bis zu 32000 transformierte Vektoren abgelegt werden, wird im Double-Buffered Mode betrieben. In jeweils eine 128 KByte große Seite schreibt der Displayprozessor das neu berechnete Bild, während der Displaygenerator über den Refreshcontroller aus der anderen Seite Vektoren liest und den Random Stroke Monitor treibt. Sobald der Displayprozessor mit der Berechnung des neuen Bildes fertig ist, werden die Seiten umgeschaltet. Dadurch können beide Seiten unbeeinträchtigt auf den Speicher mit voller Geschwindigkeit zugreifen und ein kontinuierlicher Bildrefresh ist gewährleistet. Der Refreshspeicher

besitzt noch einen dritten Port zum Systembus. Über diesen Systembus-Port können die Vektoren, die als Endpunktpaare mit Farbinformation in einem 32 Bit Wort abgelegt sind, vom lokalen Mikroprozessor ausgelesen und auf einen Plotter ausgegeben werden.

Bild 8: Refreshspeicherformat bei schwarz/weiß Darstellung

Bild 9: Refreshspeicherformat bei Farbdarstellung

Da sich der Refreshspeicher als Buffer zwischen Displayprozessor und Displaygenerator befindet, ist der Bildrefresh nicht, wie bei vielen anderen Vektormaschinen, von der Rechengeschwindigkeit des Displayprozessors abhängig. Ein Bild muß immer erst dann neu berechnet werden, wenn die Displayliste geändert wurde. Selbst komplexe Darstellungen mit 32000 Vektoren von 5 mm Länge werden flimmerfrei und mit einer konstanten Bildwiederholrate von 50 Hz gezeichnet. Der Displayprozessor trägt, nachdem er die Displayliste abgearbeitet hat, im letzten Refreshspeicherwort die Anzahl der darzustellenden Move- und Drawvektoren ein. Der Refreshcontroller übernimmt dieses Wort und übergibt die Wertepaare der Endpunkte an den Displayprozessor kontinuierlich. Die digitalen X, Y und Farb/Intensitätswerte werden vom Displaygenerator in analoge Signale umgewandelt. X- und Y-Wert sind 12 Bit lang. Damit können insgesamt 4096 x 4096 Punkte mit jeweils 128 Graustufen bzw. 8 Farben mit jeweils 16 Intensitäten adressiert werden. Der Elektronenstrahl des Bildschirms wird direkt im Random-Stroke Verfahren mit einer konstanten Geschwindigkeit von 14 mm/usec getrieben.

Bis zu vier Farb- und S/W-Monitore lassen sich gemischt gleichzeitig am Displaygenerator betreiben. Dabei können auf den Monitoren verschiedene Bilder oder parallel dasselbe Bild dargestellt werden.

Bild 10: Blockschaltbild der Vektormaschine

Anwendung

Vektormaschinen werden in der Luft- und Raumfahrtindustrie, bei Animationen in der Filmindustrie und in der Automobilindustrie für dynamische Echtzeit-Simulationen eingesetzt. Durch das ICM-3216 verfügt der Arbeitsplatz über eine große Massenspeicherkapazität - bis zu 500 MByte Magnetplatte und ein leistungsfähiges virtuelles Betriebssystem, UNIX System V. Die Applikationssoftware wird fast ausschließlich in "C" und FORTRAN 77 geschrieben. Der NS32081 Floatingpointprozessor stellt für die Anwendungsprogramme die Fließkommarechenleistung zur Verfügung. Die zu animierenden bzw. zu simulierenden dreidimensionalen Objekte werden am Anfang mit den Grundtransformationen in die Displayliste eingetragen. Danach berechnet der lokale Mikrorechner nur noch die Parameter der einzelnen Transformationsmatritzen und trägt sie in der Displayliste ein. Die Grafikmaschine übernimmt dann die eigentliche Aufgabe der Objekttransformation. Um der Forderung nach Oberflächendarstellung zu entsprechen arbeitet man an Raster-Frame-Speichern und Rastercontrollern, die direkt aus dem Displayprozessor getrieben werden. Diese Kombination wird in Rastertechnologie bei geringerer Auflösung ähnliche Geschwindigkeiten erreichen, wie die Random-Scan-Technologie.

```

=====
I Mnemonic      I Beschreibung      I
=====
I AD21 - AD00   I Address/Data/Control I
I BCOD0, BCOD1 I Bus Code          I
I BUSERR        I Bus Error         I
I BUSPAR        I Bus Parity        I
I BREQ0 - BREQ7 I Bus Requests      I
I MBICCLK       I MBIC Bus Clock    I
I BCLK, BCLK_2  I System Bus Clocks I
I INT00 - INT07 I Interrupt Requests I
I IO            I I/O Transfer      I
I PFAIL         I Power Fail        I
I PWAIT         I Peripheral Wait    I
I RESET         I Reset              I
I RFRSHTIM      I Refresh Interval   I
=====

```

Bild 2 MiniBus Signale

.31	.30	24.23	12.11	0.
move/		y	x	
draw	Intensität	Vektor	Vektor	
Flag		Wert	Wert	

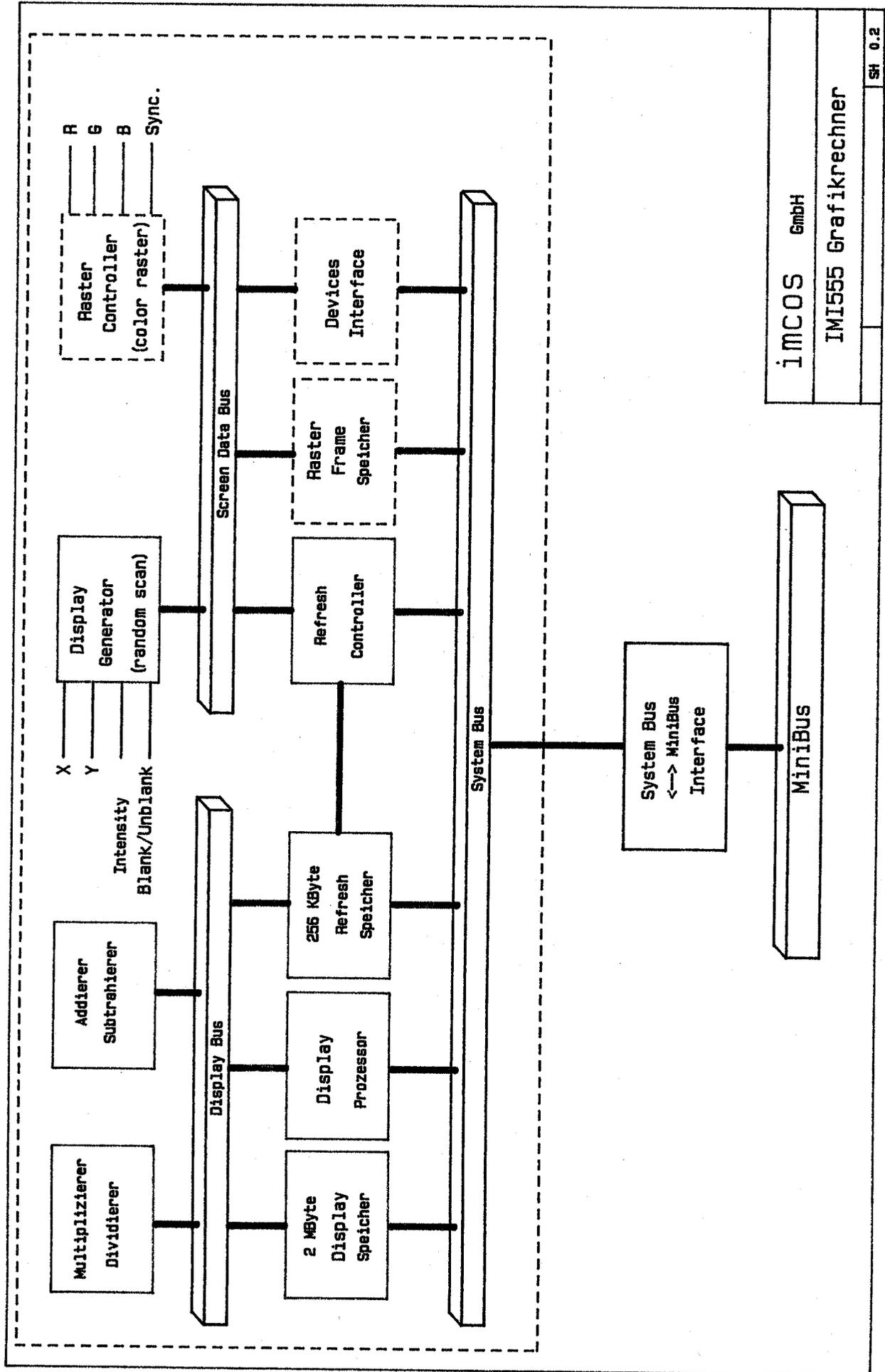
Bild 8: Refreshspeicherformat bei schwarz/weiß Darstellung

.31	.30	27.26.25.24.23	12.11	0.
move/	r g b	y	x	
draw	Inten- o r l	Vektor	Vektor	
Flag	sität t ü a	Wert	Wert	
	n u			

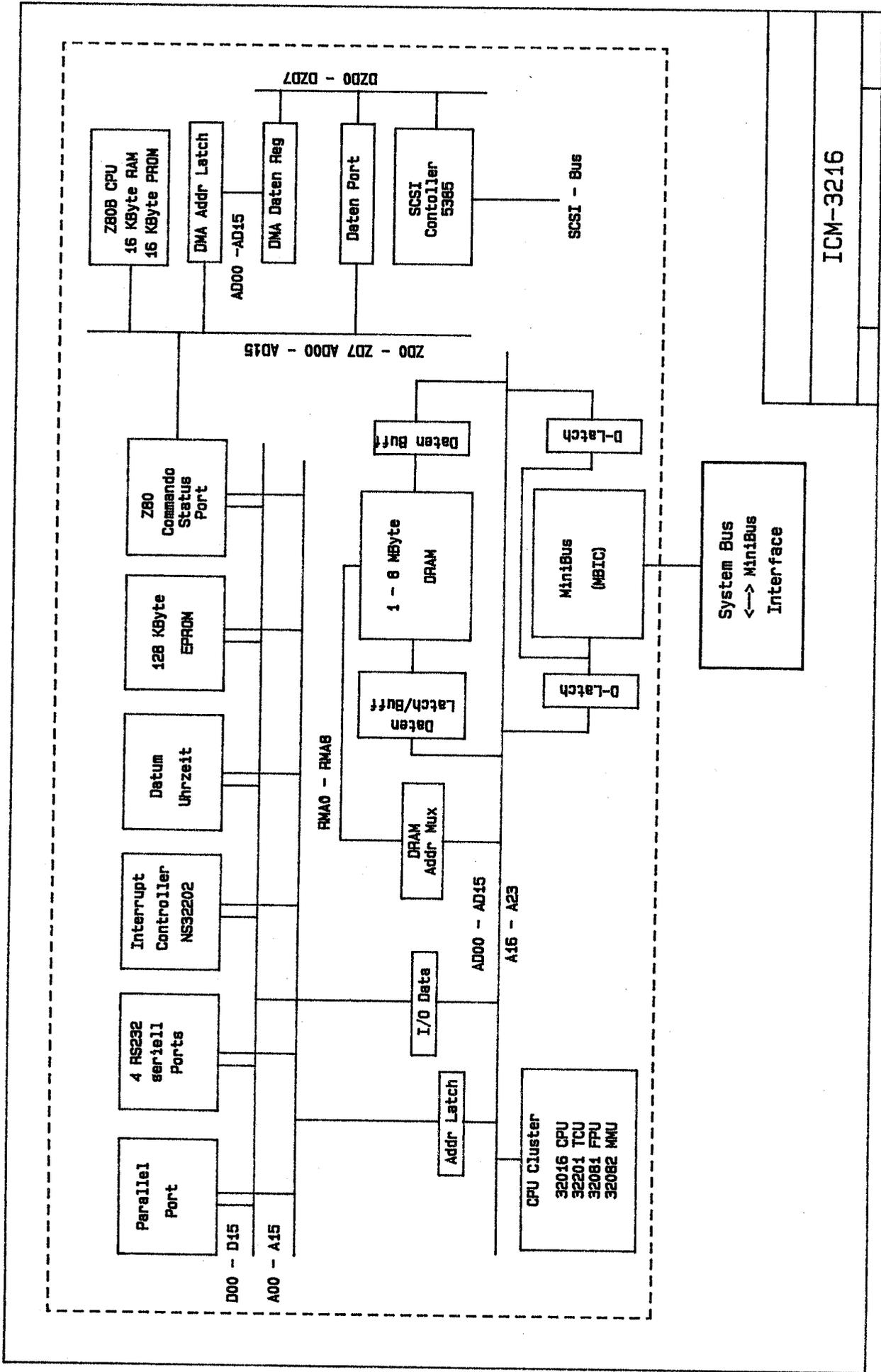
Bild 9: Refreshspeicherformat bei Farbdarstellung.

Zur Person:

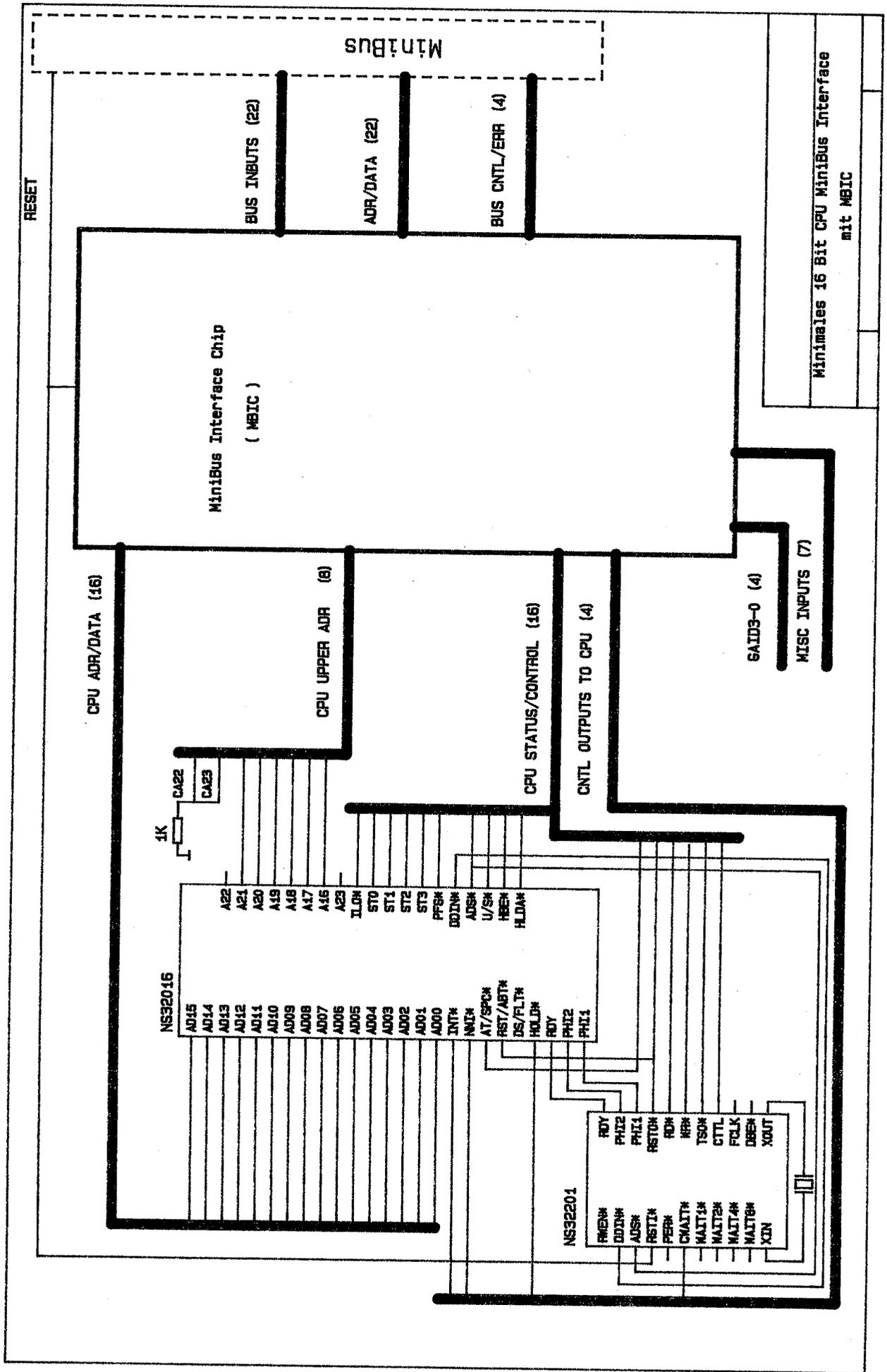
Cand. Ing. Uwe Bannow, geboren in Hamburg. Studium der Elektrotechnik an der TH Darmstadt. Praktikum bei der Fa. Interactive Machines Inc., CA, U.S.A. im Sommer 1984 und 1985. Mitarbeit bei der Integration des ICM-3216 in die Workstation IMI555.



IMCOS GmbH
IMI555 Grafikrechner



ICM-3216



Bus Clock		0	1	2	3
IO [~]	XXXXX				XXXXXXXXXXXXX
BCOD1 [~]	XXXXX		1		XXXXXXXXXXXXX
BCOD0 [~]	XXXXX	1 (master)	0 (slv)		XXXXXXXXXXXXX
AD21-19	XXXXX	M A21-19	M 110, ev, odd wait, burst	M 110, ev, odd burst	XXXXXXXXXXXXX
AD18-16	XXXXX	M A18-16	S 0 - wait	S data word parerror	XXXXXXXXXXXXX
AD15-00	XXXXX	M A15-01, wr [~]	S XXXX	S data	XXXXXXXXXXXXX
BUSPAR	XXXXX	M A21-0 par	XXXXXXXXXXXXX	S A15-0 par	XXXXXXXXXXXXX
Bus Clock		0	1	2	3

M Master

S Slave

burst Burst request

ev even

odd odd

par Parity

synchroner Lesezyklus auf dem MiniBus

Bus Clock		0	1	2	3
IO [~]	XXXXX			XXXXXXXXXXXXX	XXXXXXXXXXXXX
BCOD1 [~]	XXXXX		1	XXXXXXXXXXXXX	XXXXXXXXXXXXX
BCOD0 [~]	XXXXX	1 (master)	0 (slv)	XXXXXXXXXXXXX	XXXXXXXXXXXXX
AD21-19	XXXXX	M A21-19	M 110, ev, odd wait, burst	XXXXXXXXXXXXX	XXXXXXXXXXXXX
AD18-16	XXXXX	M A18-16	S data acc data burst	XXXXXXXXXXXXX	XXXXXXXXXXXXX
AD15-00	XXXXX	M A15-01, wr [~]	S Data	XXXXXXXXXXXXX	XXXXXXXXXXXXX
BUSPAR	XXXXX	M A21-0 par	S A15-0 par	XXXXXXXXXXXXX	XXXXXXXXXXXXX
BUSERR	XXXXX	XXXXXXXXXXXXX		error ?	XXXXXXXXXXXXX
Bus Clock		0	1	2	3

M Master

S Slave

burst Burst request

ev even

odd odd

par Parity

synchroner Schreibzyklus auf Minibus

Bus Clock		0	1	2	3	4	5	6	
IO [~]	xxxxx								xxxxx
BCOD1 [~] , 0 [~]	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx
AD15-00		port (15-08) / data (07-00)							xxxxx
AD17 - AD16 wr [~] rd [~]	xxxxx			pwait [!] ?					xxxxx
Bus Clock		0	1	2	3	4	5	6	

asynchroner 8 Bit I/O Zyklus auf dem MiniBus