

Hspeed-Modem für den EPP

Jürgen Hasch, DG1SCR, Meisenstraße 23, 73066 Uhingen

Motivation

Durch die Einführung eines 200kHz breiten Kanals für Digitalanwendungen im 70cm-Band ergibt sich ein wachsender Bedarf an Packet-Radio-Komponenten für Datenraten deutlich größer 9600 bps, vorallem auf der Benutzerseite.

Seit kurzem sind Datentransceiver mit einer entsprechend großen Bandbreite und einer S/E-Umschaltzeit < 1ms 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 an die Schnittstelle eines PCs angeschlossen werden kann und ohne größere Probleme Datenraten bis über 100kBit 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 > 500kB/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 günstig realisieren ?

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

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. Analoge Pedants zu den FPGAs sind aber ebenfalls bekannt.

Es gibt eine Reihe von Herstellern derartiger Bausteine, die beiden namhaftesten Vertreter sind Xilinx und Altera. Für das Modem wurde ein XC5204-Baustein von Xilinx verwendet, Hauptgrund dafür war [4], sowie das kürzlich eingeführte, für den Amateur-Geldbeutel erschwingliche Entwicklungspaket von Xilinx. Andere Hersteller bieten allerdings mittlerweile ähnliche Pakete an, mit denen die einfachen Vertreter der jeweiligen Bausteinserie programmiert werden können.

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 Programmierschaltung notwendig, jedoch kann hier die Programmierung nachträglich auch wieder geändert werden.

Die dritte wichtige Variante stellen SRAM-basierte FPGAs dar. Diese müssen jedesmal beim Anlegen der Versorgungsspannung aus einem externen Speicher (z.B. ein kleines 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 Abbildung 1 gezeigt, miteinander durch logische Kanäle verbunden werden können:

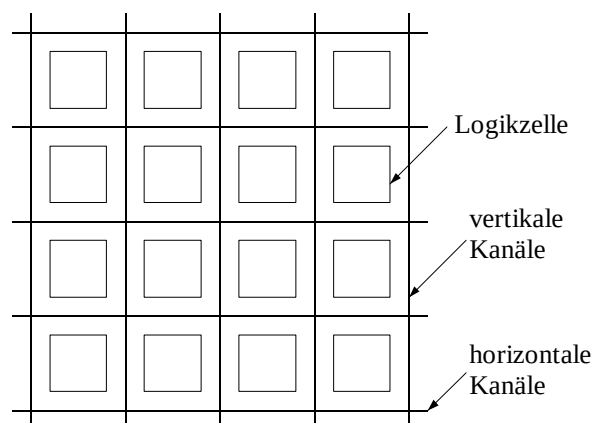


Abbildung 1: Struktur eines FPGAs

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 XC5200-Familie von Xilinx besteht die Gatterlogik einfach aus einer 16 Bit großen Speicherzelle. Für eine beliebige

Eingangskombination an den vier Eingängen kann damit jeder gewünschte Ausgangswert einprogrammiert werden:

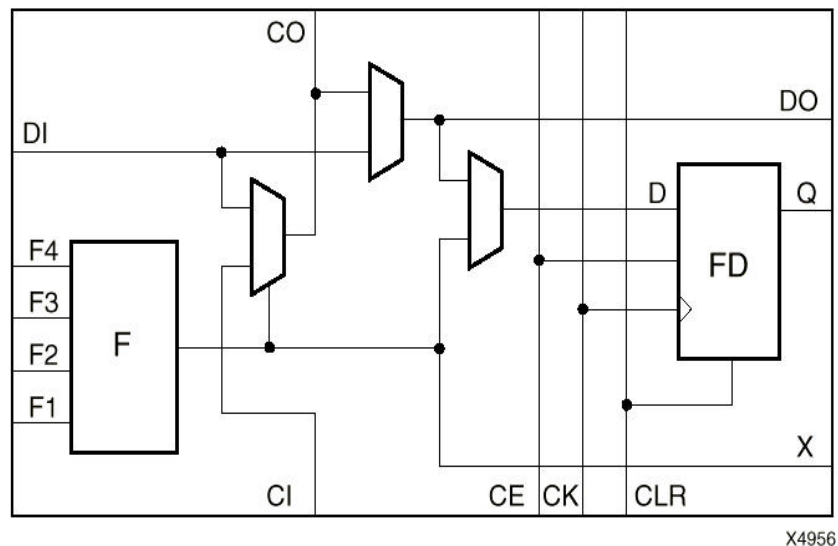


Abbildung 2: Aufbau einer Logikzelle bei der XC5200-Familie

Jede Logikzelle ist einzeln für sich programmierbar. Der ausgewählte Baustein XC5204 von Xilinx enthält 480 solcher Logikzellen und erreicht damit eine Komplexizität von etwa 6000 Gattern.

Konzept

Bei einer Bitrate von etwa 100kBit/s soll das Modem dem PC eine möglichst geringe Rechenleistung abverlangen, da moderne (Fenster-)Betriebssysteme schon einen Großteil der Rechenleistung benötigen.

Um die Interrupt-Last des PC's bei Empfang niedrig zu halten, ist eine Pufferung der Empfangsdaten notwendig, möglichst sollte im Modem ein komplettes Frame dekodiert werden und erst nach Verifizierung der CRC-Prüfsumme für den PC freigegeben werden. Da bei dem verwendeten XC5204-FPGA Speicherplatz eine wertvolle Resource ist, wurde ein externes 8k*8 RAM zur Speicherung vorgesehen.

Der Sendefall ist ein wenig einfacher aufgebaut, hier erzeugt der PC den Sendedatenstrom inklusive HDLC-Kodierung und Bitstuffung. Lediglich NRZI-Kodierung und Scrambler sind Bestandteil des Modems. Vorteil dieser Methode ist, daß man einerseits durch Voranstellen von HDLC-Flags den benötigten TX-Delay erzeugen kann und trotzdem die Sendedaten im RAM des Modems speichern kann.

Die Realisierung des EPP-Interface ist sehr einfach gelöst. Ein Datenzugriff vom PC aus liest oder schreibt ein Zeichen aus dem RAM des Modems aus und setzt gleichzeitig einen Zeiger auf das nächste Zeichen im RAM. Bei einem Adresszugriff wird das Statusregister im Modem neu gesetzt, bzw. ausgelesen.

Die Abbildung zeigt das Blockschaltbild des Modems:

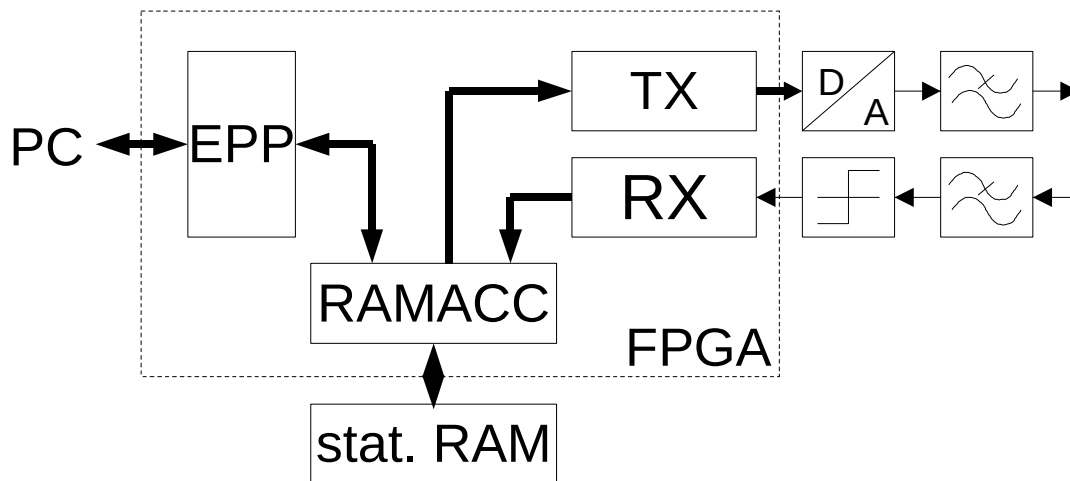


Abbildung 3: Funktionsblöcke des Modems

Gut erkennbar ist die Entkopplung zwischen dem eigentlichen Modem-Teil (RX und TX) und der EPP-Schnittstelle durch die Zwischenspeicherung der Daten im RAM.

Der Empfänger (RX) besteht aus fünf Funktionsblöcken:

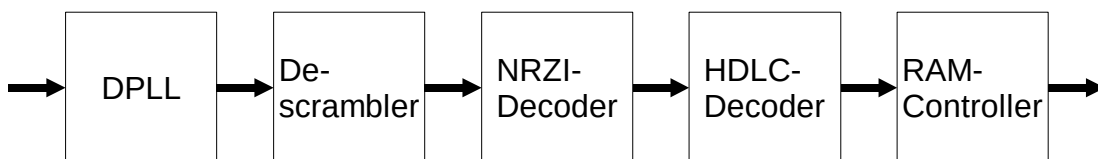


Abbildung 4: Funktionsblöcke des Empfängers

In der DPLL geschieht die Taktrückgewinnung und DCD-Erkennung. Dazu wird der 64fache Empfangstakt zugeführt und versucht, auf die Flanken des Datensignals zu regeln. Die DCD-Erkennung geschieht, indem überprüft wird, ob in der Mitte der Augenöffnung eine Änderung der Empfangsdaten vorkommt. Falls dies geschieht, wird die DCD 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.

Der HDLC-Decoder erkennt empfangene HDLC-Flags und zählt die Anzahl der nach einem Flag empfangenen Bytes mit. Stimmt am Ende eines Frames die übertragene Prüfsumme mit der berechneten Prüfsumme überein, wird die Länge des empfangenen Frames vor die Frame-Daten geschrieben und dem PC über einen Interrupt signalisiert, daß ein Frame empfangen wurde.

Der Sender ist sehr einfach aufgebaut:

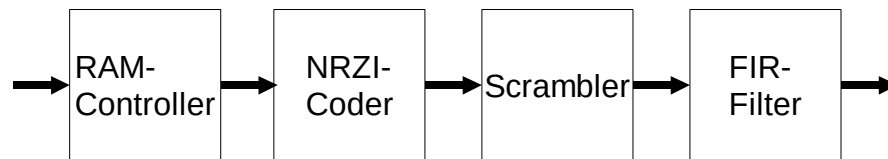


Abbildung 6: Funktionsblöcke des Senders

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

Hardware

Im Anhang A ist das Schaltbild des EPP-Modems abgebildet. Zentraler Baustein der Schaltung ist natürlich das FPGA, das direkt an die EPP-Schnittstelle des PCs angeschlossen ist. Wird der Modemtreiber auf der PC-Seite gestartet, so wird das FPGA mittels Reset-Leitung zurückgesetzt und über die Write- und D0-Leitung der Druckerschnittstelle werden anschließend die Konfigurationsdaten seriell in das FPGA übertragen. 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 einem 4,9152MHz Quarz verwendet.

Die aus dem internen FIR-Filter kommenden Sendedaten, die an den Ausgängen O0-O7 anliegen, werden über ein als D/A-Wandler beschaltetes Widerstandsnetzwerk zu einem aktiven Tiefpaßfilter geführt und gehen, nach einem Impedanzwandler und einem Trimmer zur PegelEinstellung, auf den Ausgangsstecker.

Das vom Funkgerät kommende Empfangssignal wird zuerst auf einen Impedanzwandler geführt, gefolgt von einem Tiefpaßfilter. Danach 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 5V 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 ?

Software

Wie schon angedeutet, besteht die Erstellung Software aus zwei Teilen, der Programmierung des FPGAs und dem Schreiben des Modem-Treibers für den PC.

Die FPGA-Programmierung geschieht mit Hilfe der vom Hersteller gelieferten Software „Xilinx Foundation 1.4“, in einer Mischung zwischen Schaltbildeingabe und VHDL-Unterfunktionen.

Die Schaltbildeingabe wird für die oberste Hierarchieebene der Schaltung verwendet, in der alle Komponenten nur noch miteinander verdrahtet werden müssen. Die grafische Darstellung erhöht dabei die Übersichtlichkeit.

Die eigentlichen Baugruppen im Modem sind mittels der Beschreibungssprache „VHDL“ (Very High Speed Integrated Circuit High Definition Language) realisiert. Das ist eine Hardware-Beschreibungssprache, die, freundlich formuliert, auch für digitale Schaltungen in FPGAs geeignet ist. Gegenüber der Schaltbildeingabe erleichtert sich hier die Realisierung algorithmischer Baugruppen erheblich, allerdings um den Preis eines erhöhten Ressourcenverbrauchs und des geringen Zugriffs auf spezifische Eigenschaften der gewählten Bausteinfamilie.

Auf der PC-Seite ist ein Flexnet-Treiber in der Entwicklung. Der Treiber führt nach Programmstart zuerst die Konfiguration des FPGAs durch, im Betrieb reicht er dann die vom Modem dekodierten Pakete direkt zu Flexnet durch.

Im Sendebetrieb erzeugt der Treiber die für das Modem benötigten Rohdaten (d.h. er erzeugt die HDLC-Flags, fügt das Bitstuffing ein und berechnet die CRC-Prüfsumme). Ein beliebiges TX delay kann durch hinzufügen von genügend Flags vor die Nutzdaten erreicht werden.

Ausblick

Die Entwicklung des Modems ist noch nicht abgeschlossen, da bis zum Redaktionsschluß der Empfangsteil des Modems nicht verifiziert war. Bis zum Vortrag sind hier hoffentlich neuere Erkenntnisse vorhanden.

Durch die Möglichkeit zu Software-Updates sind Verbesserungen am Modem ohne großen Aufwand möglich, so kann beispielsweise ein verbesserter Sendeteil dem PC die Arbeit vereinfachen. Dazu muß jedoch noch eine einfache Möglichkeit für die Erzeugung des gewünschten TX delay gefunden werden.

Literatur

- [1] A Kurpiers DL8AAU und Martin Liebeck DL2ZBN: Hochgeschwindigkeits-Packet-Radio - ein Transceiverkonzept für das 70cm Band
Skriptum zur 13. Internationalen Packet-Radio Tagung
- [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] Nico Palermo: Yet Another 9k6 Modem
Beschreibung unter <http://www.microlet.com/yam>

Anhang A: Schaltbild des EPP-Modems

Anhang B: Bestückungsplan des EPP-Modems