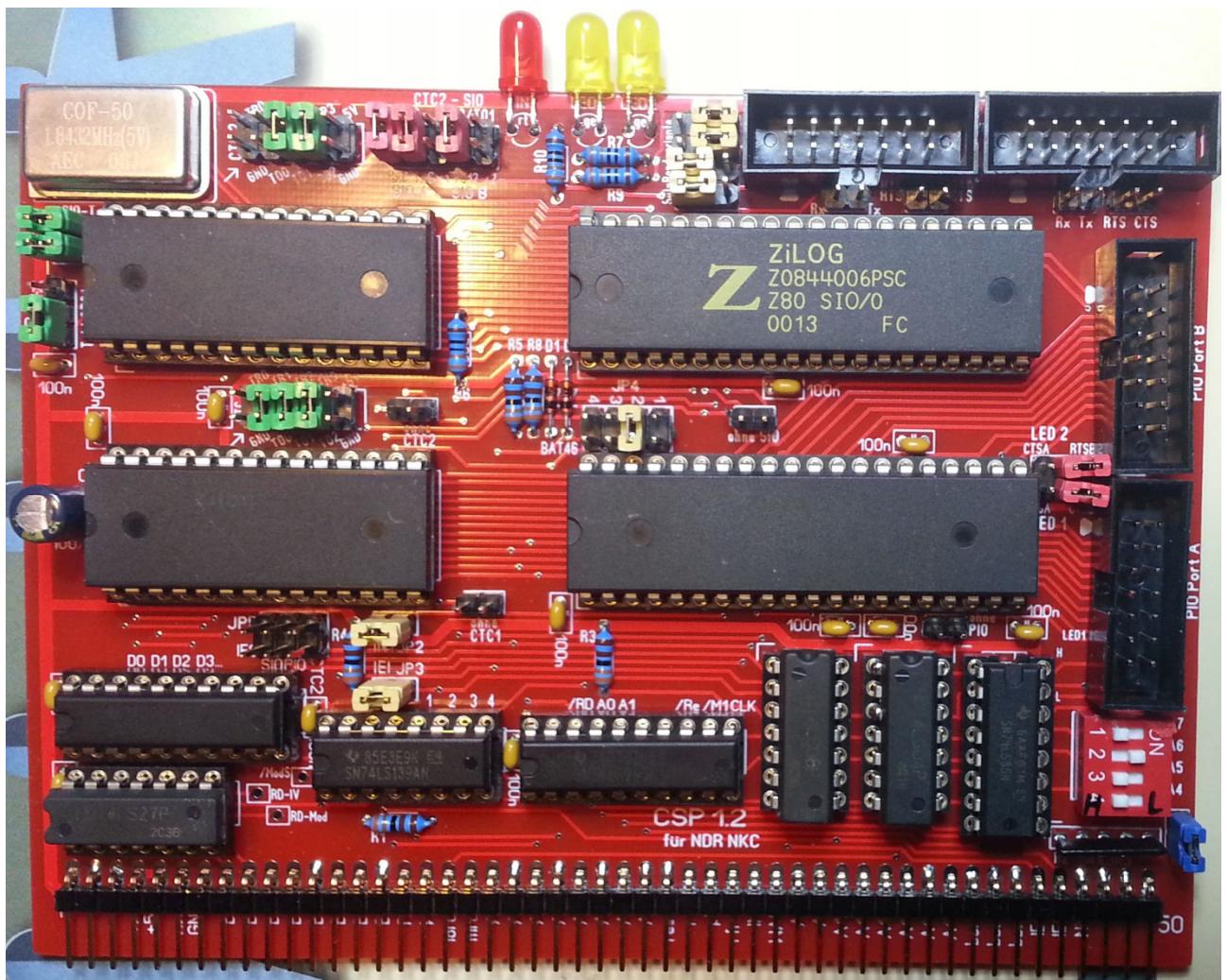


# C S P

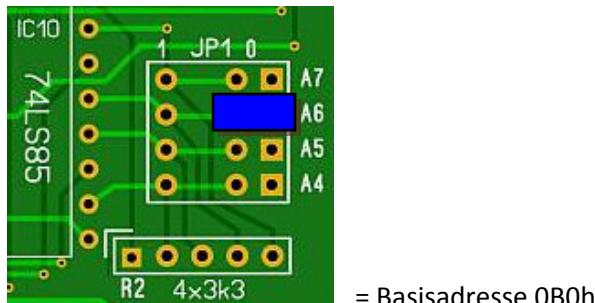
Counter-Seriell-Parallel V 1.2b



Das Modul stellt zwei CTC, eine SIO und eine PIO (*nachf. Bausteine genannt*) am NKC-Bus bereit. In erster Linie ist das Modul (*diese Leiterplatte*) für den Z80 im Interruptmode IM2 vorgesehen. Darüber hinaus sind jedoch auch Modifikationen vorbereitet um per Jumper andere Interruptmodes zu unterstützen.

## Beschreibung CSP 1.2b

Das Modul belegt 16 IO-Port-Adressen und kann mit Jumper **J 1** auf eine beliebige Basisadresse eingestellt werden. Dabei gelten für die Jumper: offen = H / geschlossen = L. Der einzige momentan freie und durchgängige IO-Adressbereich beginnt im NKC bei 0B0h. Will man diese Basisadresse einstellen ist folgender Jumper zu setzen:



Damit ergeben sich:	0B0h	CTC1 Kanal0	0B4h	CTC2 Kanal0
	0B1h	CTC1 Kanal1	0B5h	CTC2 Kanal1
	0B2h	CTC1 Kanal2	0B6h	CTC2 Kanal2
	0B3h	CTC1 Kanal3	0B7h	CTC2 Kanal3
	0B8h	SIO PortA-Daten	0BCh	PIO PortA-Daten
	0B9h	SIO PortA-Cmd	0BDh	PIO PortA-Cmd
	0BAh	SIO PortB-Daten	0BEh	PIO PortB-Daten
	0BBh	SIO PortB-Cmd	0BFh	PIO PortB-Cmd

Um die Interrupt-Priorität zu steuern sind die Bausteine miteinander in einer Interrupt-Prioritätskette der sog. „daisy-chain“ verknüpft. Jeder Baustein hat einen **IEI**-Eingang (**InterruptEnableIn**) und einen **IEO**-Ausgang (**InterruptEnableOut**). Nur wenn am **IEI** ein H anliegt ist es dem jeweiligen Baustein erlaubt, einen Interrupt anzumelden. Das tut er, in dem er die **/INT**-Leitung auf dem Bus auf L-Potential zieht. Der Baustein der einen Interrupt anmeldet zieht zusätzlich seinen **IEO** Ausgang auf L-Potential und sperrt damit die Interrupts der folgenden niedrigwertigeren Bausteine.

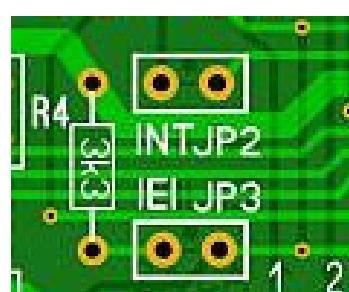
Wenn das CSP-Modul das einzige oder höherwertigste Modul ist, welches Interrupt auslösen darf, muss das Bussignal **IEI** für den Steckplatz des Modules mit einem 3k3 Widerstand auf H-Potential gezogen werden. Damit hat jedes Modul, welches auf diesen Steckplatz gesteckt wird, automatisch die höchste Priorität.

Alternativ kann auch der Jumper **J3** gesetzt werden. Dann hat das Modul, egal wo es auf dem Bus eingeschoben wird immer die höchste Priorität. Bei mehreren CSP-Modulen im Einsatz darf der Jumper **J3** nur bei dem ersten Modul gesetzt sein!

Um Konflikte zu vermeiden, sollte man sich strikt für eine der beiden Varianten entscheiden.

**JP3** schließen um IEI des ganzen Moduls auf H-Potential zu ziehen

**JP2** schließen um dem Modul überhaupt das Auslösen von Interrupts zu ermöglichen



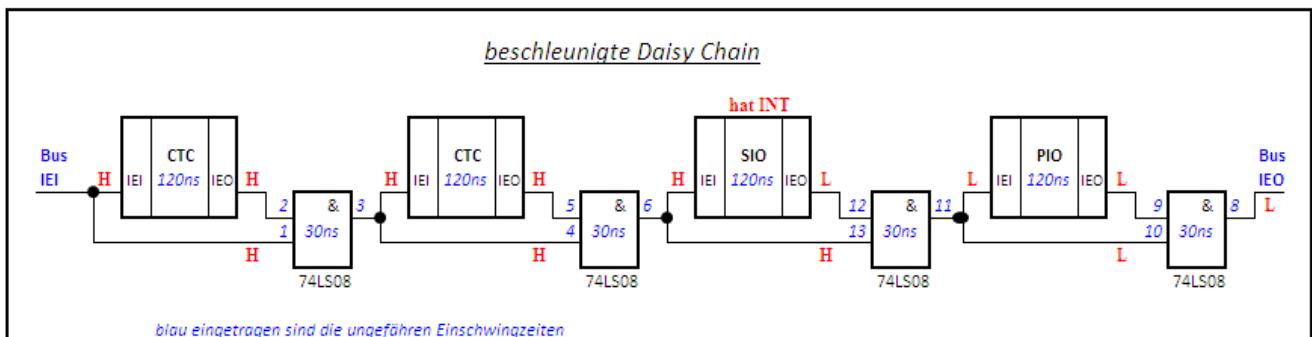
Wichtig ist auch, dass die Module, die Interrupt anmelden dürfen, in direkter Reihenfolge auf dem Bus so platziert werden, dass jeweils der IEO-Modulausgang mit dem IEI-Moduleingang des Folgemoduls verbunden ist.

## Beschreibung CSP 1.2b

### Beschleunigte Daisy Chain

Sobald mehrere Bausteine in der Interrupt-Prioritätskette eingeschleift sind, besteht zunehmend die folgende Gefahr: Ein Baustein mit hoher Priorität meldet einen Interrupt an, doch es braucht eine gewisse Zeit bis durch alle nachfolgenden Bausteine und Module sein IEO=L-Signal den letzten Baustein erreicht hat. Dann könnte der letzte Baustein auch zeitgleich einen Interrupt angemeldet haben. In der Folge würden beide im selben Moment ihren Interruptvektor auf den Bus legen, was zu Abstürzen führen würde. Um das zu vermeiden, kann man die Interrupt-Prioritätskette des einzelnen Moduls beschleunigen.

Im CSP-Modul ist das mit UND-Gattern wie folgt realisiert:



Damit verkürzt sich die Durchleitezeit im Modul von 480ns auf 120ns.

Um die Daisy Chain aufrecht zu erhalten und evtl. wilde Schwingungen durch offene Eingänge zu vermeiden, ist in der Nähe jedes Bausteins ein Jumper „ohne CTC“, „ohne SIO“ und „ohne PIO“ vorhanden, der jeweils dann zu stecken ist, wenn der zugehörige Baustein nicht bestückt wird.

### Richtungssteuerung des Datenbustreibers

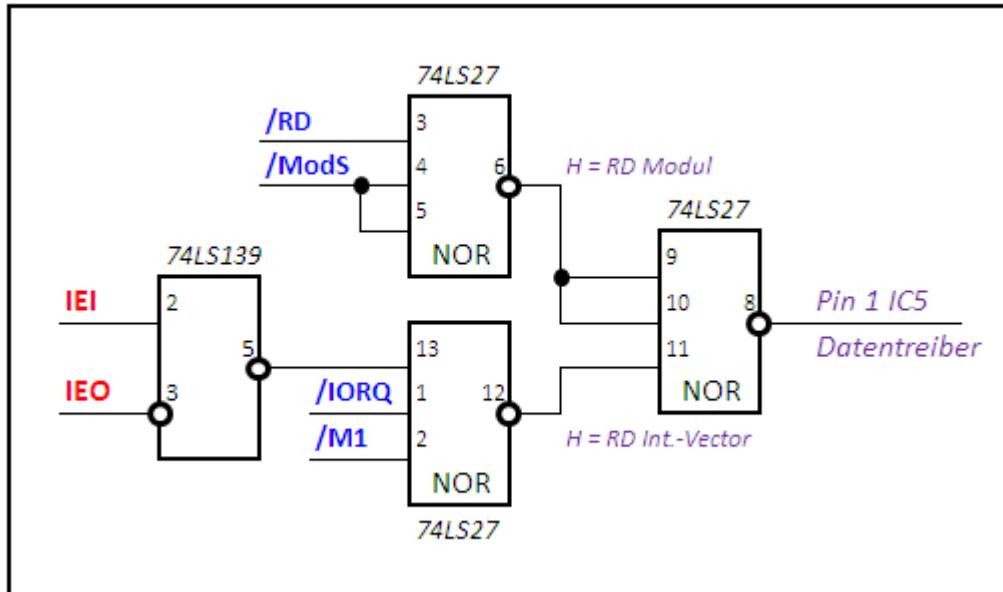
Im Interrupt-Mode IM2 gibt es zwei Besonderheiten:

1. Der interruptauslösende Baustein hält sein IEO-Ausgang solange auf L-Potential, und sperrt damit die Interrupts der nachfolgenden Bausteine, bis er auf dem Datenbus die Befehlsfolge „0EDh“ „4Dh“ (= RETI-Befehl) - als Zeichen, dass die/seine Interruptbearbeitung beendet ist - liest. Das ist jedoch nur möglich, wenn der Datenbus ständig an die Bausteine durchgeleitet wird und diese ständig den Daten-Bus „mitlessen“ können, auch wenn sie nicht angesprochen sind.
2. Die Z80 CPU „fragt“ nach einer Interruptanmeldung – INT = L - „Wer war das?“. Das macht die Z80 mit dem zeitgleichen Aktivierung der Steuerleitungen **/IORQ** und **/M1** (/RD ist dabei inaktiv!). Daraufhin legt der auslösende Baustein den zuvor für diesen Fall im Baustein programmierten Interruptvektor auf den Datenbus. Dieser Interruptvektor bildet zusammen mit dem I-Register der Z80 CPU einen Adresszeiger in eine Tabelle mit Einsprungadressen für die jeweilige Interruptroutine (Programm das im Interruptfall starten soll).

Der Datenbustreiber muss (wie oben ausgeführt) den Datenbus ständig an die Bausteine durchleiten, bis auf die beiden Fälle, wenn von einem Baustein gelesen werden soll oder dieser seinen Interruptvektor an die Z80-CPU senden will. Das bedingt eine besondere Richtungs-Steuerung des Datentreibers.

## Beschreibung CSP 1.2b

Beide Fälle werden durch die folgende Schaltung gewährleistet:



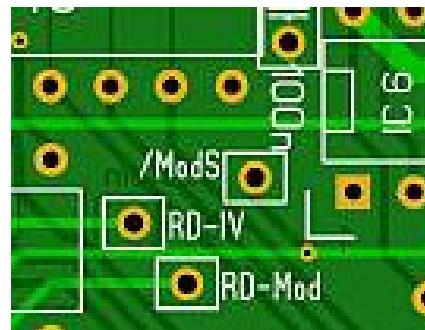
Ein L Potential am Pin1 des Datenbustreivers (IC5) bedeutet, dass der Datenstrom von den Bausteinen zum NKC-Bus geleitet wird.

Um die korrekte Arbeitsweise überprüfen zu können, sind in der Nähe des Datenbustreivers drei Kontroll-Pins angebracht.

**/ModS = L** wenn IO-Adresse angesprochen wird

**RD-IV = H** wenn vom Modul der Interruptvektor gelesen wird

**RD-Mod = H** wenn mit einem IN-Befehl vom Modul gelesen wird



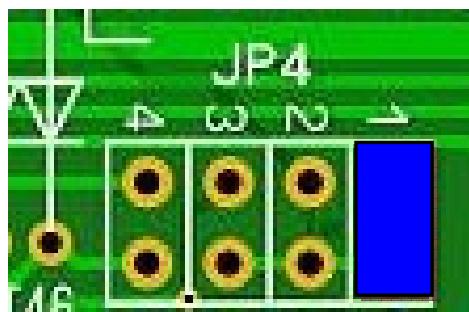
Die Kontroll-Pins kann natürlich jeder nach seinem Gusto ausführen oder ganz weg lassen. Denkbar ist, hierfür jeweils einen ausgesägten Pin einer Präzisions IC-Fassung zu verwenden.

## Beschreibung CSP 1.2b

### RESET-Steuerung der PIO

Eine Besonderheit weist noch die PIO-Baustein auf. Aufgrund der beschränkten Anzahl von Pins haben die Konstrukteure auf einen separaten RESET-Anschluss verzichtet. Das Rücksetzen der internen PIO-Register und der Abbruch der Übertragung erfolgt hier durch ein L-Potential am **/M1**-Anschluss, ohne dass **/RD** oder **/WR** aktiv sind (diese bleiben auf H-Potential).

Das bedeutet, dass die PIO am Pin **/M1** in einem Z80-System die Bussigale **/M1** und **/RESET** mit einer ODER-Verknüpfung erhalten muss. Um nicht einen weiteren IC einsetzen zu müssen, wurde diese Funktion mit den Dioden **D1** und **D2** realisiert. Der Jumper **J4** muss dafür in Position 2 gesteckt werden.



reguläres Z80-System

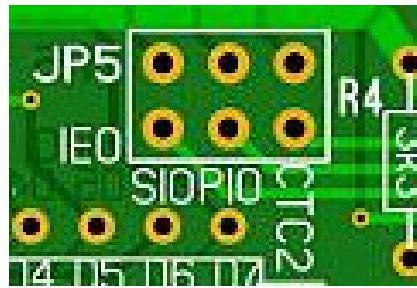
- Pos.1: nur **/M1** an **/M1**-Pin der PIO
- Pos.2: **/M1 ODER /RESET** an **/M1**-Pin der PIO
- Pos.3: **/RESET** an **/M1**-Pin der PIO
- Pos.4: Das **/M1**-Bussignal wird auf H-Potential gezogen  
(in Systemen anderer Prozessoren, die kein **/M1**-Signal haben)

Die Jumperstellungen 2,3 und 4 sind für Nicht-Z80-Systeme gedacht, in denen das **/M1**-Signal u.U. nicht vorhanden ist. Hier ist noch ein Experimentierfeld, es liegen dafür noch keine Erfahrungswerte vor.

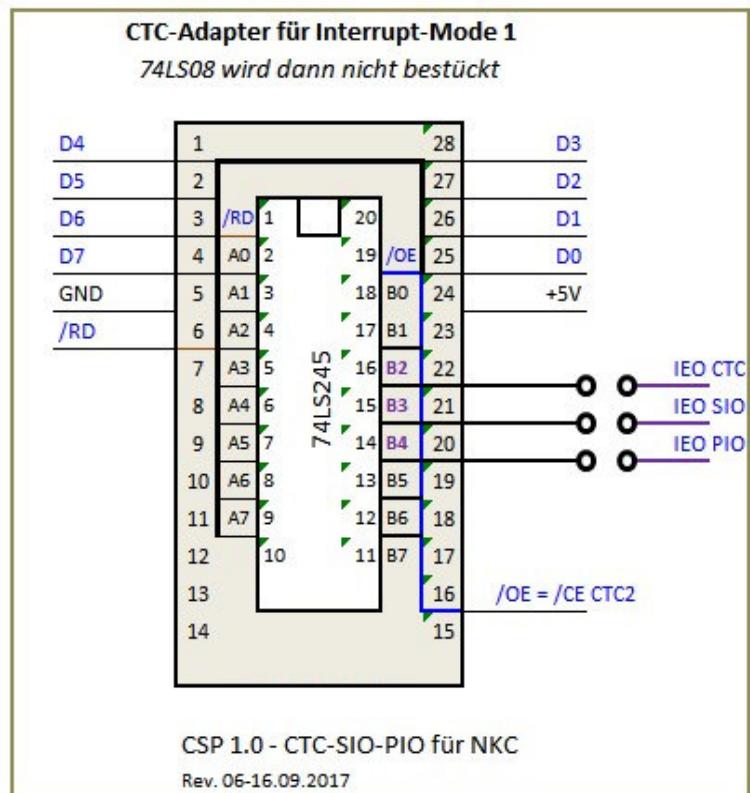
### Andere Interrupt-Modi

Werden Interruptmodi eingesetzt, bei denen die CPU selbst nach dem Auslöser des Interrupts „fahnden“ muss, kann statt der CTC1 ein Port-IC auf einer Adapter-Platine eingesetzt werden.

Dafür ist der Jumper **J5** vorgesehen



Mit diesem können die **IEO**-Ausgänge von CTC2, SIO und PIO auf die Pins 20,21,22 des CTC1-Steckplatzes gelegt werden. Ein 74LS245 kann dann z.B. unter Ausnutzung des **/CE**-Signals für CTC1 (**IO-**Adresse 0B0h) zum Auslesen der **IEO**-Signale

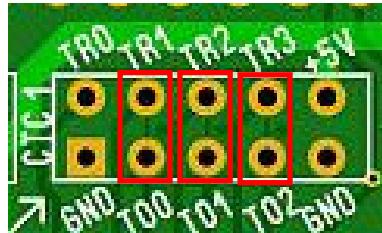


## Beschreibung CSP 1.2b

der Bausteine Verwendung finden. Der Jumper „**ohne CTC1**“ ist in dem Falle zu setzen.

### CTC-Port

Die CTC-Ports sind derart gestaltet, dass für eine Kaskadierung der CTC-Kanäle jeweils direkt ein Jumper gesetzt werden kann:

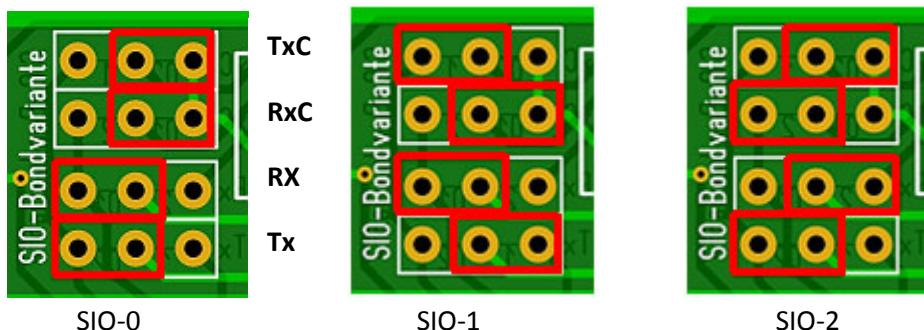


Brückt man **TO-0 mit TR-1**, **TO-1 mit TR-2** und **TO-2 mit TR-3**, so sind die Kanäle 0, 1, 2 und 3 miteinander kaskadiert. D.h. der jeweilige Ausgang TO (Takt-Out) wurde mit dem folgenden TR (Trigger)-Eingang verbunden. Damit lassen sich sehr langsame Takte, bis in den Minutenbereich, aus dem Systemtakt generieren.

### Besonderheiten SIO

Die SIO wird im Bereich der Port B Pins in unterschiedlichen Bondvarianten angeboten, die sich auf die Belegung der Peripheriepins beim Port B auswirken.

Dem wurde beim Layout des Moduls mit einem Jumpfeld Rechnung getragen:



Damit haben die Wannenstecker für den SIO-Port A & B gleiche Belegungen und anzuschließende Geräte können prinzipiell auf beide Anschlüsse gesteckt werden. Geraden findet diese Wahlfreiheit, da der SIO-Port B nicht alle Signale in den einzelnen Bondavarianten anbietet. Die Signale **Tx**, **Rx**, **RTS** und **CTS** sind



jedoch bei entsprechender Stellung der Jumper bei beiden Ports identisch.

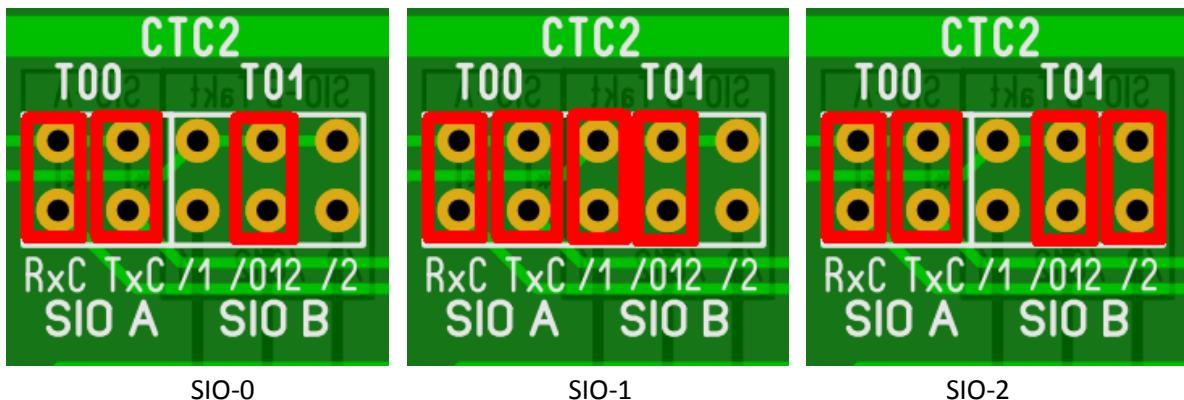
Die Jumper RX TX und RTS CTS dienen dazu, zu Testzwecken jeweils den Ausgang mit dem Eingang zu verbinden.

Damit erhält ein Programm ein direktes Echo auf die seriellen Ausgaben in selben Port.

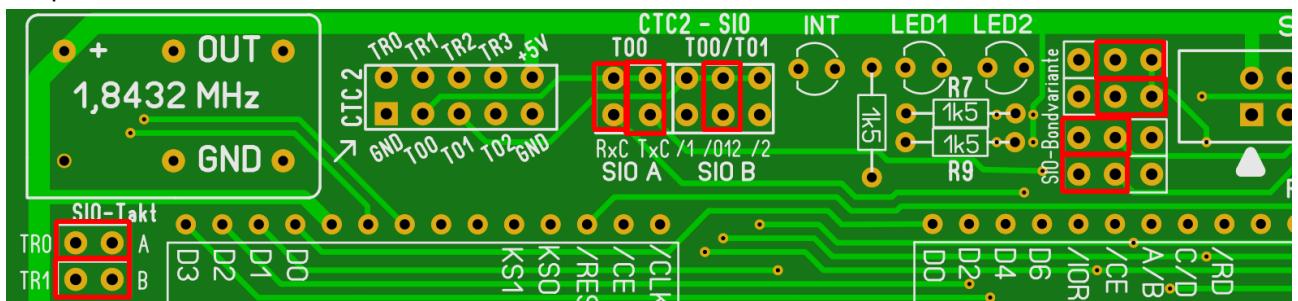
## Beschreibung CSP 1.2b

Mit dem Jumper **CTC2 SIOA SIOB** kann man den mit der CTC generierten Baud-Takt (**T00** und **T01**) nach Belieben an die SIO anlegen.

Die folgenden Beispiele zeigen die Jumperstellungen für verschiedene Bondvarianten, wenn der Rx- und der Tx-Takt von der CTC generiert werden soll.



Beispiel SIO-0 - unterschiedlicher Takt für Port A und B



Die Jumper „SIO-TAKT“ sind beide gesetzt, d.h. der Takt des Oszillators gelangt in der CTC2 gleichzeitig an TR0 und TR1. Jumper auf „CTC2“ sind dafür nicht nötig. Auf CTC2-SIO werden die CTC2 Ausgänge T00 und T01 mittels der Jumper T00 an SIOA-TxC-RxC und T01 auf mitte gesetzt. Damit erhalten für Port A Sender und Empfänger den gleichen Takt. Der Jumper CTS2-SIO T01 wird auf die Bondvarianten /012 gesetzt. Ebenso muss das Jumperfeld SIO-Bondsvariante auf die Variante-0stellung gesetzt werden.

Der Oszillator liefert 1,8432 MHz zur CTC2-TR0+TR1. Auf CTC-Kanal 0 und 1 wird über den programmierten Zähler für Kanal0 = SIO-PortA und Kanal1 = SIO-PortB und den internen Verteiler 1:16 der SIO (!) Baudraten zwischen 57k6 und 600 Bd programmiert werden.

## BAUDRATEN

Eine weitere Besonderheit stellt im Detail die Takterzeugung für die Datenübertragung der SIO dar.

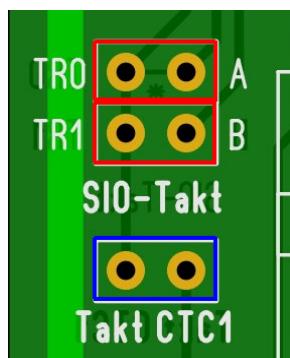
Ausgehend von einem 1,8432 MHz Oszillator kann dessen Takt je nach Wunsch mit dem Jumper **SIO-Takt** an den Triggereingang des Kanals 0, **TR0** oder/und **TR1** der CTC2 gelegt werden. Daraus lassen sich sodann in der CTC2 für SIO-PortA und PortB getrennt die wichtigsten Taktraten generieren.

## Beschreibung CSP 1.2b

Für die asynchrone Datenübertragung benötigt die SIO einen eine höhere Daten-Taktrate für die Datenübertragung, als die tatsächliche Baudrate im Empfangstakt sein soll. Nur so ist es der SIO möglich den Empfangsdatenstrom innerhalb eines Bit's mehrfach abzutasten um die Bit-Flanke sicher erkennen zu können. Dafür lässt sich über das Schreibregister WR4 der SIO ein Taktteiler auf x1, x16, x32 oder x64 einstellen. Für den Takteiler x1 ist die SIO auf externe syncronisierung des Datenstroms angewiesen; d.h. es muss neben der Datenleitung eine Synchronsignal-Leitung angeschlossen und ein syncroner Übertragungsmodus gewählt werden.

Wird der SIO-Taktteiler x16 verwendet, lassen sich z.B. aus der CTC-Taktrate von 1,8432 MHz nebenstehende Baudraten generieren.

Takt	SIO-Teiler	Zähler	BAUD
1,8432	16	2	57600,00
1,8432	16	3	38400,00
1,8432	16	4	28800,00
1,8432	16	6	19200,00
1,8432	16	8	14400,00
1,8432	16	12	9600,00
1,8432	16	16	7200,00
1,8432	16	24	4800,00
1,8432	16	32	3600,00
1,8432	16	48	2400,00
1,8432	16	64	1800,00
1,8432	16	96	1200,00
1,8432	16	192	600,00



Im nebenstehenden Bild sind die Jumper rot eingezeichnet, die den Oszillator-Takt 1,8432 MHz an die Trigger-Eingänge 0 und 1 der CTC1 leiten. Dort können diese für den SIO-Port A bzw. B weiter geteilt werden.

**Damit kann die SIO auf den beiden Ports unterschiedliche Baudraten arbeiten.**

**Der blaue Jumper leitet den 1.8432 MHz Takt an den Trigger-Eingang 0 der CTC2, falls dieser Takt dort z.B. ein Zeitsignal erzeugen soll.**

$$1,8432 \text{ MHz} = 0,54253472 \mu\text{s} / 192 / 96 = 10 \text{ ms} / 100 = 1 \text{ s}$$

Wird also der CTC2 der Oszillatortakt auf TRO zugeführt und alle Kanäle kaskadiert, so erhält man:

CTC2 – Kanal 0: Zähler = 192 TO0 = 0104,166667 µs

CTC2 – Kanal 1: Zähler = 96 TO1 = 10 ms

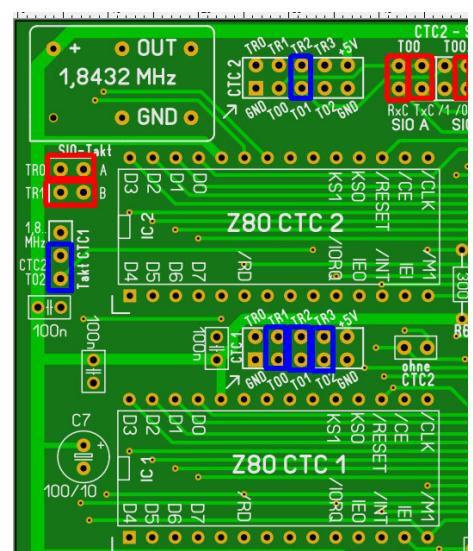
CTC2 – Kanal 2: Zähler = 100 TO2 = 1 s

CTC2 – Kanal 3: Zähler = 60 INT-Kanal3 = 1 Minute

(TO3 nicht vorhanden!)

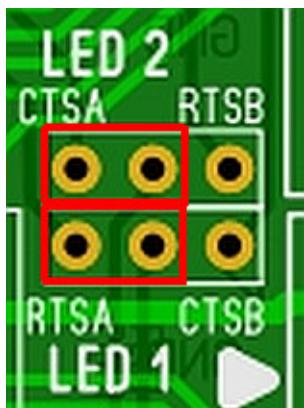
Wird jedoch für den PortB der SIO eine Baudrate von 9k6 verwendet, so hat man folgende Möglichkeit (blaue Jumper):

Der Oszillator		1,8432 MHz wird mit
CTC2 Kanal 1,	Zähler 12 auf	153.600 p.sec
CTC 2 Kanal 2,	Zähler 16 auf	9600 p.sec
CTC 1 Kanal 0,	Zähler 96 auf	100 p.sec = 10ms
CTC 1 Kanal 1,	Zähler 100 auf	1 sec
CTC 1 Kanal 2,	Zähler 60 auf	1 min
CTC 1 Kanal 3,	Zähler 60 auf	1 Std

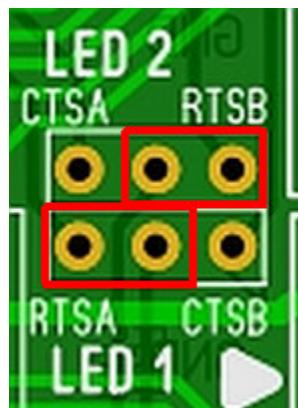


### LED – Statusanzeigen

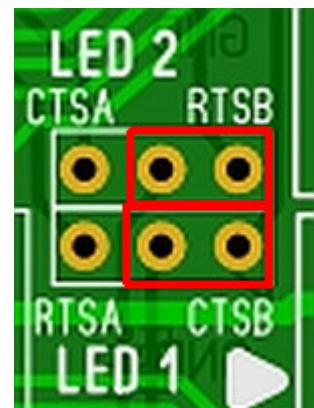
Im oberen Bereich des Moduls sind noch zwei Status-LED vorgesehen, die nach Belieben bestückt werden können um damit die Datenübertragung der SIO überwachen zu können. Über einen Jumper lässt sich deren Bedeutung variieren um Wahlweise die Datenfluss Steuersignale RTS und/oder CTS anzeigen zu lassen.



LED 1 zeigt RTS Port A  
LED 2 zeigt CTS Port A



LED1 zeigt RTS Port A  
LED 2 zeigt RTS Port B



LED1 zeigt CTS Port B  
LED 2 zeigt RTS Port B

Bei der asynchronen Verbindung mit Rx-Tx und RTS-CTS (z.B. RS232) von zwei Computern können die Signale RTS und CTS zur Datenflußsteuerung verwendet werden.

Kieser/Meder „Mikroprozessortechnik“:

„**Clear To Send** - Eingang, L-aktiv: Diese Modemeingänge können programmiert werden als automatische Freigabe. Ein L an diesen Eingängen gibt den entsprechenden Senderfrei.“

„**Request To Send** Ausgang, L-aktiv: Diese Modemsteuerausgänge dienen zur Sendeausforderung. Der RTS-Ausgang wird L, wenn das RTS-Bit gesetzt ist. „

In der Literatur werden die Signale oft missverständlich verwendet, da diese meist aus Sicht des Computers für den Computer und für das Modem stehen.

Aus Sicht der SIO steht fest:

RTS = „Redy to send“ ist ein schaltbarer Ausgang (besser: Redy to Receive)  
CTS = „Clear to send“ ist ein Eingang der den Sender stoppt (besser: Tx-Enable)

Die Funktion:

RTS-Ausgang = L → durch Bit 1 im Schreibregister WR5 d.SIO kann der Gegenstelle die Empfangsbereitschaft angezeigt werden  
CTS-Eingang = L → Sender der SIO wird aktiviert (soweit Daten vorhanden)

Verbindet man also RTS und CTS zweier RS232, so zeigt der DatenEmpfänger mit seinem Ausgang RTS = L an, dass er bereit ist Daten zu empfangen, z.B. nach dessen Initialisierung. Dieses Signal ist mit dem CTS-Eingang des DatenSenders verbunden und aktiviert mit dem L-Pegel den Sender, falls Daten zum Senden bereit stehen.

Ist der Empfänger beschäftigt um z.B. die eben empfangenen Daten zu verteilen/verarbeiten/drucken muss er nur sein RTS-Ausgang auf HL-Pegel schalten (SIO: WR5 D1) damit stellt der DatenSender mit einem H-

## Beschreibung CSP 1.2b

Pegel an seinem CTS-Eingang das Senden ein. Je nach Software wird dies als kurzzeitige Unterbrechung oder als Abbruch der Übertragung gewertet.

Also ganz einfach:

RTS-Ausgang    wird durch Steuerbefehl an SIO WR5 D1 beliebig auf L/H geschaltet  
CTS-Eingang    = H, eigener Sender aus / = L, eigener Sender ein

Damit wird klar, dass

**RTS-Signal wenn die LED leuchtet** (L-Pegel)  
bedeutet, dass die eigene Stelle anzeigt, dass sie Daten empfangen möchte

**CTS-Signal wenn die LED leuchtet** (L-Pegel)  
bedeutet, dass eine Gegenstelle da ist, die Daten empfangen kann.  
Dies ist ein guter Indikator dafür, ob die Verbindung steht.

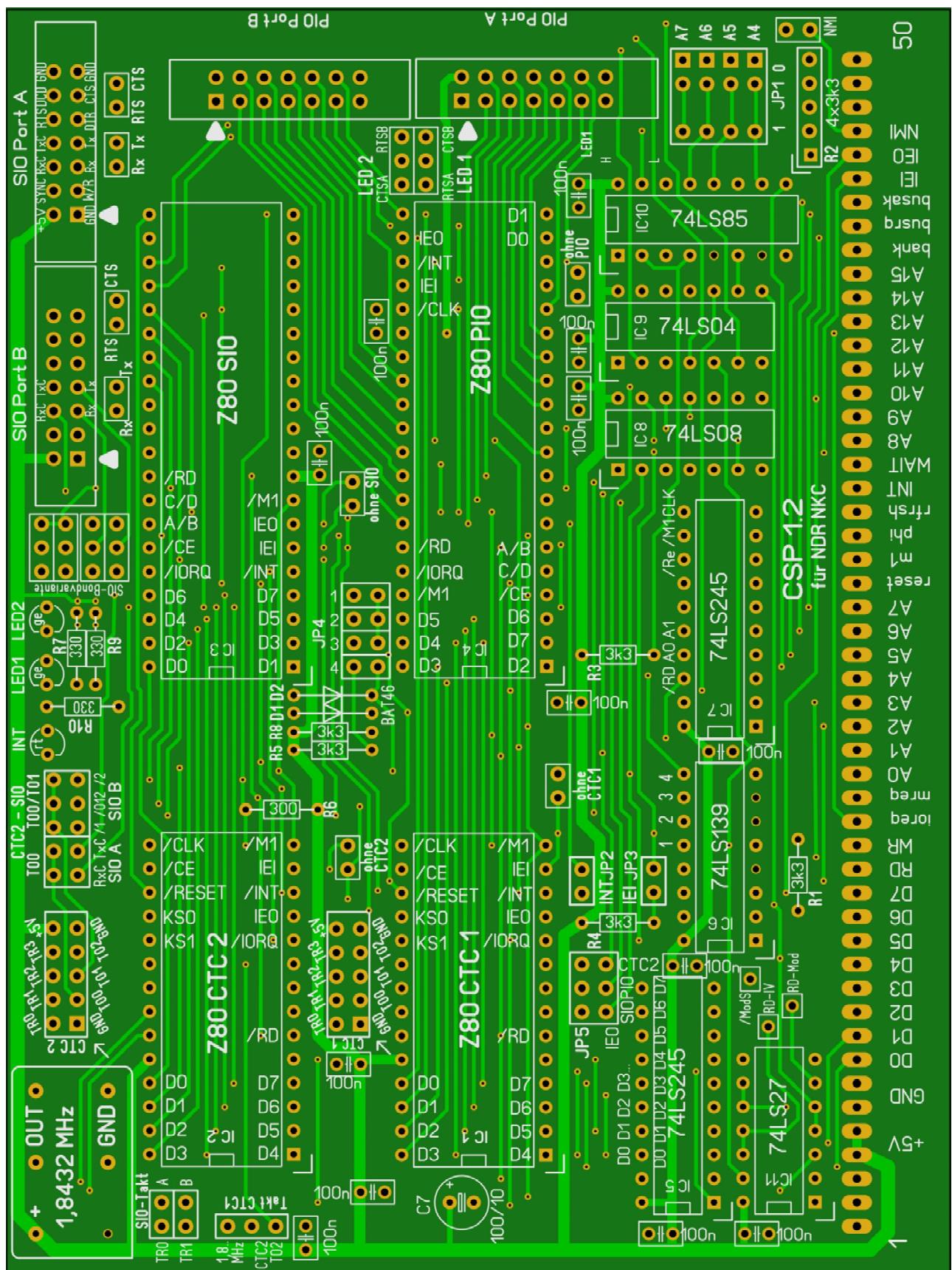
Die LED's liegen über einen Vorwiderstand an GND.

## Layoutfehler:

CSP 1.1b       IC 7 Pin10 muss an Masse

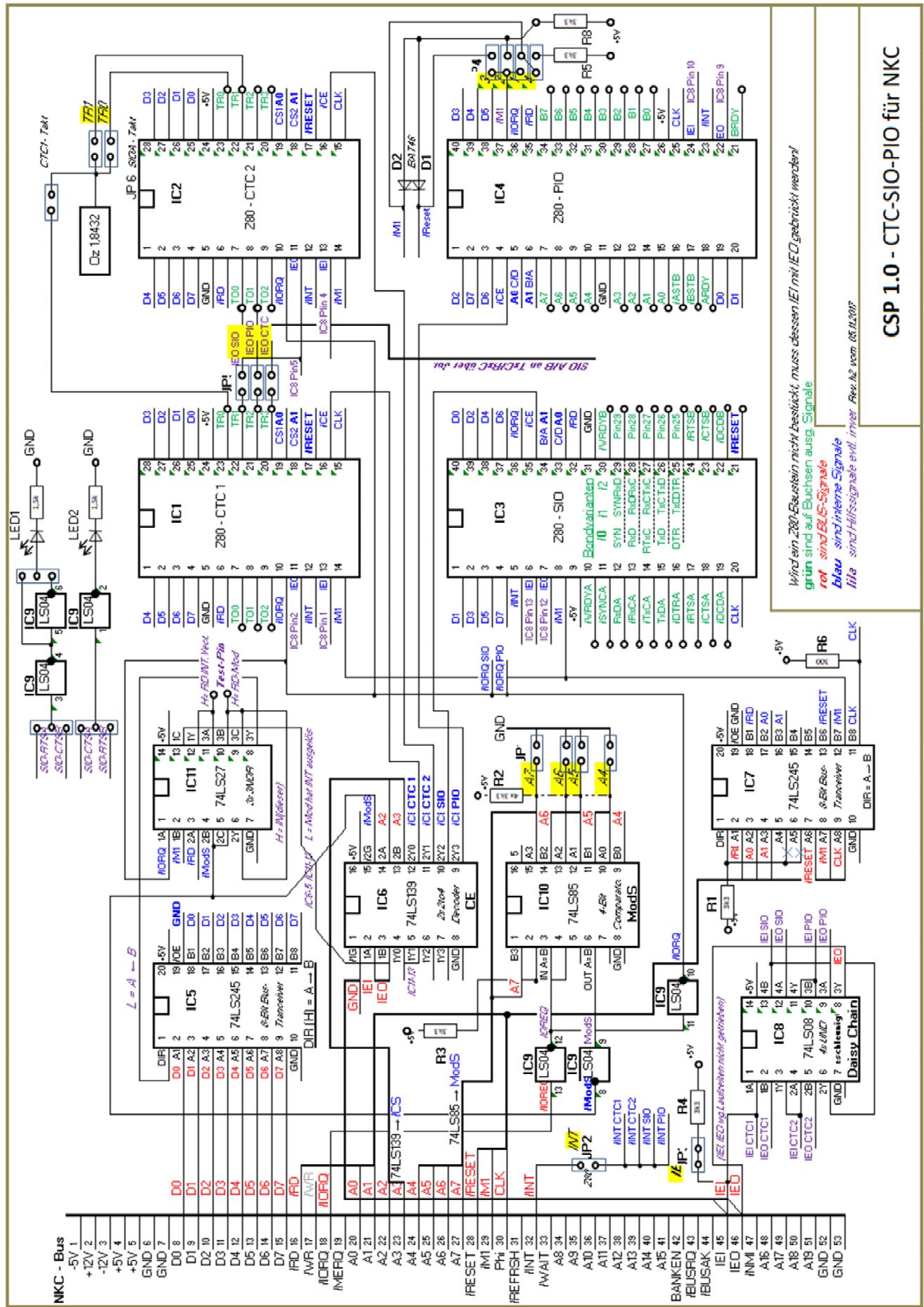
## Beschreibung CSP 1.2b

Die Leiterplatte in Fotoansicht:



## Beschreibung CSP 1.2b

## Schematischer IC-Schaltplan:



## Beschreibung CSP 1.2b

### Beispiel- und Testprogramm:

(Mit diesem Programm wurde die grundsätzliche Funktion der CSP an einer Z80 CPU im IM2 getestet.)

```
;-----  
; Interrupttest der CSP 1.0 f auf HEXIO      DL2LCE'17  
;-----  
;  
; Jumperstellung wie Foto erste Seite dieser Schrift  
;  
; Anzeige 1,2 Stelle zeigt den Interrupt der CTC1  
; Anzeige 3,4 Stelle zeigt den Interrupt der CTC2  
; Anzeige 5,6 Stelle zeigt den Wert eines CTC1 Kanals der  
;           von der SIO Port B ausgesendet  
;           und vom Port B wieder empfangen wurde  
; Am Port A der PIO wird zugleich der Zähler der CTC1 ausgegeben  
;  
  
CTC1-0 .equ 0B0h  
CTC1-1 .equ 0B2h  
CTC1-2 .equ 0B1h  
CTC1-3 .equ 0B3h  
  
CTC2-0 .equ 0B4h  
CTC2-1 .equ 0B6h  
CTC2-2 .equ 0B5h  
CTC2-3 .equ 0B7h  
  
SIOAD .equ 0B8h  
SIOAC .equ 0B9h  
SIOBD .equ 0BAh  
SIOBC .equ 0BBh  
  
PIOAD .equ 0BCh  
PIOAC .equ 0BDh  
PIOBD .equ 0BEh  
PIOBC .equ 0BFh  
  
PRINTA .EQU 0018h  
;Anzeige Akku im HEXMON A --> (IX)+(IX+1)  
  
.ORG 9000h  
;  
LD A,90h  
LD I,A  
IM 2  
  
CTCINI:  
;----- CTC1 -----  
LD A,0B0h      ;xxxx x000  
OUT (CTC1-0),A ;INT-Vektor Kanal 0  
  
;Kan.0: DI, Zeit, VT256, neg, start, ZK folgt, warte, 1  
LD A,27h      ;0010 1111  
OUT (CTC1-0),A  
  
LD A,0FAh      ;Zeitkonstante  
OUT (CTC1-0),A ;Kan.0: FAh= 1ms bei 4MHz  
  
;Kan.1: DI, Zähl, x, neg, x, ZK folgt, warte, 1  
LD A,47h      ;0100 0111  
OUT (CTC1-1),A  
  
LD A,64h      ;Zeitkonstante  
OUT (CTC1-1),A ;65h= x100 = 0,1s bei 4MHz  
  
;Kan.2: EI, Zähl, x, neg, x, ZK folgt, warte, 1  
LD A,0C7h      ;1100 0111 ;INT z.Anz. CTC1-3
```

## Beschreibung CSP 1.2b

```

OUT (CTC1-2),A      ;C7= EI / 47= DI #####
LD A,0Ah            ;Zeitkonstante
OUT (CTC1-2),A      ;Kan.2: 0Ah= x10 = 1s bei 4MHz

;Kan.3: DI, Zähl, x, neg, x, ZK folgt, warte, 1
LD A,047h           ;0100 0111
OUT (CTC1-3),A
LD A,0Ah            ;Zeitkonstante
OUT (CTC1-3),A      ;Kan.3: 3Ch= x60 = 1min b.4MHz

----- CTC2 -----
LD A,0B8h           ;xxxxx x000
OUT (CTC2-0),A      ;INT-Vektor Kanal 0

;Kan.0: DI, Zeit, VT256, neg, start, ZK folgt, warte, 1
LD A,07h             ;0010 1111 27=VT256 / 07=VT16
OUT (CTC2-0),A      ;#####
LD A,0FAh           ;Zeitkonstante
OUT (CTC2-0),A      ;Kan.0: FAh= 16ms bei 4MHz

;Kan.1: DI, Zähl, x, neg, x, ZK folgt, warte, 1
LD A,47h             ;0100 0111
OUT (CTC2-1),A
LD A,64h             ;Zeitkonstante
OUT (CTC2-1),A      ;Kan.1: 65h= x100 = 1,6s b.4MHz

;Kan.2: EI, Zähl, x, neg, x, ZK folgt, warte, 1
LD A,0C7h           ;1100 0111 ;INT z.Anz. CTC2-3
OUT (CTC2-2),A
LD A,0Ah            ;Zeitkonstante
OUT (CTC2-2),A      ;Kan.2: 0Ah= x10 = 16s bei 4MHz

;Kan.3: DI, Zähl, x, neg, x, ZK folgt, warte, 1
LD A,047h           ;0100 0111
OUT (CTC2-3),A
LD A,0Ah            ;Zeitkonstante
OUT (CTC2-3),A      ;Kan.3: 02h= x2 = 32sec b.4MHz

----- SIO B -----
LD A,18h             ;Reset SIO
OUT (SIOBC),A
;-----
LD A,02h             ;WR2
OUT (SIOBC),A
LD A,0C0h           ;INT-Vector
OUT (SIOBC),A
;-----
LD A,04h             ;WR2
OUT (SIOBC),A
LD A,44h             ;x16, 1stB, noP
OUT (SIOBC),A
;-----
LD A,03h             ;WR3
OUT (SIOBC),A
LD A,0C1h           ;Rx 8Bit, Go
OUT (SIOBC),A        ;Rx-Steuerung
;-----
LD A,05h             ;WR5
OUT (SIOBC),A
LD A,68h             ;Tx 8Bit, Go
OUT (SIOBC),A        ;Tx-Steuerung
;-----
LD A,01h             ;WR1
OUT (SIOBC),A
LD A,18h             ;INT b.jed.Zeichen
OUT (SIOBC),A
;-----
EI
RET

```

## Beschreibung CSP 1.2b

```
;----- PIO A -----  
; LD A,4Fh ;Betriebsart Byteausgabe  
; OUT (PIOAC),A  
  
; LD A,07h ;Interrupt-Kontrollwert = DI  
; OUT (PIOAC),A  
  
;----- PIO B -----  
; LD A,0C0h ;Interrupt-Vektor  
; OUT (PIOBC),A  
  
; LD A,4Fh ;Betriebsart Byteeingabe  
; OUT (PIOBC),A  
  
; LD A,17h ;Interrupt-Kontrollwert = EI  
; OUT (PIOBC),A  
  
.ORG 90B0h  
;  
;  
;Interrupt-Vector-Area  
A90b0:  
CTC1: .DEFW 0 ;INT-Adresse CTC1 Kanal 0  
      .DEFW 0 ;INT-Adresse CTC1 Kanal 1  
      .DEFW INT1 ;INT-Adresse CTC1 Kanal 2  
      .DEFW 0 ;INT-Adresse CTC1 Kanal 3  
A90b8:  
CTC2: .DEFW 0 ;INT-Adresse CTC2 Kanal 0  
      .DEFW 0 ;INT-Adresse CTC2 Kanal 1  
      .DEFW INT2 ;INT-Adresse CTC2 Kanal 2  
      .DEFW 0 ;INT-Adresse CTC2 Kanal 3  
A90c0:  
SIOB: .DEFW INT4 ;INT-Adresse SIOB Rx  
      .DEFW 0  
      .DEFW 0  
      .DEFW 0  
  
;  
;  
;CTC-Kanäle 2 und 3 OUT() manuell 41h = DI / C1h = EI  
;  
  
.ORG 90D0h  
;  
;  
;Interrupt-Routinen (für andere Betriebssysteme anpassen)  
  
INT1: PUSH AF  
      PUSH IX  
      LD IX, 8000h ;1 u. 2. Stelle  
      IN A, (CTC1-3) ;Anzeige CTC1-3  
      JR INT1  
  
INT2: PUSH AF  
      PUSH IX  
      LD IX, 8002h ;2. u. 3. Stelle  
      IN A, (CTC2-3) ;Anzeige CTC2-3  
      OUT (SIOBD),A ;auf Serielle Ausgabe  
  
INTE: CALL PRINTA ;Anzeige laden  
      POP IX  
      POP AF  
      EI  
      RETI  
  
INT4: PUSH AF  
      PUSH IX  
      LD IX, 8004h ;4. u. 5. Stelle  
      IN A, (SIOBD) ;Anzeige PIOB In  
      JR INT1
```