

# FSK-Modem für die EPP-Schnittstelle

*Jürgen Hasch, DG1SCR @ DBØGPP, Meisenstr. 23, 73066 Uhingen*

## **1. Motivation**

Durch die Einführung eines 200kHz breiten Duplexkanals für Digitalanwendungen im 70-cm-Band ergibt sich ein wachsender Bedarf an Packet-Radio-Komponenten für Datenraten deutlich größer 9600Bit/s, vor allem auf der Benutzerseite. Seit kurzem sind Datentransceiver mit einer entsprechend großen Bandbreite und einer S/E-Umschaltzeit  $< 1$  ms verfügbar [1], jedoch besteht noch ein gewisser Engpaß bei den benötigten Modem-Komponenten. Während auf der Digi-Seite ein RMNC gerade noch den Kanal bedienen kann, ist der Benutzer auf relativ teure Lösungen wie TNC3 oder USCC-Einsteckkarte mit zusätzlichem Modem angewiesen.

Um hier Abhilfe zu schaffen, sollte ein FSK-Modem entwickelt werden, das direkt an die Schnittstelle eines PCs angeschlossen werden kann und ohne größere Probleme Datenraten bis über 100kBit/s verarbeitet.

Der Argumentation der Autoren aus [2] folgend, bietet sich die parallele Schnittstelle des PC als preisgünstige und fast überall vorhandene Anschlußmöglichkeit an. Mit dem EPP-Standard ist auch eine für den Modem-Betrieb geeignete, mit einer Datendurchsatzrate von  $> 500$  kB/s von der möglichen Übertragungsbandbreite her ausreichende Schnittstelle vorhanden. Mittlerweile sind außerdem PC-Einsteckkarten mit 2 oder mehr EPP-Schnittstellen verfügbar, so daß ältere PCs aufgerüstet werden können und auch ein eventuell vorhandener Drucker weiter betrieben werden kann. Der Preis solcher Zusatzkarten ist deutlich unter 50 DM angesiedelt.

Die Frage, die sich nun stellt, ist: Wie läßt sich ein solches Modem als kompakte Baugruppe und dazu möglichst preisgünstig realisieren?

Ein Blick in verschiedene Fachzeitschriften, sowie [3] und [4] zeigen einen erfolgversprechenden Weg: FPGAs!

## **2. Was ist ein FPGA?**

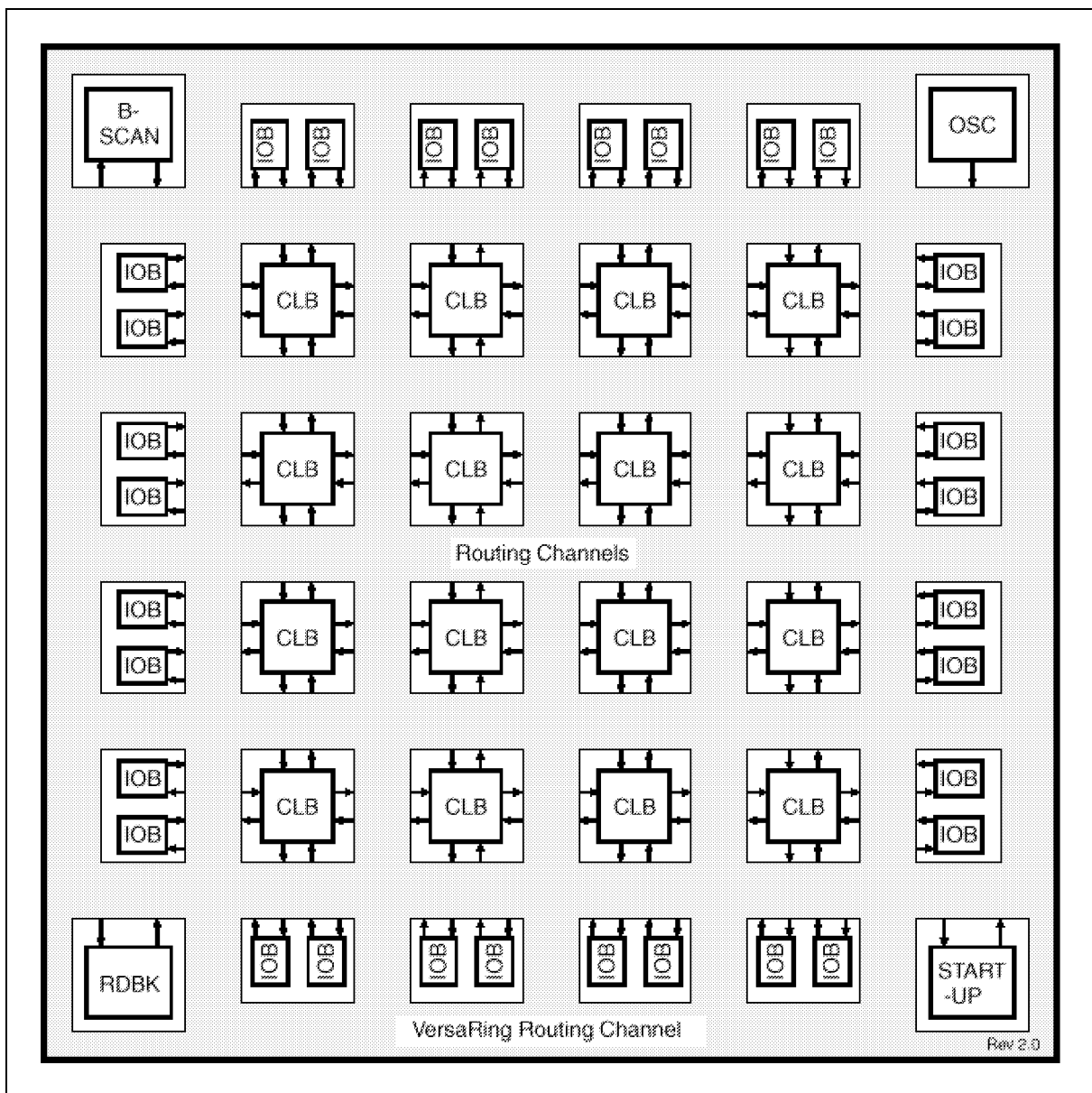
Die Abkürzung FPGA steht für "Field Programmable Gate Array", grob übersetzt "in der Anwendung programmierbare Ansammlung von Logikgattern". Ein solcher Baustein ist ein digitales Bauelement, d.h. damit können nur digitale Schaltungen realisiert werden. Es gibt eine Reihe von Herstellern derartiger Bausteine; die beiden namhaftesten Vertreter sind Xilinx und Altera. Für das Modem wurde ein Baustein der Spartan-Familie des Herstellers Xilinx verwendet. Hauptgrund dafür waren einmal das in [4] beschriebene YAM-Modem, sowie das kürzlich eingeführte, für den Amateur-Geldbeutel erschwingliche Entwicklungspaket von Xilinx. Andere Hersteller bieten ähnliche Pakete an, wobei zumindest die einfacheren Vertreter der jeweiligen Bausteinserie unterstützt werden.

Kauft man ein FPGA vom Hersteller, so ist es unprogrammiert, d.h. es sind Bausteine von der Stange, die erst vom Schaltungsentwickler für eine spezifische Aufgabe eingesetzt werden.

Dies ist natürlich auch ein wichtiger Punkt, der für den Einsatz in einem Amateurfunk-Projekt spricht.

Betrachtet man die Technologien zur Programmierung verschiedener FPGA-Familien, so kann man eine weitere Einteilung vornehmen. Einige Familien, beispielsweise die Cypress pASIC380, verwenden die Antifuse-Technologie. Dabei werden die Bausteine in einem speziellen Programmiergerät programmiert, eine nachträgliche Änderung der Programmierung ist dann nicht mehr möglich.

Eine andere Technologie stellt die Verwendung von Flash-Speicherzellen, ähnlich denen eines Flash-EPROMs, dar. Dabei ist ebenfalls ein spezielles Programmiergerät oder eine Programmerschaltung notwendig, jedoch kann hier die Programmierung nachträglich auch wieder geändert werden.



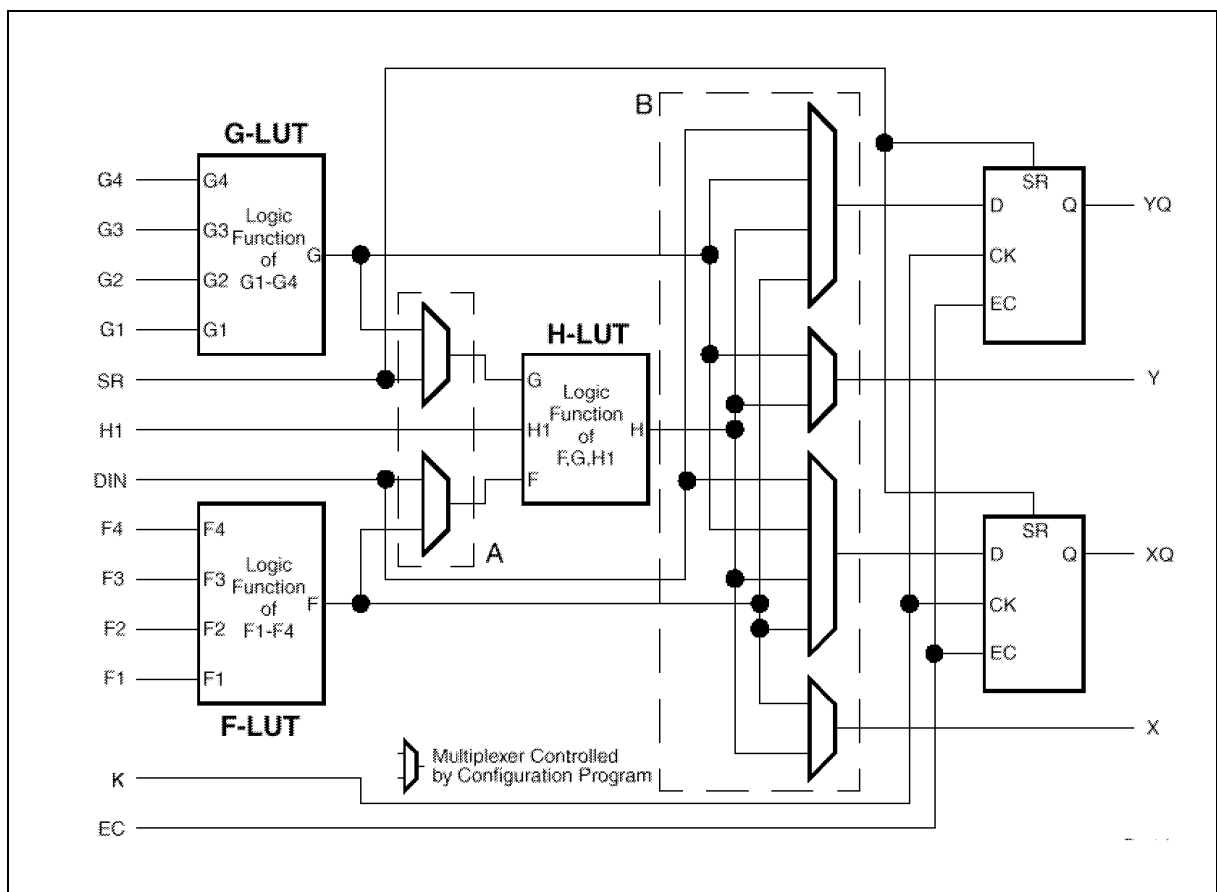
**Bild 1** Struktur eines FPGA

Die dritte wichtige Variante stellen SRAM-basierte FPGAs dar. Diese müssen jedesmal beim Anlegen der Versorgungsspannung aus einem externen Speicher (z.B. einem kleinen EEPROM) neu programmiert werden. Für unsere Anwendung bietet diese Möglichkeit viele Vorteile. Sorgt man dafür, daß die Initialisierung des FPGA auch vom PC aus vorgenommen werden kann, dann ist eine Initialisierung durch den Modem-Treiber des PCs möglich. Somit sind Treiber-Updates immer passend zur aktuellen Programmierung des FPGAs, und Verbesserungen im Modem können durch einfaches Software-Update des Treibers vorgenommen werden.

Doch jetzt zur Funktionsweise eines FPGAs:

Ein FPGA enthält eine große Anzahl an grundlegenden Logikfunktionen (Logikzellen), die, wie in Bild 1 gezeigt, miteinander durch logische Kanäle verbunden werden können. Die Programmierung eines FPGAs bedeutet also einmal die Herstellung von Verbindungen (Routing) zwischen den einzelnen Logikzellen.

Eine Logikzelle selbst besteht meist aus einer oder mehreren logischen Grundfunktionen und einem Flipflop. Bei der hier eingesetzten Spartan-Familie von Xilinx besteht die Gatterlogik aus mehreren Speicherzellen (lookup table) und zwei Flipflops.



**Bild 2** Aufbau einer Logikzelle bei der Spartan-Familie

Jede Logikzelle ist einzeln für sich programmierbar. Der ausgewählte Baustein XCS10 von Xilinx enthält 466 solcher Logikzellen und erreicht damit eine Komplexizität von etwa 10000 Gattern.

### 3. Konzept

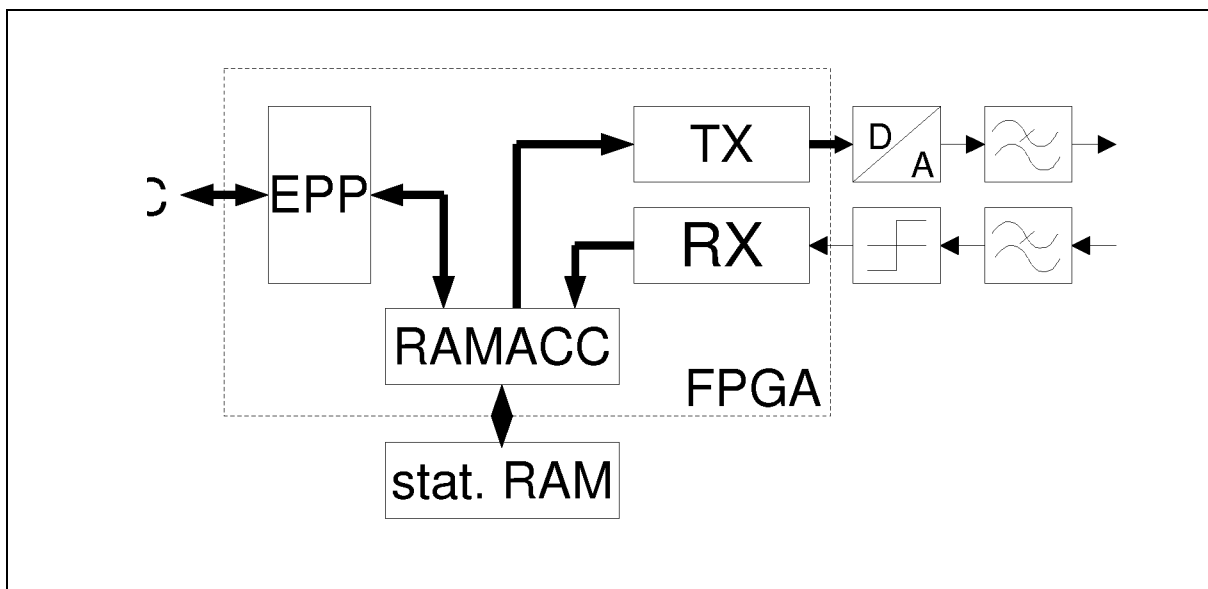
Bei einer Bitrate von 76,8kBit/s soll das Modem dem PC eine möglichst geringe Rechenleistung abverlangen, da moderne (Fenster-)Betriebssysteme schon einen Großteil der Rechenleistung benötigen.

Um die IO-Last des PCs bei Empfang niedrig zu halten, ist eine Pufferung der Empfangsdaten notwendig, möglichst sollte im Modem ein komplettes Frame dekodiert werden, um erst nach Verifizierung der CRC-Prüfsumme zum PC übertragen zu werden. Da bei FPGAs RAM-Speicherplatz eine wertvolle Resource ist, wurde ein externes 8k\*8 RAM zur Speicherung der Sende- und Empfangsdaten vorgesehen.

Der Sendeteil ist relativ einfach aufgebaut, hier erzeugt der PC den Sendedatenstrom inklusive HDLC-Kodierung und Bitstuffing. Diese Rohdaten werden zum Modem geschickt, dort zwischengespeichert und zum richtigen Zeitpunkt ausgesendet. Lediglich NRZI-Kodierer, Scrambler und ein digitales Tiefpaßfilter sind noch Bestandteil des Modems. Vorteil dieser Methode ist, daß man durch Voranstellen von HDLC-Flags ein beliebiges TX-Delay erzeugen kann, ohne den PC mit einer Zeitmessung zu beaufschlagen.

Das EPP-Interface des Modems ist sehr einfach gelöst. Ein Datenzugriff (data strobe) vom PC aus liest oder schreibt ein Byte aus oder in das RAM des Modems und setzt gleichzeitig einen Zeiger auf die nächste Adresse im Speicher. Nach 4096 Speicherzugriffen gelangt man wieder an den Ausgangspunkt zurück. Der Speicher ist in zwei Blöcke aufgeteilt, in den ersten Block schreibt der Empfänger seine Empfangsdaten, die anschließend vom PC ausgelesen werden können. Der zweite Block wird vom Sender ausgelesen und kann entsprechend vom PC aus beschrieben werden.

Bei einem Adresszugriff (address strobe) wird ein Statusregister im Modem neu gesetzt, bzw. ausgelesen. Beim Schreibzugriff kann damit die PTT aktiviert sowie die Interrupt-Leitung freigeschaltet werden. Ein Lesezugriff liefert das DCD-Signal sowie die Statusflags "Empfangsdaten vorhanden" und "Sendespeicher leer".



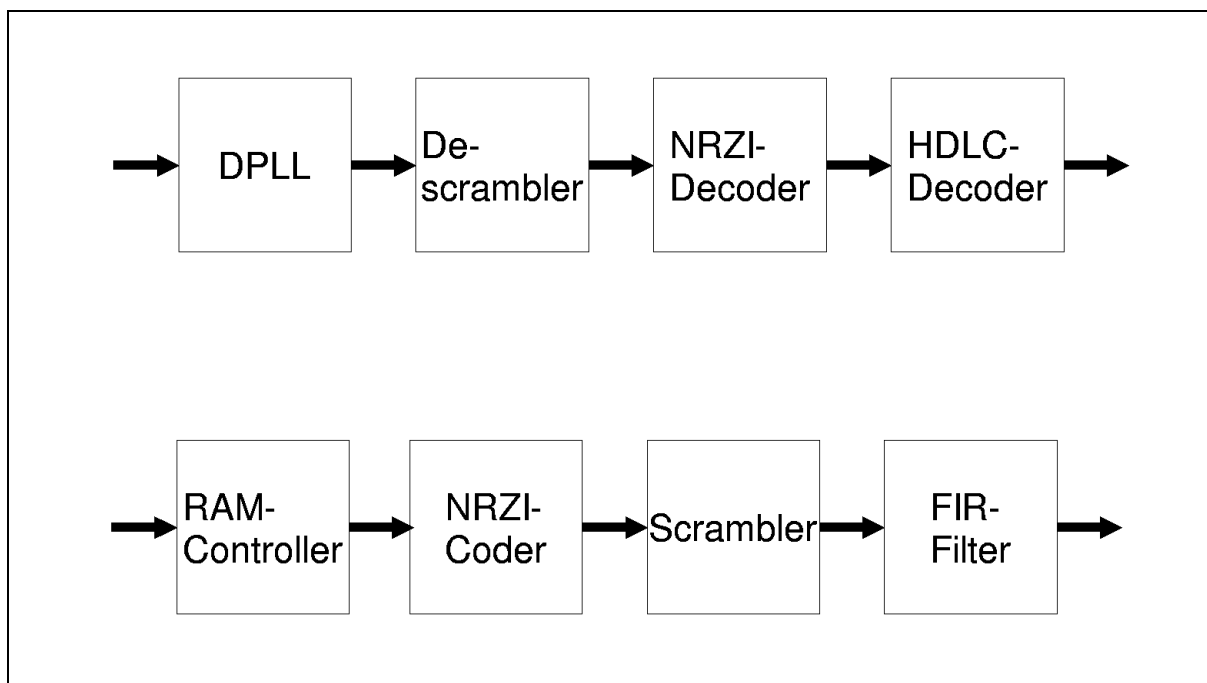
**Bild 3** Funktionsblöcke des Modems

In Bild 3 ist das Blockschaltbild des Modems gezeigt. Man erkennt die Entkopplung zwischen EPP-Interface, Sende- und Empfangsteil des Modems durch die Zwischenspeicherung der Daten in RAM.

Der Empfänger (RX) besteht aus fünf Funktionsblöcken (Bild 4):

In der DPLL geschieht die Taktrückgewinnung und DCD-Erkennung. Dazu wird der 64-fache Empfangstakt zugeführt und versucht, auf die Flanken des Datensignals zu regeln. Kommt eine Änderung des Eingangssignals nicht zum erwarteten Zeitpunkt (also möglichst genau zwischen zwei Augenöffnungen), wird der Empfangstakt entweder um  $1/64$  verzögert oder um  $1/64$  beschleunigt. Die DCD-Erkennung überprüft, ob in der Mitte der Augenöffnung eine Änderung der Empfangsdaten vorkommt. Ist diese Bedingung eine gewisse Zeit erfüllt, wird die DCD gesetzt, andernfalls wird die DCD sofort zurückgesetzt. Um ein Hängenbleiben der DCD ohne Signal zu vermeiden, kann die DCD-Erkennung durch den HDLC-Decoder zurückgesetzt werden, sobald mehr als 7 aufeinanderfolgende Einsen festgestellt wurden (was bekanntlich in gültigen Daten dank Bitstuffing nie vorkommen darf).

Die Empfangsdaten werden inklusive Bitstuffing ins RAM geschrieben, gleichzeitig werden sie ohne Bitstuffing in ein Schieberegister zur Ermittlung der CRC-Prüfsumme geschoben. Nach 8 empfangenen Bits wird der Empfangszähler um eins erhöht. Wird ein HDLC-Flag erkannt, also Anfang oder Ende eines Datenpaketes, wird der Empfangszähler entweder zurückgesetzt oder (bei einem als gültig erkannten Paket) um die Paketlänge erhöht.



**Bild 4** Funktionsblöcke des Empfängers (oben) und des Senders (unten)

Der Sender ist sehr einfach aufgebaut (Bild 4):

Der RAM-Controller liest alle 8 Bit ein neues Zeichen aus dem externen RAM in ein Schieberegister ein. Nach NRZI-Kodierung und Scrambling, wird der Datenstrom auf ein FIR-Filter zur Tiefpaßfilterung gegeben. Außerhalb des FPGAs wird der Filterausgang auf einen 8-Bit-D/A-Wandler geführt.

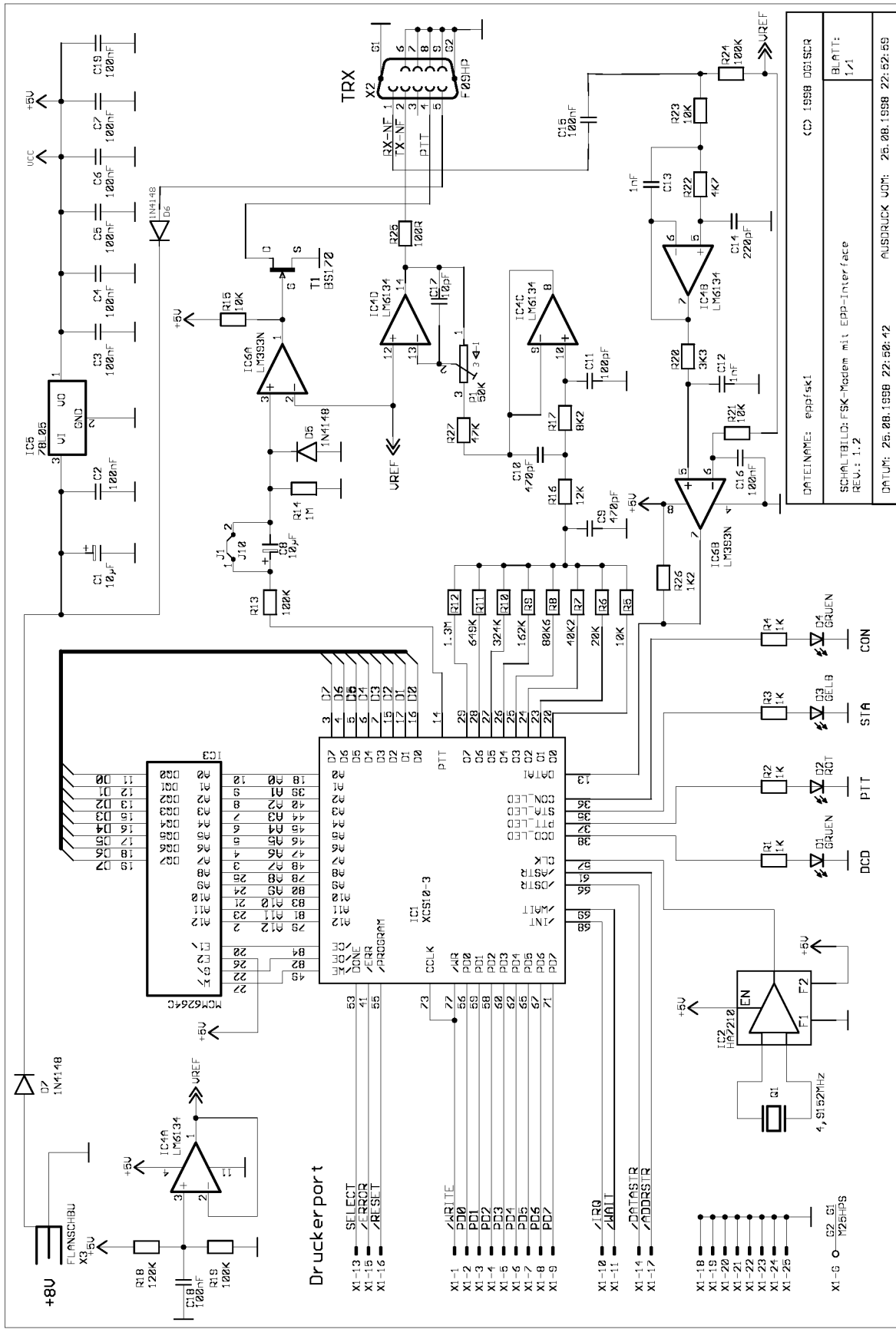


Bild 5 Schaltbild des Modems

## **4. Hardware**

### **4.1 Schaltung**

Bild 5 zeigt das Schaltbild des EPP-Modems. Zentraler Baustein der Schaltung ist natürlich das FPGA, welches direkt an die EPP-Schnittstelle des PCs angeschlossen wird. Wird auf dem PC der Modemtreiber gestartet, so wird das FPGA zuerst mittels Reset-Leitung zurückgesetzt und anschließend über die Write- und D7-Leitung der Druckerschnittstelle konfiguriert. Nach erfolgreicher Initialisierung meldet sich das FPGA, indem die Select-Leitung auf +5V gelegt wird. Da das FPGA keine eigene Oszillatorschaltung besitzt, wird als Taktquelle ein HA7210-Oszillatorbaustein von Harris mit externem 4,9152MHz Quarz verwendet.

Die aus dem internen Sendefilter kommenden Daten sind an den Ausgängen O0-O7 herausgeführt. Dort gelangen sie auf ein als D/A-Wandler beschaltetes Widerstandsnetzwerk und danach zu einem Tiefpaßfilter. Über den Trimmer P1 kann die Amplitude des Ausgangssignals eingestellt werden.

Das vom Funkgerät kommende Empfangssignal wird zuerst auf einen Impedanzwandler geführt, gefolgt von einem Tiefpaßfilter. Anschließend folgt ein Komparator, der, je nachdem, ob der empfangene Spannungswert oberhalb oder unterhalb der halben Betriebsspannung liegt, den Datenstrom zu 1 oder 0 entscheidet.

Die PTT-Leitung ist zusätzlich mit einer Watchdog-Schaltung versehen, die verhindern soll, daß ein Sender auf Dauersendung geht. Die komplette Schaltung wird über einen 5-V-Spannungsregler versorgt. Ein Betrieb ohne eigene Spannungsversorgung ist leider nicht möglich, da alle Ausgangspins des Druckerports benutzt werden, und wer möchte schon, daß, sobald Nullen übertragen werden, das Modem aussteigt ?

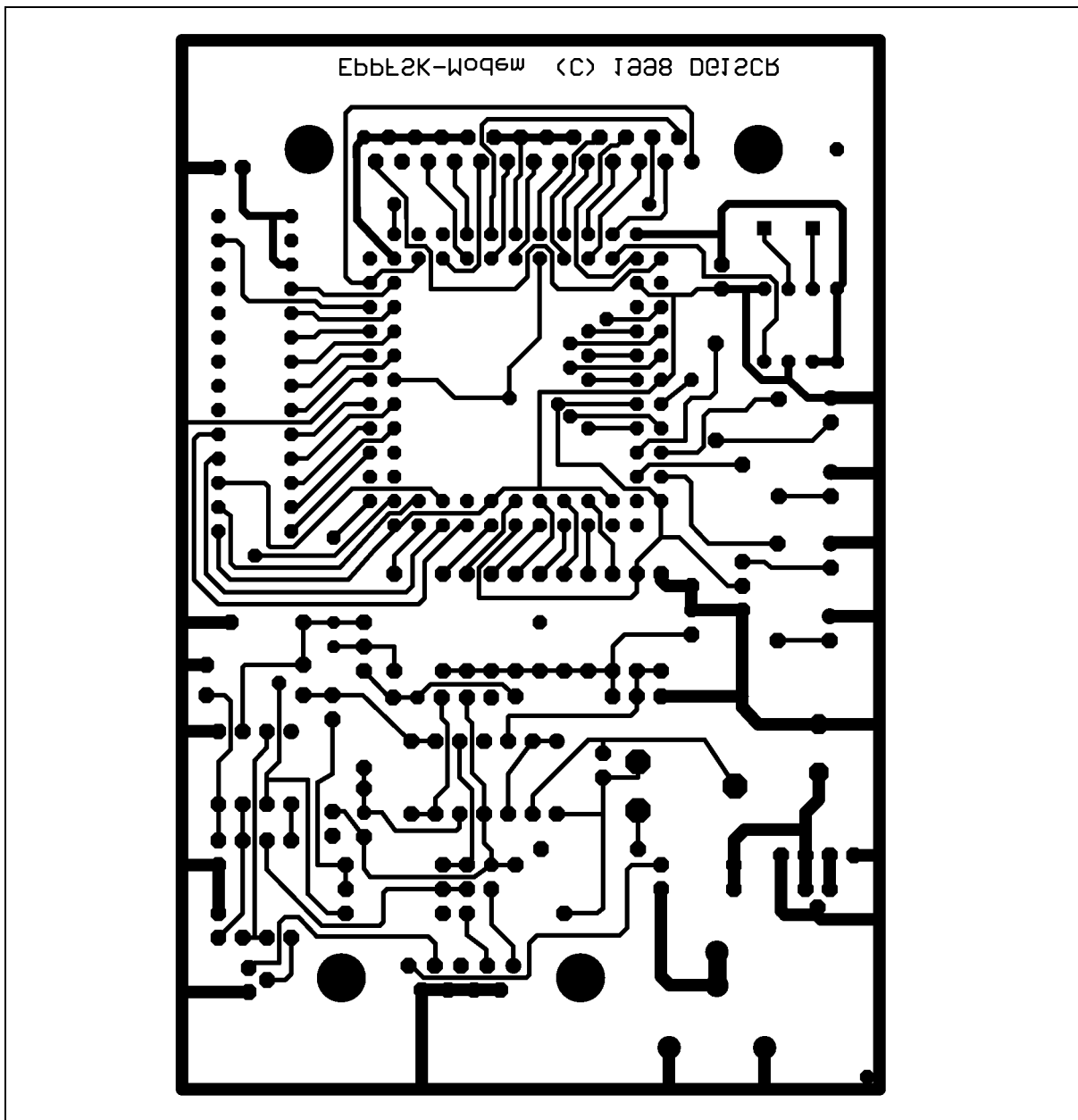
### **4.2 Aufbau**

In Bild 6, Bild 7 und Bild 8 sind Layout und Bestückungsplan des Modems abgebildet. Das Layout wurde so erstellt, daß sich das Modem auch ohne professionelle Durchkontaktierungen aufbauen läßt. Dazu sollte man die Durchkontaktierungen unter dem FPGA vor dem Einlöten des PLCC-Sockels mit Drahtstücken verlöten und die Bauteile mit Leiterbahnanschlüssen auf der Platinenoberseite entsprechend verlöten.

Die Spannungsversorgung Modems kann entweder vom Funkgerät aus über die Sub-D-Buchse des Modems geschehen oder über ein zusätzliches Steckernetzteil mit mindestens 8 V Gleichspannung.

## **5. Software**

Zur Zeit existiert ein PC/Flexnet-Treiber für das Modem. Die Inbetriebnahme geschieht, wie bei jedem anderen Modem, durch Laden des entsprechenden Treibers (hier EPPFSK). Der Treiber übernimmt auch die Initialisierung des FPGAs, so daß dies für den Benutzer völlig transparent geschieht.



**Bild 6** Layout der Lötseite des Modems

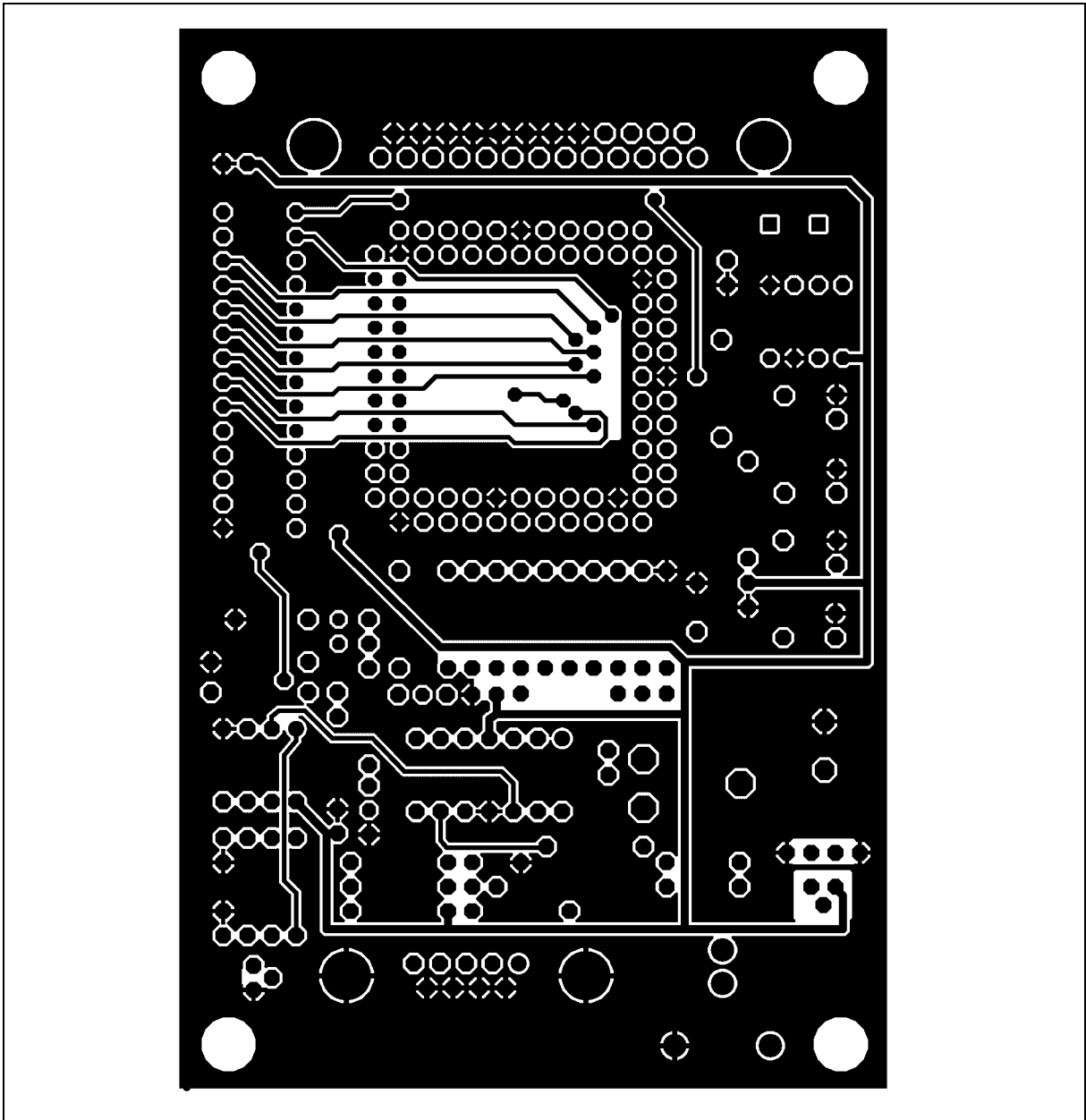
Beim Start versucht der Treiber die wichtigsten Chipsätze zu erkennen (Winbond, SMC) und die parallele Schnittstelle explizit in den EPP-Modus zu schalten. Wird eine andere Hardware verwendet, läßt sich diese Erkennung unterdrücken, IO-Adresse sowie Interrupt können dann manuell angegeben werden.

Vielen Dank an DL8AAU, DL2ZBN und DF9IC für die zahlreichen Anregungen.

## **6. Literatur**

- [1] A. Kurpiers DL8AAU und M. Liebeck DL2ZBN: Hochgeschwindigkeits-Packet-Radio - ein Transceiverkonzept für das 70cm Band. ADACOM-Magazin 10 (1997)





**Bild 7** Layout der Bestückungsseite des Modems

- [2] W.-H. Rech DF9IC et al : Ein Modemadapter für den EPP. Skriptum zur 13. Internationalen Packet-Radio Tagung.
- [3] John Wiseman: Modern Digital Design for the Radio Amateur. QEX magazine, Dezember 1997.
- [4] N. Palermo IV3NWV: Das YAM-Modem - FSK-Modem für die serielle Schnittstelle. ADACOM-Magazin 11 (1998); weitere Beschreibung bei <http://www.microlet.com/yam>

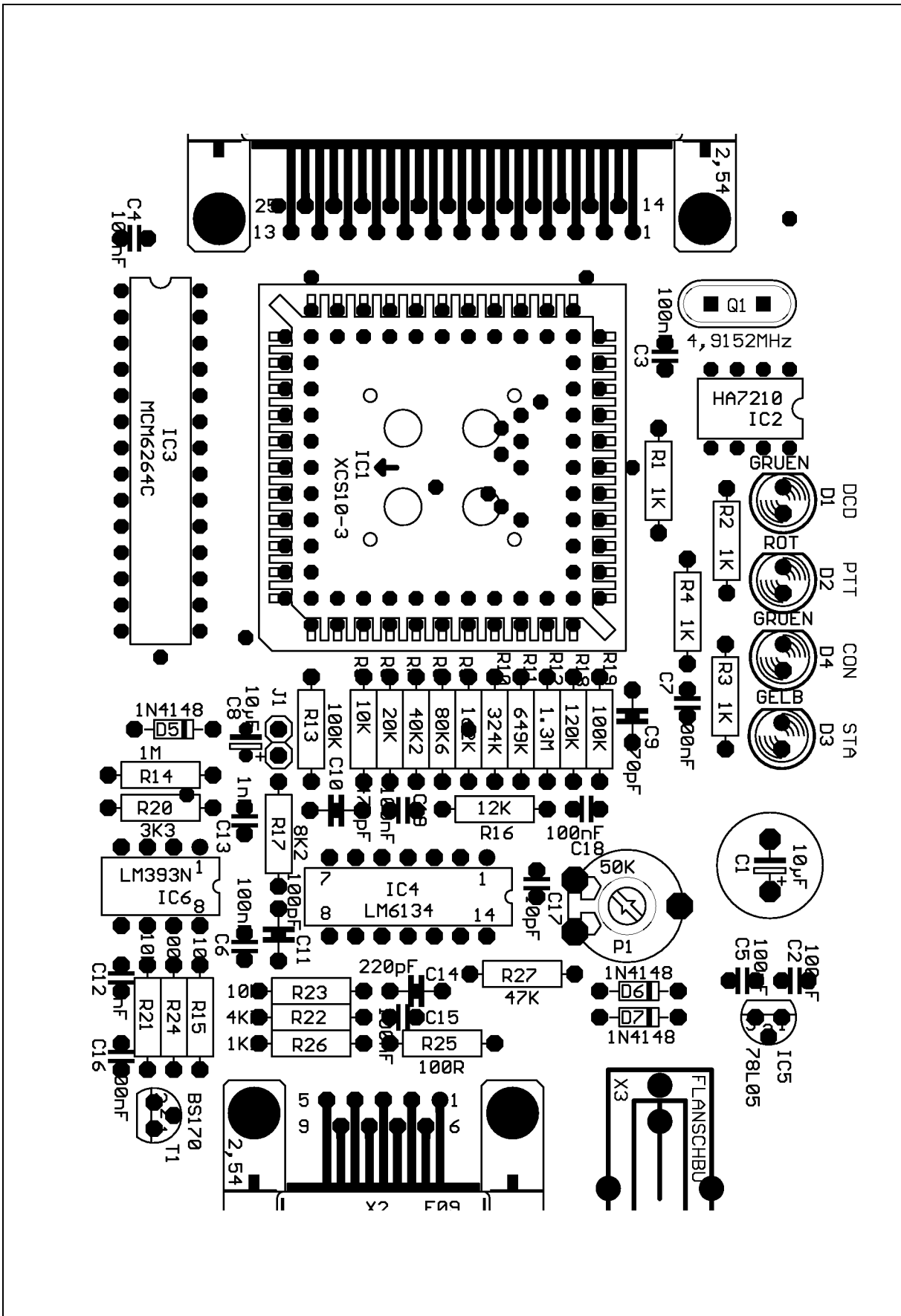


Bild 8 Bestückungsplan des Modems

C1	10µF	R1	1k
C2	100nF	R2	1k
C3	100nF	R3	1k
C4	100nF	R4	1k
C5	100nF	R5	10k
C6	100nF	R6	20k
C7	100nF	R7	40k2
C8	10µF	R8	80k6
C9	470pF	R9	162k
C10	470pF	R10	324k
C11	100pF	R11	649k
C12	1nF	R12	1.3M
C13	1nF	R13	100k
C14	220pF	R14	1M
C15	100nF	R15	10k
C16	100nF	R16	12k
C17	10pF	R17	8k2
C18	100nF	R18	120k
C19	100nF	R19	100k
		R20	3k3
D1	LED GRUEN	R21	10k
D2	LED ROT	R22	4k7
D3	LED GELB	R23	10k
D4	LED GRUEN	R24	100k
D5	1N4148	R25	100R
D6	1N4148	R26	1k2
D7	1N4148	R27	47k
T1	BS170	P1	50k
IC1	XCS10-3	Q1	Quarz 4,9152MHz
IC2	HA7210		
IC3	MCM6264C	X1	Sub-D-Stecker 25polig
IC4	LM6134	X2	Sub-D-Buchse 9polig
IC5	78L05	X3	AC-Buchse
IC6	LM393N	J1	Steckbrücke

**Tabelle 1** Stückliste