

IBS-COMPUTERTECHNIK

ARTUR-LADEBECK-STR. 143, 4800 BIELEFELD 14  
TELEFON: 0521-152807

APPLE II - INTERFACES

AP4 - PARALLEL-SCHNITTSTELLE

INSTALLATIONS- UND BENUTZUNGSHINWEISE

BITTE ERST SORGFÄLTIG DURCHLESEN,

DANN DIE KARTE BENUTZEN!

## AP4 - PARALELL I/O-KARTE

### HARDWARE

Das Herz der AP4 ist der VIA (Versatile Interface Adapter) 6522. Er hat zwei 8-bit bidirektionale I/O-Ports, zwei 16-bit programmierbare Zeitgeber bzw. Zaehler, serielle Datenports und ist handshake-faehig. In einem 2K Eprom kann der Anwender seine spezifische Software ablegen, so dass der Anwender bestimmen kann, fuer welchen Zweck er diese universelle Karte einsetzen moechte.

Die folgenden Beispiele zeigen typische Anwendungsfaelle

- a) paralleles Druckerinterface
- b) digitale Messwertfassung, Ein- und Ausgabe
- c) Kopplung des Apple mit anderen DV-Geraeten
- d) Realisierung von Interrupts zur Echtzeitverarbeitung

Neben dem VIA und dem Eprom befinden sich auf der Karte einige TTL-IC's, die der Address-Dekodierung und Datenpufferung dienen. Die Verbindung der Karte mit der gewuenschten Peripherie geschieht ueber den 26-pol. Stecker auf der Karte. Die Tabelle 1 zeigt dessen Belegung.

### INSTALLATION

**SCHALTEN SIE ZUNAECHST IHREN APPLE AUS, SOWIE ALLE AN DIE KARTE ANZUSCHLIESSENDEN GERAETE!**

Nach Abnahme des Deckels werden die acht Verbindungsstecker (Slots) des APPLE sichtbar. Die Karte kann in jeden der Slots 1 bis 6 eingesetzt werden, nicht jedoch in Slot 0 oder Slot 7. Slot 0 ist fuer die 16K-Speichererweiterung (AP1) und Slot 7 fuer die Pal-Coder-Karte (AP3) reserviert.

Setzen Sie also die Karte in einen der Slots 1 bis 6. Danach koennen Sie das Verbindungskabel an Ihre aeuessere Hardware anschliessen.

Wenn alle Verbindungen erstellt sind, den APPLE schliessen.

Sie koennen nun wieder die Geraete einschalten, Ihre AP4 ist einsatzbereit.

Voraussetzung zur Funktion der Karte ist natuerlich, dass eine geeignete Software vorhanden ist, mit der der VIA etwas anfangen kann, denn die Funktionen der Karte haengen im Wesentlichen von der Programmierung des VIA ab.

## PROGRAMMIERUNG

Die Software fuer die Karte kann sowohl im Speicherbereich des APPLE stehen oder aber in einem 2K-Eprom, welches auf der Karte sich befindet. Wird ein Eprom benutzt, so sind die Hinweise im Kapitel 'PROGRAMMIERUNG DES EPROMS' zu beachten.

Mittels der Software kann jedes Bit der beiden 8-bit I/O-Ports wahlweise als Eingang (I) oder Ausgang (O) programmiert werden. Zusaetzlich koennen zwei 16-bit Timer programmiert werden, sowie ein Schieberegister, welches als serielles Ein- wie auch als Ausgaberegister arbeiten kann.

Mit Hilfe einer leistungsfaeihigen Interruptstruktur koennen saemtliche Bausteinkomponenten ueber Interrupt bedient werden.

Die Addressierung der insgesamt 16 8-bit Register des VIA durch den APPLE erfolgt wie bei einem gewoehnlichen Speicherplatz. Bei der Initialisierung des VIA von der Software wird dessen Betriebsart durch Laden der VIA-Register mit entsprechenden Daten festgelegt. Diese Daten richten sich nach dem Anwendungsfall. Ebenfalls findet der Datenaustausch zwischen APPLE und VIA ueber einige dieser Register statt.

Die Adressen der Register sind von der Slot-Nummer abhaengig, in den die Karte eingesetzt wurde. Fuer das niedrigste Register wird die Grundadresse mit  $49152 + (n * 16)$  berechnet, wobei  $n = \text{Slotnr.} + 8$ . Fuer diejenigen, die stets bis \$F zaehlen ist die Grundadresse \$C0n0,  $n = \text{Slotnr.} + 8$ .

Fuer die folgenden 15 Register erhoehrt sich die Adresse um jeweils 1, d.h. fuer das letzte Register lautet die Adresse  $49152 + (n*16) + 15$ .

Der Tabelle 2 sind die Registerbezeichnungen, Adressen und die Registerfunktionen beschrieben.

## PROGRAMMIERUNG DES EPROMS

Das Eprom muss ein INTEL 2716, TMS 2516 oder ein aequivalenter Typ sein.

Das Eprom liegt mit seinem Bereich \$700 - \$7FF (256 Bytes) im Bereich Cn00 bis CnFF, wobei  $n = \text{Slotnr.}$ , der Eprom-Bereich 000 bis 6FF liegt im APPLE-Address-Bereich C800 - CEFF.

Der Bereich 700-7FF (Cn00-CnFF) wird angesprochen, wenn der Basic-Befehl IN# oder PR# gegeben wird. Im diesem Bereich sollte demnach die VIA-Initialisierung liegen, sowie die apple-interne Berechnung des Slots, dann folgt ein Sprung in den Bereich 000-6FF (C800-CEFF). Im APPLE-Referenz-Manual (Nr. A2L0001A) steht naeheres auf den Seiten 80 ff.

Um das Eprom mit Ende des Programms zu deselektieren, muss vor dem Programmende-Befehl, meistens ein RETURN (OP-CODE \$60), der Befehl STA \$CFFF stehen.

Mit IN#n, PR#n oder JMP \$Cn00 wird das Eprom wieder eingeschaltet und das Programm abgearbeitet.

Tabelle 1

Steckerbelegung (26-pol Stecker, B kennzeichnet Pin 1)

Pin Funktion = Pin an VIA

1...	+5V (Vcc)	...n/a
2...	+5V	...n/a
3...	NC	
4...	NC	
5...	CB1	...18
6...	CB2	...19
7...	PB6	...16
8...	PB7	...17
9...	PB4	...14
10...	PB5	...15
11...	PB2	...12
12...	PB3	...13
13...	PB0	...10
14...	PB1	...11
15...	PA6	... 8
16...	PA7	... 9
17...	PA4	... 6
18...	PA5	... 7
19...	PA2	... 4
20...	PA3	... 5
21...	CA2	...39
22...	PA1	... 3
23...	CA1	...40
24...	PA0	... 2
25...	0V (GND)	...n/a
26...	0V	

Tabelle 2

Adresse (n=Slotnr. + 8)

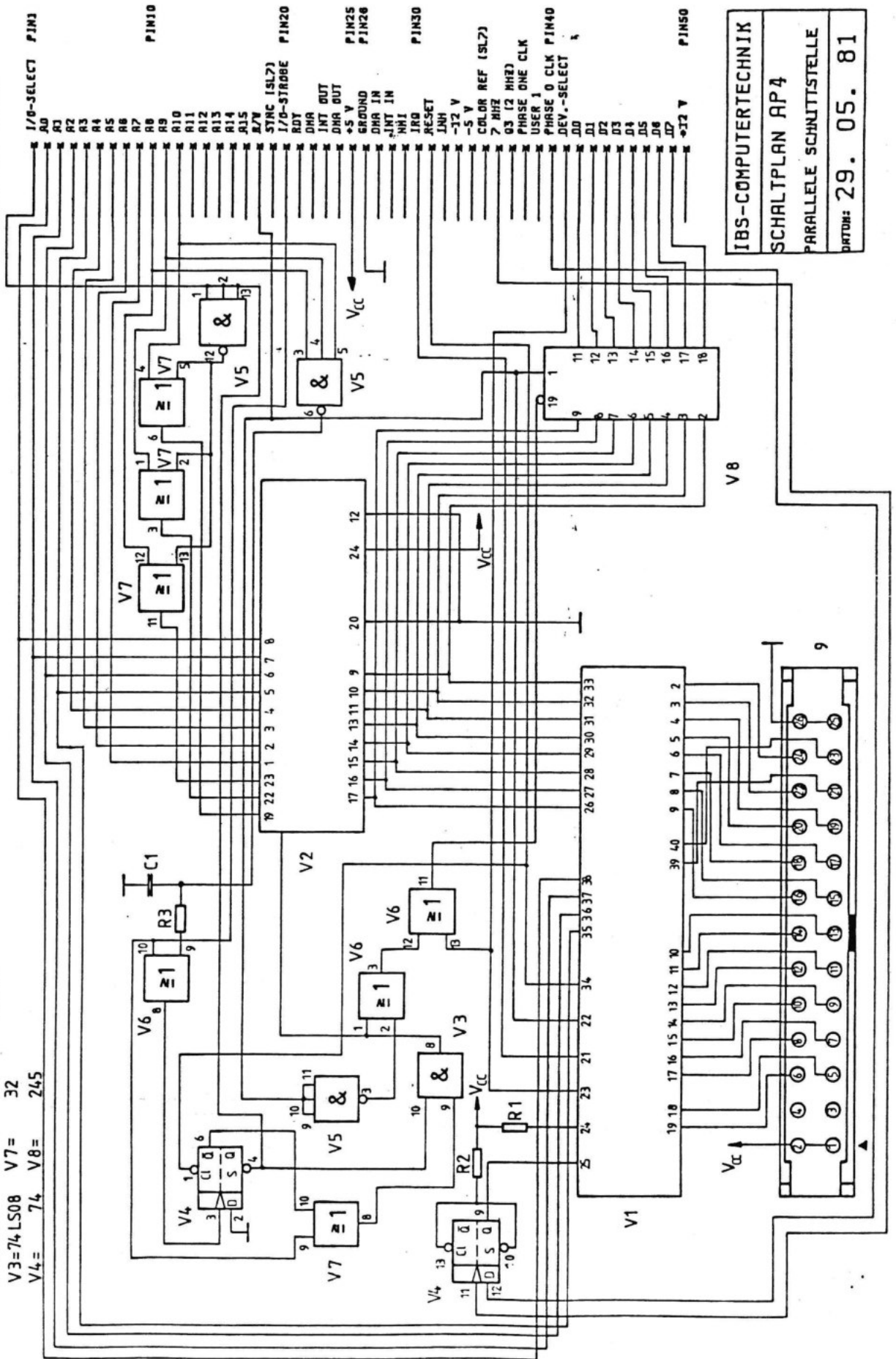
hex	dezimal	Register	Funktion
C0n0	49152+n*16	ORB	Datenkanal B
C0n1	49153+n*16	ORA	Datenkanal A
C0n2	49154+n*16	DDRB	Datenrichtung Kanal B
C0n3	49155+n*16	DDRA	Datenrichtung Kanal A
C0n4	49156+n*16	T1 Counter Register	Low-Byte
C0n5	49157+n*16	T1 Counter Register	High-Byte
C0n6	49158+n*16	T1 Latch Register	Low-Byte
C0n7	49159+n*16	T1 Latch Register	High-Byte
C0n8	49160+n*16	T2 Counter Register	Low-Byte
C0n9	49161+n*16	T2 Counter Register	High-Byte
C0nA	49162+n*16	SR	Schieberegister 8 Bit
C0nB	49163+n*16	ACR	Kontrolle von T2 und SR
C0nC	49164+n*16	PCR	CA- und CB-Signal-Kontrolle
C0nD	49165+n*16	IFR	Interrupt-Flag-Register
C0nE	49166+n*16	IER	Interrupt-Enable-Register
C0nF	49167+n*16	ORG/wo	Funktion wie ORG, jedoch ohne Handshake-Funktionen

Eine ähnliche Tabelle, aber ohne Adressenangabe befindet sich in der VIA-Beschreibung auf Seite 9. In dieser Beschreibung sind auch die Register genau erläutert.



# APPLE-SLUT

- V1= V5=74LS10 9 = Steckerleiste 9 = 26 Pins
- V2=2716 V6= 32 26 Pins
- V3=74LS08 V7= 32
- V4= 74 V8= 245



IBS-COMPUTERTECHNIK  
 SCHALTPLAN AP4  
 PARALLELE SCHNITTSTELLE  
 DATUM: 29. 05. 81

SY6522/SY6522A

PACKAGE OUTLINE

NOTE: Pin No. 1 is in lower left corner when symbolization is in normal orientation.

ORDERING INFORMATION

Order Number	Package Type	Frequency Option
SYP 6522	Plastic	1 MHz
SYP 6522A	Plastic	2 MHz
SYC 6522	Ceramic	1 MHz
SYC 6522A	Ceramic	2 MHz

PIN CONFIGURATION

VIS	1	40	CA1
PA0	2	39	CA2
PA1	3	38	RA0
PA2	4	37	RA1
PA3	5	36	RA2
PA4	6	35	RA3
PA5	7	34	RES
PA6	8	33	D0
PA7	9	32	D1
PA8	10	31	D2
PA9	11	30	D3
PA10	12	29	D4
PA11	13	28	D5
PA12	14	27	D6
PA13	15	26	D7
PA14	16	25	D8
PA15	17	24	D9
PA16	18	23	CS1
PA17	19	22	CS2
PA18	20	21	R/W
VCC	20	21	IRQ

Versatile Interface Adapter (VIA)

EINSTEINSTRASSE 127  
8000 MÜNCHEN 80  
TELEFON 089/4702098  
TELEX 5 212 931 bl/v

MICROPROCESSOR PRODUCTS  
Preliminary  
APRIL 1979

- Two 8-Bit Bidirectional I/O Ports
- Two 16-Bit Programmable Timer/Counters
- Serial Data Port
- Single +5V Power Supply
- TTL Compatible
- CMOS Compatible Peripheral Control Lines

- Expanded "Handshake" Capability Allows Positive Control of Data Transfers Between Processor and Peripheral Devices
- Latched Output and Input Registers
- 1 MHz and 2 MHz Operation

The SY6522 Versatile Interface Adapter (VIA) is a very flexible I/O control device. In addition, this device contains a pair of very powerful 16-bit interval timers, a serial-to-parallel/parallel-to-serial shift register and input data latching on the peripheral ports. Expanded handshaking capability allows control of bi-directional data transfers between VIA's in multiple processor systems.

Control of peripheral devices is handled primarily through two 8-bit bi-directional ports. Each line can

be programmed as either an input or an output. Several peripheral I/O lines can be controlled directly from the interval timers for generating programmable frequency square waves or for counting externally generated pulses. To facilitate control of the many powerful features of this chip, an interrupt flag register, an interrupt enable register and a pair of function control registers are provided.

Figure 1. SY6522 Block Diagram

SYNERTEK® INC. • P.O. BOX 552 • SANTA CLARA, CALIFORNIA 95052 • TEL. 408/986-5600 • TWX: 910-338-0135

SYNERTEK® INC. • P.O. BOX 552 • SANTA CLARA, CALIFORNIA 95052 • TEL. 408/986-5600 • TWX: 910-338-0135

B-15K-4778



**ABSOLUTE MAXIMUM RATINGS**

Rating	Symbol	Value	Unit
Supply Voltage	V <sub>CC</sub>	-0.3 to +7.0	V
Input Voltage	V <sub>IN</sub>	-0.3 to +7.0	V
Operating Temperature Range	T <sub>A</sub>	0 to +70	°C
Storage Temperature Range	T <sub>stg</sub>	-55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages. However, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages.

**ELECTRICAL CHARACTERISTICS (V<sub>CC</sub> = 5.0V ± 5%, T<sub>A</sub> = 0-70°C unless otherwise noted)**

Symbol	Characteristic	Min.	Max.	Unit
V <sub>IH</sub>	Input High Voltage (all except φ2)	2.4	V <sub>CC</sub>	V
V <sub>CH</sub>	Clock High Voltage	2.4	V <sub>CC</sub>	V
V <sub>IL</sub>	Input Low Voltage	-0.3	0.4	V
I <sub>IN</sub>	Input Leakage Current - V <sub>IN</sub> = 0 to 5 Vdc R/W, RES, RS0, RS1, RS2, RS3, CS1, CS2, CA1, φ2	-	±2.5	μA
I <sub>TSI</sub>	Off-state Input Current - V <sub>IN</sub> = .4 to 2.4V V <sub>CC</sub> = Max, D0 to D7	-	±10	μA
I <sub>IH</sub>	Input High Current - V <sub>IH</sub> = 2.4V PA0-PA7, CA2, PBO-PB7, CB1, CB2	-100	-	μA
I <sub>IL</sub>	Input Low Current - V <sub>IL</sub> = 0.4 Vdc PA0-PA7, CA2, PBO-PB7, CB1, CB2	-	-1.6	mA
V <sub>OH</sub>	Output High Voltage V <sub>CC</sub> = min, I <sub>load</sub> = -100 μA PA0-PA7, CA2, PBO-PB7, CB1, CB2	2.4	-	V
V <sub>OL</sub>	Output Low Voltage V <sub>CC</sub> = min, I <sub>load</sub> = 1.6 mA	-	0.4	V
I <sub>OH</sub>	Output High Current (Sourcing) V <sub>OH</sub> = 2.4V V <sub>OH</sub> = 1.5V (PBO-PB7)	-100	-	μA
I <sub>OL</sub>	Output Low Current (Sinking) V <sub>OL</sub> = 0.4 Vdc	-1.0	-	mA
I <sub>OFF</sub>	Output Leakage Current (Off state) I <sub>FO</sub>	-	10	μA
C <sub>IN</sub>	Input Capacitance - T <sub>A</sub> = 25°C, f = 1 MHz (R/W, RES, RS0, RS1, RS2, RS3, CS1, CS2, D0-D7, PA0-PA7, CA1, CA2, PBO-PB7) (CB1, CB2) (φ2 Input)	-	7.0	pF
C <sub>OUT</sub>	Output Capacitance - T <sub>A</sub> = 25°C, f = 1 MHz	-	10	pF
P <sub>D</sub>	Power Dissipation	-	700	mW

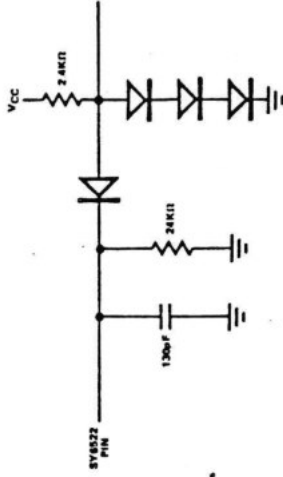


Figure 2. Test Load (for all Dynamic Parameters)

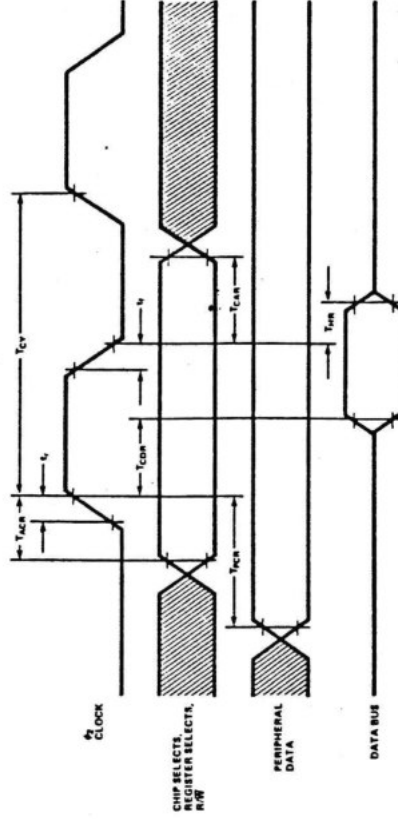


Figure 3. Read Timing Characteristics

**READ TIMING CHARACTERISTICS (FIGURE 3)**

Symbol	Parameter	SY6522		SY6522A		Unit
		Min.	Max.	Min.	Max.	
T <sub>CY</sub>	Cycle Time	1	50	0.5	50	μs
T <sub>ACR</sub>	Address Set-Up Time	180	-	90	-	ns
T <sub>CAR</sub>	Address Hold Time	0	-	0	-	ns
T <sub>PCR</sub>	Peripheral Data Set-Up Time	300	-	300	-	ns
T <sub>CDR</sub>	Data Bus Delay Time	-	395	-	200	ns
T <sub>HR</sub>	Data Bus Hold Time	10	-	10	-	ns

NOTE: tr, tf = 10 to 30ns.



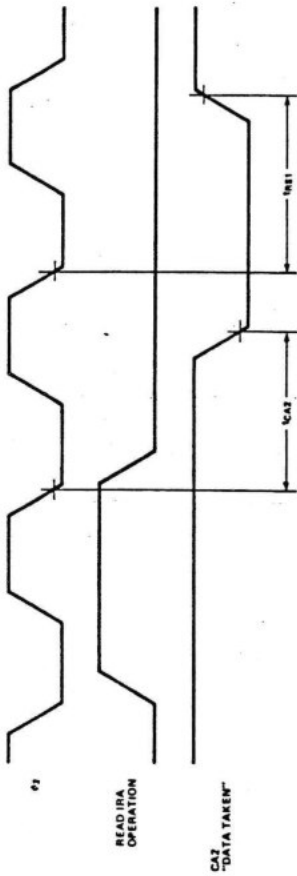


Figure 5a. CA2 Timing for Read Handshake, Pulse Mode

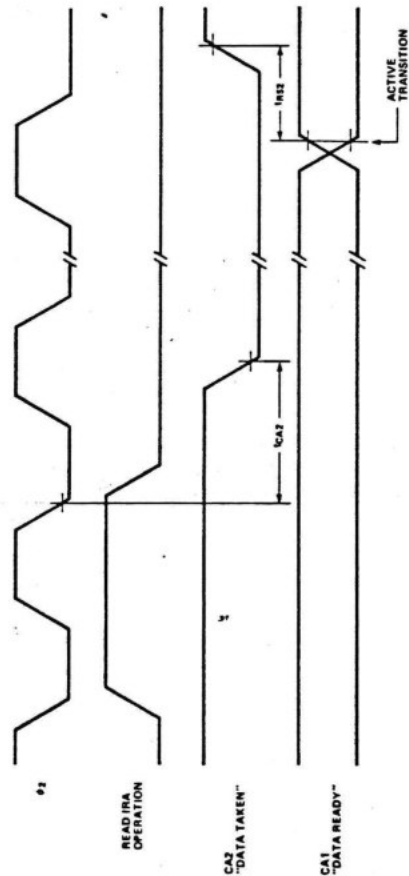


Figure 5b. CA2 Timing for Read Handshake, Handshake Mode

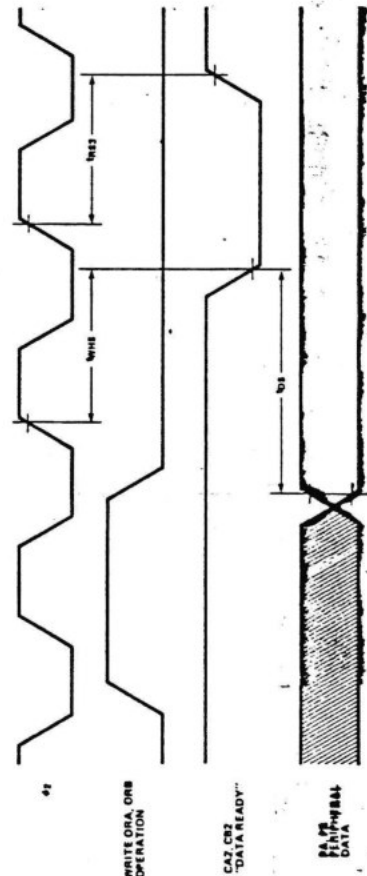


Figure 5c. CA2, CB2 Timing for Write Handshake, Pulse Mode

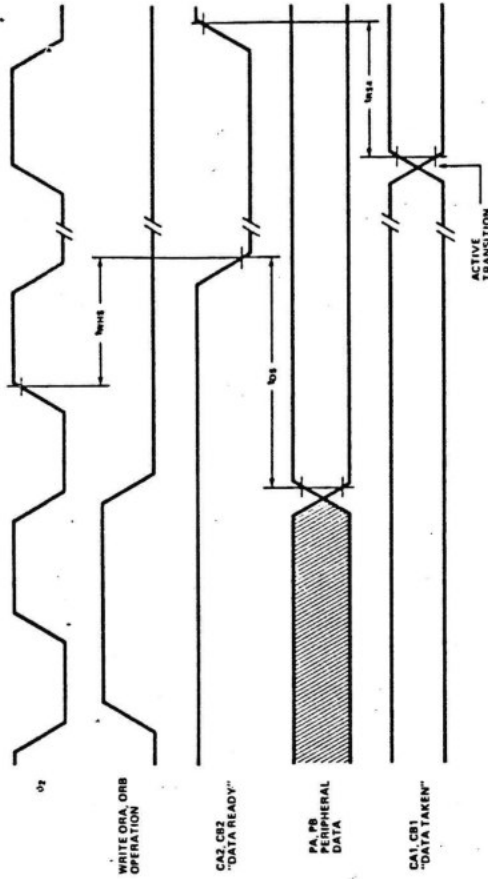


Figure 5d. CA2, CB2 Timing for Write Handshake, Handshake Mode

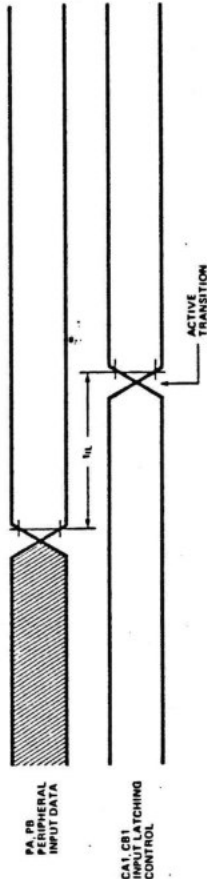


Figure 5e. Peripheral Data Input Latching Timing

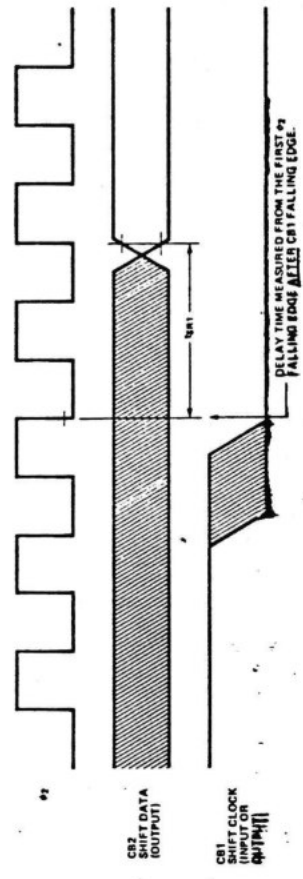


Figure 5f. Timing for Shift Out with Internal or External Shift Clcking



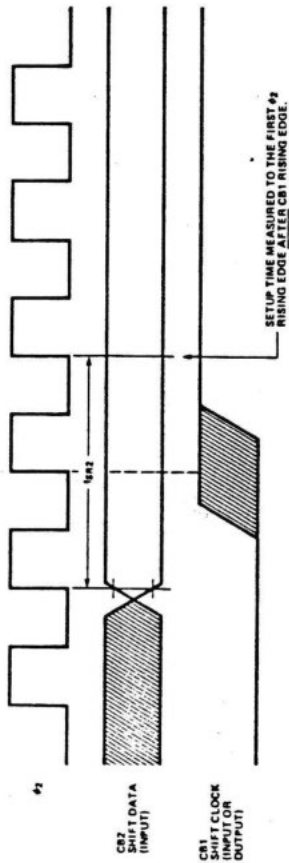


Figure 5g. Timing for Shift in with Internal or External Shift Clocking

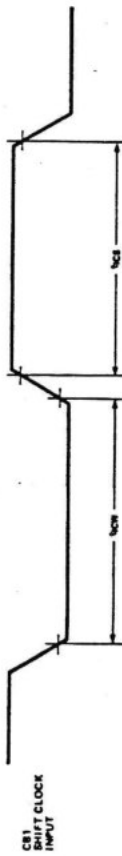


Figure 5h. External Shift Clock Timing



Figure 5i. Pulse Count Input Timing

PIN DESCRIPTIONS

RES (Reset)

The reset input clears all internal registers to logic 0 (except T1 and T2 latches and counters and the Shift Register). This places all peripheral interface lines in the input state, disables the timers, shift register, etc. and disables interrupting from the chip.

phi2 (Input Clock)

The input clock is the system phi2 clock and is used to trigger all data transfers between the system processor and the SY6522.

R/W (Read/Write)

The direction of the data transfers between the SY6522 and the system processor is controlled by the R/W line. If R/W is low, data will be transferred out of the processor into the selected SY6522 register (write operation). If R/W is high and the chip is selected, data will be transferred out of the SY6522 (read operation).

DB0-DB7 (Data Bus)

The eight bi-directional data bus lines are used to transfer data between the SY6522 and the system processor. During read cycles, the contents of the selected SY6522 register are placed on the data bus lines and transferred into the processor. During write cycles, these lines are high-impedance inputs and data is transferred from the processor into the selected register. When the SY6522 is unselected, the data bus lines are high-impedance.

CS1, CS2 (Chip Selects)

The two chip select inputs are normally connected to processor address lines either directly or through decoding. The selected SY6522 register will be accessed when CS1 is high and CS2 is low.

RS0-RS3 (Register Selects)

The four Register Select inputs permit the system processor to select one of the 16 internal registers of the SY6522, as shown in Figure 6.

Register Number	RS Coding				Register Desig.	Description	
	RS3	RS2	RS1	RS0		Write	Read
0	0	0	0	0	ORB/IRB	Output Register "B"	Input Register "B"
1	0	0	0	1	ORA/IRA	Output Register "A"	Input Register "A"
2	0	0	1	0	DRB	Data Direction Register "B"	
3	0	0	1	1	DDRA	Data Direction Register "A"	
4	0	1	0	0	T1C-L	T1 Low-Order Latches	T1 Low-Order Counter
5	0	1	0	1	T1C-H	T1 High-Order Counter	
6	0	1	1	0	T1L-L	T1 Low-Order Latches	
7	0	1	1	1	T1L-H	T1 High-Order Latches	
8	1	0	0	0	T2C-L	T2 Low-Order Latches	T2 Low-Order Counter
9	1	0	0	1	T2C-H	T2 High-Order Counter	
10	1	0	1	0	SR	Shift Register	
11	1	0	1	1	ACR	Auxiliary Control Register	
12	1	1	0	0	PCR	Peripheral Control Register	
13	1	1	0	1	IFR	Interrupt Flag Register	
14	1	1	1	0	IER	Interrupt Enable Register	
15	1	1	1	1	ORA/IRA	Same as Reg 1 Except No "Handshake"	

Figure 6. SY6522 Internal Register Summary

**IRQ (Interrupt Request)**

The Interrupt Request output goes low whenever an internal interrupt flag is set and the corresponding interrupt enable bit is a logic 1. This output is "open-drain" to allow the interrupt request signal to be "wire-or'ed" with other equivalent signals in the system.

**PA0-PA7 (Peripheral A Port)**

The Peripheral A port consists of 8 lines which can be individually programmed to act as inputs or outputs under control of a Data Direction Register. The polarity of output pins is controlled by an Output Register and input data may be latched into an internal register under control of the CA1 line. All of these modes of operation are controlled by the system processor through the internal control registers. These lines represent one standard TTL load in the input mode and will drive one standard TTL load in the output mode. Figure 7 illustrates the output circuit.

**CA1, CA2 (Peripheral A Control Lines)**

The two Peripheral A control lines act as interrupt inputs or as handshake outputs. Each line controls an internal interrupt flag with a corresponding interrupt enable bit. In addition, CA1 controls the latching of data on Peripheral A port input lines. CA1 is a high-impedance input only while CA2 represents one standard TTL load in the input mode. CA2 will drive one standard TTL load in the output mode.

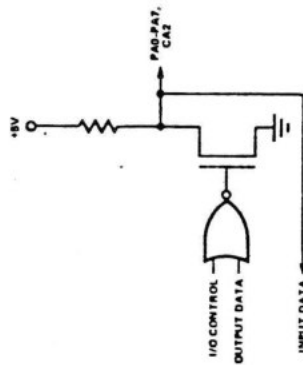


Figure 7. Peripheral A Port Output Circuit

**PB0-PB7 (Peripheral B Port)**

The Peripheral B port consists of eight bidirectional lines which are controlled by an output register and a data direction register in much the same manner as the

PA port. In addition, the polarity of the PB7 output signal can be controlled by one of the interval timers while the second timer can be programmed to count pulses on the PB6 pin. Peripheral B lines represent one standard TTL load in the input mode and will drive one standard TTL load in the output mode. In addition, they are capable of sourcing 1.0mA at 1.5VDC in the output mode to allow the outputs to directly drive Darlington transistor circuits. Figure 8 is the circuit schematic.

**CB1, CB2 (Peripheral B Control Lines)**

The Peripheral B control lines act as interrupt inputs or as handshake outputs. As with CA1 and CA2, each line controls an interrupt flag with a corresponding interrupt enable bit. In addition, these lines act as a serial port under control of the Shift Register. These lines represent one standard TTL load in the input mode and will drive one standard TTL load in the output mode. Unlike PB0-PB7, CB1 and CB2 cannot drive Darlington transistor circuits.

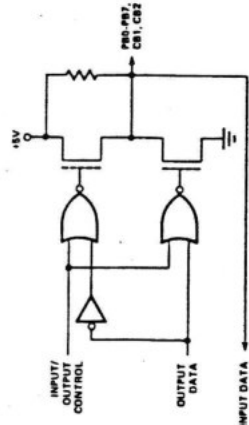


Figure 8. Peripheral B Port Output Circuit

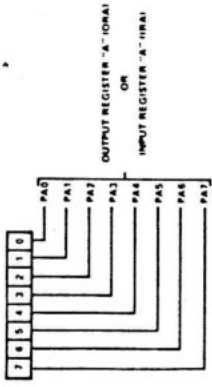
**FUNCTIONAL DESCRIPTION**

**Port A and Port B Operation**

Each 8-bit peripheral port has a Data Direction Register (DDRA, DDRB) for specifying whether the peripheral pins are to act as inputs or outputs. A 0 in a bit of the Data Direction Register causes the corresponding peripheral pin to act as an input. A 1 causes the pin to act as an output.

Each peripheral pin is also controlled by a bit in the Output Register (ORA, ORB) and an Input Register (IRA, IRB). When the pin is programmed as an output, the voltage on the pin is controlled by the cor-

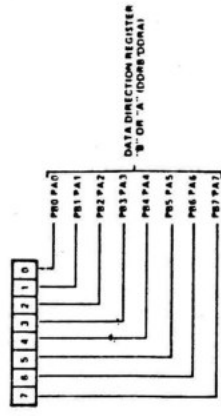
REG 1 - ORA/IRA



Data Direction	Pin	WRITE	READ
DDRA = 1 (OUTPUT)	PA0	MPU writes Output Level (Input latching disabled)	MPU reads level on PA pin
DDRA = 1 (OUTPUT)	PA1	MPU writes Output Level (Input latching disabled)	MPU reads IRB bit which is the level of the PA pin at the time of the last CA1 active transition
DDRA = 0 (INPUT)	PA2	MPU writes into ORB, but no effect on pin level, until DORA changed	MPU reads level on PA pin
DDRA = 0 (INPUT)	PA3	MPU writes into ORB, but no effect on pin level, until DORA changed	MPU reads IRB bit which is the level of the last CA1 active transition
DDRA = 0 (INPUT)	PA4	MPU writes into ORB, but no effect on pin level, until DORA changed	MPU reads level on PA pin
DDRA = 0 (INPUT)	PA5	MPU writes into ORB, but no effect on pin level, until DORA changed	MPU reads IRB bit which is the level of the last CA1 active transition
DDRA = 0 (INPUT)	PA6	MPU writes into ORB, but no effect on pin level, until DORA changed	MPU reads level on PA pin
DDRA = 0 (INPUT)	PA7	MPU writes into ORB, but no effect on pin level, until DORA changed	MPU reads IRB bit which is the level of the last CA1 active transition

Figure 10. Output Register A (ORA), Input Register A (IRA)

REG 2 (DDRB) AND REG 3 (DDRA)



'0' ASSOCIATED PA PIN IS AN INPUT, HIGH IMPEDANCE  
'1' ASSOCIATED PA PIN IS AN OUTPUT, WHOSE LEVEL IS DETERMINED BY ONE ORA REGISTER BIT

Figure 11. Data Direction Registers (DDRB, DDRA)

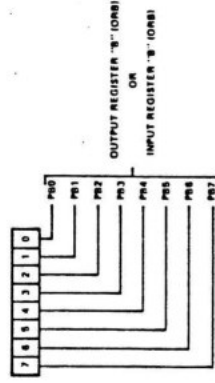
through the operation of "handshake" lines. Port A lines (CA1, CA2) handshake data on both a read and a write operation while the Port B lines (CB1, CB2) handshake on a write operation only.

**Read Handshake**

Positive control of data transfers from peripheral devices into the system processor can be accomplished very effectively using Read Handshaking. In this case, the peripheral device must generate the equivalent of a "Data Ready" signal to the processor signifying that valid data is present on the peripheral port. This signal normally interrupts the processor, which then reads the data, causing generation of a "Data Taken" signal. The peripheral device responds by making new data available. This process continues until the data transfer is complete.

Figure 9. Output Register B (ORB), Input Register B (IRB)

REG 0 - ORB/IRB



Data Direction	Pin	WRITE	READ
DDRB = 1 (OUTPUT)	PB0	MPU writes Output Level (Input latching disabled)	MPU reads Output Register Bit (ORB)
DDRB = 1 (OUTPUT)	PB1	MPU writes into ORB, but no effect on pin level, until DORA changed	MPU reads level on PB pin
DDRB = 0 (INPUT)	PB2	MPU writes into ORB, but no effect on pin level, until DORA changed	MPU reads IRB bit, which is the level of the PB pin at the time of the last CB1 active transition
DDRB = 0 (INPUT)	PB3	MPU writes into ORB, but no effect on pin level, until DORA changed	MPU reads level on PB pin
DDRB = 0 (INPUT)	PB4	MPU writes into ORB, but no effect on pin level, until DORA changed	MPU reads IRB bit, which is the level of the last CB1 active transition
DDRB = 0 (INPUT)	PB5	MPU writes into ORB, but no effect on pin level, until DORA changed	MPU reads level on PB pin
DDRB = 0 (INPUT)	PB6	MPU writes into ORB, but no effect on pin level, until DORA changed	MPU reads IRB bit, which is the level of the last CB1 active transition
DDRB = 0 (INPUT)	PB7	MPU writes into ORB, but no effect on pin level, until DORA changed	MPU reads level on PB pin



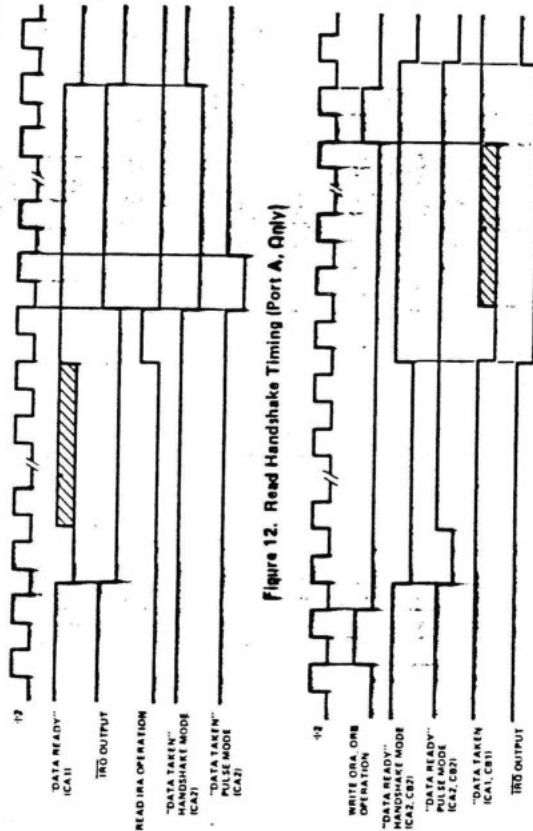


Figure 12. Read Handshake Timing (Port A, Only)

Figure 13. Write Handshake Timing

In the SY6522, automatic "Read" Handshaking is possible on the Peripheral A port only. The CA1 interrupt input pin accepts the "Data Ready" signal and CA2 generates the "Data Taken" signal. The "Data Ready" signal will set an internal flag which may interrupt the processor or which may be polled under program control. The "Data Taken" signal can either be a pulse or a level which is set low by the system processor and is cleared by the "Data Ready" signal. These options are shown in Figure 12 which illustrates the normal Read Handshaking sequence.

Write Handshake

The sequence of operations which allows handshaking data from the system processor to a peripheral device is very similar to that described for Read Handshaking. However, for Write Handshaking, the SY6522 generates the "Data Ready" signal and the peripheral device must respond with the "Data Taken" signal. This can be accomplished on both the PA port and the PB port on the SY6522. CA2 or CB2 act as a "Data Ready" output in either the handshake mode or pulse mode and CA1 or CB1 accept the "Data Taken" signal from the peripheral device, setting the interrupt flag and cleaning the "Data Ready" output. This sequence is shown in Figure 13.

Selection of operating modes for CA1, CA2, CB1, and CB2 is accomplished by the Peripheral Control Register (Figure 14).

**Timer Operation**  
Interval Timer T1 consists of two 8-bit latches and a 16-bit counter. The latches are used to store data which is to be loaded into the counter. After loading, the counter decrements at  $\phi 2$  clock rate. Upon reaching zero, an interrupt flag will be set, and TRQ will go low if the interrupt is enabled. The timer will then disable any further interrupts, or will automatically transfer the contents of the latches into the counter and will continue to decrement. In addition, the timer may be programmed to invert the output signal on a peripheral pin each time it "times-out". Each of these modes is discussed separately below.

The T1 counter is depicted in Figure 15 and the latches in Figure 16.

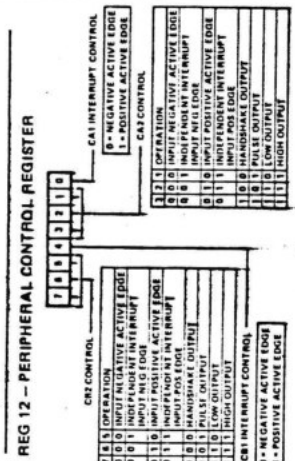


Figure 14. CA1, CA2, CB1, CB2 Control

Two bits are provided in the Auxiliary Control Register (bits 6 and 7) to allow selection of the T1 operating modes. The four possible modes are depicted in Figure 17.

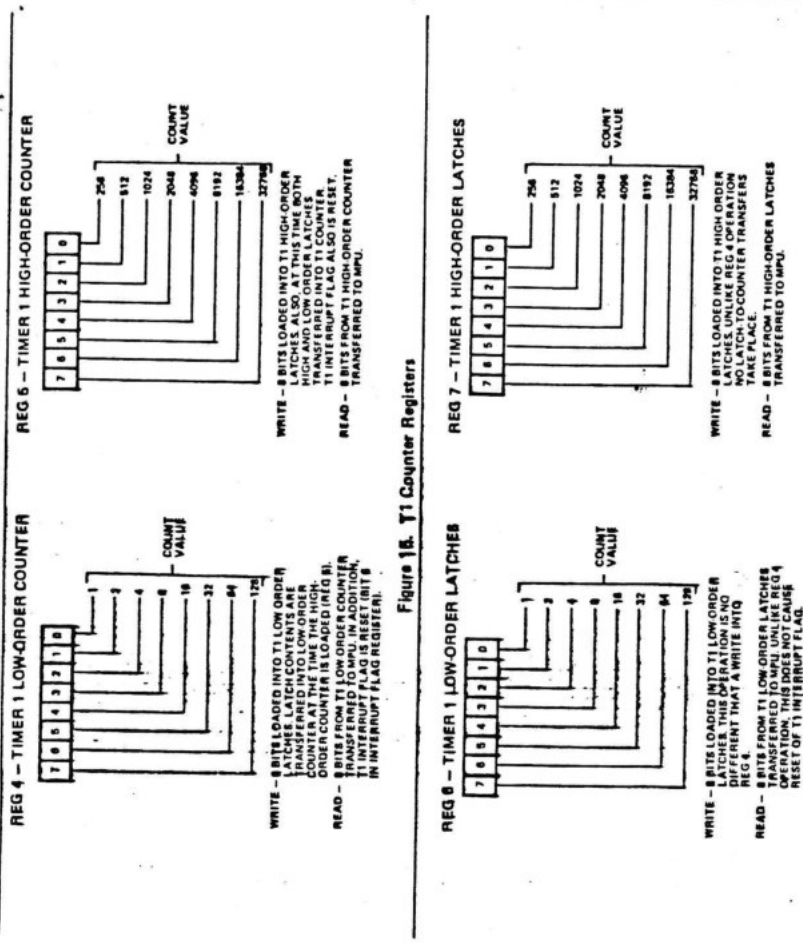


Figure 16. T1 Counter Registers

Figure 16. T1 Latch Registers

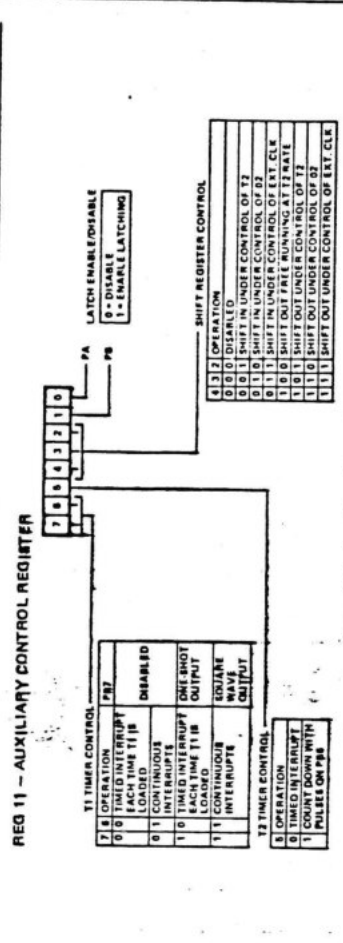


Figure 17. Auxiliary Control Register

Note: The processor does not write directly into the low order counter (T1C-L). Instead, this half of the counter is loaded automatically from the low order latch when the processor writes into the high order counter. In fact, it may not be necessary to write to the low order counter in some applications since the timing operation is triggered by writing to the high order counter.

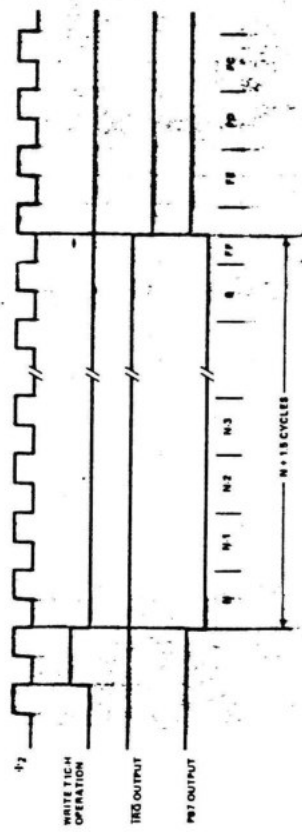


Figure 18. Timer 1 and Timer 2 One-Shot Mode Timing

**Timer 1 One-Shot Mode**

The interval timer one-shot mode allows generation of a single interrupt for each timer load operation. As with any interval timer, the delay between the "write T1C-H" operation and generation of the processor interrupt is a direct function of the data loaded into the timing counter. In addition to generating a single interrupt, Timer 1 can be programmed to produce a single negative pulse on the PB7 peripheral pin. With the output enabled (ACR7=1) a "write T1C-H" operation will cause PB7 to go low. PB7 will return high when Timer 1 times out. The result is a single programmable width pulse.

In the one-shot mode, writing into the high order latch has no effect on the operation of Timer 1. However, it will be necessary to assure that the low order latch contains the proper data before initiating the count-down with a "write T1C-H" operation. When the processor writes into the high order counter, the T1 interrupt flag will be cleared, the contents of the low order latch will be transferred into the low order counter, and the timer will begin to decrement at system clock rate. If the PB7 output is enabled, this signal will go low on the phase two following the write operation. When the counter reaches zero, the T1 interrupt flag will be set, the  $\overline{TRC}$  pin will go low (interrupt enabled), and the signal on PB7 will go high. At this time the counter will continue to decrement at system clock rate. This allows the system processor to read the contents of the counter to determine the time since interrupt. However, the T1 interrupt flag cannot be set again unless it has been cleared as described in this specification.

Timing for the SY6522 interval timer one-shot modes is shown in Figure 18.

**Timer 1 Free-Run Mode**

The most important advantage associated with the latches in T1 is the ability to produce a continuous

series of evenly spaced interrupts and the ability to produce a square wave on PB7 whose frequency is not affected by variations in the processor interrupt response time. This is accomplished in the "free-running" mode.

In the free-running mode, the interrupt flag is set and the signal on PB7 is inverted each time the counter reaches zero. However, instead of continuing to decrement from zero after a time-out, the timer automatically transfers the contents of the latch into the counter (18 bits) and continues to decrement from there. The interrupt flag can be cleared by writing T1C-H, by reading T1C-L, or by writing directly into the flag as described later. However, it is not necessary to rewrite the timer to enable setting the interrupt flag on the next time-out.

All interval timers in the SY6522 are "re-triggerable". Rewriting the counter will always re-initialize the time-out period. In fact, the time-out can be prevented completely if the processor continues to rewrite the timer before it reaches zero. Timer 1 will operate in this manner if the processor writes into the high order counter (T1C-H). However, by loading the latches only, the processor can access the timer during each down-counting operation without affecting the time-out in process. Instead, the data loaded into the latches will determine the length of the next time-out period. This capability is particularly valuable in the free-running mode with the output enabled. In this mode, the signal on PB7 is inverted and the interrupt flag is set with each time-out. By responding to the interrupt with new data for the latches, the processor can determine the period of the next half cycle during each half cycle of the output signal on PB7. In this manner, very complex waveforms can be generated. Timing for the free-running mode is shown in Figure 19.

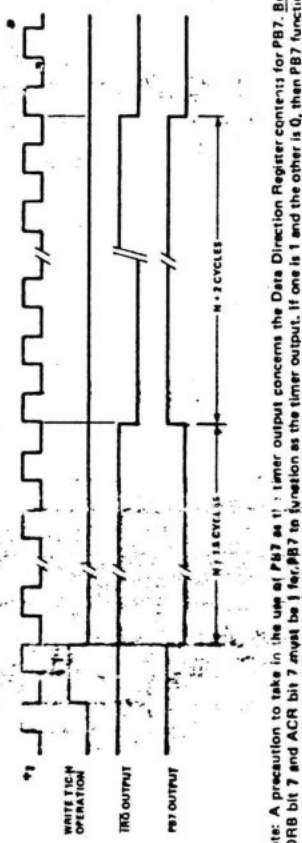


Figure 19. Timer 1 Free-Run Mode Timing

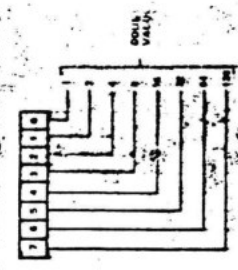
**Timer 2 Operation**

Timer 2 operates as an interval timer (in the "one-shot" mode only), or as a counter for counting negative pulses on the PB6 peripheral pin. A single control bit is provided in the Auxiliary Control Register to select between these two modes. This timer is comprised of a "write-only" low-order latch (T2L-L), a "read-only" low-order counter and a read/write high order counter. The counter registers act as a 18-bit counter which decrements at  $\phi_2$  rate. Figure 20 illustrates the T2 Counter Registers.

**Timer 2 One-Shot Mode**

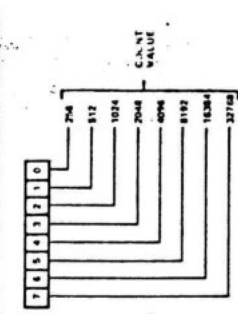
As an interval timer, T2 operates in the "one-shot" mode similar to Timer 1. In this mode, T2 provides a single interrupt for each "write T2C-H" operation. After timing out, the counter will continue to decrement. However, setting of the interrupt flag will be disabled after initial time-out so that it will not be set by the counter continuing to decrement through zero. The processor must rewrite T2C-H to enable setting of the interrupt flag. The interrupt flag is cleared by reading T2C-L or by writing T2C-H. Timing for this operation is shown in Figure 18.

REG 8 - TIMER 2 LOW-ORDER COUNTER



WRITE - 8 BITS LOADED INTO T2 LOW-ORDER LATCHES FROM T1 LOW-ORDER COUNTER. FLAG IS RESET.  
READ - 8 BITS FROM T2 HIGH-ORDER COUNTER TRANSFERRED TO MPU.

REG 9 - TIMER 2 HIGH-ORDER COUNTER



WRITE - 8 BITS LOADED INTO T2 HIGH-ORDER COUNTER ALSO LOW-ORDER LATCHES FROM T1 LOW-ORDER COUNTER. FLAG IS RESET.  
READ - 8 BITS FROM T2 HIGH-ORDER COUNTER TRANSFERRED TO MPU.

Figure 20. T2 Counter Registers

Timer 2 Pulse Counting Mode

In the pulse counting mode, T2 serves primarily to count a predetermined number of negative-going pulses on PB6. This is accomplished by first loading a number into T2. Writing into T2CH clears the interrupt flag and allows the counter to decrement each time a pulse is applied to PB6. The interrupt flag will be set when T2 reaches zero. At this time the counter will continue to decrement with each pulse on PB6. However, it is necessary to rewrite T2CH to allow the interrupt flag to set on subsequent down-counting operations. Timing for this mode is shown in Figure 21. The pulse must be low on the leading edge of  $\phi_2$ .

Shift Register Operation

The Shift Register (SR) performs serial data transfers into and out of the CB1 pin under control of an internal module-8 counter. Shift pulses can be applied to the CB1 pin from an external source or, with the proper mode selection, shift pulses generated internally will appear on the CB1 pin for controlling external devices.

The control bits which select the various shift register operating modes are located in the Auxiliary Control Register. Figure 22 illustrates the configuration of the SR data bits and the SR control bits of the ACR. Figures 23 and 24 illustrate the operation of the various shift register modes.

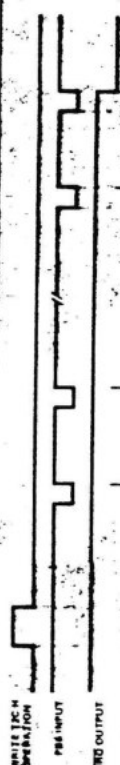
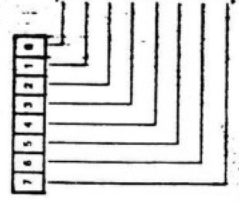


Figure 21. Timer 2 Pulse Counting Mode

REG 10 - SHIFT REGISTER



NOTE: 1. WRITING ONE BIT 7 AS THE FIRST BIT INTO ACR SIGNAL ALLOWONLY IS ROTATED BACK INTO BIT 0. 2. WRITING OTHER BITS INITIALLY ENTERED INTO ACR ARE SHIFTED TOWARDS BIT 7.

REG 11 - AUXILIARY CONTROL REGISTER

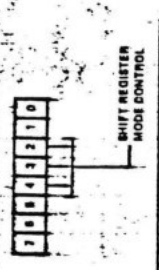


Table with 8 rows and 2 columns: Bit number and Operation. Bit 7: 0 = DISABLED, 1 = SHIFT UNDER CONTROL OF T2. Bit 6: 0 = 0, 1 = SHIFT UNDER CONTROL OF T2. Bit 5: 0 = 0, 1 = SHIFT UNDER CONTROL OF EXT CLK. Bit 4: 0 = 0, 1 = SHIFT OUT FALL RISING ALTERNATE. Bit 3: 0 = 0, 1 = SHIFT OUT UNDER CONTROL OF T2. Bit 2: 0 = 0, 1 = SHIFT OUT UNDER CONTROL OF T2. Bit 1: 0 = 0, 1 = SHIFT OUT UNDER CONTROL OF EXT CLK. Bit 0: 0 = 0, 1 = SHIFT OUT UNDER CONTROL OF EXT CLK.

Figure 22. SR and ACR Control Bits

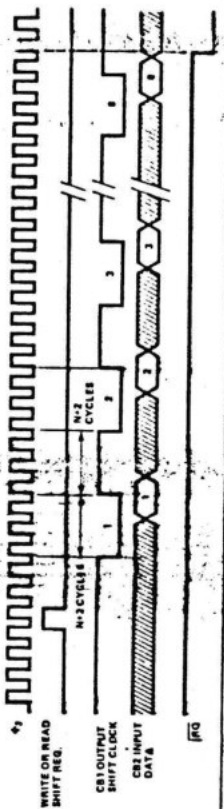
SR Disabled (000)

The 000 mode is used to disable the Shift Register. In this mode the microprocessor can write or read the SR, but the shifting operation is disabled and operation of CB1 and CB2 is controlled by the appropriate bits in the Peripheral Control Register (PCR). In this mode the SR Interrupt Flag is disabled (held to a logic 0).

Shift in Under Control of T2 (001)

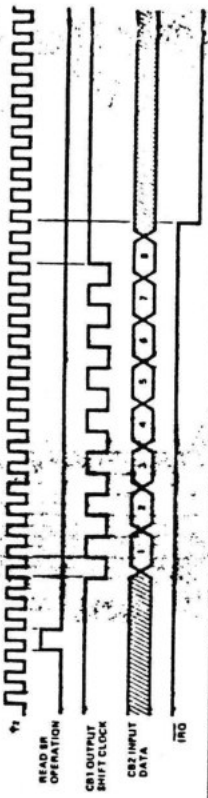
In the 001 mode the shifting rate is controlled by the low order 8 bits of T2. Shift pulses are generated on the CB1 pin to control shifting in external devices. The time between transitions of this output clock is a function of the system clock period and the contents of the low order T2 latch (N).

The shifting operation is triggered by writing or reading the shift register. Data is shifted first into the low order bit of SR and is then shifted into the next higher order bit of the shift register on the negative-going edge of each clock pulse. The input data should change before the positive-going edge of the CB1 clock pulse. This data is shifted into the shift register during the  $\phi_2$  clock cycle following the positive-going edge of the CB1 clock pulse. After 8 CB1 clock pulses, the shift register interrupt flag will be set and IRQ will go low.



Shift in Under Control of  $\phi_2$  (010)

In mode 010 the shift rate is a direct function of the system clock frequency. CB1 becomes an output which generates shift pulses for controlling external devices. Timer 2 operates as an independent interval timer which has no effect on SR. The shifting operation is triggered by reading or writing the Shift Register. Data is shifted into bit 0 and is then shifted into the next higher order bit of the shift register on the trailing edge of each  $\phi_2$  clock pulse. After 8 clock pulses, the shift register interrupt flag will be set, and the output clock pulses on CB1 will stop.



Shift in Under Control of External CB1 Clock (011)

In mode 011 CB1 becomes an input. This allows an external device to load the shift register at its own pace. The shift register counter will interrupt the processor each time 8 bits have been shifted in. However, the shift register counter does not stop the shifting operation; it acts simply as a pulse counter. Reading or writing the Shift Register resets the Interrupt flag and initializes the SR counter to count another 8 pulses.

Note that the data is shifted during the first system clock cycle following the positive-going edge of the CB1 shift pulse. For this reason, data must be held stable during the first full cycle following CB1 going high.

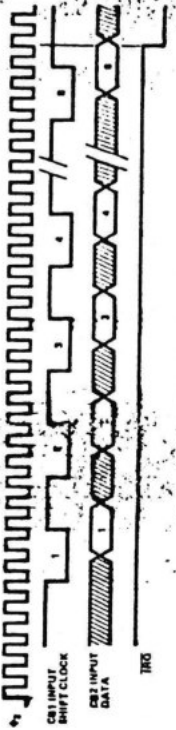
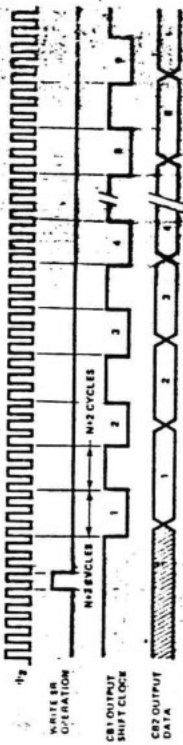


Figure 23. Shift Register Input Modes



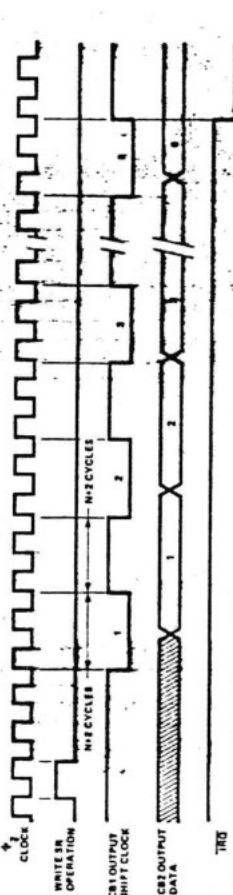
Shift Out Free-Running at T2 Rate (100)

Mode 100 is very similar to mode 101 in which the shifting rate is set by T2. However, in mode 100, the SR Counter does not stop the shifting operation. Since the Shift Register bit 7 (SR7) is recirculated back into bit 0, the 8 bits loaded into the shift register will be clocked onto CB2 repeatedly. In this mode the shift register counter is disabled.



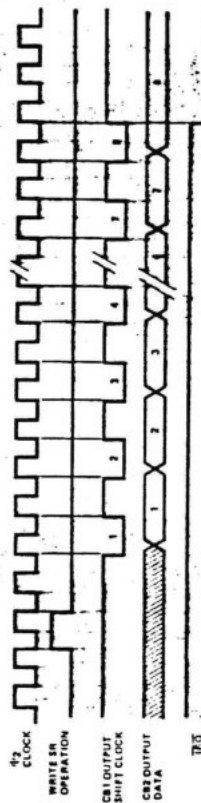
Shift Out Under Control of T2 (101)

In mode 101, the shift rate is controlled by T2 (as in the previous mode). However, with each read or write of the shift register the SR Counter is reset and 8 bits are shifted onto CB2. At the same time, 8 shift pulses are generated on CB1 to control shifting in external devices. After the 8 shift pulses, the shifting is disabled, the SR Interrupt Flag is set and CB2 remains at the last data level.



Shift Out Under Control of φ2 (110)

In mode 110, the shift rate is controlled by the φ2 system clock.



Shift Out Under Control of External CB1 Clock (111)

In mode 111 shifting is controlled by pulses applied to the CB1 pin by an external device. The SR counter sets the SR Interrupt flag each time it counts 8 pulses but it does not disable the shifting function. Each time the microprocessor writes or reads the shift register, the SR interrupt flag is reset and the SR counter is initialized to begin counting the next 8 shift pulses on pin CB1. After 8 shift pulses, the interrupt flag is set. The microprocessor can then load the shift register with the next byte of data.

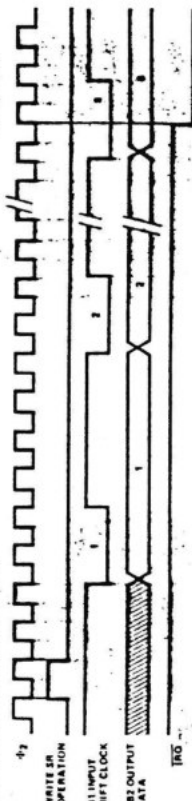


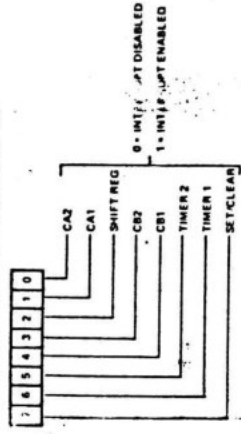
Figure 24. Shift Register Output Modes

address 1110 (IER address). If bit 7 of the data placed on the system data bus during this write operation is a 0, each 1 in bits 6 through 0 clears the corresponding bit in the Interrupt Enable Register. If each zero in bits 6 through 0, the corresponding bit is unaffected.

Setting selected bits in the Interrupt Enable Register is accomplished by writing to the same address with bit 7 in the data word set to a logic 1. In this case, each 1 in bits 6 through 0 will set the corresponding bit. For each zero, the corresponding bit will be unaffected. This individual control of the setting and clearing operations allows very convenient control of the interrupts during system operation.

In addition to setting and clearing IER bits, the processor can read the contents of this register. To do this, the proper address on the register select and chip select inputs with the R/W line high. Bit 7 will be read as a logic 0.

REG 14 - INTERRUPT ENABLE REGISTER



NOTES:  
1. IF BIT 7 IS A "0", THEN EACH "1" IN BITS 6 - 0 DISABLES THE CORRESPONDING INTERRUPT.  
2. IF BIT 7 IS A "1", THEN EACH "1" IN BITS 6 - 0 ENABLES THE CORRESPONDING INTERRUPT.  
3. IF A READ OF THIS REGISTER IS DONE, BIT 7 WILL BE "0" AND ALL OTHER BITS WILL REFLECT THEIR ENABLE/DISABLE STATE.

Figure 26. Interrupt Enable Register (IER)

The Interrupt Flag Register (IFR) and the Interrupt Enable Register (IER) are depicted in Figures 25 and 26, respectively.

The IFR may be read directly by the processor. In addition, individual flag bits may be cleared by writing a "1" into the appropriate bit of the IFR. When the proper chip select and register signals are applied to the chip, the contents of this register are placed on the data bus. Bit 7 indicates the status of the IRQ output. This bit corresponds to the logic function:  $IRQ = IFR6 \times IER6 + IFR5 \times IER5 + IFR4 \times IER4 + IFR3 \times IER3 + IFR2 \times IER2 + IFR1 \times IER1 + IFR0 \times IER0$ . Note: X = logic AND, + = Logic OR.

The IFR bit 7 is not a flag. Therefore, this bit is not directly cleared by writing a logic 1 into it. It can only be cleared by clearing all the flags in the register or by disabling all the active interrupts as discussed in the next section.

REG 13 - INTERRUPT FLAG REGISTER

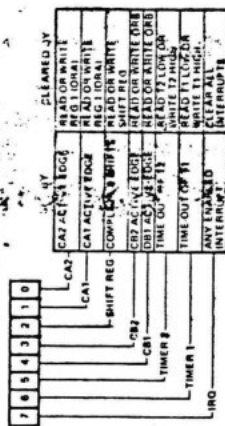


Figure 25. Interrupt Flag Register (IFR)

For each interrupt flag in IFR, there is a corresponding bit in the Interrupt Enable Register. The system processor can set or clear selected bits in this register to facilitate controlling individual interrupts without affecting others. This is accomplished by writing to

APPLE INTERFACE

AP 4-GRAFIK-DRUCKER-INTERFACE

fuer : ITOH 8510

MEC 8023B / C

INSTALLATIONS - UND BENUTZUNGSHINWEISE

BITTE ERST SORGFÄLTIG DURCHLESEN

DANN DIE KARTE BENUTZEN !

Betriebsanleitung fuer das PARALLEL-GRAFIK-DRUCKERINTERFACE

AP 4 - G - ITOH / NEC.

1. allgemeine Beschreibung
2. Einbauhinweise
3. Steuerbefehle
  - 3.1 Steuerbefehle im Textmodus
  - 3.2 Steuerbefehle im Grafikmodus
  - 3.3 Beispiele zum Grafikmodus
4. Voreinstellung der Druckparameter
  - 4.1 Kompatibilität zu PASCAL und CP/M
  - 4.2 Punktgrafikdruck im BASIC-Modus
  - 4.3 Maschinenroutinen
- 5.1 mit USR (X)-Funktion
- 5.1.2 mit CALL-Funktion
6. Elektronische Vertikale Formatsteuerung
7. Anpassung Visiplot an AP 4 G

APPLE ist eingetragenes Warenzeichen der APPLE COMPUTER INC.  
Cupertino, California

# IBM COMPUTERTECHNIK

Computersysteme · Hard- und Softwareentwicklung

## 1. Beschreibung der parallel Drucker Interface Karte AP4 - 6

Dieses Handbuch beschreibt die Anwendungsmöglichkeiten des parallel Drucker-Interfaces fuer den Apple II Microcomputer. Empfehlenswert zum Gebrauch des Apple II Systems mit diesem Drucker - Interface ist die Durchsicht der entsprechenden Kapitel des Drucker-Handbueches und der erforderlichen Kapitel der Apple II Handbuecher. Durch unsachgemaeße Handhabung koennen sowohl der Apple II Microcomputer als auch das parallel Interface beschaedigt oder zerstoert werden.

Das parallel Drucker-Interface erlaubt den Anschluss von Druckern mit parallel Interface an den Apple II Computer. Es liefert die noetigen Steuerzeichen fuer folgende Drucker :

ITOH 8510  
ITOH 8510A  
NEC 8023B/C

Durch die in die Druckerkarte eingebaute Software bietet sie eine komfortable Textausgabe sowie Punktgrafikdruck beider hochaufloesender Bildschirmseiten (HGR 1 und HGR 2).

Dieses Interface ist ohne zusaetzliche Software kompatibel mit den Betriebssystemen :

BASIC  
DOS 3.3  
PASCAL  
FORTRAN  
CP/M  
MONITOR

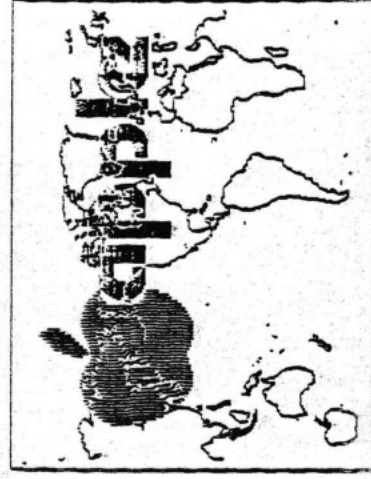
Der Betrieb dieses parallel Drucker-Interfaces ist auch ohne Disketten-Betriebssystem Software moeglich.

# IBM COMPUTERTECHNIK

Computersysteme · Hard- und Softwareentwicklung

## 2. Einbau des Parallel-Drucker-Interfaces

1. Schalten Sie die Stromzufuhr des APPLE ab.
2. Innerhalb des Apple koennen Sie jetzt an der Rueckseite der Hauptplatine 8 offene Steckverbindungen ("Slots") sehen, die von links nach rechts mit 0 bis 7 durchnummeriert sind.
3. Wenn Sie sicher sind, dass der Strom des Apple abgeschaltet ist, setzen Sie die Parallel-Karte in den 2. Interface-Steckplatz von links (Slot 1) ein. Die Druckerkarte funktioniert auch in anderen Steckplaetzen, ist dann aber nicht mehr kompatibel mit CP/M und PASCAL sowie mit den Programmen, die die Parallel-Karte in Slot 1 erwarten.  
Achten Sie darauf, dass die Karte fest in dem Slot sitzt. Die Seite der Drucker Interface-karte, die NICHT mit Bauteilen bestueckt ist, muss zum Netzteilblock des Apple Computers zeigen.
4. Fuehren Sie das Kabel durch eine der vertikalen Oeffnungen an der Rueckseite des Apple-Gehaeuses nach aussen.
5. Befestigen Sie nun wieder den Deckel Ihres APPLE's.
6. Verbinden Sie das Kabel mit Ihrem Drucker.
7. Das Interface ist nun betriebsbereit !





## 3. Steuerbefehle

Steuerbefehle zur AP4 - G muessen mit einem festgelegten Steuerwort beginnen. Dieses Steuerwort wird beim Initialisieren des parallel Drucker-Interfaces auf CTRL-I (bei CP/M und PASCAL CTRL-Y) eingestellt. Das Steuerwort kann per Befehl gesendert werden, jedoch ist hierbei auf einen eventuellen Konflikt mit anderen Befehlen (z.B. CTRL-M = Wagenruecklauf CR) zu achten. Beachten Sie in dem Zusammenhang auch die Anmerkungen zu diesem Druckkommando der Befehlstabelle.

Bei den Anweisungen, die eine Zifferneingabe erlauben, ist die Voreinstellung 00. Das heisst, wenn keine Zifferneingabe erfolgt, wird der Wert 00 angenommen. Die Zifferneingabe darf nur zweifellig erfolgen. Wenn mehr als zwei Ziffern eingegeben werden kann es zu Fehlfunktionen kommen.

Beachten Sie, dass CTRL-Zeichen nicht auf dem Bildschirm erscheinen. Um die Sonderfunktionen des Druckers durch Programmkontrolle zu steuern, benutzen Sie die CHR\$(-)-Funktion von BASIC, bzw. die CHR-Funktion von Pascal.

### 3.1 Steuerbefehle im Textmodus

**ANMERKUNG:** In den folgenden Beispielen wird davon ausgegangen, dass sich das Drucker Interface in Steckplatz 1 des Apple befindet.

Leerzeichen sind nur aus Gruenden der besseren Lesbarkeit benutzt worden. Bei Eingabe in den Computer duerfen keine Leerzeichen eingegeben werden. Das Zeichen 'n' wird bei der praktischen Anwendung durch eine oder zwei Ziffern ersetzt.

**PR#n** Schaltet das Interface ein. 'n' steht fuer die Nummer des Steckplatzes, in dem das Interface steckt. Alle Zeichen, die gedruckt werden, erscheinen auch auf dem Bildschirm des Apple.

Beispiel:  
Mit DOS 10 PRINT CHR\$(4);"PR#1"  
ohne DOS 10 PR#1

**PR#0** Schaltet das Interface aus. Ebenso wird das Interface durch RESET abgeschaltet.

Beispiel:  
Mit DOS 10 PRINT CHR\$(4);"PR#0"  
ohne DOS 10 PR#0

**CTRL-I A** Es wird nach jedem CR (Wagenruecklauf) ein LF (Zeilen-vorschub) gesendet.  
DOS 3.3 gibt kein LF nach einem CR ab. Deshalb ergaenzt das Interface nach jedem CR im Text noch ein LF. Wenn Sie doppelten Zeilenvorschub an Ihrem Drucker bemerken, pruefen Sie die Stellung der DIP-Schalter in ihrem

Beispiel: 10 PRINT CHR\$(9);"A"

**CTRL-I K** Loescht den Befehl CTRL-I A. Ein automatischer Zeilen-vorschub wird nicht mehr ausgefuehrt.

Beispiel: 10 PRINT CHR\$(9);"K"

**CTRL-INN** Setzt die Zeilenlaenge auf 'n'Z eichen vom Linksanschlag des Druckkopfes (nicht vom linken Papierrand) ausgeben. Nach diesem Kommando erfolgt kein Echo der Ausgabe auf dem Bildschirm des APPLE, sondern die Zeichen werden nur noch auf dem Drucker ausgegeben. Ein Wagenruecklauf wird automatisch ergaenzt, sobald die eingestellte Zeichenzahl erreicht ist. Bei n = 00 (Linksanschlag = 0) wird diese Funktion abgeschaltet. Die Angabe des CTRL-INN-Kommandos kann z.B. dazu verwendet werden, um BASIC-Listing nicht im 40-Zeichenformat zu Papier zu bekommen.

Beispiel: 10 PRINT CHR\$(9);"80N"  
Es werden 80 Zeichen pro Zeile ausgegeben.

**CTRL-I I** Alle Zeichen werden sowohl auf dem 40-Zeichen Bildschirm des Apple II, als auch auf dem Drucker ausgegeben.

Beispiel: 10 PRINT CHR\$(9);"I"

**CTRL-INL** Setzt den linken Leerrand auf 'n' Zeichen. Der Drucker beginnt den Ausdruck bei dieser Position.

Beispiel: 10 PRINT CHR\$(9);"6L"  
Jede Druckspalte beginnt auf der 6.Spalte.

**CTRL-INR** Setzt den rechten Anschlag auf 'n' Zeichen. Hierbei ist zu beachten, dass dieser "Anschlag" nur die Position angibt, ab der die Druckerkarte auf ein Leerzeichen (SPACE) achtet. Wird ein Leerzeichen entdeckt, so erfolgt ein Zeilenvorschub. Dadurch wird vermieden, dass Worte zerschnitten werden. Wird der rechte Anschlag auf 0 gesetzt, ist diese Funktion abgespaltet.

Beispiel: 10 PRINT CHR\$(9);"80R"  
Setzt rechten Rand auf die Spalte 80.

**CTRL-INP** Dieses Kommando setzt die Seitenlaenge auf 'n' Druckzeilen. Wird das vorgewahlte Seitenende erreicht, so werden automatisch 6 Zeilenvorschuebe ausgefuehrt, um die Papierperforation zu ueberspringen. Wird die Seitenlaenge auf 00 gesetzt, so ist diese Funktion abgespaltet.



Beispiel: 10 PRINT CHR\$(9);"66P"  
Setzt die Seitenlänge auf 66 Zeilen pro Seite.

**CTRL-I H** Schaltet die Druckerausgabe auf 8 Bit Modus um. Damit koennen Grafikzeichen ausgedruckt werden. Dieses Kommando ist nicht bei Benutzung der eingebauten HGR-Grafik Kommandos erforderlich.

Beispiel: 10 PRINT CHR\$(9);"H"

**CTRL-I CTRL-Y** Schaltet CTRL-Y als neues Steuerwort ein. Die Umschaltung auf CTRL-I muss dann durch die Sequenz CTRL-Y CTRL-I erfolgen. Die Umschaltung des Drucker Steuerwortes sollte in einem Fall unbedingt geaendert werden, und zwar immer dann, wenn im BASIC-Modus Druckdaten als Punktgrafik uebermittelt werden sollen, in denen die Zahl 9 (ASCII Wert fuer CTRL-I) vorkommt. In einem solchen Fall wird der Drucker, unvorhersehbare Muster erzeugen.

Beispiel: 10 PRINT CHR\$(9);CHR\$(25)  
Schaltet das Steuerzeichen von CTRL-I auf CTRL-Y um.

## 3-2 Steuerbefehle im GRAFIK-Modus

Mit der AP4-6 koennen Sie den Grafikbildschirm Ihres APPLE II Computers auf dem Drucker ausgeben. Alle hierzu notwendigen Grafikbefehle bestehen aus dem Steuerwort (normalerweise CTRL-I), einem "G" und dem im folgenden beschriebenen Moeglichkeiten. Wenn keine anderen Optionen benutzt werden, dass heisst, wenn nur CTRL-I G eingegeben wird, dann wird die HGR-Seite 1 horizontal ausgedruckt. Hierbei werden weisse Punkte auf dem Bildschirm schwarz ausgegeben.

Liste aller moeglichen Grafikbefehlen:

- 2 - Ausdruck der zweiten Seite
- D - Ausgeben des Bildes mit doppelter Grosse.
- E - Unidirektionaler Druck von links nach rechts.
- I - Invertieren der auszugebenden Grafikseite.
- R - Drehung des auszugebenden Bildes um 90 Grad.
- L - Ausdruck beginnt ab einer vorgegebenen Position.
- T - Erzeugt in der ausgewaehlten Bildschirmseite ein Testbild.
- A - AND Verknuepfung beider HIRES Seiten.
- X - EXKLUSIV OR Verknuepfung beider HIRES Seiten.
- O - OR Verknuepfung beider HIRES Seiten.

Eine detaillierte Beschreibung der Grafik-Befehle finden Sie nach diesem Abschnitt. Die Benutzung der Graphik-Kommandos ist ebenso einfach, wie die Benutzung der Text-Kommandos. Sie werden eingeleitet mit einem CTRL-I, gefolgt von einem 'G'. Danach koennen alle Grafik-Steuerzeichen in beliebiger Reihenfolge zum Interface gesendet werden. Erst wenn "RETURN" gesendet wird, erfolgt der Start des Grafik-Drucks.

Ein Beispiel fuer Grafik-Kommando ist PRINT CHR\$(9) "GDIR2" gefolgt von "RETURN". Dieses Kommando bewirkt, dass das Bild der zweiten Grafikseite invertiert, mit doppelter Grosse und um 90 Grad gedreht ausgegeben wird. Hierbei spielt die Reihenfolge der Buchstaben und Ziffern keinerlei Rolle solange CTRL-I voransteht.

**Wichtig:** Bitte achten Sie darauf, dass eine zu grosseuegige Bemessung des freien linken Randes (eingestellt mit dem L-Befehl) dazu fuehren kann, dass der Drucker nicht alles ausdrucken kann und die Arbeit verweigert. Nach einem solchen Vorfall ist RESET fuer Computer und Drucker erforderlich.

**Bemerkung:** Wenn Sie diese Kommandos ueber die Tastatur eingeben wollen, wird der Apple die Fehlermeldung "SYNTAX ERROR" ausgeben. Das ist vollkommen normal, da der Computer den Befehl in dieser Form nicht versteht. Um diesen Effekt zu vermeiden, schreiben Sie folgende kurze Programmzeile:

# IBM COMPUTERTechnik

Computersysteme · Hard- und Softwareentwicklung

CTRL-I GE

Der Druck erfolgt nur von Links nach Rechts. Durch diese Anweisung wird vermieden, dass Verschiebungen im Papier zu einem verzerrten Ausdruck fuhren.

Beispiel: 10 PRINT CHR\$(9);"GE"

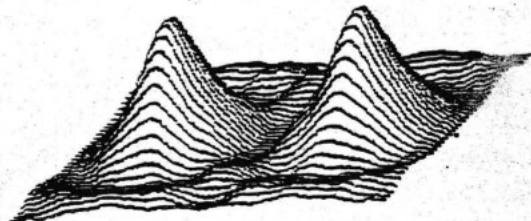
Hier ist deutlich anhand der senkrechten Umrandung zu sehen, dass keine Verschiebung mehr auftritt.



CTRL-I GR

Dreht das Bild vor der Ausgabe um 90 Grad im Uhrzeigersinn.

Beispiel: 10 PRINT CHR\$(9); "GR"



# IBM COMPUTERTechnik

Computersysteme · Hard- und Softwareentwicklung

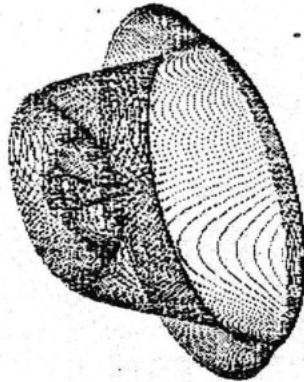
10 PRINT CHR\$(9);"G";PR#0 (RETURN)  
und starten sie mit "RUN" (RETURN).

### 3.3 Beispiele zum Grafikmodus

CTRL-I G

Im RAM-Bereich des Apple II Systems gibt es 2 Bereiche, in denen die Daten fuer die hochauflösende Grafik (HGR) gespeichert werden. Wenn Sie den Befehl CTRL-I G benutzen wird als Voreinstellung die Seite 1 ausgewählt. Die hochauflösende Bildschirmseite 1 (HGR PAGE 1) wird ausgedruckt, wobei jeder weisse Punkt auf dem Bildschirm schwarz gedruckt wird.

Beispiel: 10 PRINT CHR\$(9);"G"



CTRL-I G2

Es gilt dasselbe wie fuer das Kommando CTRL-I G, mit dem Unterschied dass statt Seite 1 Seite 2 ausgegeben wird.

Beispiel: 10 PRINT CHR\$(9);"G2"



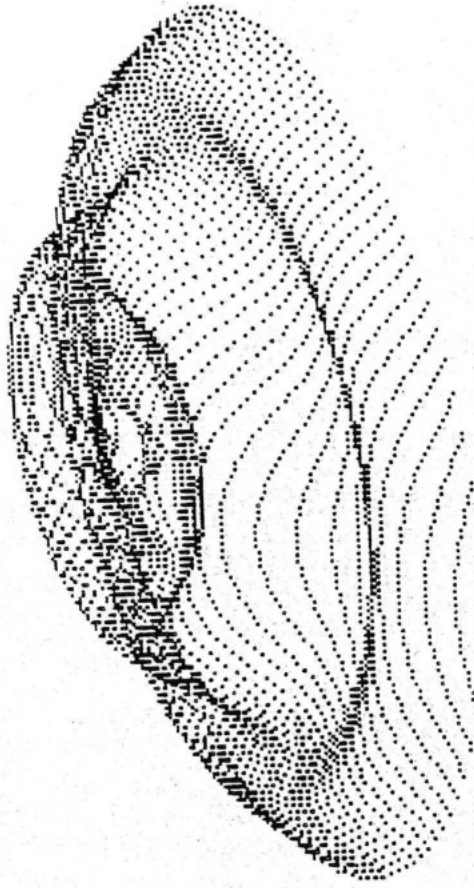
# IBS COMPUTERTechnik

Computersysteme · Hard- und Softwareentwicklung

CTRL-I 6D

Gibt das Bild in doppelter Grösse wieder. Bildgrösse wird horizontal und vertikal verdoppelt. (=4-fache Bildflaeche).

Beispiel: 10 PRINT CHR\$(9);"6D"



# IBS COMPUTERTechnik

Computersysteme · Hard- und Softwareentwicklung

CTRL-I 6L

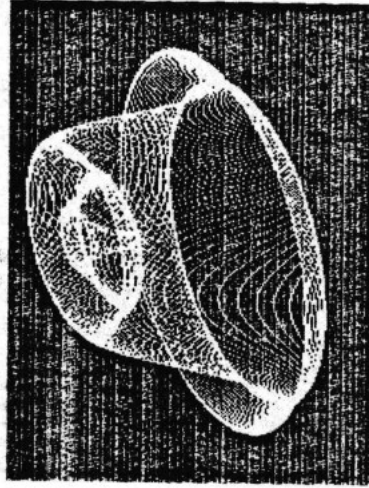
Druckt das Bild beginnend an der Position, die vorher durch den CTRL INL Befehl (linker Randabstand) vorgegeben wurde. Wird diese Option nicht benutzen, erfolgt die Darstellung zentriert, wobei von DIN-A4 Papierbreite ausgegangen wird.

Beispiel: 10 PRINT CHR\$(9); "6L"

CTRL-I 6I

Invertiert das Druckbild. Hierbei werden nicht die weissen Bildpunkte, sondern die auf dem Bildschirm schwarz erscheinenden Flaechen gedruckt

Beispiel: 10 PRINT CHR\$(9); "6I"





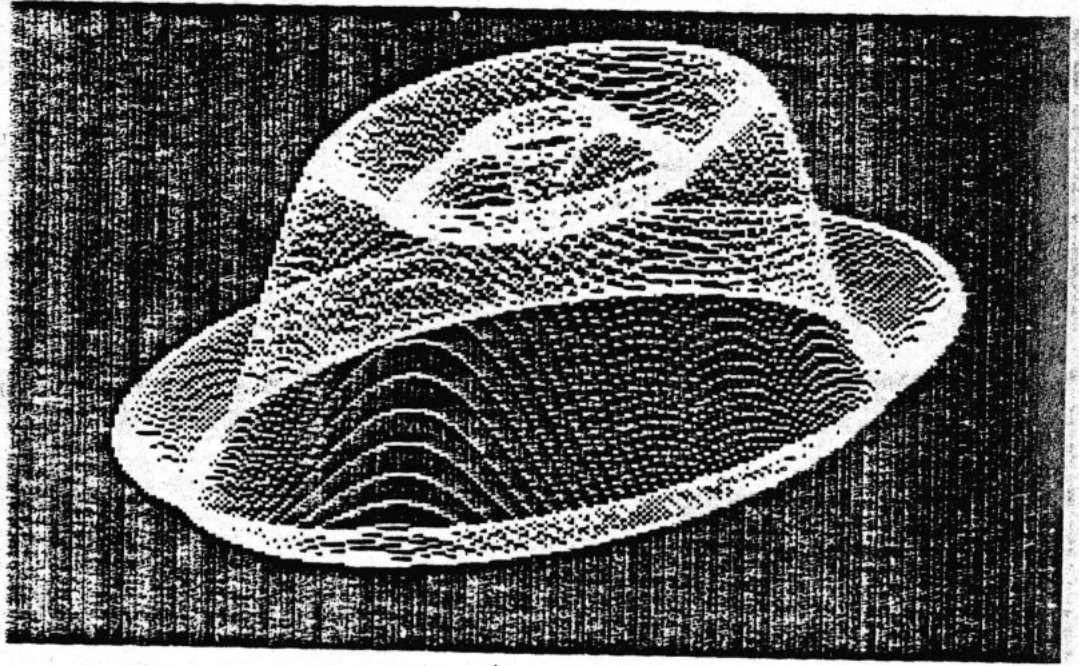
# IBS COMPUTERTECHNIK

Computersysteme · Hard- und Softwareentwicklung

Alle Graphik-Kommandos koennen kombiniert werden:

Beispiel: 10 PRINT CHR\$(9);"GERD12"

Dieser Befehl erzeugt ('E')inrichtung, um 90 Grad ('R')otierten, die ('B')oppelte Grosse des Bildschirms darstellenden ('I')nversen Ausdruck der ('G')raphik-Seite ('2').



# IBS COMPUTERTECHNIK

Computersysteme · Hard- und Softwareentwicklung

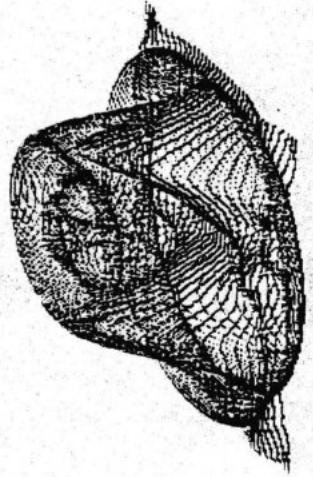
Von den nachstehenden Bildschirmsteuerzeichen darf nur eines in einer Druckeranweisung vorkommen. Es handelt sich bei diesen HGR-Druck Optionen um Logische Operationen zwischen den beiden hochauflösenden Grafikseiten. Die Daten der HGR-Seite 1 werden mit den korrespondierenden Daten der HGR-Seite 2 entsprechend der Logischen Operation verknuepft. Das Ergebnis dieser Operation wird ausgedruckt. CTRL-I GA entspricht Verknuepfung beider Bildschirmseiten durch ein logisches "UND". Nur wenn auf beiden Bildschirmseiten an der selben Stelle ein Bildpunkt ist, wird ein Bildpunkt ausgedruckt.

Beispiel: 10 PRINT CHR\$(9);"GA"



CTRL-I G0 Verknuepfung beider Bildschirmseiten durch ein logisches "ODER". Jeder Bildpunkt von beiden HGR-Seiten wird ausgedruckt.

Beispiel: 00 PRINT CHR\$(9);"G0"



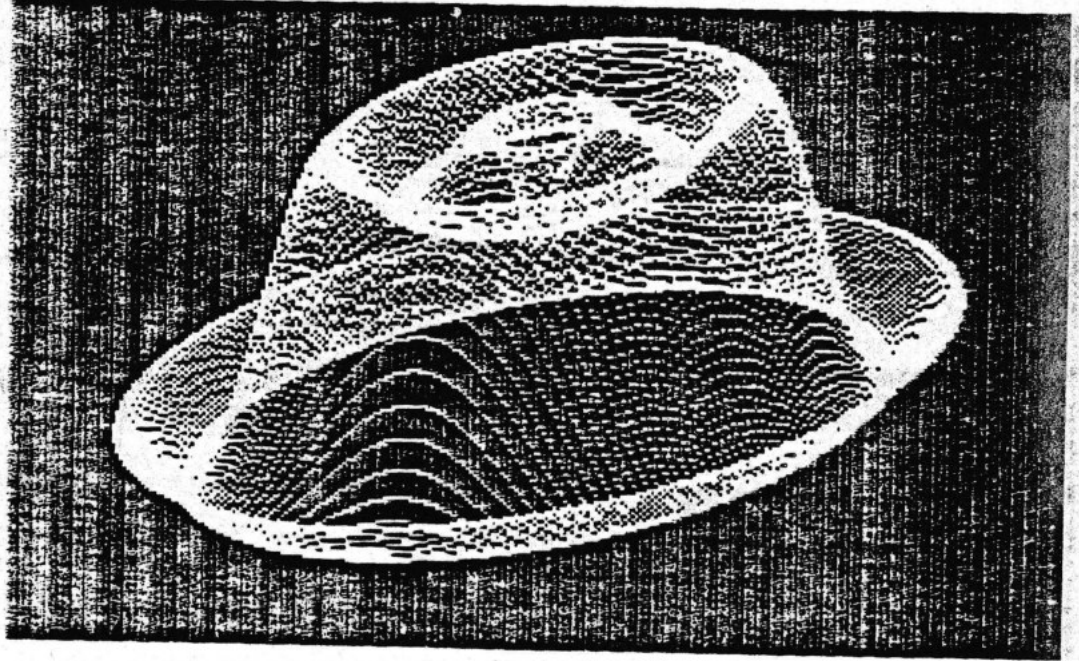
# IBS COMPUTERTECHNIK

Computersysteme · Hard- und Softwareentwicklung

Alle Graphik-Kommandos koennen kombiniert werden:

Beispiel: 10 PRINT CHR\$(9);"GERD12"

Dieser Befehl erzeugt ('E')inrichtung, um 90 Grad ('R')otierten, die ('D')oppelte Groesse des Bildschirms darstellenden ('1')nversen Ausdruck der ('G')raphik-Seite ('2').



# IBS COMPUTERTECHNIK

Computersysteme · Hard- und Softwareentwicklung

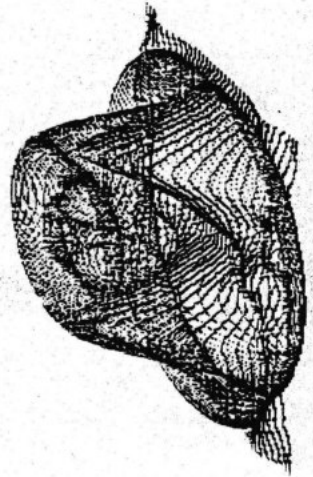
Von den nachstehenden Bildschirmsteuerzeichen darf nur eines in einer Druckeranweisung vorkommen. Es handelt sich bei diesen HGR-Druck Optionen um Logische Operationen zwischen den beiden hochauflösenden Grafikseiten. Die Daten der HGR-Seite 1 werden mit den korrespondierenden Daten der HGR-Seite 2 entsprechend der Logischen Operation verknuepft. Das Ergebnis dieser Operation wird ausgedruckt. CTRL-I GA entspricht Verknuepfung beider Bildschirmseiten durch ein logisches "UND". Nur wenn auf beiden Bildschirmseiten an der selben Stelle ein Bildpunkt ist, wird ein Bildpunkt ausgedruckt.

Beispiel: 10 PRINT CHR\$(9);"GA"



CTRL-I G0 Verknuepfung beider Bildschirmseiten durch ein logisches "ODER". Jeder Bildpunkt von beiden HGR-Seiten wird ausgedruckt.

Beispiel: 00 PRINT CHR\$(9);"G0"





# IBS COMPUTERTECHNIK

Computersysteme · Hard- und Softwareentwicklung

CTRL-I GX

Verknuepfung beider Bildschirmseiten durch ein logisches "EXCLUSIV ODER". Wenn an der selben Stelle im Bildschirm Gleichheit herrscht, wird kein Bildpunkt ausgedruckt. Nur wenn Ungleichheit gegeben ist wird ein Bildpunkt gedruckt.

Beispiel: 10 PRINT CHR\$(9);"GX"



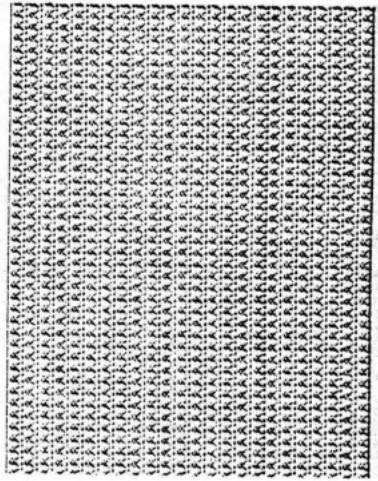
CTRL-I GT Dieses Kommando erzeugt in der ausgewachten HGR Seite ein elektronisches Testbild. Soll die HGR2 benutzt werden, so muss die Ziffer 2 vor dem Befehl eingegeben werden, da nach CTRL IG die Seite 1 angewacht ist.

Beispiel: 10 PRINT CHR\$(9);"GT"

Erzeugt ein "Testbild" in HGR1.

10 PRINT CHR\$(9);"GT2"

Erzeugt ein "Testbild" in HGR2.



# IBS COMPUTERTECHNIK

Computersysteme · Hard- und Softwareentwicklung

## 4. Voreinstellung der Druckparameter

Automatische Voreinstellung bei Ansprechen des Interfaces durch BASIC oder Initialisierung durch PASCAL und CP/M.

- Linker Rand = 0
- Zeilenlaenge = 0
- Papierlaenge = 0
- Bildschirm = an
- Drucker-Bell = aus
- Achtes Bit = aus
- Zeilenvorschub nach CR = an bei BASIC
- = aus bei PASCAL und CP/M

## 4.1 Kompatibilitaet mit PASCAL und CP/M

Das Parallel Interface arbeitet sowohl mit PASCAL als auch mit CP/M. Da beide Systeme den Befehl CTRL-I als einen Tabulatorbefehl interpretieren, wurde er durch CTRL-Y ersetzt. Wenn Sie Files ausdrucken, die Tabulatoren enthalten, pruefen Sie vorher, dass die Druckertabulatoren schon gesetzt sind. PASCAL kennt keine Monitor Routinen. Aus diesem Grund sollten Sie nicht versuchen, den Schirm mit CTRL-Y I ausschalten, oder mit CTRL-Y S den Bildschirm zu kopieren. Beides haette zur Folge, dass sowohl Drucker als auch APPLE II Computer "haengen". Fuer beide Gerate ist dann ein RESET erforderlich.

# IBM COMPUTERTECHNIK

Computersysteme · Hard- und Softwareentwicklung

## 2. Punktgraphik im BASIC-Modus

In diesem Kapitel wird beschrieben, wie man Punktgraphik von einem BASIC-Programm aus steuert.

Die folgende Abbildung zeigt den Zusammenhang zwischen den Drucknadeln und den Daten-Bits, die gedruckt werden sollen.

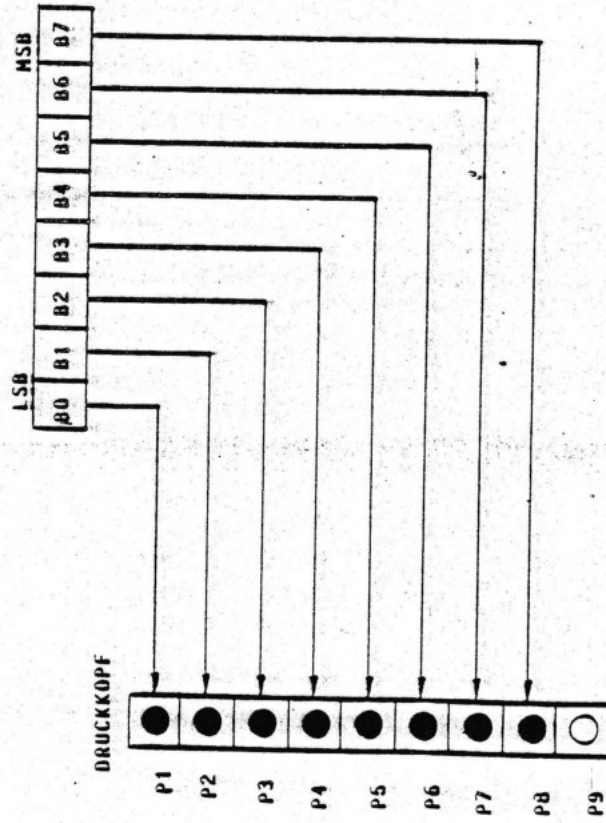


Abbildung: Zusammenhang zwischen Druckdaten und Drucknadeln.  
Anmerkung: Nadel 9 wird nur zum Unterstreichen benutzt. (siehe M8510 Bedienungshandbuch)

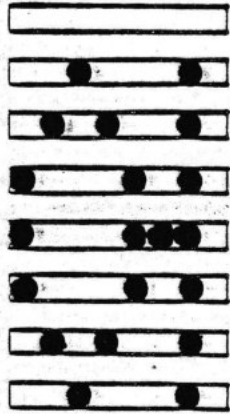
# IBM COMPUTERTECHNIK

Computersysteme · Hard- und Softwareentwicklung

Der Punktgraphik-Modus wird mit einem sechsbyte Befehl selektiert. (ESC S n<sub>1</sub> n<sub>2</sub> n<sub>3</sub> n<sub>4</sub> n<sub>5</sub> n<sub>6</sub>). Es koennen danach 1 bis 9999 Bytes transferiert werden. Die 4 Bytes n<sub>1</sub> n<sub>2</sub> n<sub>3</sub> n<sub>4</sub> stellen den ASCII-Wert der 4 Ziffern einer Dezimalzahl dar, welche die Anzahl der zu uebertragenden Bytes angibt.

Beispiel: um 10 Byte zu uebertragen  
PRINT CHR\$(27);CHR\$(83);CHR\$(48);CHR\$(49);CHR\$(48)  
oder einfacher: PRINT CHR\$(27);"S0010"

Beispiel -Programm zum Ausdrucken folgender Punktgraphik-Daten



```
10 REM PUNKTGRAPHIK-DRUCK
20 PRINT CHR$(4)"PR#1"
30 PRINT CHR$(9)"72N": REM BILDSCHIRM ABSCHALTEN
40 PRINT CHR$(27)"S0008",
50 PRINT CHR$(68);CHR$(74);CHR$(81);CHR$(81);CHR$(241);
60 PRINT CHR$(81);CHR$(10)
70 PRINT CHR$(9)"I": REM BILDSCHIRM EINSCHALTEN
80 PRINT CHR$(4)"PR#0"
```

Anmerkung: 1. Der Anwender sollte beachten, dass im obigen BASIC-Programm das "most significant bit" (MSB, 8. Bit) nicht mitgedruckt wird.

2. Im Apple-Basic-System wird dem MSB stets eine "0" bei der uebertragung zum Drucker zugewiesen. Daten, die eine "1" als MSB enthalten, koennen nicht uebertragen werden (ausser beim Druck von Kanazeichen).

3. Stehe fruhere Abschnitte zum uebertragen von Kontroll-Charakteren wie "I" als Punktgraphik-Daten.

4. Eine uebertragung von mehr als 255 Bytes ist in diesem BASIC-Programmbeispiel nicht moeglich. Wuenschen Sie eine uebertragung von mehr als 255 Bytes, benutzen Sie bitte das nachfolgende Maschinenprogramm.



# IBS COMPUTERTECHNIK

Computersysteme · Hard- und Softwareentwicklung

## 5.1 Maschinensprachen-Routine zur Übertragung von Drucker-Daten

### 5.1.1 Methode mit USR (X) - Funktion

Wenn die folgenden Routinen ausgeführt werden, können alle Daten zwischen \$00 und \$FF auf den Drucker geschickt werden. Mit USR ( ) (10K-byte BASIC)

1. 4C 00 03 bei Adresse \$A-\$C eintragen.
2. Die folgende Maschinenroutine bei \$300 eintragen:

```
START: A9 A0 LDA #SA0
        8D 9C C0 STA $C09C
        A9 FF LDA #FFF
        8D 92 C0 STA $C092
        20 0C E1 JSR $E10C
        A5 A1 LDA $A1
OUT:    8D 90 C0 STA $C090
LOOP:   AD 9D C0 LDA $C09D
        29 10 AND # $10
        F0 F9 BEQ LOOP
        60 RTS
```

Diese Adressen gelten, wenn das Interface in Slot 1 steckt.

3. Um den Code "75" dreimal von einem BASIC-PROGRAMM aus zu uebertragen, wird eine Programmzeile gebraucht, die etwa so aussieht:

```
DUMMY =USR(75)+USR(75)+USR(75)
```

(DUMMY ist ein beliebiger Variablenname)

```
10 REM PUNKTGRAFIK-DRUCK 2
20 REM MASCHINEN-UNTERPROGRAMM
30 RESTORE
40 FOR AD = 768 TO 768+25
50 READ A : POKE AD,A
60 NEXT AD
70 DATA 169,160,141,156,192,169,255,141,146,192,32,12,225,
165,161,141,144,192,173,156,192,41,16,240,249,96
80 REM USR ( ) - VEKTOR
90 POKE 10,76:POKE 11,0:POKE 12,3
100 REM ESC $ (48) (48) (56)
110 Z = USR(27)+USR (83)+USR (48)+ USR (48)+ USR (56)
120 REM Punktgrafik - Daten
130 Z = USR (68) + USR (74) + USR (81) + USR (241) + USR (81) +
USR (74) + USR (68) + USR (0)
140 Z = USR (10) + USR (10):REM ZEILENVORSCHUB
```

Erklärung des Programmbeispiels:

LIST 10-90: die Maschinenroutine wird in Speicherplatz \$300 ff. gepopt und der USR ( ) Vektor in \$A-C

LIST 100-140: Punktgrafik-Daten werden ausgedruckt.

# IBS COMPUTERTECHNIK

Computersysteme · Hard- und Softwareentwicklung

## 5.1.2 Methode mit CALL - Funktion

1. Die folgende Maschinenroutine wird ab Adresse \$300 geschrieben:

```
START: A9 A0 LDA #SA0
        8D 9C C0 STA $C09C
        A9 FF LDA #FFF
        8D 92 C0 STA $C092
        A5 00 LDA $00
        8D 90 C0 STA $C090
        AD 9D C0 LDA $C09D
        29 10 AND # $10
        F0 F9 BEQ LOOP
        60 RTS
;Interface in Slot 1
```

2. Um den Code "75" dreimal von einem BASIC-Programm aus zu uebertragen, wird etwa folgendes Programm gebraucht:

```
10 POKE 0,75 : CALL 768
20 POKE 0,75 : CALL 768
30 POKE 0,75 : CALL 768
```

Anmerkung: "\$00" ist eine voruebergehende Speicheradresse, in der die zu druckenden Daten abgelegt werden.

```
10 REM PUNKTGRAFIK-DRUCK
20 REM MASCHINEN-UNTERPROGRAMM
30 RESTORE
40 FOR AD = 768 TO 768 + 22
50 READ A : POKE AD,A
60 NEXT AD
65 DATA 169,160,141,156,192,169,255,141,146,192,
165,0,141,144,192,173,156,192,41,16,240,249,96
70 REM ESC $ (48) (48) (56)
80 POKE 0,27 : CALL 768
90 POKE 0,83 : CALL 768
100 POKE 0,48 : CALL 768
110 POKE 0,48 : CALL 768
120 POKE 0,48 : CALL 768
130 POKE 0,56 : CALL 768
140 REM PUNKTGRAFIK-DATEN
150 POKE 0,68 : CALL 768
160 POKE 0,74 : CALL 768
170 POKE 0,81 : CALL 768
180 POKE 0,241 : CALL 768
190 POKE 0,81 : CALL 768
200 POKE 0,74 : CALL 768
210 POKE 0,68 : CALL 768
220 POKE 0,0 : CALL 768
230 REM ZEILENVORSCHUB
240 POKE 0,10 : CALL 768
250 POKE 0,10 : CALL 768
```

Anmerkungen zum Gebrauch

Endet eine PRINT-Anweisung mit einem Semikolon, wird kein CR an den Drucker gesendet. Das bedeutet, dass bei der Uebergabe von nichtdruckenden Kontrollcharakteren an den Drucker ohne Semikolon Abschluss ein Zeilenvorschub erfolgt. D.h., dass nach

```
10 PRINT CHR$(9) "80"
20 PRINT CHR$(9) "6L"
30 PRINT CHR$(9) "60P"
```

vorab drei Zeilenvorschuebe erfolgen.

Vermeiden koennen Sie dies, wenn Sie in jeder Programmzeile am Ende ein Semikolon einfuegen.

6. Elektronische Vertikale formatsteuerung (EVFU)

Dieses BASIC-Programmbeispiel erstellt eine fuer eine einfache EVFU. Diese Formatierung des Druckers ist dann zu benutzen, wenn Sie nicht die Befehlssequenz CTRL-Inf verwenden koennen, weil Sie entweder andere Papierformate haben oder eine von 6 abweichende Anzahl von Zeilenvorschueben wuenschen.

```
10 REM FORMATIERROUTINE
20 REM MASCHINEN-UNTERPROGRAMM
30 RESTORE
40 FOR AD = 768 TO 768 + 22
50 READ A: POKE AD,A
60 NEXT AD
70 DATA 169,160,141,156,192,169,255,141,146,192,32,12,225,
165,161,141,144,192,173,156,192,41,16,240,249,96
80 REM USR() -VEKTOR
90 POKE 10,76: POKE 11,0: POKE 12,3
100 REM FORMATDATEN GS-LADEBEFEHL (29)
TOF-FORMATANFANG (65) (64)
110 Z = USR (29) + USR (65) + USR (64)
120 FE = 66: REM FORMATEMDE
130 FL = 72: REM FORMATLAENGE
140 FOR I = 2 TO FE
150 Z = USR (64) + USR (64)
160 NEXT I
170 REM BOF - FORMATEMDE WIRD GESETZT (67) (64)
180 Z = USR (67) + USR (64)
190 FOR I = FE + 2 TO FL
200 Z = USR (64) + USR (64)
210 NEXT I
220 REM FORMATDATEN: TOF-FORMATANFANG (65) (64)
RS-LADEABSCHLUSS (30)
230 Z = USR (65) + USR (64) + USR (30)
240 END
```

Anpassung Visiplot an AP 4 G

Anpassung:

1. APPLE mit DOS 3.3 booten
2. Visiplotdiskette einlegen
3. Das File "VISIPL0T" einladen (LOAD VISIPL0T)
4. Zeile ... aendern (Liste 1)
5. Das File "VISIPL0T" auf Diskette ablegen (SAVE VISIPL0T)

Bedienung:

Nach dem Anwaehlen des "PRINT" von Visiplot erscheint die Zeile  
COMMAND (AP4G):

Nun gibt man die Art des Druckbildes an.  
z.B. GE CR Seite 1 mit unidirektionalem Druck  
GDE CR Seite 1 mit doppelter Groesse und unidirektionalem Druck

Liste 1

```
385 IF PEEK (944)= 0 THEN POKE 975,PS:
POKE 39118,PS :HOME = VTAB 23:HTAB1:
INPUT "COMMAND (AP4G) ?" ; SH$
386 PRINT D$: "PRW": PS:PRINT CHR$(9); SH$:
PRINT D$: "PRW0"
```