

Nintendo GameBoy



Bemerkung der Autoren

Die Information in dieser Ausarbeitung werden ohne Rücksicht auf einen Patentschutz veröffentlicht.

Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen.

Trotzdem können Fehler nicht vollständig ausgeschlossen werden.

Die Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgend eine Haftung übernehmen.

Für Verbesserungsvorschläge und Hinweise auf Fehler sind die Autoren dankbar.

Inhaltsverzeichnis:

1. Einleitung	4
2. Der interne Aufbau des GameBoy	5
2.1 Der CPU-Kern des GameBoy	6
2.1.1 <i>Der geänderte OpCode des GameBoy</i>	7
2.2 Die Speicherbelegung des GameBoy	9
2.2.1 <i>Die Speicherauswahllogik des GameBoy</i>	10
2.2.2 <i>Die Speicherauswahl des GameBoy</i>	10
2.3 Der Soundgenerator des GameBoy	11
2.3.1 <i>Die Soundgeneratorschaltkreise des GameBoy</i>	12
2.3.2 <i>Die Mono/Stereo Auswahleinheit des GameBoy</i>	13
2.4 Das Videosystem des GameBoy	14
2.4.1 <i>Der Aufbau von Hintergrundbildschirm und Fenster des GameBoy</i>	15
3. Die Cartridge des GameBoy	16
4. Die serielle Schnittstelle des GameBoy	18
5. Die PowerUp-Sequenz des GameBoy	19
6. Die Entwicklungsoberflächen für den GameBoy	20
7. Quellen	22
8. Anhang	23
Anhang A (Sound Register)	23
Anhang B (Display Register)	26
Anhang C (SIO Register)	28
Anhang D (Schaltpläne)	29

1. Einleitung

Der GameBoy ist, nach Game&Watch, das zweite Handheld von Nintendo. Er erschien am 18. April 1989 zum ersten Mal auf dem japanischen Markt. Obwohl im Erscheinungsjahr schon zwei Handheld, das Atari Lynx und das Turbo Express GT von NEC auf dem Markt waren, ist der GameBoy fast 10 Jahre später immer noch Marktführer, was Handheld betrifft.

Wie konnte sich der GameBoy mit einem vier Graustufen-Display durchsetzen, obwohl die Konkurrenz damals schon mit Farbbildschirmen arbeitete.

Atari war der erste, der sein Handheld auf dem Markt hatte. Dieses war schon mit einem Farbbildschirm ausgestattet, aber der gravierendste Nachteil war die Größe des Handheld. NEC hatte in diesem Punkt Vorteile und zudem auch eine riesige Palette an Spielen, da man die Spiele der großen NEC-Konsole auch auf dem Handheld spielen konnte. Trotzdem konnte sich NEC auf dem Markt nicht durchsetzen, da Nintendo gerade in den USA besonders populär war. Dies wird besonders deutlich, wenn man bedenkt, daß jede dritte Familie in den Staaten ein NES besaß. Damit war der Erfolg vom GameBoy eigentlich klar und dieser Erfolg hält bis heute an.

Für den GameBoy gibt es mittlerweile nicht nur Spiele, sondern auch die verschiedensten Zusatzgeräte, wie z.B. die vor kurzem erschienene Kamera, der Drucker und die externe Tastatur mit Textverarbeitungssoftware.

Fast 10 Jahre nach Markteinführung brachte Nintendo einen verbesserten GameBoy heraus, den GameBoy Pocket, kurz darauf wurde auch eine Variante mit Farbdisplay auf den Markt gebracht, der GameBoy Color.

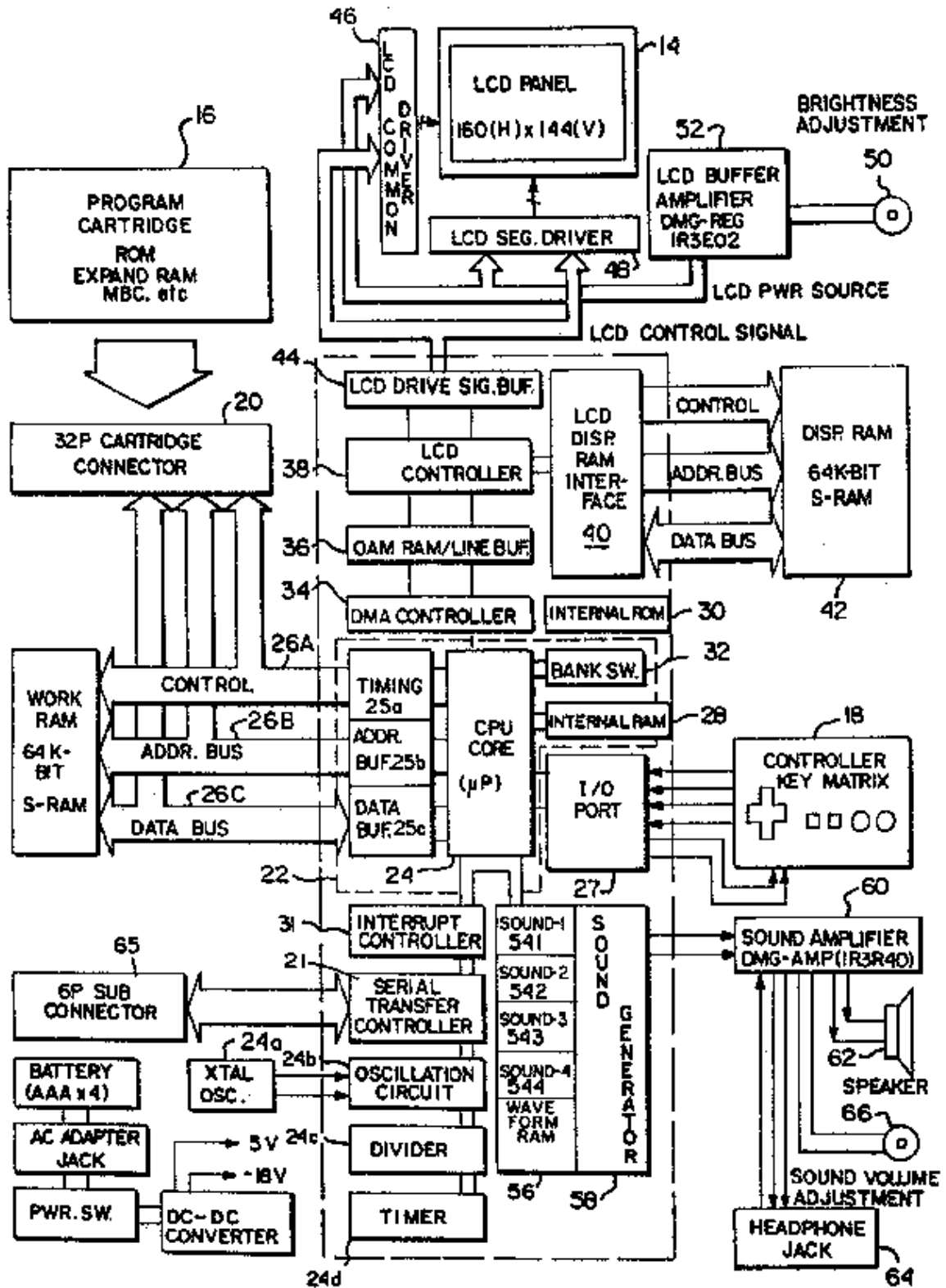
Folgend ein Vergleich zwischen den GameBoy (Die Unterschiede zum Gameboy Classic sind farblich markiert):

	GameBoy Classic	GameBoy Pocket	GameBoy Color
CPU	8bit (Vergleichbar mit Z80) mit 4.194304 MHz	8bit (Vergleichbar mit Z80) mit 4.194304 MHz	8bit (Vergleichbar mit Z80) mit 4.194304 / 8.388 MHz
RAM	8 KByte	8 KByte	32 KByte
Video RAM	8 KByte	8 KByte	16 KByte
Auflösung	160x144 Pixel 20x18 Zeilen	160x144 Pixel 20x18 Zeilen	160x144 Pixel 20x18 Zeilen
Farbe	4 Graustufen	4 Graustufen	256 Farben
Sound	4 Kanäle Pseudo Stereo	4 Kanäle Pseudo Stereo	4 Kanäle Pseudo Stereo
Power	DC 6.0V 0.7W	DC 3.0V 0.7W	DC 3.0V 0.6W

In dieser Ausarbeitung halten wir uns vorrangig an den GameBoy Classic, da das Grundprinzip dieses GameBoy für alle weiteren GameBoy dasselbe geblieben ist.

2. Der interne Aufbau des GameBoy

Der GameBoy besteht aus den verschiedensten Komponenten, wie der schematische Aufbau im folgenden Bild zeigt.



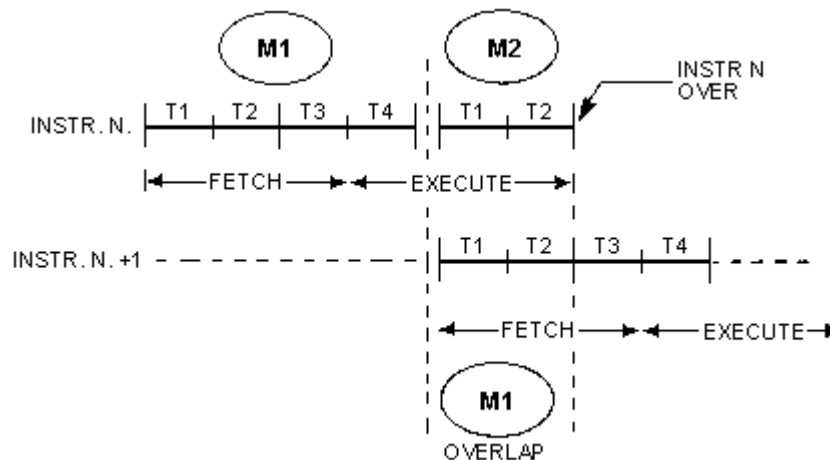
Quelle: U.S. Patent 5.184.830 Fig. 4

In der Mitte des Bildes ist der CPU-Baustein LR 35902 zu erkennen. Dieser beinhaltet neben dem CPU-Kern den 4-Kanal-Soundgenerator, den LCD-Controller, einen Hardwaretimer, sowie die DMA-, Interrupt- und I/O Controller. Außerhalb dieses CPU-Bausteins verfügt der GameBoy über zwei verschiedene Systembusse. Der erste Systembus dient zur Kommunikation mit dem Arbeitsspeicher sowie der externen Programm-Cartridge. Der zweite Systembus dient ausschließlich zur Kommunikation zwischen LCD-Controller und Video-RAM. Des Weiteren sind am CPU-Baustein das Display, die Tastatur, der Lautsprecher, sowie die 6polige serielle Schnittstelle angeschlossen. Diese Schnittstelle dient z.B. für den Anschluß an einen zweiten GameBoy.

2.1 Der CPU-Kern des GameBoy

Der CPU-Kern des GameBoy ist mit dem des Z80 vergleichbar. Die Unterschiede liegen in den Registern, sowie bei dem OpCode. Da Nintendo nähere Auskünfte über den CPU-Kern und seine Architektur verweigerte, gehen wir an dieser Stelle kurz auf die Architektur des Z80 ein und versuchen die Unterschiede zum GameBoy zu verdeutlichen.

Der Z80 ist mit einer einfachen Pipelinearchitektur versehen, die das „fetch/execute overlapping“ ermöglicht. Das bedeutet, daß es möglich ist, die nächste Anweisung aus dem Speicher zu holen, während die aktuelle Anweisung gerade ausgeführt wird.



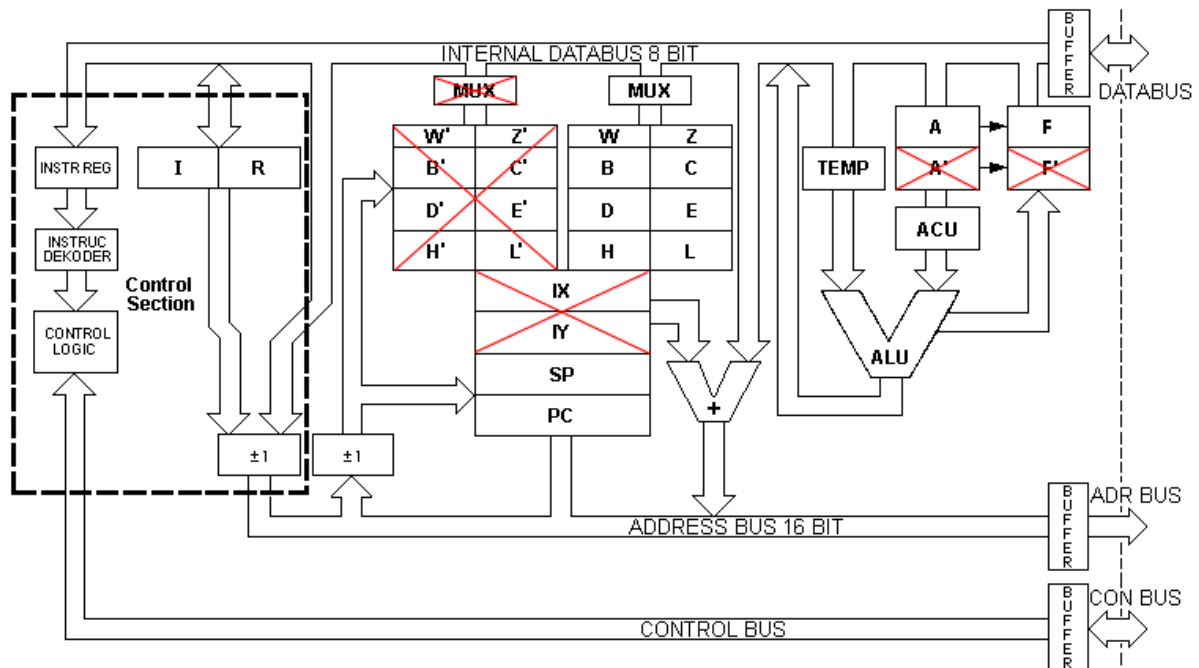
Quelle: <http://www.geocities.com/SiliconValley/Peaks/3938/Z80ARKI.HTM> Fig. 1

Eine typische Eigenschaft der CISC-Prozessoren ist die variable Länge der Anweisungen. Beim Z80, wie beim GameBoy, kann die Länge der Anweisungen zwischen 1-4 Byte betragen. Um die Geschwindigkeit der Z80 CPU zu erhöhen, beträgt die Länge der meisten Anweisungen jedoch nur 1Byte.

Eine vollständige Auflistung aller Anweisungen würde an dieser Stelle den Rahmen dieser Ausarbeitung sprengen. Wir verweisen daher auf das Buch:

„Programming the Z80“ von Rodney Zaks
SYBEX (ISBN: 0-89588-094-6)

Im folgenden Bild sollen die Unterschiede zwischen dem internen Aufbau des GameBoy CPU-Kern und dem des Z80 verdeutlicht werden.



Quelle: <http://www.geocities.com/SiliconValley/Peaks/3938/Z80ARKI.HTM> Fig. 2

Der GameBoy verfügt, im Gegensatz zum Z80, nur über 10 8Bit-Register und über 2 16Bit-Register. Die beiden 16Bit-Register sind zum einen der StackPointer(SP) und der ProgramCounter(PC). Die beiden Indexregister IX und IY des Z80 wurden genauso entfernt, wie die „Shadow-Register“ BC', DE', HL', AF'.

Das interne Abarbeiten der Befehle erfolgt ansonsten wie beim Z80.

2.1.1 Der geänderte OpCode des GameBoy

Da einige Register des Z80 nicht mehr zur Verfügung stehen, wurden einige OpCodes entfernt, bzw. mit neuen Funktionen belegt.

Bei dem OpCode wurden folgende Anweisungen entfernt:

- Alle Befehle, welche die IX und IY Register betreffen.
- Alle IN/OUT Anweisungen.
- Alle „exchange“ Anweisungen.
- Alle Befehle mit der Vorsilbe ED
(Welcher OpCode genau damit gemeint ist, konnten wir aus den uns zur Verfügung stehenden Unterlagen nicht ermitteln)
- Alle Sprungbefehle, Unterprogrammaufrufe und Rücksprünge in Abhängigkeit von Parität bzw. Überlauf und des Vorzeichenflags.

Folgender OpCode wurden gelöscht:

OpCode	Z80 Anweisungen
D3	OUTA (byte)
DB	INA (byte)
DD	Prefix DD
E3	EX HL, (SP)
E4	CALL PO,word
EB	EX DE,HL
EC	CALL PE,word
F2	JP P,word
F4	CALL P,word
FC	CALL M,word
FD	Prefix FD

Folgender OpCode hat eine neue Funktion erhalten:

OpCode	Z80 Anweisungen	GameBoy Anweisungen	
08 xx xx	EX AF,AF'	LD (word),SP	Save SP at given address
10 xx	DJNZ offset	STOP	Meaning unknown
22	LD (word),HL	LD (HLI),A	Save A at (HL) and increment HL
2A	LD HL,(word)	LD A,(HLI)	Load A from (HL) and increment HL
32	LD (word),A	LD (HLD),A	Save A at (HL) and decrement HL
3A	LD A,(word)	LD A,(HLD)	Load A from (HL) and decrement HL
D9	EXX	RETI	Enable interrupts and return
EO xx	RET PO	LD (byte),A	Save A at (FF00+byte)
E2	JP PO,word	LD (C),A	Save A at (FF00+C)
E8 xx	RET PE	ADD SP,offset	Add signed offset to SP
EA xx xx	JP PE,word	LD (word),A	Save A at given address
F0 xx	RET P	LD A, (byte)	Load A from (FF00+byte)
F8 xx	RET M	LDHL SP,offset	Load HL with SP + signed offset
FA xx xx	JP M,word	LD A, (word)	Load A from given address

Anzumerken ist, daß für die Soundgenerierung sowie für die Videodarstellung kein eigener OpCode implementiert wurde.

Die Erzeugung wird nur mit Hilfe der Spezialregister gesteuert. Diese Spezialregister werden in den folgenden Abschnitten näher betrachtet.

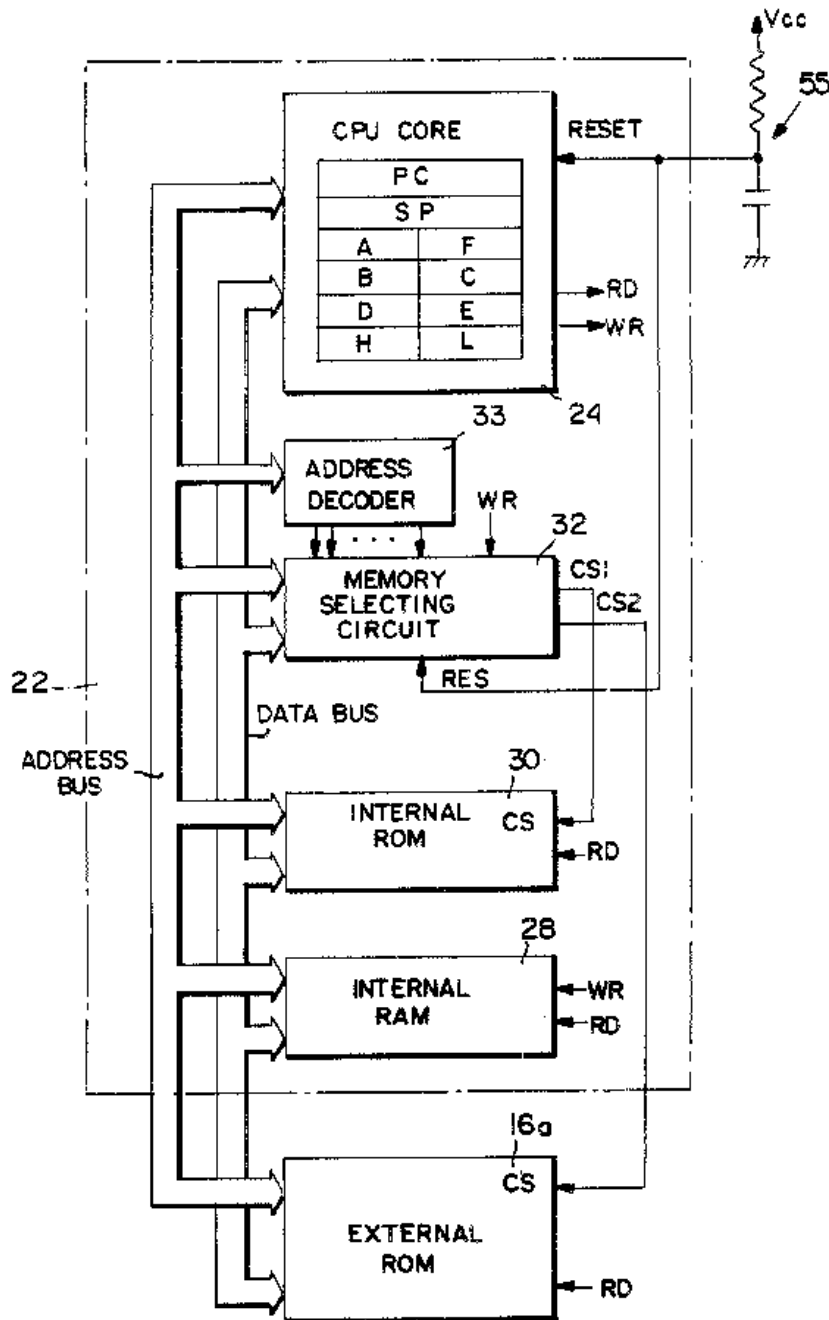
Jedes dieser Spezialregister verfügt über eine feste Adresse, über die dieses direkt angesprochen wird.

2.2 Die Speicherbelegung des GameBoy

In diesem Abschnitt gehen wir näher auf den Speicher, seine Aufteilung und auf die Speicherauswahllogik des GameBoy ein.

Im folgenden Bild ist der Speicheraufbau des GameBoy dargestellt. Wie man leicht erkennen kann, beinhaltet der Aufbau einen Adreßdeko­der, eine Speicherauswahllogik, ein internes RAM und ein internes sowie externes ROM. Diese sind über den Adreßbus und den Datenbus mit dem CPU-Kern verbunden.

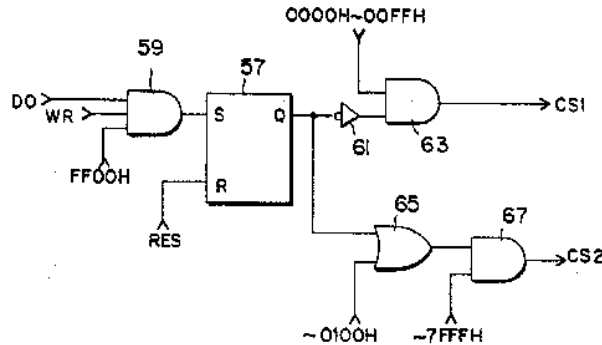
Das externe ROM befindet sich in der austauschbaren Programm-Cartridge. Auf diese Cartridge gehen wir später näher ein.



Quelle: U.S. Patent 5.095.798 Fig. 2A

2.2.1 Die Speicherauswahllogik des GameBoy

Das folgende Bild zeigt die Speicherauswahllogik, die für die Generierung der ChipSelect für das interne bzw. das externe ROM zuständig ist.

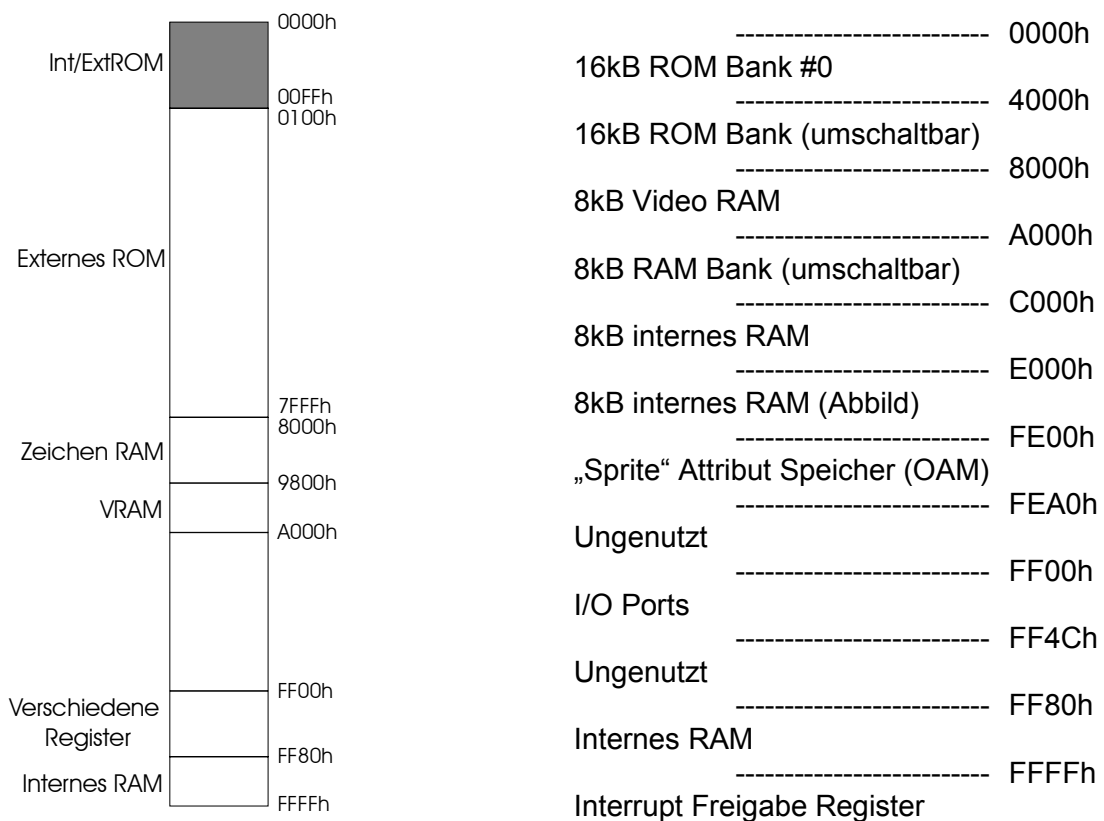


Quelle: U.S. Patent 5.184.830 Fig. 6

2.2.2 Die Speicheraufteilung des GameBoy

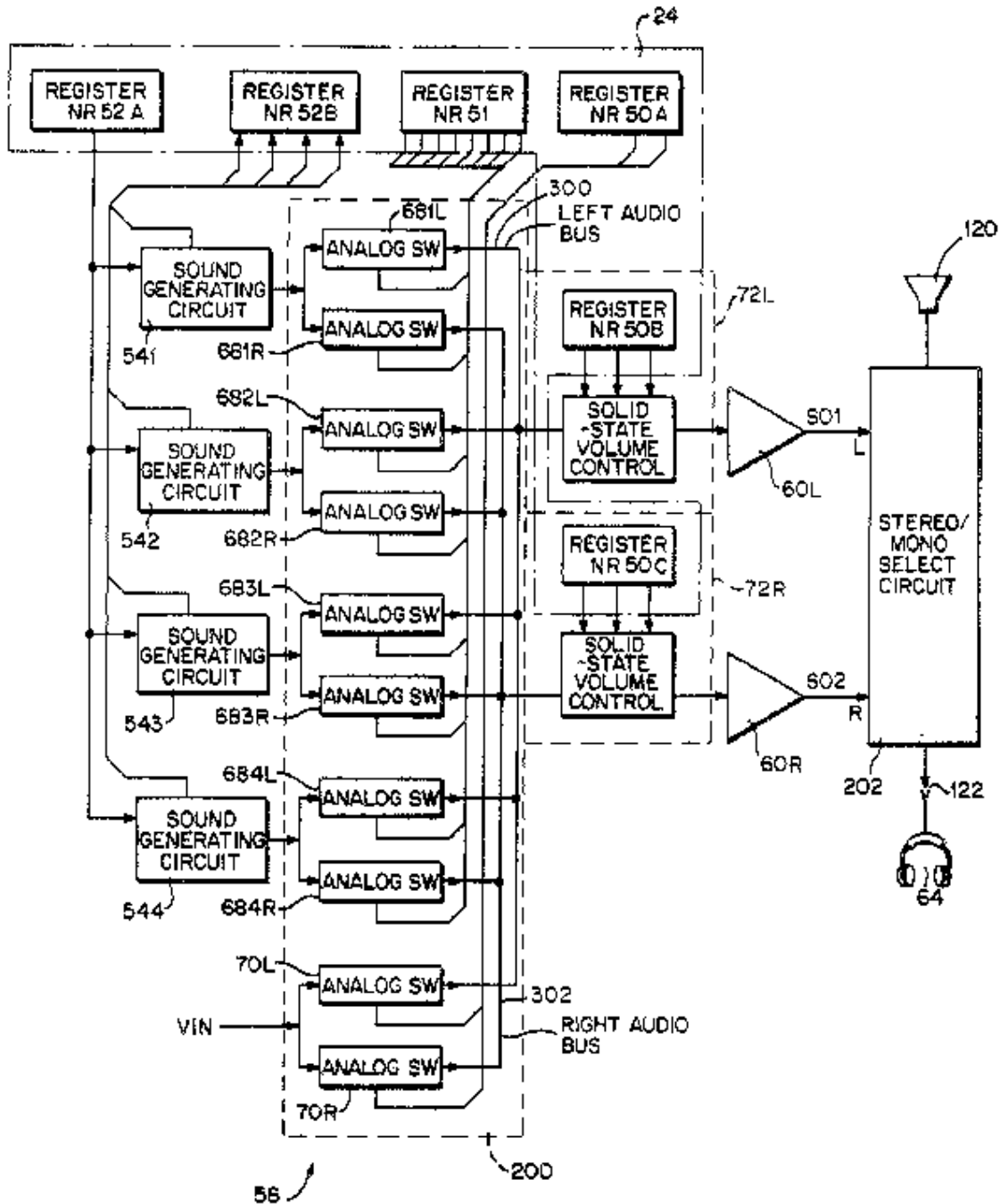
Im folgenden wird die Aufteilung des im GameBoy zur Verfügung stehenden Speichers näher beschrieben.

Die grau unterlegte Fläche zeigt den Speicherbereich, der vom internen sowie vom externen ROM gemeinsam (in Abhängigkeit der Speicherauswahllogik) genutzt wird. In der tabellarischen Auflistung sind die umschaltbaren Speicherbereiche zu erkennen sowie weitere Angaben zur Speicherverteilung.



2.3 Der Soundgenerator des GameBoy

Der GameBoy hat einen 4 Kanal Pseudo-Stereo-Soundgenerator, wie er in der folgenden Abbildung zu sehen ist. Pseudo bedeutet in diesem Fall, dass kein echter Stereo-Sound, sondern nur ein einfaches Mischsignal erzeugt wird.



Quelle: U.S. Patent 5.095.798 Fig. 4

In der Abbildung ist zu erkennen, daß der Soundgenerator vier Schaltkreise zur Sounderzeugung hat. Zusätzlich besitzt der Soundgenerator einen Multiplexer mit 10 *analogen* Schaltern (laut Patent) zum Weiterleiten der Soundsignale. Weiterhin sind die Lautstärkeregelungen der beiden Audiokanäle sowie die Mono/Stereo Auswahlschaltung zu erkennen.

Die vier Schaltkreise zur Soundgenerierung sind voneinander völlig unabhängig und werden durch die Register (NR10-NR44) gesteuert. Der Multiplexer wird durch die Register (NR50-NR52) gesteuert und dient dazu, die Ausgänge der einzelnen Schaltkreise entweder gar nicht, nur an einen oder an beide Lautstärkeregelungen (linker/rechter Kanal) weiterzuleiten.

Folgend sind die einzelnen Register NR50-NR52 explizit aufgezeigt

	REGISTER	ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
S • C O N T R O L	NR 50 S01/S02 LEVEL	FF 24	VIN INPUT 0: OFF 1: TO S02	S02 OUTPUT LEVEL CONTROL FROM 000:(MIN) TO 111:(MAX)			VIN INPUT 0: OFF 1: TO S01	S01 OUTPUT LEVEL CONTROL FROM 000:(MIN) TO 111:(MAX)		
	NR 51 S01/S02 ASSIGNMENT	FF 25	ASSIGN SOUND-4 TO S02	ASSIGN SOUND-3 TO S02	ASSIGN SOUND-2 TO S02	ASSIGN SOUND-1 TO S02	ASSIGN SOUND-4 TO S01	ASSIGN SOUND-3 TO S01	ASSIGN SOUND-2 TO S01	ASSIGN SOUND-1 TO S01
	NR 52 SOUND END FLAG	FF 26	ALL SOUND 0: STOP 1: START				SOUND-4 ON-FLAG	SOUND-3 ON-FLAG	SOUND-2 ON-FLAG	SOUND-1 ON-FLAG

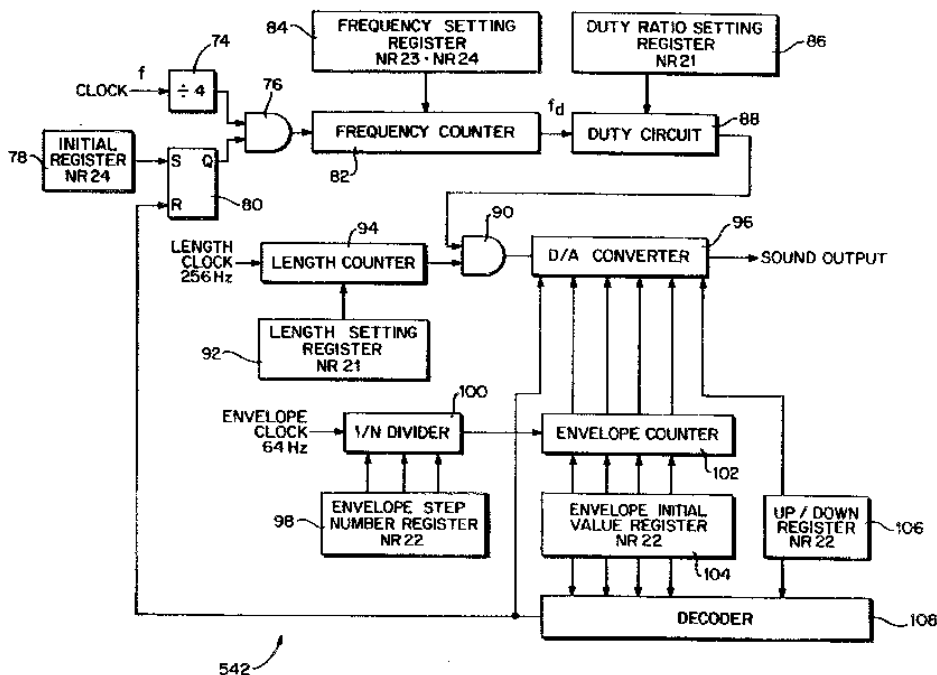
Quelle: U.S. Patent 5.095.798 Fig. 4A

Das Ausgangssignal der Lautstärkeregelung wird durch den Linken bzw. Rechten Audioverstärker verstärkt. Die Ausgangssignale der beiden Verstärker werden nun an die Mono/Stereo Auswahleinheit weitergeleitet, wo sie entweder an dem internen Lautsprecher (Mono) oder an dem externen Kopfhörer (Stereo) ausgegeben werden.

2.3.1 Die Soundgeneratorschaltkreise des GameBoy

Da die vier Schaltkreise zur Soundgenerierung in ihrem Aufbau und Funktionsweise fast identisch sind, betrachten wir folgend nur einen der vier Schaltkreise näher. Sie sind nicht vollständig identisch, da jeder der Schaltkreise für eine spezielle Soundart konzipiert wurde.

Betrachten wir hier nun den zweiten Soundgeneratorschaltkreis:



Quelle: U.S. Patent 5.095.798 Fig. 5

Wie im vorangegangenen Bild zu erkennen ist, beinhaltet dieser Schaltkreis verschiedene Zähler, Teiler sowie einige andere Komponenten, die durch die Register NR21-NR24 gesteuert werden.

In der folgenden Abbildung werden die Register NR21-NR24 weiter aufgeschlüsselt:

REGISTER	ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0	
NR 21 SOUND /SOUND L	FF 16	WAVE FORM DUTY			SOUND LENGTH					
		00: 12.5% 00: 50%			11: 0-63					
NR 22 ENVELOPE	FF 17	ENVELOPE INITIAL VALUE 0-FH			ENVELOPE	ENVELOPE STEP NUMBER N 0-7				
		0: NONE F: MAXIMUM			0: ATTENUAT 1: AMPLIFY	LENGTH OF 1-STEP = $N \cdot (1/64) \text{sec}$ N=0: STOP ENVELOPE OPERATION				
NR 23 FREQUENCY LOWER DATA	FF 18	104			106					
		FREQUENCY LOWER DATA 8-BIT								
NR 24 FREQUENCY UPPER...ETC	FF 19	INIT. FLAG	LENGTH				FREQUENCY UPPER DATA 3-BIT			
		1: RESTART	0: CONTINUE 1: NR 21				IF X IS 11-BIT AS FREQUENCY DATA $f = 4194304 / (4 \cdot 2^3 \cdot (2048 - X)) \text{Hz}$			

Quelle: U.S. Patent 5.095.798 Fig. 5B

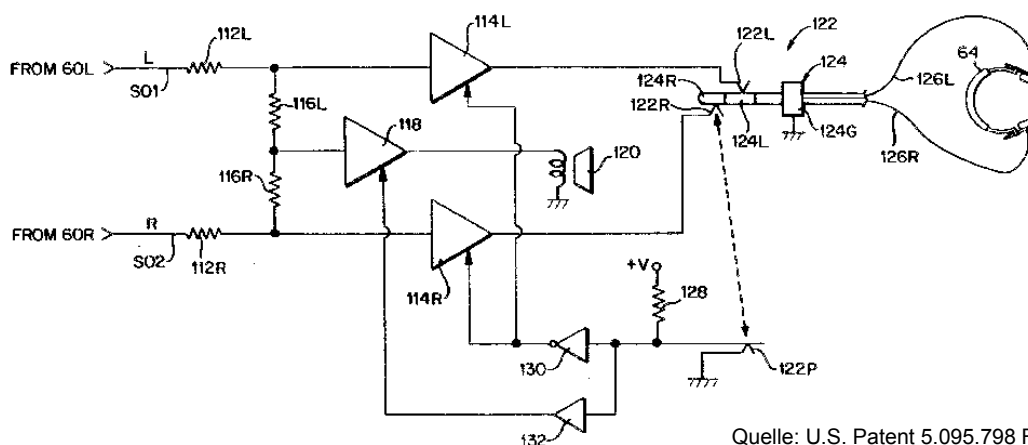
Zu den drei weiteren Soundgeneratorschaltkreisen gehören ebenfalls die entsprechenden Register. Diese sind für den Schaltkreis 1 die Register NR11-NR14, für den Schaltkreis 3 die Register NR31-NR34 und für den Schaltkreis 4 die Register NR41-NR44.

Wir wollen hier nicht im einzelnen auf diese Register eingehen, da sie sich nur in geringen Teilen unterscheiden. Die Register sind im Anhang A aufgeführt.

Die Komponenten mit den laufenden Nummern 74 bis 94 erzeugen das Taktsignal zur Steuerung der Umwandlung der Daten des Envelope Counters. Durch den D/A Wandler werden diese Daten in das analoge Soundsignal umgewandelt. Zusätzlich kontrollieren diese Komponenten die notwendige Zeitparametersteuerung für die Erzeugung des Soundsignals. Der Envelope Counter und die zugehörigen Komponenten mit den laufenden Nummern 98 bis 106 sind für die Regelung der Amplitude des Soundsignals zuständig. Der Dekoder dient zum Rücksetzen der gesamten Schaltung, nach Beendigung der Soundgenerierung.

2.3.2 Die Mono/Stereo Auswahleinheit des GameBoy

Die Mono/Stereo Auswahleinheit dient, wie bereits vorher erwähnt, zur Ansteuerung des internen Lautsprechers oder eines externen Kopfhörers.

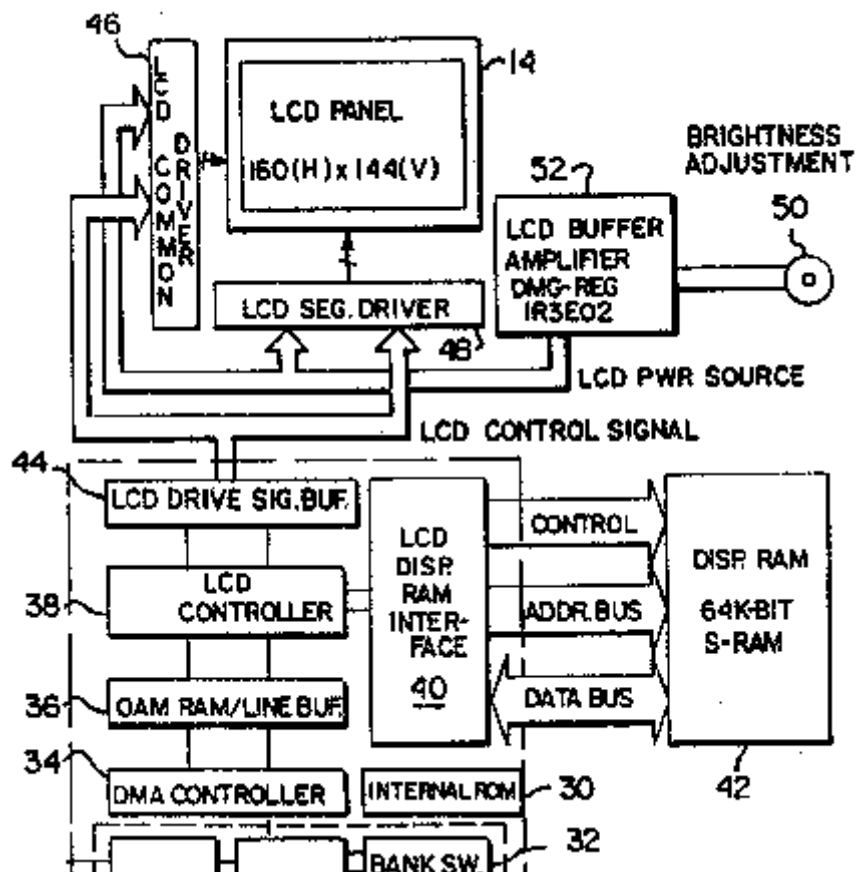


Quelle: U.S. Patent 5.095.798 Fig. 8

Registriert diese Auswahleinheit das Vorhandensein eines Kopfhörers, so leitet sie das Soundsignal, getrennt nach dem linken und rechten Audiokanal, an diesen Kopfhörer weiter. Sollte kein Kopfhörer angeschlossen sein, so wird das Soundsignal der beiden Audiokanäle gemischt, und als Monosignal an den internen Lautsprecher weitergeleitet, der dieses Signal ausgibt.

2.4 Das Videosystem des GameBoy

In diesem Abschnitt gehen wir auf das Videosystem des GameBoy ein. Betrachten wir das folgende Bild etwas näher.



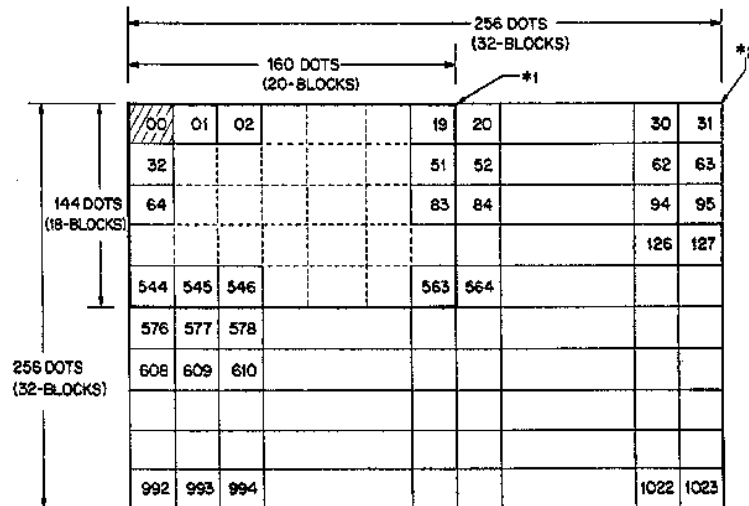
Quelle: U.S. Patent 5.184.830 Fig. 4

Wie in diesem Ausschnitt zu erkennen ist, werden die Daten für das Display, die von der CPU kommen, mittels des DMA-Controllers an den LCD-Controller weitergegeben. Dieser LCD-Controller ist über das LCD-RAM-Interface mit dem Display-RAM verbunden. Für diese Verbindungen zwischen den LCD-RAM-Interface und dem Display-RAM hat der GameBoy einen zweiten vollständigen Systembus, wie bereits am Anfang dieser Ausarbeitung erwähnt wurde.

2.4.1 Der Aufbau von Hintergrundbildschirm und Fenster des GameBoy

Der Bildschirmspeicher für den Hintergrund hat eine Größe von 256x256 Pixel oder 32x32 Blöcke, von denen jeder aus 8x8 Pixel besteht. Von dieses 256x256 Pixel können jedoch nur 160x144 Pixel gleichzeitig dargestellt werden.

In dem folgenden Bild der Hintergrundmatrix ist der sichtbare Bereich gestrichelt dargestellt.



Quelle: U.S. Patent 5.184.830 Fig. 14

Dieser sichtbare Bereich kann durch die Änderung der Scrollregister SCX und SCY verschoben werden, wobei in den Scrollregistern die Koordinaten der linken oberen Ecke dieses Bereichs eingetragen sind.

Der Bereich des VRAM, in dem die Hintergrundmatrix abgelegt ist, wird als Hintergrund-Block-Verzeichnis bezeichnet. Sie ist, wie bereits im oberen Bild dargestellt, in 32x32 Blöcken organisiert. In diesem Hintergrund-Block-Verzeichnis werden nur die Nummern der unterschiedlichen Blöcke abgelegt, die angezeigt werden sollen. Der Aufbau eines Blocks mit 8x8 Pixel wird einer Block-Daten-Tabelle entnommen. Diese Tabelle befindet sich an der Speicherstelle 8000h-8FFFh bzw. 8800h-97FFh im RAM. Welche dieser beiden Adreßbereiche benutzt werden soll, wird im LCDC-Register eingestellt.

Es gibt zwei verschiedene Hintergrund-Block-Verzeichnisse im VRAM. Diese Verzeichnisse liegen in den Adreßbereichen 9800h-9BFFh sowie 9C00h-9FFFh. Es ist nur möglich, eine der beiden Tabellen zum jeweiligen Zeitpunkt anzuzeigen. Welche der beiden Tabellen angezeigt werden soll, wird ebenfalls mit dem LCDC-Register eingestellt.

Zusätzlich zum Hintergrund des GameBoy kam man ein Fenster erzeugen, welches über dem Hintergrundbild liegt. Dieses Fenster kann allerdings nicht verschoben werden. Die Position dieses Fensters wird in den Registern WNDPOSX und WNDPOSX eingestellt. Die Blöcke, aus denen dieses Fenster zusammengesetzt wird, werden ebenfalls der Block-Daten-Tabelle entnommen.

Der Hintergrund sowie das Fenster können durch das LCDC-Register unabhängig voneinander ein- bzw. ausgeschaltet werden.

3. Die Cartridge des GameBoy

Für den GameBoy gibt es verschiedene Cartridges. Damit die CPU des GameBoy weiß, um welche Art Cartridge es sich handelt, ist im ROM der Cartridge an der Adresse 0147h ein Identifikationscode gespeichert.

Die folgende Tabelle zeigt, welche Cartridge-Typen der GameBoy unterscheidet.

Code	Beschreibung
00	ROM
01	ROM + MBC1
02	ROM + MBC1 + RAM
03	ROM + MBC1 + RAM + Batterie
05	ROM + MBC2
06	ROM + MBC2 + Batterie
08	ROM + RAM
09	ROM + RAM + Batterie
0B	ROM + MMM01
0C	ROM + MMM01 + SRAM
0D	ROM + MMM01 + SRAM + Batterie
0F	ROM + MBC3 + Timer + Batterie
10	ROM + MBC3 + Timer + RAM + Batterie
11	ROM + MBC3
12	ROM + MBC3 + RAM
13	ROM + MBC3 + RAM + Batterie
19	ROM + MBC5
1A	ROM + MBC5 + RAM
1B	ROM + MBC5 + RAM + Batterie
1C	ROM + MBC5 + RUMBLE
1D	ROM + MBC5 + RUMBLE + SRAM
1E	ROM + MBC5 + RUMBLE + SRAM + Batterie
1F	Pocket Kamera
FD	Bundai TAMA5
FE	Hudson HuC-3
FF	Hudson HuC-1

MBC	= Memory Bank Controller
MMM01	= nicht bekannt
RUMBLE	= Eine Cartridge mit Vibrationsfunktion (eingebauter Motor)
TAMA5	= nicht bekannt
HuC	= Memory Bank und Infrarot Controller

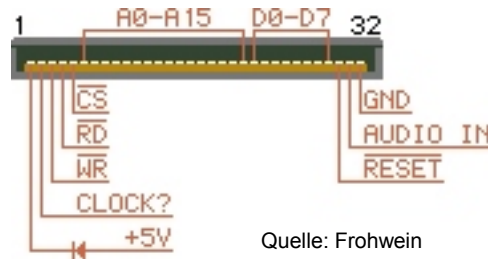
In den Cartridges ist meistens neben dem ROM auch ein Memory Bank Controller (MBC) vorhanden, wie aus der Tabelle zu entnehmen ist.

Der Memory Bank Controller dient zum Umschalten zwischen den einzelnen Speicherbänken.

Für den Fall, daß sich ein RAM auf der Cartridge befindet, wird auch dieses vom MBC gesteuert. Der RAM-Baustein kann an einer Batterie angeschlossen sein, um eine Sicherung der Daten auch über einen längeren Zeitraum zu gewährleisten. Die verschiedenen MBC unterscheiden sich hauptsächlich darin, daß sie verschiedene Größen von ROM und RAM steuern können.

Es gibt auch spezielle Cartridges, die z.B. mit einer Kamera, einer Infrarot Schnittstelle (zur Kommunikation) oder einem Motor (sog. Rumble Pack) ausgestattet sind.

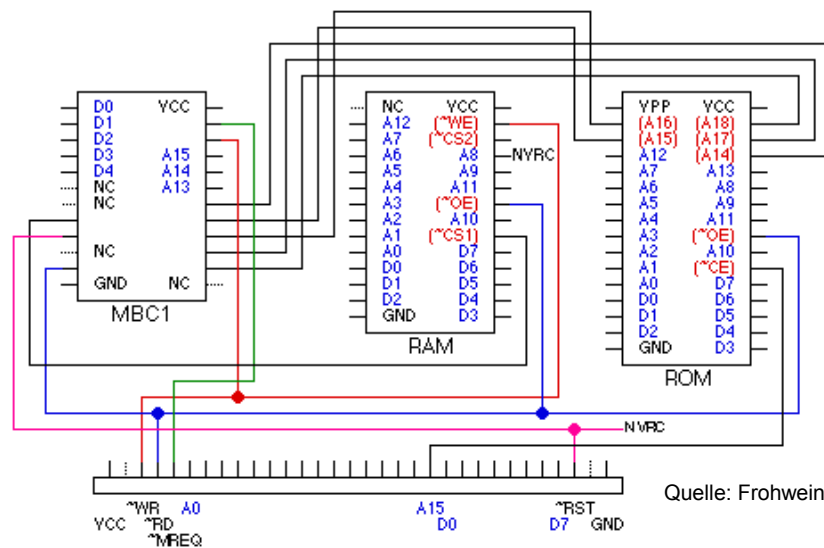
Die Cartridge ist über eine 32polige Steckverbindung (siehe Abbildung) mit dem GameBoy verbunden.



Quelle: Frohwein

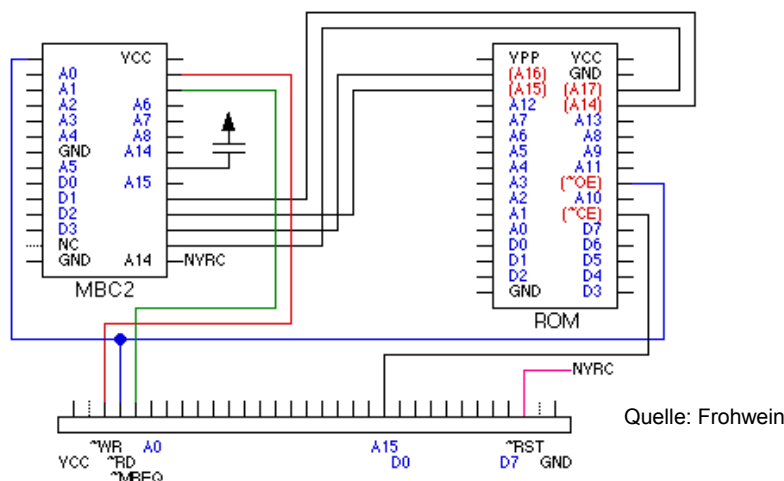
In den folgenden Abbildungen werden zwei Schaltpläne unterschiedlicher Cartridges dargestellt.

Der hier dargestellte Schaltplan einer Cartridge beinhaltet einen MBC1, der ein 4Mbit (512KB) ROM und ein 64Kbit (8KB) RAM steuert.



Quelle: Frohwein

Folgend ein Beispiel mit einem MBC2. Der MBC2 enthält ein internes RAM von 512x4 Bit im Controller, so daß kein RAM zusätzlich angeschlossen werden muß. Das externe ROM hat eine Größe von 2Mbit (265KB). An dem MBC2 ist hier noch an eine Batterie zur Sicherung der Daten über ein längeren Zeitraum angeschlossen.



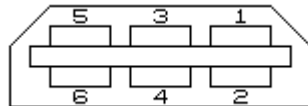
Quelle: Frohwein

4. Die serielle Schnittstelle des GameBoy

Die serielle Schnittstelle des GameBoy ist kompatibel mit der RS-232 (IBM PC) und der RS-485 (Mac). Sie ist über Register leicht einstellbar. Sie arbeitet aber ohne Start- und Stopbit, daher muß man sich bei Bedarf geeignete Übertragungsprotokolle einfallen lassen.

Die serielle Schnittstelle kann intern bzw. extern getaktet werden. Bei interner Taktung wird mit einer Geschwindigkeit von 8192 Hz übertragen.

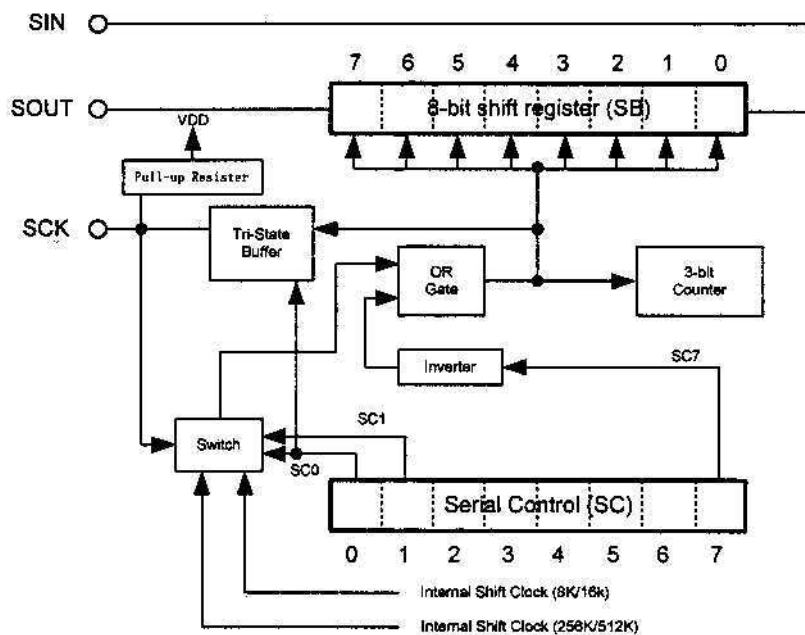
In diesem Bild ist der 6polige Anschluß mit der entsprechenden Pinbelegung dargestellt.



Quelle: Frohwein

Pin	Funktion
1	VCC
2	Seriell OUT
3	Seriell IN
4	Port 14 (CPU-Port)
5	CLK
6	GND

Folgend das Blockdiagramm der seriellen Schnittstelle:



Quelle: Color GameBoy User Guide Ver. 1-3

Wie man in diesem Bild erkennen kann, werden die Daten, die seriell übertragen werden sollen, in das Shift-Register geladen. Mittels des Serial-Control-Register kann die serielle Übertragung gestartet bzw. kontrolliert werden.

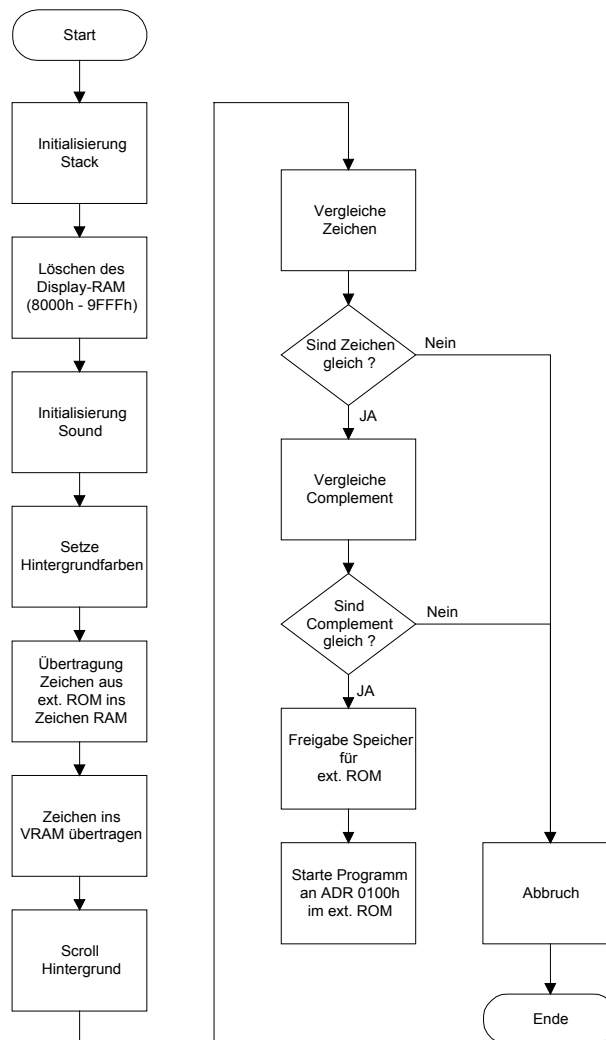
Die serielle Schnittstelle wurde von Nintendo im GameBoy eingebaut, um eine Kommunikation zwischen zwei GameBoy zu ermöglichen.

Die Grundidee war es dabei, mehrere GameBoy in einer Art Netzwerk zu verbinden und so mit mehreren Spielern gemeinsam ein Spiel spielen zu können. Die Voraussetzung dabei ist, daß jeder GameBoy mit der gleichen Spiel-Cartridge ausgestattet ist.

5. Die PowerUP-Sequenz des GameBoy

Beim Einschalten des GameBoy beginnt die CPU, eine Startsequenz zu durchlaufen. Diese Startsequenz dient der Prüfung, ob die eingesteckte Cartridge gültig ist.

Diese Startsequenz ist im folgenden Ablaufplan dargestellt.



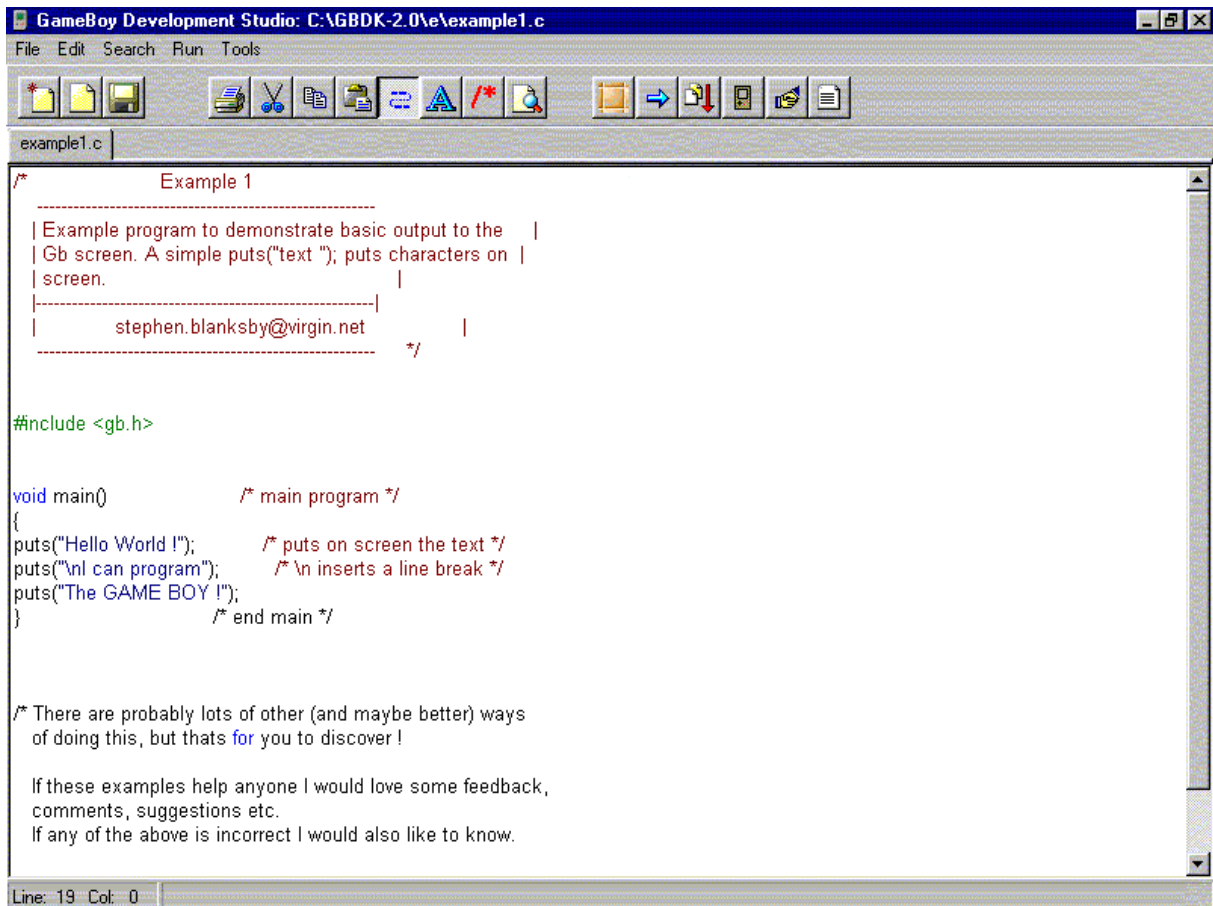
Nach Einschalten des GameBoy startet ein 256 Byte großes Programm an der Adresse 0000h des internen ROM. Dieses Programm initialisiert den StackPointer und löscht anschließend das Display-RAM. Im nächsten Schritt wird der Soundgenerator initialisiert und die Hintergrundfarben werden geladen. Nun liest das Programm die Speicherstellen 0104h bis 0133h aus dem externen ROM, und überträgt deren Inhalt in das Display-RAM. Diese Daten ergeben ein Nintendo-Logo, welches am oberen Rand des Display angezeigt wird. Dieses Logo wird nun in die Mitte des Display gescrollt. Wenn dieses erfolgt ist, werden zwei Musiknoten ausgegeben. Die Speicherstellen 0104h bis 0133h werden erneut aus dem externen ROM gelesen und mit einer Tabelle im internen ROM verglichen. Sollte nur ein Byte falsch sein, bricht der GameBoy die Startsequenz ab.

Ist der Vergleich erfolgreich gewesen, addiert das Programm die Werte der Speicherstellen 0134h bis 014Dh aufeinander. Zu diesem Wert wird dann der Wert 25 hinzuaddiert. Ergibt das niederwertigste Byte keine Null, wird wiederum die Startsequenz abgebrochen. Sind die oben genannten Vergleiche erfolgreich, wird das internen ROM abgeschaltet, und das Programm an der Speicherstelle 0100h in dem externen ROM wird gestartet.

6. Die Entwicklungsoberflächen für den GameBoy

Für den GameBoy gibt es eine Vielzahl von Entwicklungsoberflächen. Es fängt beim (einfachen) Assembler an und geht bis zum relativ komfortablen Entwicklungsstudio unter C.

Hier ein Beispiel für eine Entwicklungsoberfläche unter Windows:



The screenshot shows the GameBoy Development Studio interface. The title bar reads "GameBoy Development Studio: C:\GBDK-2.0\example1.c". The menu bar includes "File", "Edit", "Search", "Run", and "Tools". The toolbar contains various icons for file operations and development. The main text area displays the following C code:

```
/*      Example 1
-----
| Example program to demonstrate basic output to the  |
| Gb screen. A simple puts("text "); puts characters on |
| screen.                                           |
|-----|
|      stephen.blanksby@virgin.net      |
|-----*/

#include <gb.h>

void main()          /* main program */
{
puts("Hello World!");    /* puts on screen the text */
puts("\nI can program"); /* \n inserts a line break */
puts("The GAME BOY!");
}                      /* end main */

/* There are probably lots of other (and maybe better) ways
of doing this, but thats for you to discover !

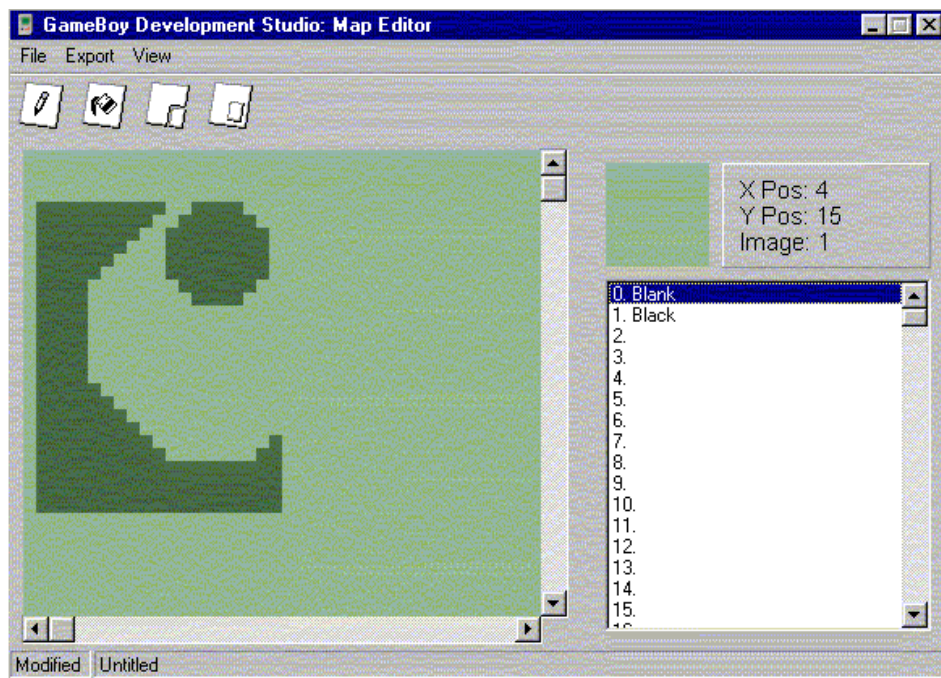
If these examples help anyone I would love some feedback,
comments, suggestions etc.
If any of the above is incorrect I would also like to know.*/
```

The status bar at the bottom left shows "Line: 19 Col: 0".

Dieses GameBoy Development Studio ist eine reine grafische Entwicklungsoberfläche, in die noch ein geeigneter Compiler, z.B. der des GameBoy Developer's Kit, eingebunden werden muß. Nach dem Einbinden eines Compiler hat man nun die Möglichkeit, in ANSI C seine GameBoy-Programme zu entwickeln. Genauso könnte man unter dieser Oberfläche einen Assembler Compiler einbinden, und in Assembler Programme entwickeln.

Bei dem GameBoy Developer's Kit sind auch einige Header-Dateien, die speziell für den GameBoy entwickelt wurden, dabei. Im Internet gibt es noch ein großes Angebot an Header-Dateien, welche die Programmentwicklung vereinfachen.

Ein nützliches Tool ist der MapEditor, der beim Development Studio dabei ist.



Mit diesem MapEditor kann man schnell ein Hintergrundbild bzw. Grafikblöcke erstellen. So bleibt dem Entwickler viel Handarbeit beim Erstellen dieser sogenannten Map's erspart.

Weitere nützliche Tools sind die GameBoy Emulatoren, die als Freeware, in einigen Fällen allerdings auch als Shareware, im Internet erhältlich sind. Mit diesen Emulatoren kann man seine erstellten Programme schnell testen, ohne den Umweg über den GameBoy gehen zu müssen.

Hier z.B. ein ScreenShot des Emulators GB97:



7. Quellen

United States Patent

- Patent Nummer 5.095.798
(Electronic gaming device with pseudo-stereophonic sound generating capabilities
http://patent.womplex.ibm.com/details?patent_number=5095798)
- Patent Nummer 5.184.830
(Compact hand-held video game system)
http://patent.womplex.ibm.com/details?patent_number=5184830)

Z80 CPU architecture

<http://www.geocities.com/SiliconValley/Peaks/3938/Z80arki.htm>
Allgemeine Infos zur Z80 Architektur

Nintendo frequently asked questions

<http://www.cs.umd.edu/users/fms/>
Antworten auf die meisten Fragen in Sachen GameBoy

Jeff Frohwein

<http://home.hiwaay.net/~jfrowei/gameboy/home.html>
GameBoy Seite des „GameBoy Guru“ im Internet.
Schaltpläne, technische Daten, Dokumente, viele Links

Black Thunder

<http://www.blackthunder.demon.nl/>
Dokumente & Beispiele zur Programmierung

GameBoy Emulation Heaven

<http://qbemuheaven.cjb.net/>
Emulatoren für fast jedes Betriebssystem

GameBoy Development Studio

<http://www.geocities.com/eureka/9827/>
Win95 Interface zur Entwicklung von Programmen in C für den GameBoy

Andrew March's GB Page

<http://www.icenet.com.au/~amarch/>
ROM-Emulator für den GameBoy

Jason's GBDK Page

<http://www.gbdev.org/news/>
Dokumente & Beispiele zum GBDK

Kashi's GameBoy Page

<http://www.keisei.tsukuba.ac.jp/~kashima/games/gbread2e.html>
Informationen über verschiedene Cartridges

CGB – Color Game Boy - User's Guide Vers. 1-3

Handbuch zum GameBoy Color

Azrail's GameBoy Section

Allgemeine Infos zum GameBoy (leider ist uns die URL abhanden gekommen)

8. Anhang

In den folgenden Anhängen werden nur die Register des GameBoy näher aufgeführt, die auch in dieser Ausarbeitung angesprochen worden sind. Zusätzlich haben wir in Anhang D noch die Schaltpläne für den GameBoy und der Cartridge beigefügt.

Anhang A

In diesem Anhang sind die Soundregister aus dem U.S. Patent 5.095.798 *-Electronic gaming device with pseudo-stereophonic sound generating capabilities-* der Vollständigkeit halber aufgeführt.

Soundgeneratorschaltkreis 1:

REGISTER	ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0	
S O U N D 1	NR 10 SWEEP	FF 10	SWEEP TIME 00: OFF 01: 23.4ms 10: 39.1ms 11: 46.9ms 00: 7.8ms 10: 31.3ms 11: 54.7ms	SWEEP			NUMBER OF SWEEP SHIFT: 0-7 0: + f hi 1: - f low			
	NR 11 DUTY/ SOUND LENGTH	FF 11		WAVE FORM DUTY 00: 12.5% 01: 25% 00: 50% 11: 75%		SOUND LENGTH DATA 11: 0-63 SOUND LENGTH = (64 - 11) * (1/256)sec				
	NR 12 ENVELOPE	FF 12	ENVELOPE INITIAL VALUE 0-FH 0: NO SOUND F: MAXIMUM			ENVELOPE 0: ATTENUAT 1: AMPLIFY	NUMBER OF ENVELOPE STEP (N): 0-7 LENGTH OF 1-STEP = N * (1/64)sec N=0: STOP ENVELOPE OPERATION			
	NR 13 FREQUENCY LOWER DATA	FF 13	FREQUENCY LOWER DATA 8-BIT							
	NR 14 FREQUENCY UPPER/.ETC	FF 14	INITIAL FLAG 1: RESTART	LENGTH 0: CONTINUE 1: NR 11				FREQUENCY UPPER DATA 3-BIT IF X IS 11-BIT AS FREQUENCY DATA $f = 4194304 / (4 * 2^3 * (2048 - X))$ Hz		

FIG. 5A

Soundgeneratorschaltkreis 2:

REGISTER	ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0	
S O U N D 2	NR 21 SOUND /SOUND L	FF 16	WAVE FORM DUTY 00: 12.5% 01: 25% 00: 50% 11: 75%		SOUND LENGTH 11: 0-63 SOUND LENGTH = (64 - 11) * (1/256)sec					
	NR 22 ENVELOPE	FF 17	ENVELOPE INITIAL VALUE 0-FH 0: NONE F: MAXIMUM			ENVELOPE 0: ATTENUAT 1: AMPLIFY	ENVELOPE STEP NUMBER N 0-7 LENGTH OF 1-STEP = N * (1/64)sec N=0: STOP ENVELOPE OPERATION			
	NR 23 FREQUENCY LOWER DATA	FF 18	FREQUENCY LOWER DATA 8-BIT							
	NR 24 FREQUENCY UPPER/.ETC	FF 19	INIT. FLAG 1: RESTART	LENGTH 0: CONTINUE 1: NR 21				FREQUENCY UPPER DATA 3-BIT IF X IS 11-BIT AS FREQUENCY DATA $f = 4194304 / (4 * 2^3 * (2048 - X))$ Hz		

FIG. 5B

Soundgeneratorschaltkreis 3:

REGISTER	ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
SOUND 3	NR 30 SOUND OFF	FF 1A	SOUND OFF 0: OFF 1: ON						
	NR 31 SOUND LENGTH	FF 1B	SOUND LENGTH DATA 11: 0-255 SOUND LENGTH = (256-t1)*(1/256)sec						
	NR 32 OUTPUT LEVEL	FF 1C		OUTPUT LEVEL 00: NONE 10: 1/2 01: MAX 11: 1/4					
	NR 33 FREQUENCY LOWER DATA	FF 1D	FREQUENCY LOWER DATA-BIT						
	NR 34 FREQUENCY UPPER/.ETC	FF 1E	INITIAL FLAG 1:RESTART	LENGTH 0:CONTINUE 1:NR 31				FREQUENCY UPPER DATA 3-BIT IF X IS 11-BIT AS FREQUENCY DATA $f = 4194304 / (4 * 2^3 (2048 - X))$ Hz	

FIG. 5C

Soundgeneratorschaltkreis 4:

REGISTER	ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0	
SOUND 4	NR 41 SOUND LENGTH	FF 20			SOUND LENGTH DATA 11:0-63 SOUND LENGTH = (64-t1)*(1/256)sec					
	NR 42 ENVELOPE	FF 21	ENVELOPE INITIAL VALUE 0 - FH 0 : NONE F : MAX			ENVELOPE 0:ATTENUAT 1:AMPLIFIC	ENVELOPE STEP NUMBER N 0-7 LENGTH OF 1-STEP = N*(1/64)sec N=0: STOP ENVELOPE OPERATION			
	NR 43 POLYNOMIAL COUNTER	FF 22	POLYNOMIAL COUNTER CLOCK FREQUENCY SELECTION 0000: $f_b * 1/2$ 1100: $f_b * 1/2^{13}$ 0001: $f_b * 1/2^2$ FROM 1101: $f_b * 1/2^{14}$		PROHIBITED CODE 1110 1111	# OF STEPS 0: 15 1: 7	DIVIDING RATIO FREQUENCY f_b SELECTION 000: $f * 1/2^3 * 1/2$ 110: $f * 1/2^3 * 1/5$ 001: $f * 1/2^3 * 1/1$ FROM 111: $f * 1/2^3 * 1/7$			
	NR 44 INITIAL LENGTH	FF 23	INITIAL FLAG 1:RESTART	LENGTH 0:CONTINUE 1:NR 11						

FIG. 5D

Soundkontroll-Register:

	REGISTER	ADDRESS	D7	D6	D5	D4	D3	D2	D1	D0
S • C O N T R O L	NR 50 S01/S02 LEVEL	FF 24	VIN INPUT 0: OFF 1: TO S02	S02 OUTPUT LEVEL CONTROL FROM 000:(MIN) TO 111:(MAX)			VIN INPUT 0: OFF 1: TO S01	S01 OUTPUT LEVEL CONTROL FROM 000:(MIN) TO 111:(MAX)		
	NR 51 S01/S02 ASSIGNMENT	FF 25	ASSIGN SOUND-4 TO S02	ASSIGN SOUND-3 TO S02	ASSIGN SOUND-2 TO S02	ASSIGN SOUND-1 TO S02	ASSIGN SOUND-4 TO S01	ASSIGN SOUND-3 TO S01	ASSIGN SOUND-2 TO S01	ASSIGN SOUND-1 TO S01
	NR 52 SOUND END FLAG	FF 26	ALL SOUND 0: STOP 1: START				SOUND-4 ON-FLAG	SOUND-3 ON-FLAG	SOUND-2 ON-FLAG	SOUND-1 ON-FLAG

FIG. 4A

Anhang B

In diesem Anhang sind die LCD-Display-Register aus dem U.S. Patent 5.184.830 - *Compact hand-held video game system* - der Vollständigkeit halber aufgeführt.

FIG. 15A LCD DISPLAY REGISTER

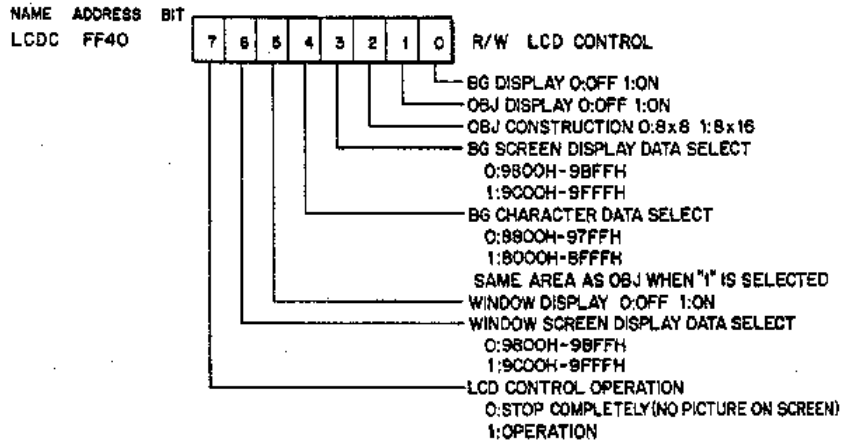


FIG. 15B

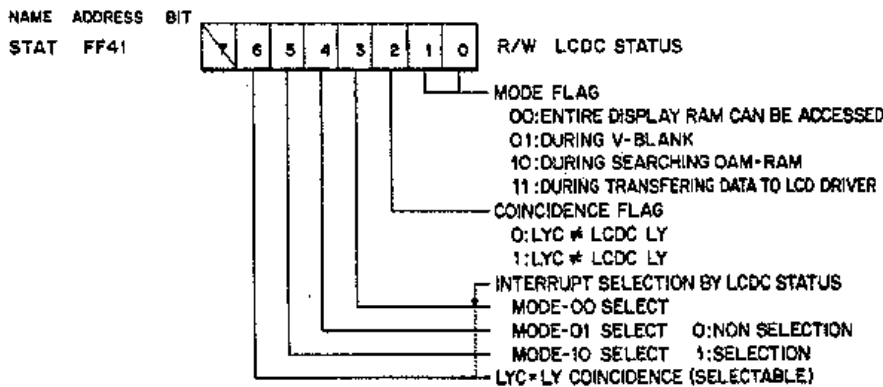


FIG. 15C

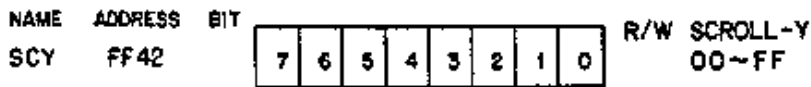


FIG. 15D

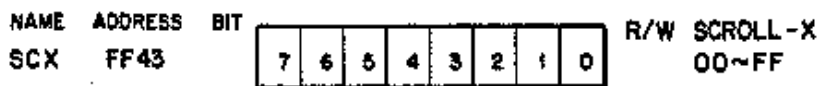


FIG. 15E

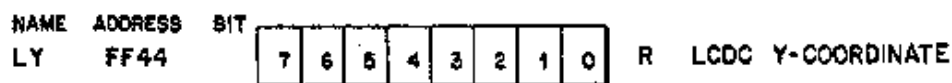


FIG. 15F

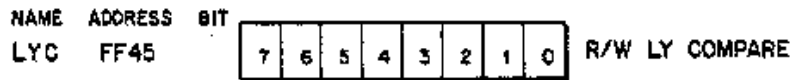


FIG. 15G

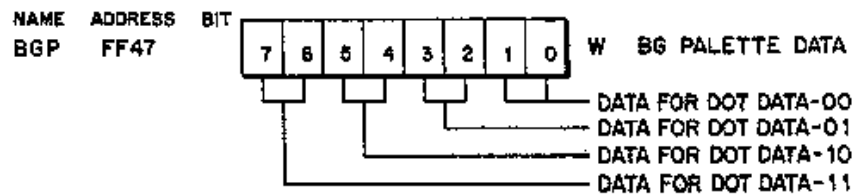


FIG. 15H

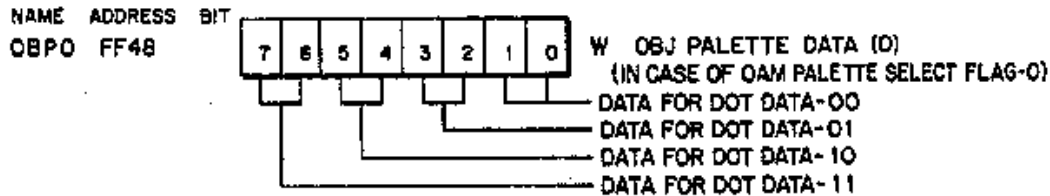


FIG. 15I

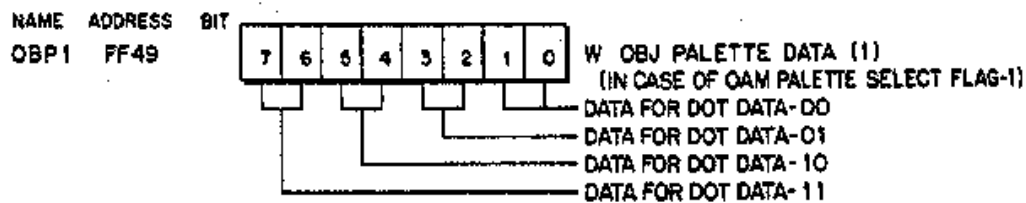


FIG. 15J

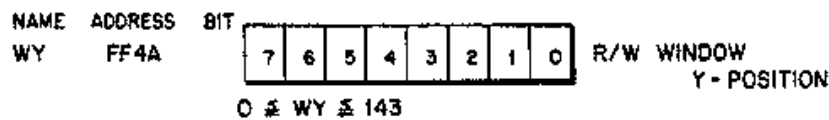
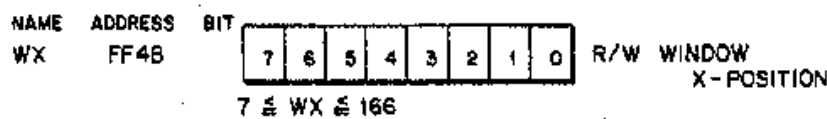


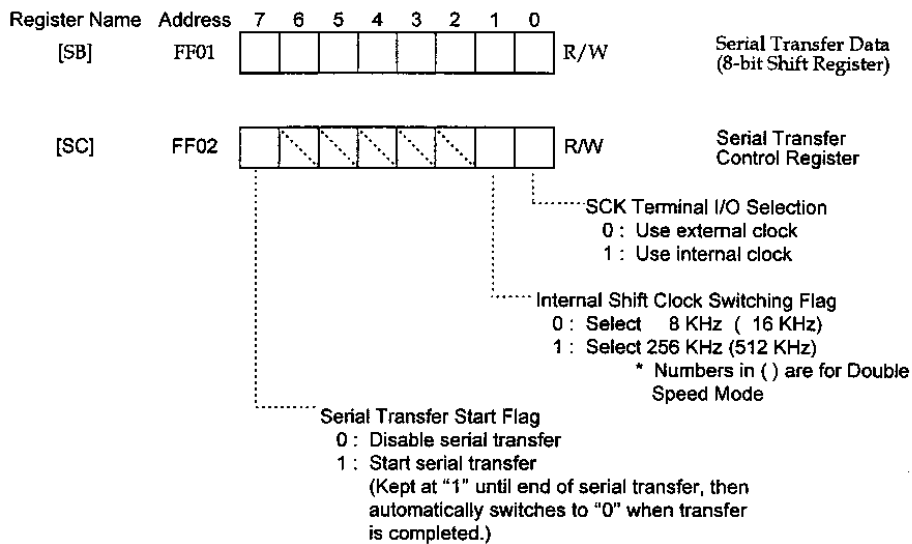
FIG. 15K



Anhang C

In diesem Anhang sind die Serial-Input/Output-Register (SIO) aus dem - *Color Game Boy User's Guide ver.1-3* - der Vollständigkeit halber aufgeführt.

Serial Input/Output (SIO)



Anhang D

In diesem Anhang sind die detaillierten Schaltpläne des GameBoy aufgeführt.

