

Labor für Rechnerstrukturen



Schriftliche Ausarbeitung des Vortrags

Wie 64kByte die Welt eroberten

Dozent:

Prof. Risse

Gruppe:

Manfred Benten

Matr.-Nr.: 499200434

Jens Willkowsky

Matr.-Nr.: 499218522

Inhalt

1.	EINLEITUNG	1
	1.1 DER C64 HEUTE	1
	1.2 DIE FIRMA COMMODORE	1
	1.3 ENTWICKLUNGSGESCHICHTE	2
2.	DAS SYSTEM	3
	2.1 SPEICHER	4
	2.1.1 ROM-Speicher.....	4
	2.1.2 RAM-Speicher.....	5
	2.2 BUS	5
	2.3 SID (SOUND INTERFACE DEVICE)	6
	2.4 VIC (VIDEO INTERFACE CONTROL / MOS 6569)	6
	2.5 SCHNITTSTELLEN, SPEICHERGERÄTE UND PERIPHERIE	7
3.	CPU MOS6510	9
	3.1 ALLGEMEINES	10
	3.2 REGISTER	10
	3.3 ALU	10
	3.4 BEFEHLSSATZ	10
4.	FAZIT	13
5.	ANHANG	14
	5.1 ADRESSIERUNGSARTEN	14
	5.2 STATUSREGISTER	14
	5.3 BEFEHLSÜBERSICHT NACH KATEGORIEN	15
	5.3.1 Vergleichsbefehle	15
	5.3.2 Sprungbefehle	15
	5.3.3 Unterbrechungsbefehle	15
	5.3.4 In- / Dekrementierbefehle	15
	5.3.5 Transferbefehle	16
	5.3.6 Arithmetische Befehle	16
	5.3.7 Logische Befehle	16
	5.3.8 Schiebe- und Rotierbefehle	16
	5.4 QUELLENANGABEN	17
	5.4.1 Literatur	17
	5.4.2 Internet.....	17

1. Einleitung

Als einer der erfolgreichsten Computer überhaupt, haben wir als Teenager den c64 kennen gelernt. Es war für uns der erste Kontakt mit dem Medium Computer.

Er war in der Lage eine Faszination auszustrahlen, die uns dazu bewegte sich näher mit diesem Bereich zu befassen und uns vielleicht dazu veranlasste heute derartiges zu studieren.

Bemerkenswert war damals (1982 Markteinführung), dass man schlicht davon ausging, das dieser Rechner das leistungsstärkste war, was es auf dem Sektor gab.

Aus einem Buch namens ‚Super Grafik, das Buch zum Commodore 64‘:

*„...hochauflösende Graphik mit 320*200 Punkten, Multicolorgraphik (160*200), 16 Farben.....So oder ähnlich wird – wohl auch zu Recht – unser guter alter c64 angepriesen. Kaum jemand, der sich auch nur ein wenig mit Computern auskennt, wird an solchen Anzeigen vorbeischaun, denn Ihr c64 kann wirklich viel.“*

Solche vollmündigen Versprechen wollen wir nicht einfach hinnehmen und deshalb untersuchen wir was technisch dahinter steckt. Vorweg gibt es noch eine Betrachtung seiner aktuellen Nutzung und Szene, sowie eine kurze Biographie der Firma Commodore und des Familienstammbaumes des c64.

1.1 Der C64 heute

Gibt man einmal den Begriff „c64“ in eine Suchmaschine ein, so bekommt man diverse Links die auf Emulatoren und damit auf Spiele verweisen. Die Spiele waren es, die diesen Rechner populär machten. Es gibt eine Vielzahl von Emulatoren die nicht nur zu spielerischen Zwecken angewendet werden.

Zum anderen gibt es den legendären Soundchip SID (Sound Interface Device). Seine Anhänger treffen sich heute noch, und zwar zuletzt in England im Februar diesen Jahres. Die Komponisten der Spiele und Demo Klassiker werden verehrt, gewürdigt.

Ansonsten gibt es viele, viele 64'er, die auf irgendwelchen Dachböden irgendwelcher Eltern, Omas oder Opas, einst vom Taschengeld abgespart, ihr tristes Dasein fristen.

Man findet im Netz etliches an Dokumentation zum Rechner selbst. Darüber hinaus gibt es allerhand ehrgeizige Projekte, wie etwa die Entwicklung eines C-Kompilers oder Versuche einen http-Browser dafür zu entwickeln. Vorwiegend sind es wohl Liebhaber“ dieses Rechners, die sich heute noch mit ihm auseinandersetzen.

1.2 Die Firma Commodore

Jack Tramiel gründete Mitte der fünfziger Jahre seine Firma, die Büromöbel und Schreibmaschinen produzieren sollte. Seiner ersten Schreibmaschine gab er den Namen Commodore nachdem er 1958 in Berlin war und ein Auto mit dem Schriftzug Commodore entdeckte. Seine eigentlichen Wunschnamen waren leider schon vergeben und somit hieß auch die neue Firma Commodore, die eigentlich erst 1962 rechtmäßig gegründet wird und Commodore Business Machines Ltd. heißt.

Tramiel lernt Anfang der 60er Irving Gould, einen tüchtigen Geschäftsmann, kennen, den er in die Firma Commodore einbringt. Gould rettet Commodore mehrmals vor der Pleite. Durch das viele Geld, das Gould mitbringt, hat er in der Firma das Sagen und Ende 1983 wird er den eigentlichen Gründer von Commodore aus der Firma eckeln! Er bringt die Philosophie ein: "Wer einen Commodore haben will, kauft sich einen! Somit brauchen wir keine Werbung!". Auch ist Gould der Ansicht: ein Produkt das sich gut verkaufe, brauche noch nicht ersetzt zu

werden. Doch gerade in der Computerbranche ist die Weiterentwicklung das A und O zum Überleben. Diese konsequente Einhaltung dieser Einstellung wird Commodore jedoch Anfang 1994 das Genick brechen.

Nachdem Commodore seine Schreibmaschinen gut verkaufen kann, möchte Tramiel auch in den Tischrechner-Markt einsteigen. Die ersten Rechenmaschinen erscheinen Ende der 60er Jahre. Anfang der siebziger entschließt sich Commodore in den Taschenrechnermarkt einzusteigen. Commodore verkauft diese Taschenrechner recht erfolgreich.

Als der Markt für Taschenrechner jedoch erhebliche Einbüssungen für Commodore bekommt, entwickeln Tramiels Experten unter Führung von Chuck Peddle den PET 2001-8, den ersten Homecomputer!

Er wird intern 1976 vorgestellt und erscheint 1977 recht ausgereift auf dem Markt als Komplettsystem in einem Gehäuse bestehend aus Monitor, Datensette und Tastatur. Inzwischen wurde die Firma MOS-Technology gekauft, nun war man nicht mehr von Texas Instruments abhängig.

Der PET Computer erfährt in den nächsten Jahren umfangreiche Verbesserungen (aus dem PET wird aus rechtlichen Gründen der CBM) und wird schnell zum Renner unter den Computerspezialisten. Werden der PET und die CBM-Serie anfangs in Amerika entwickelt, trennt Tramiel Commodore in 2 Bereiche auf: Amerika entwickelt ab 1980 die Homecomputer wie den VIC=20 und den Amiga 500, und Commodore Deutschland in Braunschweig entwickelt die Bürocomputer wie später den Commodore PC-10 oder den Amiga 2000. Bis Ende der 80er Jahre wird diese Linie beibehalten.

In Deutschland wird ab 1981 unter dem Namen VC-20 (VC= Volks Computer) der erste richtig erschwingliche Homecomputer angeboten und trotz des schon 1982 vorgestellten C64 kann er sich bis 1985 auf dem Markt behaupten.

Mit dem Commodore C64 veröffentlicht Commodore den erfolgreichsten Homecomputer der Welt. Er wird 1982 eingeführt und hält sich bis 1993. Mit über 22 Millionen (!) Exemplaren ist dieser kleine 8Bit-Rechner eine Sensation.

Am 29. April 1994 gibt Commodore USA offiziell bekannt, dass sie nicht mehr in der Lage seien, die Geschäfte weiterzuführen!

1.3 Entwicklungsgeschichte

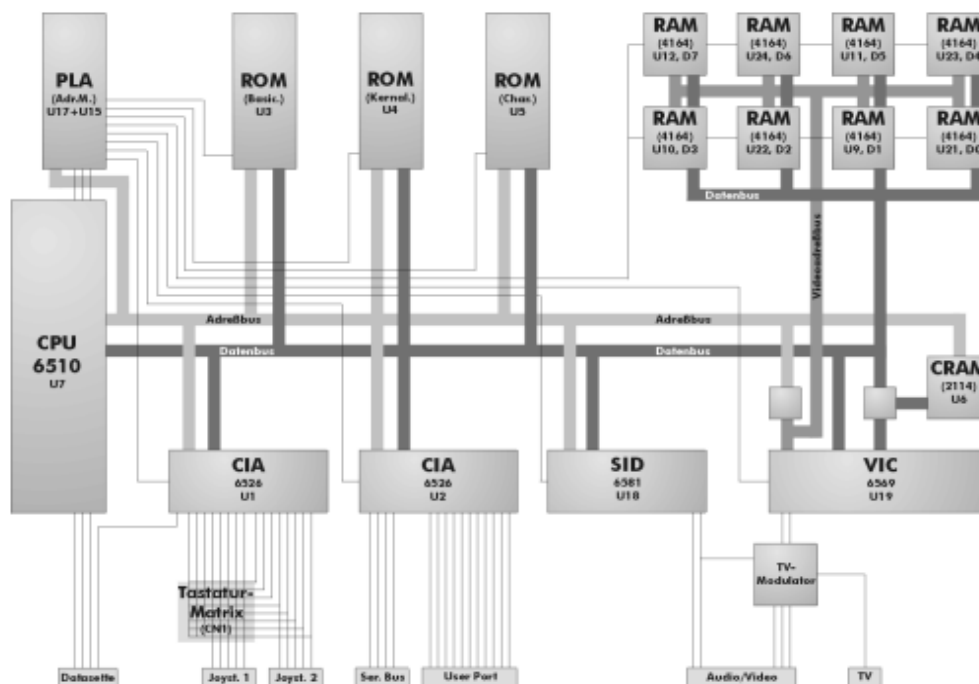
Der erste Rechner, den Commodore entwarf und auch baute war die alles in einem Maschine PET (Personal Electronic Transactor). Dies war 1977, fünf Jahre bevor der C64 auf den Markt kam. Der PET war bei den Kunden ziemlich beliebt und konkurrierte mit dem damals aktuellen Apple II, beide waren mit der 6502 MOS-CPU ausgestattet. Er war vor allem für den privaten Gebrauch gedacht. Er wurde auch in kleineren Unternehmen eingesetzt. Der PET war der erste Computer mit einer Tastatur, Kassettendeck und Bildschirm in einem Gehäuse. Später baute Commodore auch diese Art von Rechnern für den Bürogebrauch, wobei es eine Namensänderung von PET zu CBM (Commodore Büro Maschinen) gab. Dann baute Commodore ihren ersten Heimcomputer VC/VIC 20. Den kleinen Bruder des C64s, der sich Anfang der 80er Jahre großer Beliebtheit erfreute. Als der C64 erschien wurde der VC 20 vom Markt genommen, denn der C64 hatte weitaus mehr RAM, bessere Grafik und Soundmöglichkeiten. Für den kleinen Geldbeutel baute Commodore zwei low-cost Geräte, den C16 und 116 als Ersatz zum recht günstigen VC20.

2. Das System

Als typischer Vertreter des Homecomputer-Bereiches, kam der C64 in seiner kompakten Bauform daher. Nun fragt man sich vielleicht, was unter dem Bereich Homecomputer zu verstehen ist. Hier siedelten sich vor allem die Rechner kompakter Bauweise an, zu relativ erschwinglichen Preisen, die für jedermann zu erwerben waren. Sie ließen sich aufgrund der Bauweise nur sehr bedingt aufrüsten. Es war auch die Zeit der elektronischen Bastler und Entdecker. Es wurden aus einem 7pol. Dinstecker und einem Standard Taster (1,15DM) wunderbare RESET Taster hergestellt, die sich auf der Rückseite des c64 in dem Floppy Port einsetzen ließen. Andere hatten einen solchen einfach in das Gehäuse direkt neben dem Commodore Logo Eingebohrt. Es gab aber auch komplexere, durchaus Sinnvolle Basteleien, wie etwa die Betriebssystemerweiterung Speeddos. Es handelt sich hier um eine Erweiterung des Basicinterpreters der ein beschleunigten Zugriff auf das Floppy Speichermedium ermöglichte.

Diese Basteleien konnte man sich nur unter der Hand über Freundes Freunde besorgen. Wahre Halbgötter besaßen einen entsprechenden Eprom-Brenner und hatten alle wichtigen IC's auf Textool Sockel gesetzt, um diese komfortabler austauschen zu können. Es gibt viele andere derartige Beispiele doch zurück zum Thema.

Zum Lieferumfang des C64 Systems gehörte der eigentliche Rechner (die sogenannte Butterbrotkiste) mit dem Anschlusskabel für den Fernseher und der Spannungsversorgung. Man konnte dann weitere Zusatzgeräte kaufen, wie z.B. die allseits bekannte Datasette als einfachstes Datenspeicherungsmedium. Später wurde diese dann durch die Floppy abgelöst. Soweit zu den Äußerlichkeiten, nun wollen wir uns mit den Innereien und dem Internen Aufbau dieses Gerätes beschäftigen.



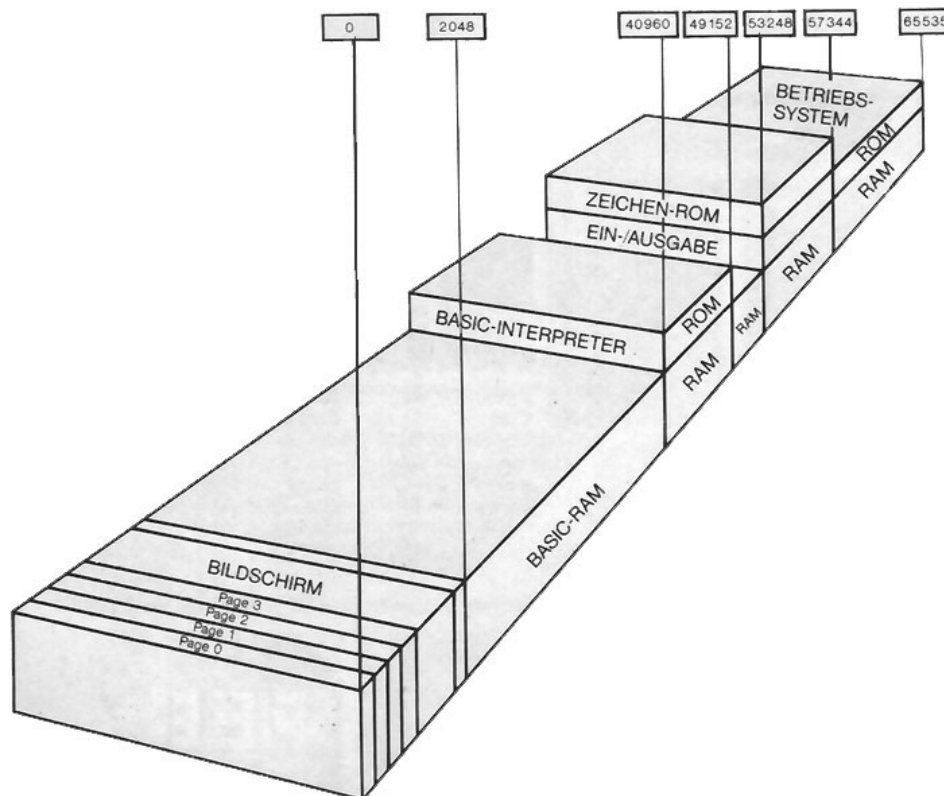
Dies ist der Schematische Aufbau. Die grundlegenden Bauteile sind in dieser Zeichnung über ihre Daten- und Kontrollwege verbunden, also durch die Systembusse. Die dargestellten Bauteile werden folgend im einzelnen in ihrer Funktionalität, in ihrer Kommunikation und in ihren Besonderheiten einzeln betrachtet.

2.1 Speicher

Die Speichermöglichkeiten dieses Systems sind in keiner Weise mit denen moderner vergleichbar, was sowohl Kapazität als auch Zugriffszeiten angeht.

Der adressierbare Bereich von 64KByte ist gleichzeitig von Rom und Ram belegt, teilweise auch dreifach. Die sich überschneidenden Adressräume von Ram und Rom werden unter Nutzung eines zu diesem Zweck kodierten programmierbaren Chip (PLA) adressiert. Die CPU hat drei Steuersignale, mit deren Signalzuständen die Zugriffsarten über das PLA auscodiert werden. Die Speicherbausteine bekommen ihr CS-Signal von dem PLA.

Das folgende Bild zeigt die generelle Aufteilung des Speichers als Blockgrafik:



2.1.1 ROM-Speicher

Der Romspeicher besteht grundlegend aus drei Segmenten, als erstes der Basic-Interpreter mit 8KByte, als zweites der Zeichensatz-Speicher mit 4KByte und zum dritten das Betriebssystemrom mit 8Kbyte.

Speicherbereich mit den Zugehörigen Adressbereichen:

1. BASIC-ROM (\$A000-\$BFFF)
2. KERNAL-ROM (\$D000-\$FFFF)
 - o Zeichensatz-ROM (\$D000-\$DFFF)
 - o I/O Bausteine (\$D000-\$DFFF)
 - o Betriebssystem (\$E000-\$FFFF)

2.1.2 RAM-Speicher

Der Ramspeicher ist insgesamt 64KByte groß und in acht Speicherbänke unterteilt. Das erste KByte wird als quasi Cache und Registererweiterung von der CPU benutzt. Es werden hier unter anderem Systemflags und Register, wie etwa der Stackpointer und der Programmcounter abgelegt. Die erste Seite, die Zero-Page ist der CPU unter direkter Adressierung zugänglich, mit nur einem Byte Adressanteil.

Die nächste Seite (Page 1) ist als Stack nutzbar. In den anderen beiden Seiten werden verschiedene Systemparameter und andere nicht so eindeutig kategorisierbare Speicherinhalte abgelegt und verwaltet (auf die wir nicht näher eingehen werden).

Als nächstes kommt nun der Bildschirmspeicher. Wie die Bezeichnung schon sagt, werden in diesem Teil des Speichers die aktuell dargestellten Zeichen abgelegt. Die dem zugrundeliegenden Bitmuster sind im Zeichensatz-Rom gespeichert.

Anschließend wäre nun der Basic-Ram von größerem Interesse, dieser Bereich ist zuständig für die aktuell geladenen Basicprogramme. Die Programme werden als Mnemonischer-Code, verständlich für den Basic-Interpreter, dort abgelegt. Es ist mit 38KByte der größte zusammenhängende Speicherbereich, der auch als Programmspeicher für Assemblerprogramme dienen kann. Der folgende Speicherbereich, der wie bereits erläutert, von den ROM-Bausteinen benutzt wird, kann durch geeignete Konfiguration auch als RAM-Speicher benutzt werden. Z.B. kann man den Bereich des Zeichensatz-ROMs zum RAM umfunktionieren, um dort eigene Zeichensätze zu platzieren. Dieses Verfahren wurde oftmals bei den damaligen „Jump and Run“ spielen angewandt, da man so den Spielfluss immens steigern konnte.

Hier nun noch einmal die wesentlichen Merkmale im Überblick:

- unterteilt in 8 Bänke zu je 8kB
- direkt adressierbar (16 Adressleitungen)
- Speicherbereich von 0 bis 65535 (64kB)
 - Zeropage (\$0000 bis \$03FF)
 - Bildschirm (\$0400 bis \$07E8)
 - Frei (\$07E9 bis \$07FF)
 - Basic RAM (\$0800 bis \$9FFF)
 - Assembler/Daten (\$C000 bis \$CFFF)

2.2 Bus

Das Bus-System im C64 ist, wenn man es mit heutigen Systemen vergleicht, etwas konfus aufgebaut. Auf den ersten Blick ist es verwunderlich, dass die CPU gar nicht für die Buskontrolle verantwortlich ist. Diese Aufgabe übernimmt der VIC (Video Interface Control), der eigentlich für die Grafik verantwortlich ist. Man hat ihm aber auch die Buskontrolle gegeben, damit er den Bildschirmspeicher direkt auslesen kann. Weiterhin ist er für den Speicherrefresh zuständig, was auch ein Grund dafür ist, dass er der Busmaster ist. Wenn die CPU auf den Bus zugreifen will, muss sie ihn beim VIC beantragen. Der VIC gibt den Bus dann bei der nächsten Möglichkeit an die CPU ab. Wenn die CPU ihre Aufgabe erledigt hat, muss sie den Bus wieder freigeben.

Der Bus besteht aus folgenden Komponenten:

- Datenbus: 8 BIT
- Adressbus: 16 BIT
- Steuerbus
 - zur Konfiguration des Speichers
 - wird über PLA realisiert

Wie im Abschnitt zum Speicher schon angesprochen, dient der PLA des Steuerbusses zur Konfiguration des Speicherzugriffs. Über ihn kann man bestimmte Speicherbereiche so umkonfigurieren, dass man nicht mehr auf den darüberliegenden ROM-Speicher zugreift, sondern den RAM-Speicher anspricht. Wenn man diese Einstellungen während des laufenden Programms vornimmt, kann man die verfügbare Speicherkapazität um einige KBytes vergrößern, da man die konfigurierbaren Speicherteile doppelt oder sogar dreifach nutzt.

2.3 SID (Sound Interface Device)

Der SID ist der Baustein, der für die Tonerzeugung zuständig ist. Er hat eine programmierbare Lautstärke-Regelung in 16 Stufen. Die drei Stimmen sind unabhängig voneinander programmierbar und besitzen je 8 Oktaven Umfang. Der Baustein kann vier Wellenformen erzeugen, die alle mit entsprechenden Parametern für ein sehr vielfältiges Klangbild sorgen können. Ausserdem entnehmen wir der technischen Beschreibung: Amplituden-Modulation, Ring-Modulation, Oszillatoren-Synchronisation, programmierbarer Filter, getaktet mit 2 MHz. Seinerzeit war dieser Baustein relativ fortschrittlich und war daher von vielen Musikern und Komponisten als Spielwiese und Instrument entdeckt worden.

2.4 VIC (Video Interface Control / MOS 6569)

Mit der Bezeichnung Video Interface Control untertreibt man fasst bei der Bezeichnung dieses Bausteins. Er hat – abgesehen von der Darstellung – noch viele für das System existenziellere Aufgaben zu erledigen, als nur die Verarbeitung und Organisation der Video Daten. Grundlegend wäre da die Buskontrolle: Als Busmaster verwaltet er die Anfragen aus dem System und teilt dem Anfragenden Systempart (CPU, IO- Controller .etc.) die Buskontrolle zu. Dies ist naheliegend, da der VIC auch eine besondere Rolle für den Speicher spielt, so ist er auch zuständig für das Speicherrefreshing.

Der VIC besitzt 47 Register und einen eigenen Adressbus, über den er den Speicher auslesen und adressieren kann. Diesen benötigt er auch, da der allgemeine Adressbus ab und zu von anderen Systemkomponenten benötigt wird und der Bildschirminhalt trotzdem auf dem aktuellen Stand gehalten werden muss. Für den VIC ist der Speicher grundlegend in zwei Bereiche aufgeteilt. Zum einen ist da der Speicher, der die Art der Zeichen enthält und zum anderen gibt es noch den Bereich mit den Farbinformationen für die Entsprechenden Zeichen. Im normalen Zweifarben-Textmodus (40 mal 24 Zeichen) liegt der Zeichenspeicher ab Adresse 1024 und der Farbspeicher ab Adresse 55296 und beträgt je 1000 Byte. Jedem Zeichen – welches aus einer 8 mal 8 Zeichenmatrix besteht – kann so eine Vorder- und Hintergrundfarbe zugeteilt werden. Der VIC baut den Bildschirminhalt dann mit Hilfe des Zeichensatz-ROM auf. Die Ablageorte für die einzelnen Informationen können dabei aber auch in andere Speicherbereiche umgelagert werden. Dieses wird vor allem dann benutzt, wenn man den Bildschirminhalt möglichst schnell komplett austauschen will. Hierzu hat man

den neuen Bildschirminhalt bereits an einer anderen Speicherstelle im RAM (sowohl Zeichen- als auch Farbinformationen) aufgebaut und dann nur noch mittels der entsprechenden Register auf die anderen Informationen umgeschaltet. Der VIC hat dann das Bild anhand der neuen Basisadressen für Farb- und Zeicheninformationen aufgebaut.

Zusätzlich zu der normalen Blockgrafik des Textmodus kann der VIC noch acht Hardwareprites generieren. Diese sind kleine 24 mal 21 Bildpunkte große Bereiche, die Pixelgenau auf dem Bildschirm positioniert werden können. Sie beziehen ihren Inhalt aus einem frei konfigurierbaren Speicherbereich. Im gerade erwähnten 24 mal 21 Modus kann man ihnen allerdings nur zwei Farben geben. Es gibt aber auch noch die Möglichkeit die Anzahl der Farben auf vier zu steigern, was allerdings zulasten der horizontalen Auflösung geht. Diese halbiert sich dann nämlich. Der Spritegenerator besitzt noch weitere Register, die Auskunft darüber geben, ob ein Sprite mit dem Hintergrund oder ob zwei Sprites miteinander kollidiert sind.

Ein weiteres Merkmal des VIC ist, dass er auch in den Grafikmodus versetzt werden kann. In diesem Modus kann jedes Pixel des gesamten Bildschirms einzeln gesetzt werden. Die Informationen für die entsprechenden Farben werden nun in dem Bereich abgelegt, wo vorher die Zeicheninformationen des Textmodus waren. Für den Grafikmodus gibt es zwei verschiedene Auflösungen. Im hochauflösenden Modus stehen dem Programmierer 320 mal 200 Punkte zur Verfügung. Für die Farbinformation werden diese wiederum zu 8*8-Matrizen zusammengefasst. Diesen kann man nun je eine Vorder- und eine Hintergrundfarbe geben. Die Farbinformationen werden dabei wieder ganz normal im Color-RAM abgelegt. Im zweiten Modus, dem sogenannten „Multi-Color“-Modus wurde die horizontale Auflösung auf die Hälfte reduziert. Die Bytes, die für die Pixelinformation zuständig sind, werden so von acht Einzelbits auf vier Wertepaare reduziert. Jedes dieser Wertepaare entscheidet nun darüber, wo die Farbinformation für das zu generierende Pixels herkommen. Die Grobmatrix für die Farberzeugung ist nun auf 4*8 Pixel geschrumpft. Ein Wert von b00 in einem Pixelfeld sorgt nun dafür, dass die Farbe für den Hintergrund zu Anzeige verwendet wird. Bei einer b01 werden die unteren vier Bits des Video-RAMs für den zugehörigen 4*8 Pixelbereich genommen, bei einer b10 die entsprechenden oberen vier Bits des gleichen Bereiches. Enthält das Pixelfeld eine b11, so wird die Farbe aus dem Color-RAM genommen. Wenn man sich einmal näher mit dieser Thematik auseinandersetzt, stellt man fest, dass sich die Farbvielfalt nicht verdoppelt, sondern dass man nur zwei zusätzliche globale Farben bekommt, die aber für den gesamten Bildschirminhalt zuständig sind.

2.5 Schnittstellen, Speichergeräte und Peripherie

Die Schnittstellen des C64 bestehen im einzelnen aus zwei Joystick-Ports, einen Seriellen-Port (Drucker, Floppy etc.), dem „Userport“ und dem Anschluss für die Datasette(Speichermedium mit Musikkassetten).

Alles was etwas mit Datentransfer zu tun hat geht bei dem C64 über Bausteine mit der Bezeichnung Common Interface Adapter. Es gibt davon zwei Stück, der erste ist zuständig für die Bearbeitung der Tastatureingaben und die der beiden Joystickanschlüsse. Der zweite nimmt sich des Userports und des Seriellen Busses an. Die Arbeitsweise dieses Bausteins ist die eines jeden I/O-Controllers, mit jeweils zwei acht Bit und einem seriellen Ein/Ausgang.

Die Speichergeräte beschränken sich auf die schon erwähnte Datasette und einem Diskettenlaufwerk, von denen bis zu vier Geräte gleichzeitig betrieben werden können. Die

Datensette hat ihren eigenen Anschluss, die Motorsteuerung kommt direkt über Treiberstufen aus der CPU. Der Datenkanal wird über den ersten CIA abgewickelt (serieller I/O-Anschluss). Die Floppylaufwerke werden an den seriellen Port angeschlossen und sind hintereinander anschließbar.

Das Herz der Floppylaufwerke leitet uns nun direkt zu dem Zentrum dieser Rechnerarchitektur, es ist eine CPU vom Typ 6502, welche das Vorgängermodell der 6510 ist.

3.1 Allgemeines

Die 6510 CPU wird mit einem Takt von ca. 1MHz (985kHz) betrieben. Sie verfügt über einen acht Bit breiten Datenbus und kann über ihren 16 Bit breiten Adressbus 65536 (64Kbyte) Speicherstellen adressieren. Sie besitzt sieben Register und eine Acht-Bit Alu. Weiterhin gibt es die sogenannte Zeropage, welche von der CPU über ein Byte adressiert werden kann und daher für Registerähnliche Funktionen verwendet wird. Für die 56 Befehle, die sie kennt, stellt sie elf verschiedene Adressierungsarten zur Verfügung. Daraus ergibt sich ein Befehlsumfang von 151 verschiedenen Kombinationen. Es gibt zwei Interrupteingänge: IRQ und NMI. Der IRQ kann von der CPU bearbeitet werden, er kann aber auch abgelehnt werden. Liegt allerdings eine Unterbrechungsanforderung am NMI an, so muss die CPU das laufende Programm nach dem aktuellen Befehl unterbrechen und die Unterbrechungsroutine des NMI ausführen. Sie kann den NMI also nicht ablehnen. Für jeden Eingang gibt es einen zwei Byte großen Vektoreintrag, der auf die entsprechende Serviceroutine zeigt. Die Routinen selber fragen dann ein paar weitere Speicherbereiche ab, um feststellen zu können, wer die Unterbrechung veranlasst hat. Um eigene Serviceroutinen zu implementieren, muss man lediglich die entsprechenden Vektoreinträge manipulieren und von seiner Routine nach Beendigung der eigenen Subroutine in die Ursprungsroutine zurück zu springen.

3.2 Register

Die Register des Prozessors haben alle eine Breite von acht Bit. Der Programmcounter, besteht daher aus einem LOW- und einem HIGH-Register. So kann er den gesamten Speicherbereich von 64Kbyte abdecken. Der Stackpointer zeigt auf einen 256Byte großen Bereich oberhalb der Zero-Page (standardmäßig Page 1).

Das Statusregister mit seinen acht einzelnen Bits, von denen aber nur sieben benutzt werden. Die Bedeutung der einzelnen Bits wird im Kapitel 6.2. erklärt.

Nun kommen die Beiden X und Y Register. Sie werden als Indizes für die indirekte und indizierten Adressierungsarten verwendet.

Das wichtigste Register ist der Akkumulator(Akku). Alle Rechenoperationen, sowie sämtliche Speicherzugriffe werden über dieses Register realisiert.

3.3 ALU

Die ALU hat ebenfalls eine Datenbreite von Acht-Bit, Zielregister aller Operationen ist das Akku-Register. Sie verfügt über fünf verschiedene Operationen, der arithmetischen Addition und Subtraktion und der logischen AND-, OR- sowie NOT-Operation. Das D-Statusregister ist für die Alu die Speicherstelle für ein Flag, welches anzeigt ob eine Rechenoperation im Binär oder im BCD Modus berechnet werden soll.

Ansonsten hat das Ergebnis der ALU Auswirkungen auf die Flags (C,Z,V,N) des oben erwähnten Statusregisters.

3.4 Befehlssatz

Der Befehlssatz Setzt sich grundlegend aus 56 verschiedenen Befehlen zusammen, von denen sich dann einige infolge ihrer verschiedenen Adressierungsarten (bis zu acht) 151

verschiedene Möglichkeiten ergeben. Insgesamt verfügt die CPU über elf Adressierungsarten, die sich auch auf die Anzahl der Taktzyklen pro Befehl auswirken. Die Befehle haben inhomogene Längen, und zwar 1, 2 und 3 Byte.

Strebt man nun eine Kategorisierung in RISC oder CISC Prozessor an, so ergeben sich Argumente die auf beide Technologien hinweisen. Es gibt aber auch Argumente, die gegenteiliges Anzeigen.

RISC Merkmale (nach D.Tabak)	Erfüllt	Auswertung
Weniger als 50 Maschinenbefehle	(+)	Wenn man die Grenze von 50 Befehlen etwas aufweicht, dann hat der C64 mit seinen 56 verschiedenen Befehlen dieses Kriterium erfüllt.
Weniger als 4 Adressierungsarten	--	11 Adressierungsarten
Speicherzugriffe nur über LOAD/STORE-Befehle	-	Der Speicher kann auch direkt manipuliert werden.
Mehr als 32 Prozessor-Register	--	Weniger als zehn Register.
Festverdrahtete Maschinenbefehle (keine Mikroprogrammierung)	+	
Ausführung der „meisten“ Befehle in einem Taktzyklus	--	Abarbeitung erfolgt in zwei bis sieben Zyklen.
Höhere Programmiersprachen werden durch optimierende Compiler unterstützt	-	Es gibt Compiler für diese CPU, die wir aber nicht klassifizieren können.

CISC Merkmale	Erfüllt	Auswertung
Befehlssätze sind groß und mikroprogrammiert	-	151 Kombinationen der 56 Befehle
Mikroprogrammierung ermöglicht Rechnerfamilien	-	
Assemblerprogrammierung vorherrschend	++	Die Basic-Programme werden vom Interpreter auch in Assembler gewandelt.
Befehlsformate sind sehr unregelmäßig, oberstes Ziel: gute Ausnutzung des HS	++	Ein bis Drei Bytes, Zwei bis Sieben Zyklen, Null bis Zwei Parameter
Befehle sind komplex (weil Mikrocode schneller ist als Maschinencode)	-	Multiplikation wird durch Maschinencode realisiert, da sie als Mikrocode nicht vorhanden ist.
Kurze Maschinenprogramme	-	Komplexere Funktionen müssen hier mit Assembler nachgebildet werden.
Nur geringe Anzahl von Registern zugunsten von Speicher-Speicher-Operationen	+/-	Direkte Manipulation von Speicherstellen ist möglich. Es gibt aber keinen Speicher-Speicher Transfer.
Caches sind sehr viel kleiner als die Mikroprogrammspeicher (Control-Memory)	++	Der Cache strebt in unserem Fall sogar gegen Null.

Eine eindeutige Klassifizierung lässt sich anhand der oben stehenden Kriterien nicht vornehmen, da weder die CISC-, noch die RISC-Merkmale zu einem Grossteil erfüllt werden. Weiterhin gibt es aber sowohl CISC-, als auch RISC-Merkmale, die als erfüllt betrachtet werden können. Aus Sicht der Programmierung lässt sich die CPU eher als ein RISC-Prozessor betrachten, da er einen übersichtlichen Befehlssatz hat.

Eine eindeutige Charakterisierung fällt natürlich auch schwer, da gewisse Bestandteile heutiger CPU's damals noch gar nicht erfunden waren (z.B. Pipelining, Caching).

Beispiele des Befehlssatzes:

■ JSR

#	address	R/W	Description
1	PC	R	fetch opcode, increment PC
2	PC	R	fetch low address byte, increment PC
3	\$0100,S	R	internal operation
4	\$0100,S	W	push PCH on stack, decrement S
5	\$0100,S	W	push PCL on stack, decrement S
6	PC	R	copy low address byte to PCL, fetch high address byte to PCH

■ RTS

#	address	R/W	description
1	PC	R	fetch opcode, increment PC
2	PC	R	read next instruction byte (and throw it away)
3	\$0100,S	R	increment S
4	\$0100,S	R	pull PCL from stack, increment S
5	\$0100,S	R	pull PCH from stack
6	PC	R	increment PC

■ AND (indiziert indirekt)

#	address	R/W	Description
1	PC	R	fetch opcode, increment PC
2	PC	R	fetch pointer address, increment PC
3	pointer	R	read from the address, add X to it
4	pointer+X	R	fetch effective address low
5	pointer+X+1	R	fetch effective address high
6	address	R	read from effective address

4. Fazit

Die Auseinandersetzung mit der Technik des c64 war für uns eine Herausforderung, das Wissen aus Verschiedenen Studienfächern diene uns hierbei als Grundlage. Ein solches ‚Komplettsystem‘ beinhaltet viele interessante Aspekte, wie etwa Realisierung der Interrupte, Anbindung, Einteilung und Verwendung der Speicher. Wer macht was im System, (ein Videocontroller als Busmaster ist doch was!), usw. Interessant ist auch der Versuch einer kategorischen Zuordnung, wie beim Vergleich von RISC vs. CISC. Wir hatten so die Möglichkeit unseren früheren Schwarm mal mit anderen Augen zu betrachten und zu Analysieren. Setzt man nun die Welt mit unseren Kinderzimmern gleich, so lässt sich der Titel natürlich nur bestätigen. Zur Weltherrschaft hat es in nie gereicht, aber er war sehr populär und weit verbreitet. Er ebnete sozusagen den Weg der Rechner in die Haushalte. So könnte man vielleicht auch eine gewisse Rechtfertigung des Titels in Betracht ziehen.

5. Anhang

5.1 Adressierungsarten

Art	Beschreibung
Unmittelbar	Es wird direkt ein Wert angegeben z.B.: #\$FF
Implizit	Bedarf keinerlei zusätzlicher Adressierung (BSP.: NOP)
Absolut	direkt zugriff auf eine Speicherstelle des RAM's
Zeropage	Speicherstelle von 0 bis 255 (benötigt ein BYTE weniger)
Absolut-X-Indiziert	Direkte Speicheradresse + Inhalt des X-Registers
Absolut-Y-Indiziert	Direkte Speicheradresse + Inhalt des Y-Registers
Zeropage-X-Indiziert	X. Register der Zeropage
Indirekt	Spricht eine Adresse an, in der wiederum eine Adresse steht
Indirekt-X-Indiziert	Spricht eine Adresse + X an, die wiederum eine Adresse enthält
Indirekt-Y-Indiziert	Spricht eine Adresse + Y an, die wiederum eine Adresse enthält
Relativ	Es wird um Anzahl Byte gesprungen (Sprungweite -128 bis 127)

5.2 Statusregister

Das Statusregister hat folgendes Format.

Bit	7	6	5	4	3	2	1	0
Abkürzung	N	V	I	B	D	I	Z	C

Das I-Flag ist dabei auf zwei Bits gelegt worden. Dieses beruht wahrscheinlich auf Schaltungstechnischen Grundlagen. Man konnte so einige Gatter und Verknüpfungen sparen. Die einzelnen Bits haben dabei die folgenden Bedeutungen:

- C, Carry Flag(Über- bzw. Unterlaufsflag)
Dieses Flag zeigt an, ob bei einer Rechenoperation ein Über- oder Unterlauf aufgetreten ist. Es liefert auch das Ergebnis eines Vergleichs, wenn beide Zahlenwerte nicht identisch sind.
- Z, Zero Flag(Null-Flag)
Diese Flag wird gesetzt, wenn das Ergebnis einer Operation Null ist und wenn bei einem Vergleich beide Zahlenwerte identisch sind.
- I, Interrupt Disable (Interrupt-Flag)
Mittels diesem Flag wird festgelegt, ob ein Maschinenprogramm durch einen Interrupt unterbrochen werden darf oder nicht.
- D, Decimal (Dezimalmodus)
Über dieses Flag wird bestimmt, ob Additionen oder Subtraktionen im Dezimalmodus durchgeführt werden sollen. Dieser Modus ermöglicht eine höhere Rechengenauigkeit.
- B, Break(Abbruch)
Dieses Flag wird gesetzt, wenn ein Maschinenprogramm durch den Befehl BRK unterbrochen wird.
- V, Overflow(Überlauf)
Diese Flag zeigt an, ob bei Rechenoperationen ein Überlauf in das Vorzeichenbit aufgetreten ist oder nicht.
- N, Negativ(Negativ-Flag)
Dieses Flag wird gesetzt, wenn das Ergebnis einer Operation größer als 127 ist und demzufolge das siebente Bit gesetzt wurde.

5.3 Befehlsübersicht nach Kategorien

5.3.1 Vergleichsbefehle

Befehl	Beschreibung
CMP	Vergleicht Speicherstelle mit Akku
CPX	Vergleicht Speicherstelle mit X-Register
CPY	Vergleicht Speicherstelle mit Y-Register

5.3.2 Sprungbefehle

Befehl	Beschreibung
BCC	Verzweigt, falls das Carry-Flag gelöscht ist.
BCS	Verzweigt, falls das Carry-Flag gesetzt ist.
BCQ	Verzweigt, falls das Zero-Flag gelöscht ist.
BNE	Verzweigt, falls das Zero-Flag gesetzt ist.
BMI	Verzweigt, falls das Negativ-Flag gelöscht ist.
BPL	Verzweigt, falls das Negativ-Flag gesetzt ist.
B	Verzweigt, falls das Overflow-Flag gelöscht ist.
BVS	Verzweigt, falls das Overflow-Flag gesetzt ist.
JMP	Verzweigt zur angegebenen Adresse
JSR	Verzweigt zur angegebenen Unterprogrammadresse und kehrt nach Beendigung des Unterprogramms wieder zurück
NOP	Keine Operation
RTS	Rücksprung aus einem Unterprogramm

5.3.3 Unterbrechungsbefehle

Befehl	Beschreibung
BRK	Programmunterbrechung
CLI	Interrupt-Flag löschen
RTI	Rücksprung aus Interrupt
SEI	Interrupt-Flag setzen

5.3.4 In- / Dekrementierbefehle

Befehl	Beschreibung
DEC	Dekrementiert den Inhalt der angegebenen Speicherstelle
DEX	Dekrementiert den Inhalt des X-Registers
DEY	Dekrementiert den Inhalt des Y-Registers
INC	Inkrementiert den Inhalt der angegebenen Speicherstelle
INX	Inkrementiert den Inhalt des X-Registers
INY	Inkrementiert den Inhalt des Y-Registers

5.3.5 Transferbefehle

Befehl	Beschreibung
LDA	Lädt den Akku mit einem angegebenen Wert
LDX	Lädt das X-Register mit einem angegebenen Wert
LDY	Lädt das Y-Register mit einem angegebenen Wert
PHA	Bringt den Akku-Inhalt auf den Stack
PHP	Bringt den Inhalt des Status-Registers auf den Stack
PLA	Lädt den Akku mit dem obersten Stack-Element
PLP	Lädt das Status-Register mit dem obersten Stack-Element
STA	Schreibt den Inhalt des Akkus in die angegebene Speicherstelle
STX	Schreibt den Inhalt des X-Registers in die angegebene Speicherstelle
STY	Schreibt den Inhalt des Y-Registers in die angegebene Speicherstelle
TAX	Kopiert den Akku-Inhalt in das X-Register
TAY	Kopiert den Akku-Inhalt in das Y-Register
TXA	Kopiert den Inhalt des X-Registers in den Akku
TYA	Kopiert den Inhalt des Y-Registers in den Akku
TSX	Schreibt den Inhalt des Stack-Pointers in das X-Register
TXS	Schreibt den Inhalt des X-Registers in den Stack-Pointer

5.3.6 Arithmetische Befehle

Befehl	Beschreibung
ADC	Addiert einen Operanden und das Carry-Flag zum Akku-Inhalt
CLC	Löscht das Carry-Flag
CLD	Löscht das Dezimal-Flag
CLV	Löscht das Overflow-Flag
SBC	Subtrahiert einen Operanden und das Carry-Flag vom Akku-Inhalt
SEC	Löscht das Carry-Flag
SED	Setzt das Dezimal-Flag

5.3.7 Logische Befehle

Befehl	Beschreibung
AND	Verknüpft den Akku-Inhalt mit einem Operanden durch ein logisches UND.
BIT	Erweiterter AND-Befehl mit setzen der entsprechenden Flags
EOR	Verknüpft den Akku-Inhalt mit einem Operanden durch ein logisches EXCLUSIV-ODER.
ORA	Verknüpft den Akku-Inhalt mit einem Operanden durch ein logisches ODER.

5.3.8 Schiebe- und Rotierbefehle

Befehl	Beschreibung
ASL	Verschiebt die Bits des Operanden um eine Stelle nach links.
LSR	Verschiebt die Bits des Operanden um eine Stelle nach rechts.
ROL	Rotiert die Bits des Operanden um eine Stelle nach links.
ROR	Rotiert die Bits des Operanden um eine Stelle nach rechts.

5.4 Quellenangaben

5.4.1 Literatur

- Martin Hecht
„Das grosse Commodore 64 Buch“
Data Becker Verlag
ISBN: 3-89011-370-2
- Axel Plenge
„Das SUPERGRAFIK Buch zum Commodore 64“
Data Becker Verlag
ISBN:3-89011-213-7
- Wester
„Das Betriebssystem des Commodore 64“
Data Becker Verlag
ISBN:3-89011-136-x
- Englisch, Gerits, Thrun, Hartwig, Löffelmann
„64 Tips & Tricks“
Data Becker Verlag
ISBN:3-89011-001-0
- Florian Müller
„c64 Tips, Tricks und Tools“
Markt & Technik Verlag
ISBN:3-89090-499-8
- Diverse Sonderhefte der Zeitschrift „64'er“
Sonderheft 35 „Assembler“
Sonderheft 65 „Tips & Tools“
Sonderheft 71 „Assembler leicht gemacht“
Markt & Technik Verlag

5.4.2 Internet

- „MJK's Commodore Hardware Overview: Commodore 64“
<http://www.student.informatik.tu-darmstadt.de/~mjk/nepa/dev/c64.html>
- „Commodore Heaven“
<http://www.cbmhardware.de/c64/c64.php3?Lang=D>
- "The Official Unofficial C=Hacking Homepage"
<http://www.ffd2.com/fridge/chacking/>

Da uns der Favoritenordner leider abhanden gekommen ist, ist es uns nicht mehr möglich alle Links anzugeben, von denen wir unsere Informationen bekommen haben.