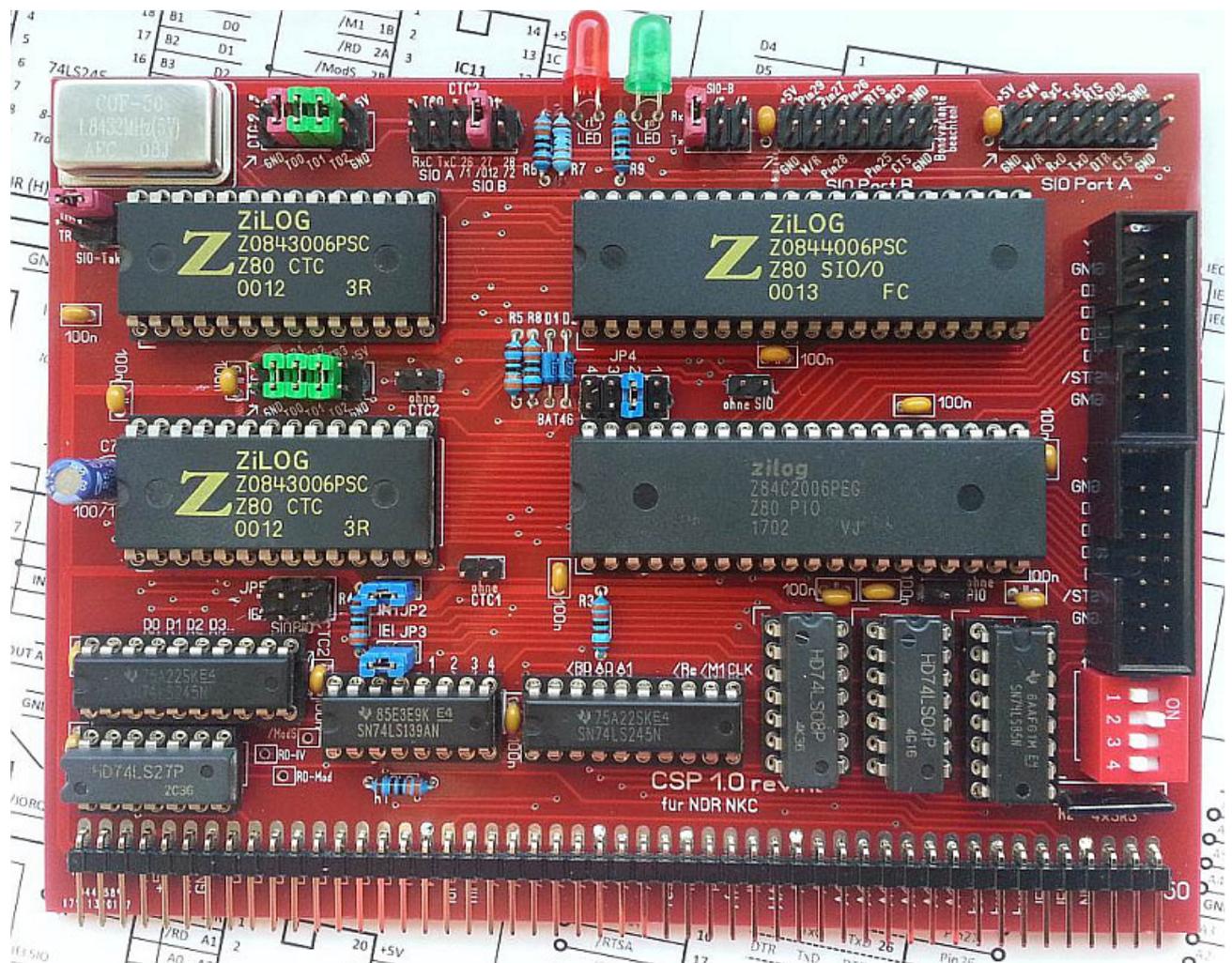


CSP

Counter-Seriell-Parallel V 1.0



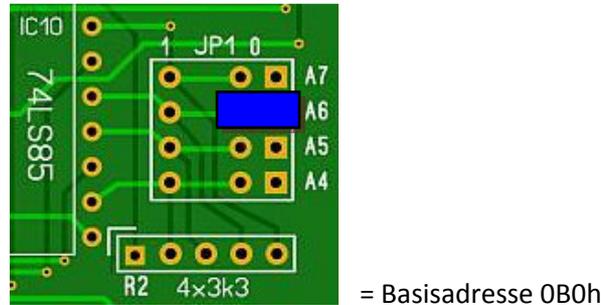
Im Bild:

- blaue Jumper für grunds. Funktion
- rote Jumper für die Sio-Funktionen
- grüne Jumper für die Kaskadierung der CTC-Kanäle

Beschreibung CSP 1.0

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.

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 (**I**nterrupt**E**nable**I**n) und einen **IEO**-Ausgang (**I**nterrupt**E**nable**O**ut). 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.

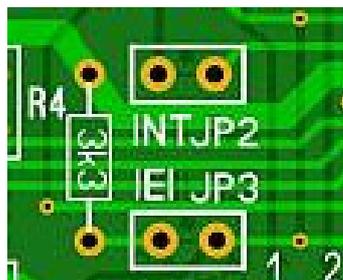
Wenn das CSP-Modul das einzige oder höherwertigste Modul ist, welches Interrupt auslösen darf, muss das Bussignal **IEI** für den Steckplatz des Modules mit einem 3k3 Widerstand auf H-Potential gezogen werden. Damit hat jedes Modul, welches auf diesen Steckplatz gesteckt wird, automatisch die höchste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

Alternativ kann auch der Jumper **J3** gesetzt werden. Dann hat das Modul, egal wo es auf dem Bus eingesteckt 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 Moduls auf H-Potential zu ziehen

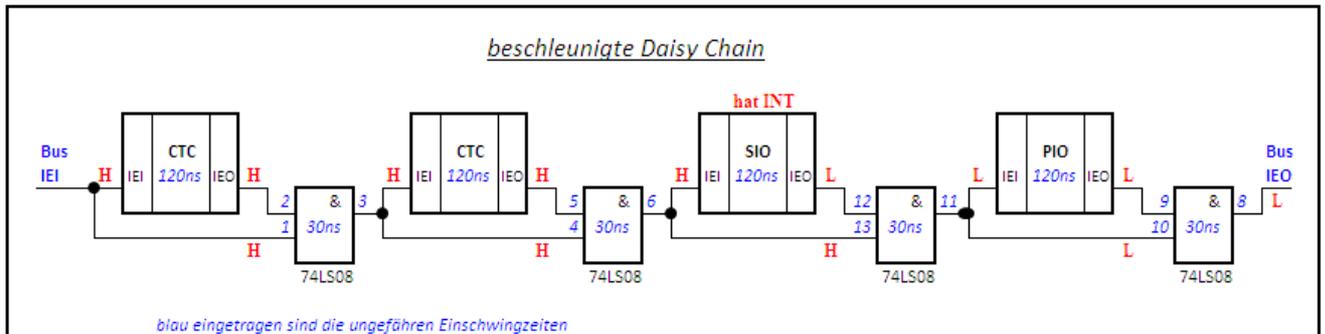
JP2 schließen um dem Modul überhaupt das Auslösen von Interrupts zu ermöglichen



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

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. 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 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

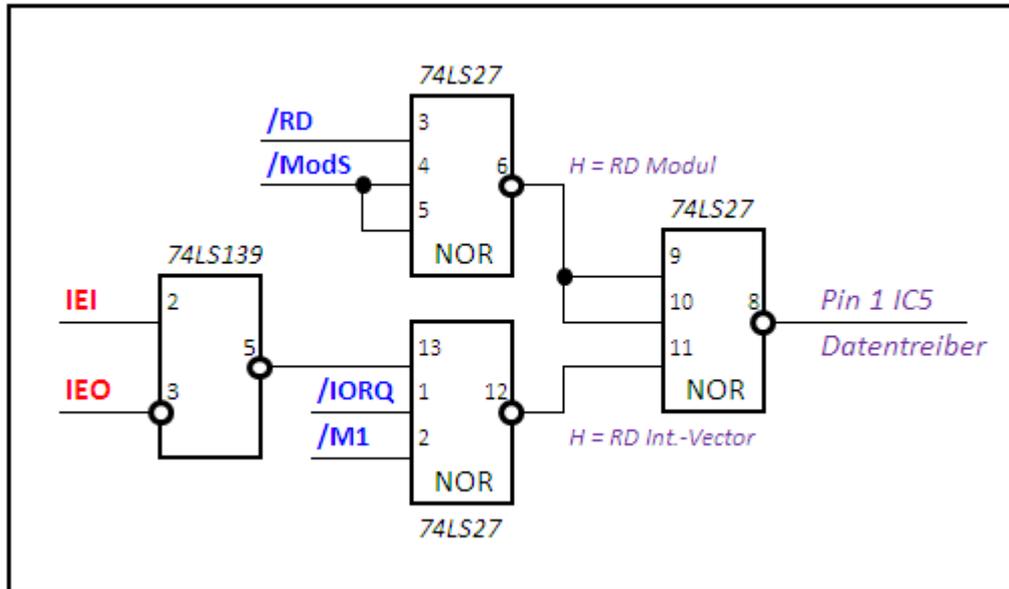
Im Interrupt-Mode IM2 gibt es zwei Besonderheiten:

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 seine Interruptbearbeitung beendet ist - liest. Das ist jedoch nur möglich, wenn der Datenbus ständig an die Bausteine durchgeleitet wird und diese den Bus „mitlesen“ können, auch wenn diese nicht angesprochen sind.
2. Die Z80 CPU „fragt“ nach einer Interruptanmeldung „Wer war das?“. Das erfolgt mit dem zeitgleichen aktivieren der Steuerleitungen **/IORQ** und **/M1** (**/RD** ist dabei inaktiv!). Daraufhin sendet der auslösende Baustein den zuvor für diesen Fall im Baustein programmierten Interruptvektor aus. Dieser Interruptvektor bildet zusammen mit dem I-Register der Z80 CPU einen Adresszeiger in eine Tabelle mit Interruptroutinen-Einsprungadressen.

Beschreibung CSP 1.0

Der Datenbustreiber muss 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.

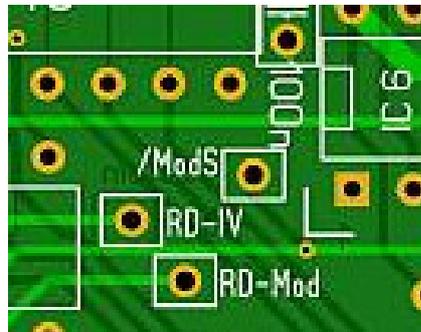
Beide Fälle werden durch die folgende Schaltung gewährleistet:



Ein L Potential am Pin1 des Datenbustreibers (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 Datenbustreibers drei Kontroll-Pins angebracht.

- /ModS = L** wenn Basisadresse angesprochen wird
- RD-IV = H** wenn vom Modul der Interruptvektor gelesen wird
- RD-Mod = H** wenn mit einem IN-Befehl vom Modul gelesen wird

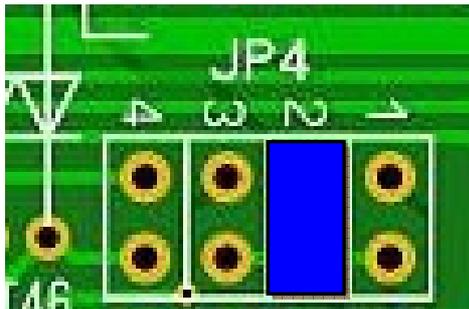


Die Kontroll-Pins kann natürlich jeder nach seinem Gusto ausführen oder ganz weg lassen. Denkbar ist, hierfür jeweils einen ausgesägten Pin einer Präzisions IC-Fassung zu verwenden.

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 (bleiben auf H).

Das bedeutet, dass die PIO am Pin **/M1** in einem Z80-System die Bussignale **/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.



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

Die Jumperstellungen 1,3 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.

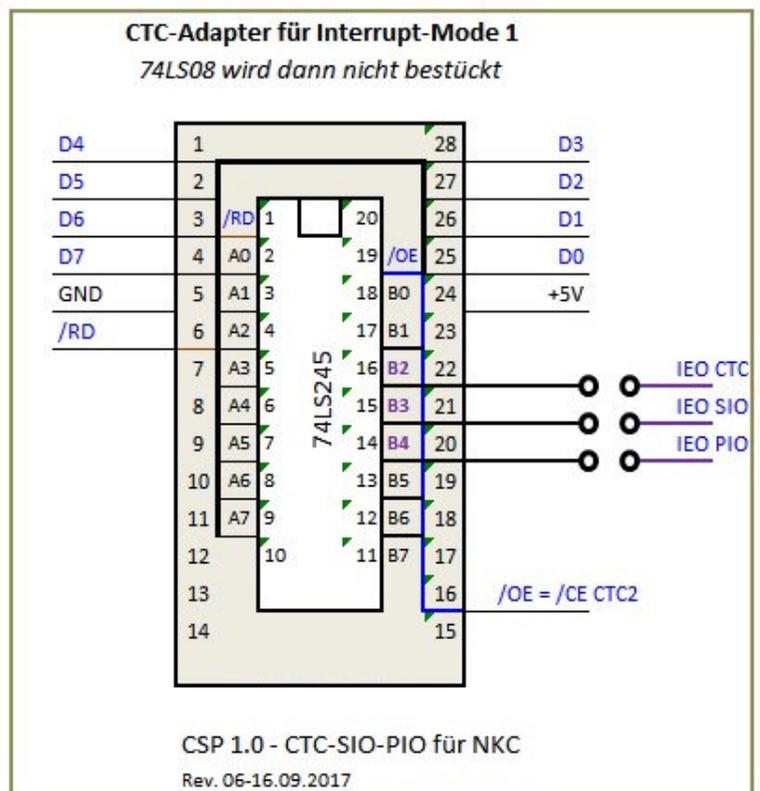
Andere Interrupt-Modi

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

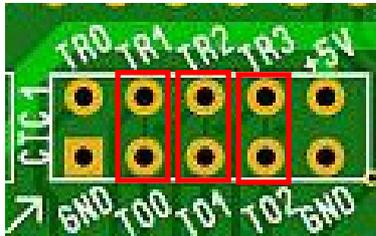


Mit diesem können die **IEO**-Ausgänge von CTC2, SIO und PIO auf die Pins 20,21,22 des CTC1-Steckplatzes gelegt werden. Ein 74LS245 kann dann z.B. unter Ausnutzung des **/CE**-Signals für CTC1 (IO-Adresse 0B0h) zum Auslesen der **IEO**-Signale der Bausteine Verwendung finden. Der Jumper „ohne CTC1“ ist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

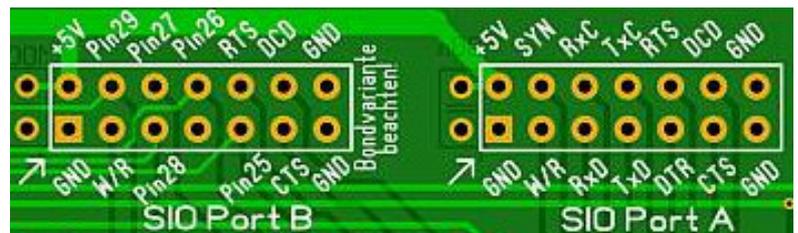


Brückt man **TO-0** mit **TR-1**, **TO-1** mit **TR-2** und **TO-2** mit **TR-3**, so sind die Kanäle 0, 1, 2 und 3 miteinander kaskadiert. D.h. der jeweilige Ausgang TO (Takt-Out) wurde mit dem folgenden TR (Trigger)-Eingang verbunden. Damit lassen sich sehr langsame Takte, bis in den Minutenbereich,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

Besonderheiten SIO

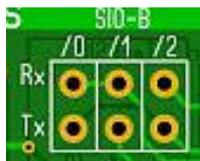
Die SIO wird im Bereich der Port B Pins in unterschiedlichen Bondvarianten angeboten.

Dem wurde beim Layout des Moduls Rechnung getragen, indem beim Port B nur die Pin-Nummern angeboten werden. Der Nutzer muss sich dann selbst informieren, welche Pins benötigt werden.



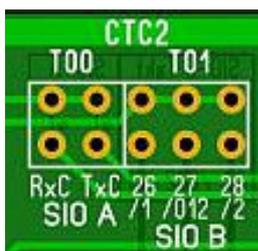
(In CSP 1.1, falls es die gibt, werden beide Ports per Jumper identisch beschaltet.)

Für einen modulinternen Test, bei dem Tx-Daten direkt mit den Rx-Daten des gleichen Kanals gebrückt werden, ist ein weiterer Jumper (**SIO-B**) vorgesehen.



Hierfür sind für die einzelnen Bondvarianten die Signale RxD und TxD des Port B herausgeführt, so dass für die entsprechende Bondvariante mit einem Jumper Tx und Rx-Daten verbunden werden können. Da in der SIO der Sende- und Empfangskanal grundsätzlich getrennt liegen, kann damit innerhalb eines Ports der Datentransfer getestet werden (Siehe Testprogramm am Ende des Scripts und Jumperstellung der bestückten Platine auf der ersten Seite).

Mit dem Jumper **CTC2 SIOA SIOB** kann man den mit der CTC generierten Baud-Takt (**TO0** und **TO1**) nach



Belieben an die SIO anlegen. Den Jumper **TO0** mit **RxC** und **TxC** der SIO A zu verbinden erklärt sich sicher von selbst. Bei der SIO B wird es etwas komplizierter, da die SIO in unterschiedlichen Bondvarianten angeboten wird.

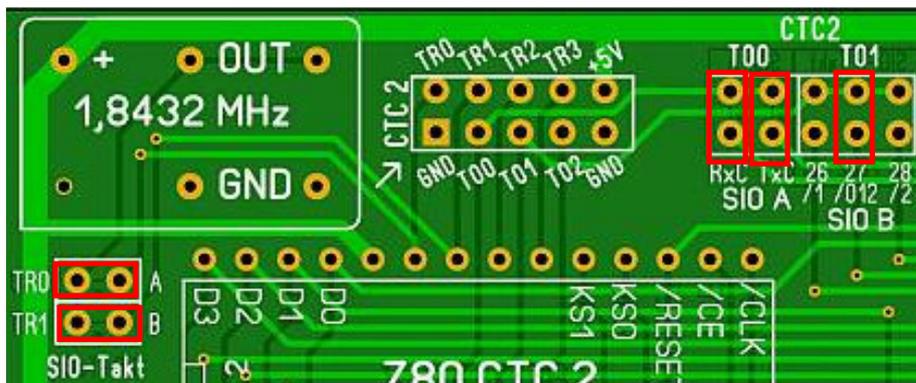
Bei der SIO/0 ist **TO1** nur mit Pin **27** zu brücken,
bei der SIO/1 ist **TO1** mit Pin **26** und Pin **27** zu brücken und
bei der SIO/2 ist **TO1** mit Pin **27** und **28** zu brücken.

(getestet wurde nur mit der SIO/0 Variante)

Eine Besonderheit stellt im Detail die Takterzeugung für die Datenübertragung der SIO:

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:

Takt	CTC Kanal		Baud
	Zähler		
1,8432	32		57600,00
1,8432	48		38400,00
1,8432	64		28800,00
1,8432	96		19200,00
1,8432	192		9600,00
1,8432	256		7200,00



Im vorstehenden Bild sind die Jumper eingezeichnet, die notwendig sind, wenn CTC2-Kanal0 den TxRx-Takt für SIO Port A und CTC2-Kanal1 den TxRx-Takt für SIO/0-Port B nach vorstehender Tabelle generieren soll.

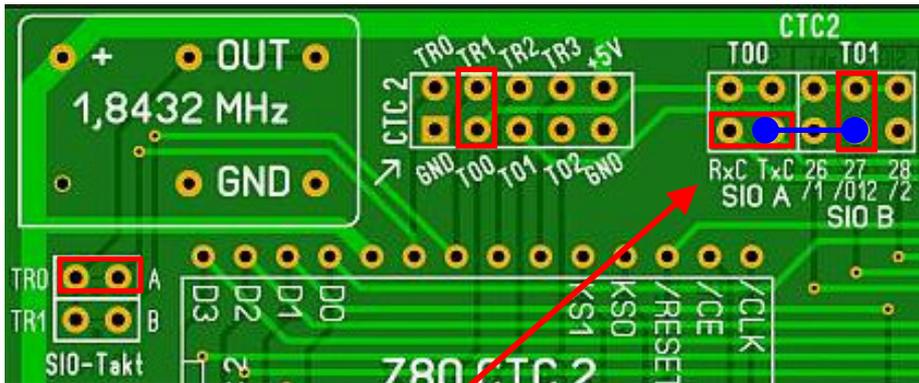
Möchte man hingegen auch niedrigere Baudraten verwenden kommt man um einen Kniff nicht umhin.

Takt	CTC Kanal		Baud
	Vorteiler	Zähler	
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
1,8432	256	24	300,00
1,8432	256	48	150,00

Man benötigt zusätzlich einen 16-er Vorteiler, den die CTC im Zählermodus leider nicht zur Verfügung stellt. Dafür muss man einen weiteren CTC Kanal einsetzen.

Beschreibung CSP 1.0

Wenn es ausreichenwürde, dass die SIO auf beiden Ports die gleiche Baudrate verwendet, könnte man die Jumper wie folgt setzen:



Dabei müssten die beiden Jumper bei „CTC2“ noch miteinander verbunden werden.

Nunmehr kann der CTC2-Kanal0 als 16-er Vorteiler programmiert werden und der CTC2-Kanal1 zur Bereitstellung einer beliebigen Baudrate nach obiger Tabelle.

Will man jedoch dennoch auf beiden Ports unterschiedliche Baudraten einsetzen wird man u.U. nicht umhin kommen, in dem Bereich kleinere Leiterzugkorrekturen vorzunehmen. Dann bliebe CTC2-Kanal0 der Vorteiler und CTC2-Kanal1 und Kanal2 getrennt zur getrennten Bereitstellung der Baudraten für SIO-Port A und B.

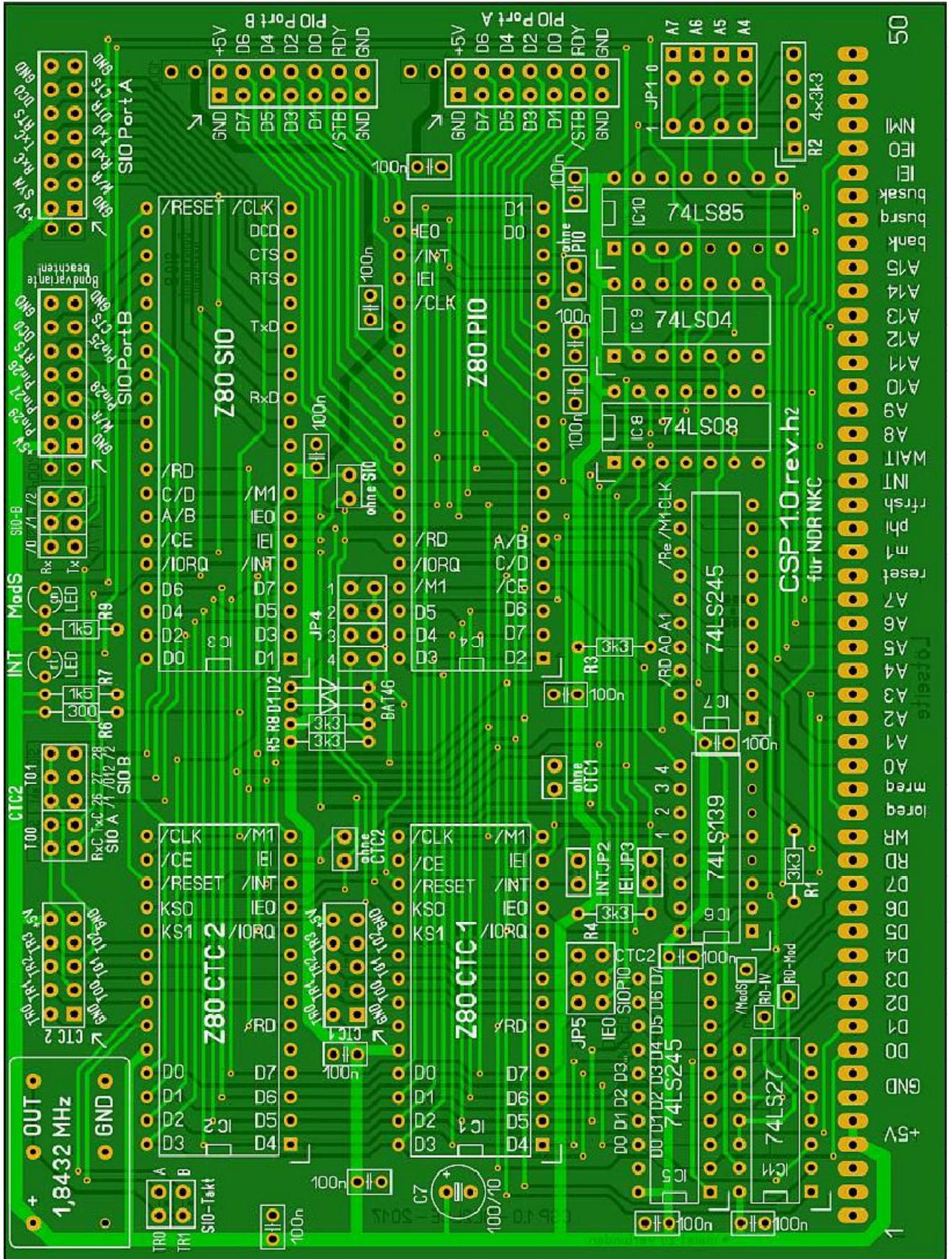
LED – Statusanzeigen

Im oberen Bereich des Moduls sind noch zwei Status-LED vorgesehen, die nach Belieben bestückt werden können.

Die LED **ModS** leuchtet wenn das Modul über seine Basisadresse angesprochen wird. Dieser Impuls ist jedoch so kurz, dass hier keine merkliche Leuchterscheinung zu verzeichnen ist. Diese LED kann man also zunächst getrost weg lassen.

Die LED - **INT** - leuchtet wenn das Modul einen Interrupt ausgelöst hat (IEI = H und IEO = L) und zwar solange, bis die Interruptroutine abgearbeitet ist. Hier sollte man abweichend vom Bestückungsplan einen 600 Ohm Widerstand verwenden. Damit kann man das Arbeiten noch ausreichend erkennen. Wichtig ist diese LED um erkennen zu können, dass der Interrupt auch arbeitet. Wird z.B. der Interruptvector nicht an die CPU gesendet, die CPU die Interruptroutine nicht findet oder am Ender der Interruptroutine der Interrupt nicht wieder frei gegeben wird (Software), leuchtet die LED dauerhaft, was einen Fehler anzeigt.

Die Leiterplatte in Fotoansicht:



Beschreibung CSP 1.0

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 1.0

```
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         ;xxxx 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 1.0

```
;----- 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 INTE

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 INTE
```