

4. Entwicklung eines ASIC für einen GPS-Empfänger

Hans-Peter Behrens, Fachhochschule Offenburg

An der FH Offenburg arbeiten seit Ende 1989 in einem Team die Professoren Dr. Jansen, Dr. Schüssele, die wissenschaftlichen Mitarbeiter Bernd Reinke, Martin Jörger und die Diplomanden Hans Fiesel, Otmar Feißt an dem Entwurf eines Navigationsempfängers. Im Rahmen dieses Projekts, genannt GPS-Projekt (GPS = Global Positioning System), wurde im Herbst 1990 ein experimenteller Empfänger in Betrieb genommen. Nachdem die Testergebnisse gezeigt hatten, daß das Konzept der Anlage stimmte, ging es nun um die Miniaturisierung, Integration und Optimierung der Schaltung. Außerdem sollte der bisher verwendete PC durch einen auf der Platine befindlichen Mikroprozessor ersetzt werden. Im Zusammenhang mit dem GPS-Projekt wurden bisher im Offenburger ASIC-Labor eine Anlogschaltung auf einem B500, drei LCA Designs und diverse GAL's entwickelt.

Zur Zeit arbeiten mehrere Diplomanden an der zweiten Generation des Empfängers. Meine Aufgabe besteht darin, die dort noch in drei LCA's untergebrachte digitale Logik sowie einen Teil des bisherigen PC-Interface in einem IMS Gate Forrest zu integrieren. Außerdem muß die Logik von 8 Bit auf einen 16 Bit breiten Datenbus umgestellt und an die neue Peripherie des Mikroprozessors angepasst werden. Damit soll die jetzige Digital-Platine noch weiter verkleinert werden (Bild 1). Wesentlich ist dabei die Umsetzung der zahlreichen Zähler- und Registerstrukturen in einen Gate Forrest. Als Arbeitsmittel stehen Apollo Workstations mit Mentor Software zur Verfügung.

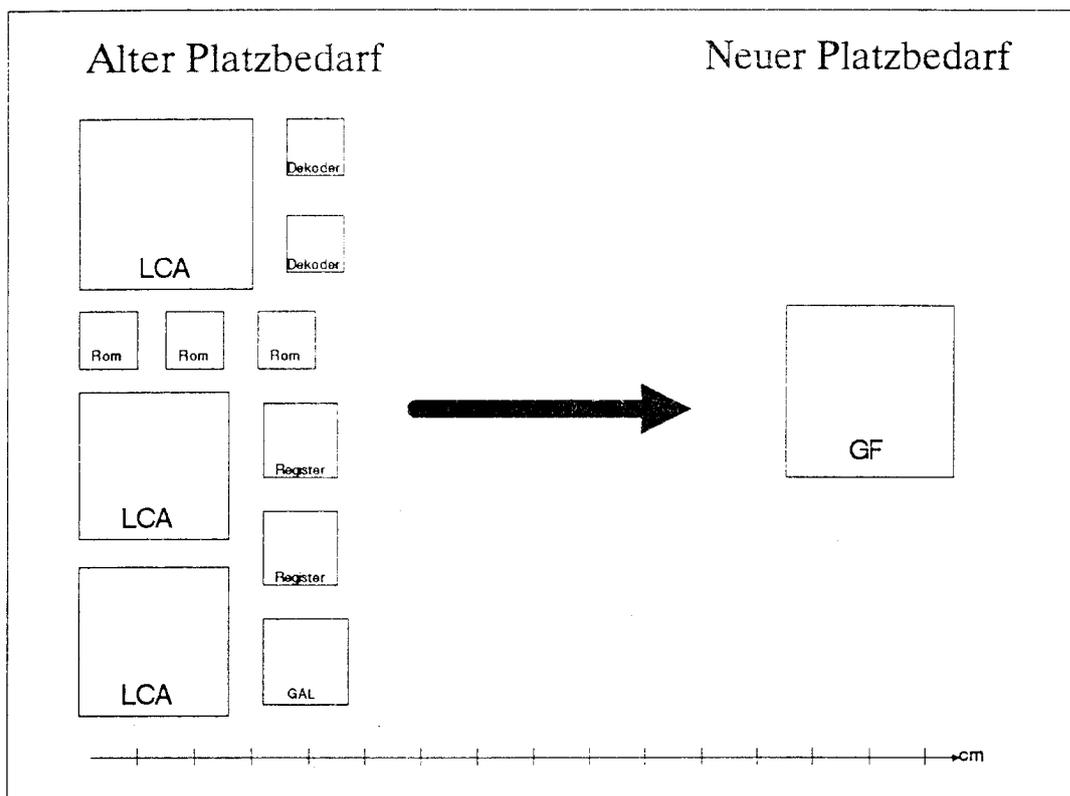


Bild 1: Platzvergleich Version 2 zu Version 3

Einführung:

Bei dem GPS-System handelt es sich um ein satellitengestütztes Navigationssystem, mit Spread Spectrum Modulation. Das System wurde ursprünglich für rein militärische Zwecke entwickelt, inzwischen aber zum Teil für zivile Zwecke freigegeben. Durch GPS wird es möglich, weltweit eine dreidimensionale Positionsbestimmung mit einer Genauigkeit von ca. 100m für zivile und 30m für militärische Anwendungen durchzuführen. Dies wird jedoch erst nach einer vollständigen Installation des Systems mit 18 Satelliten möglich sein. Diese werden die Erde in ca. 20200 km Höhe auf 6 verschiedenen Umlaufbahnen umkreisen. Drei weitere Satelliten sind als Reserve vorgesehen (Bild 2).

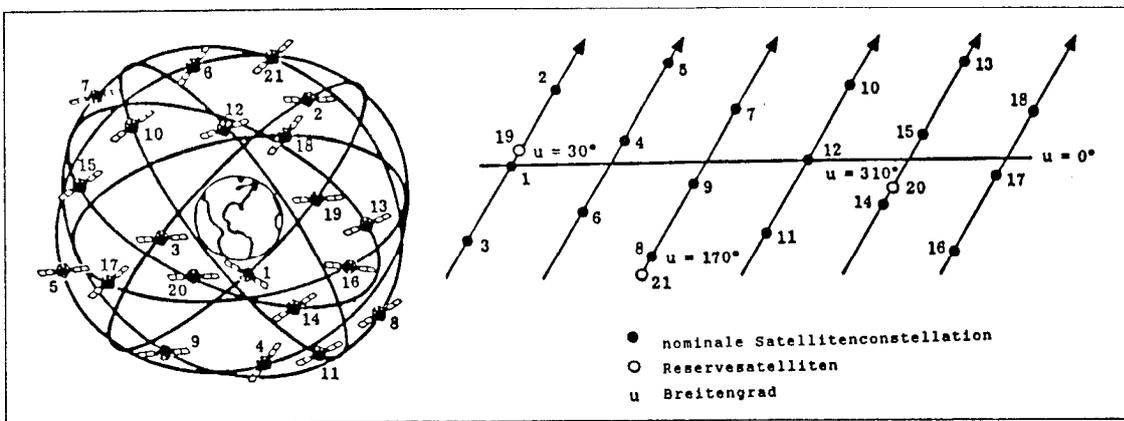


Bild 2: Satellitenlaufbahnen [2]

Um eine dreidimensionale Ortsbestimmung durchführen zu können, muß die Entfernung zu mindestens drei Satelliten sowie die genaue Position jedes dieser Satelliten bekannt sein. Zur Abstandsmessung wird die Signallaufzeit eines vom Satelliten ausgesendeten Datenstroms und die Ausbreitungsgeschwindigkeit elektromagnetischer Wellen verwendet. Da zwischen Empfänger- und Satellitenzeit keine Differenz auftreten darf, müßte auch der Empfänger genau wie die Satelliten mit einer Atomuhr ausgerüstet sein. Um diesen Aufwand zu umgehen, mißt man die Entfernung zu einem vierten Satelliten und erhält somit ein System von vier Gleichungen, aus denen Ort und Zeit bestimmt werden können.

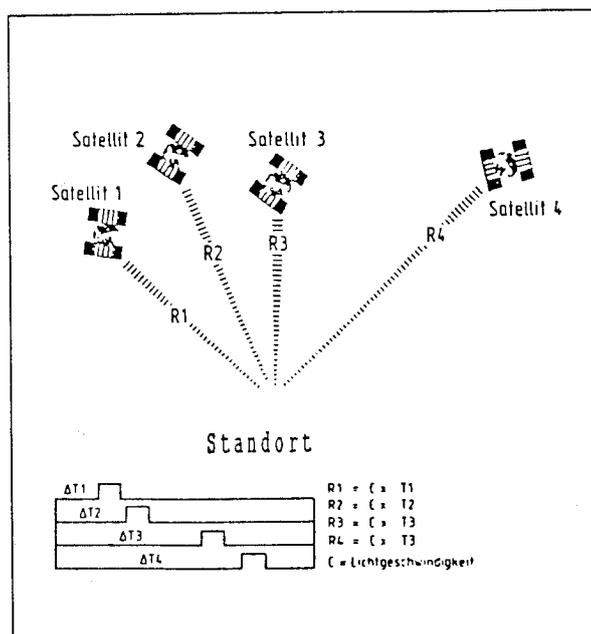


Bild 3: 3D-Ortsbestimmung [2]

Satellitensignale:

Alle Satelliten senden zwei Signale L1 und L2, zur Datenübertragung werden die Frequenzen $f_1 = 1575,42$ MHz und $f_2 = 1227,6$ MHz verwendet. Diese Frequenzen sind Vielfache des Systemgrundtaktes $F_0 = 10,23$ MHz. Für das Signal L1 wird der Träger f_1 mit zwei Codefolgen, dem C/A (Coarse Acquisition) und P (Precision) -Code moduliert und zwar durch die Quadratur-Phasenmodulation. Die Modulation aus dem Träger f_2 und dem P-Code bildet das Signal L2. Der P-Code und das Signal L2 sind jedoch der militärischen Nutzung vorbehalten. Als minimaler Empfangspegel wird für den C/A-Code -130dBm angegeben.

C/A-Code:

Der C/A Code ist ein Goldcode, der aus der Modulo-2-Addition (Exor) zweier 1023 Bit Pseudo-Random-Noise (PRN) Codes gewonnen wird. Die Folgen G1 und G2 werden von jeweils einem linear rückgekoppelten 10 Bit Schieberegister generiert. Getaktet werden die Schieberegister mit einem Zehntel des Grundtaktes, also 1,023 Mhz. Mit dem X1-Epoche Signal wird der C/A Code synchronisiert.

Jeder Satellit sendet einen anderen C/A Code. Die unterschiedlichen Codes werden dadurch erzeugt, daß die Folge G2 um eine ganzzahlige Anzahl von C/A Takten verzögert wird. Durch die Exorverknüpfung zweier Abgriffe des Schieberegisters erhält man solche verzögerten G2 Folgen.

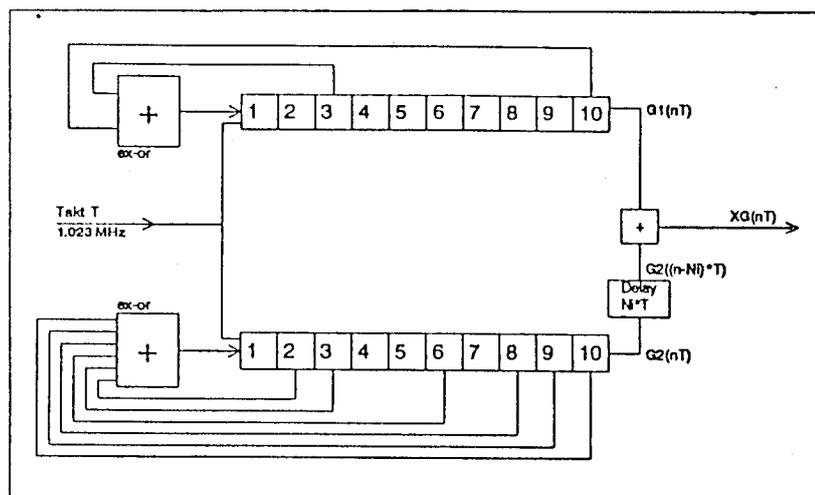


Bild 4: C/A-Code-Erzeugung [2]

Empfangskonzept der FH:

Zur Auswertung des Satellitensignals ist es notwendig, das Empfangssignal mit einem intern erzeugten C/A Code zu korrelieren. Dieser interne Code muß dem Satellitensignal entsprechen und mit ihm in Phase gebracht werden. Hierfür benötigt der GPS-Empfänger einen Codegenerator und eine Codesynchronisationseinheit. Bei der Verknüpfung beider Signale ergibt sich ein Signalanteil entsprechend der Verschiebung beider Codes zueinander. Zuerst wird die Empfängercodephase solange variiert, bis ein Korrelationssignal einen von Null verschiedenen Anteil besitzt. Der Korrelationspegel wird nach dem Pegeldetektor entsprechend der Codephase mit plus/minus 1 oder 0 bewertet. Ist der Mittelwert des Ausgangssignals dieser Bewertung gleich Null, dann sind die Codes in Phase (Prinzip des Tau-Dither-Loop).

Blockstruktur der gesamten Schaltung:

Bild 5 zeigt ein vereinfachtes Blockschaltbild des digitalen Empfängerteils. Außer dem Prozessor und dessen speziellen Peripheriebausteinen sowie einigen A/D-, D/A-Wandlern befinden sich alle digitalen Schaltungen in einem Chip.

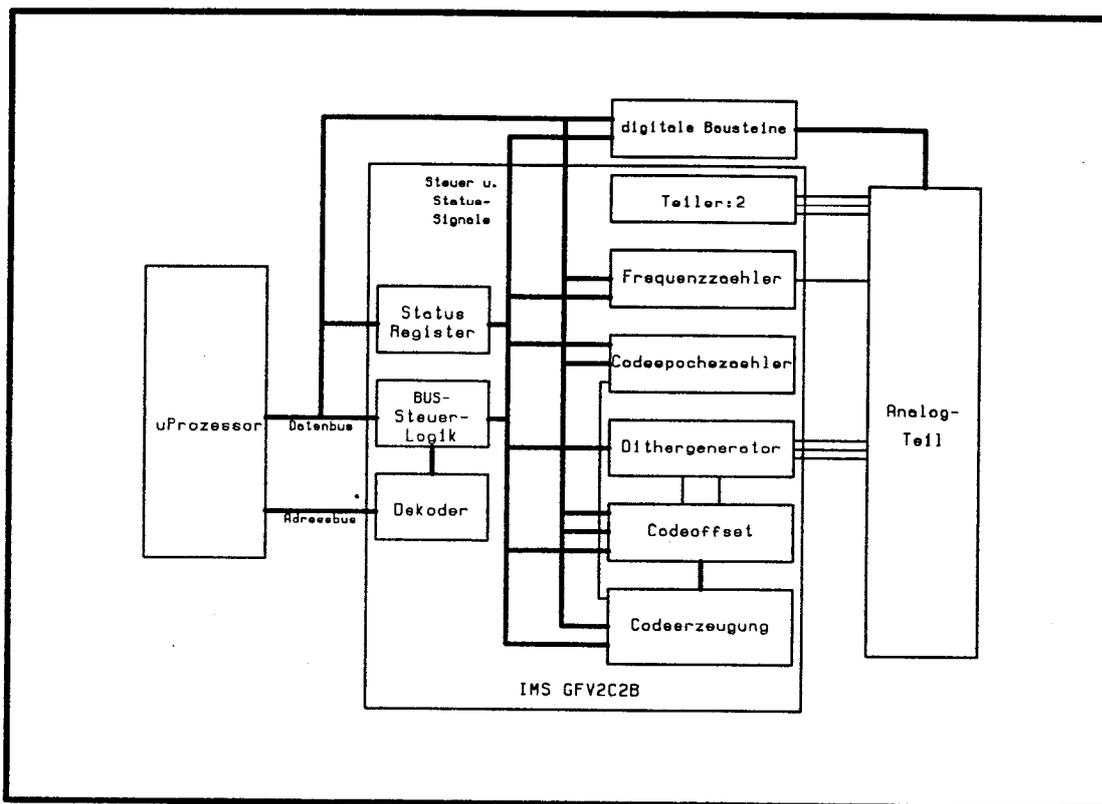


Bild 5: Blockschaltbild des ASIC

Kurze Beschreibungen der einzelnen Blöcke:

Codeerzeugung:

Mit zwei *10 Bit Schieberegistern* und jeweiligen EXOR-Matrizen wird der C/A-Code generiert. Um die je nach Satellitennummer unterschiedlichen C/A-Codes zu erhalten, wird der vom Prozessor gewünschte Parallelabgriff des *G2-Schieberegisters* im *G2-Codeselect-Register* abgespeichert, mit der *G2-Tab-Selektion* abgegriffen und EXOR-verknüpft. Der Takt für die Schieberegister wird in einem *Teiler* (Systemtakt:10) erzeugt und mit dem 1 ms Set-Enable-Signal (SE) synchronisiert.

Außerdem befinden sich im Codegenerator noch weitere Baugruppen, die zur Synchronisation des internen mit dem externen C/A-Code benötigt werden. Der *Systemtaktzähler 10,23MHz* erzeugt einen 14 Bit Vergleichswert zur Erzeugung des SE-Signals und liefert ein 1kHz Synchronsignal. Das 14-Bit-Signal wird mit dem vom *Codeoffset* gelieferten Verzögerungswert verglichen und erzeugt bei Gleichheit beider Werte das SE-Signal. Mit diesem Verzögerungswert kann der C/A-Code in 100ns-Schritten verschoben werden.

Codeoffset:

Diese Baugruppe liefert der *Codeerzeugung* einen statisch anstehenden 14 Bit Verzögerungswert. Dieser kann vom Prozessor vorgegeben und bei Dither Count Enable (DCE), in Abhängigkeit der Komperatoreingänge inkrementiert bzw. dekrementiert werden. Dabei steuert das *Stellglied* das Auf-/Abwärtszählen des Verzögerungswertzählers. Eine *Handshakeschaltung* regelt die Übernahme eines Wertes vom Prozessor in den Verzögerungswertzähler.

Dithergenerator:

Der C/A-Code durchläuft ein *11 Bit Schieberegister* mit drei Steuereingängen. Mit den im *Dithergenerator* erzeugten Steuersignalen kann der Code in 500ns-Schritten verschoben werden.

Codeepochezähler:

Der *Codeepochezähler* dient der Unterteilung der X1-Epochen in Millisekundenschritte, um eine eindeutige Zeitskalierung zu erhalten. Der Zähler durchläuft einen 0 bis 1499 Zyklus und wird mit jeder neuen Codeepoche durch das SE-Signal inkrementiert.

Frequenzzähler:

Er zählt die Frequenz des spannungsgesteuerten Quarzoszillators zur Messung der Dopplerverschiebung. Ein *Torsignalgenerator* zählt das 1ms SE-Signal und erzeugt damit alle 2 Sekunden einen Impuls. Mit diesem übernehmen die *Ausgangsregister* den Zustand des Frequenzzählers und der *Frequenzzähler* wird zurückgesetzt.

Teiler:

Dieser Block erzeugt Ansteuersignale für den Costas-Loop. Dabei handelt es sich um zwei 90° zueinander verschobene 5,115Mhz-Signale zur Versorgung der Regelschleife der Trägernachführung.

Dekoder:

Dekodierung der Adressen für die Bussteuerung.

Bussteuerlogik:

Die Bussteuerung regelt den Datenverkehr zwischen den einzelnen internen- bzw. externen Bausteinen und dem Controller.

Statusregister:

Für den Prozessor wichtige Statussignale werden hier zwischengespeichert.

Realisierung der Schaltung mit Mentor-Software:

Es bot sich an die bereits vorhandene Aufteilung in Blöcke beizubehalten und zu verfeinern, da es sich um eine sehr komplexe Schaltung mit einer großen Zahl an Registerstufen und Zählern handelt. Die IMS-Bibliothek bietet bisher keine Makros für Gate-Forrest an, deshalb gab ich häufig auftretende Strukturen einmalig ein, definierte sie als Block und verwendete sie wie ein Makro auf der untersten Hierarchieebene der Schaltplaneingabe. Zum Beispiel definierte ich häufig benötigte Ein-/Ausgaberegisterstufen, Vier-Bit-Synchronzähler, sowie weitere Zähler- und Registerseinheiten als "Makro-Block". Bei dem Aufbau dieser Blöcke wurde auf möglichst geringen Platzverbrauch geachtet. So wurden die Zählerstufen zwar den TTL Standardzählern nachempfunden, aber soweit möglich minimisiert und auf die platzsparenden negierten Standardbausteine der IMS-GF-Bibliothek umgestellt.

Die Funktion der einzelnen Blöcke wie auch der höherliegenden Schaltplanebenen wurde mit MENTOR-QUICKSIM überprüft und simuliert.

Bild 6 zeigt den *Frequenzzähler*. Dieser ist aus vier Zählermakros aufgebaut. Im Bild ist außerdem einer der 4-Bit-Synchronzähler-Blöcke sowie ein Ausschnitt aus der Quicksim-Simulation enthalten. Diese Zählerstrukturen konnten außer in dieser Baugruppe auch im *Codeepochezähler* bzw in der *Codeerzeugung* verwendet werden.

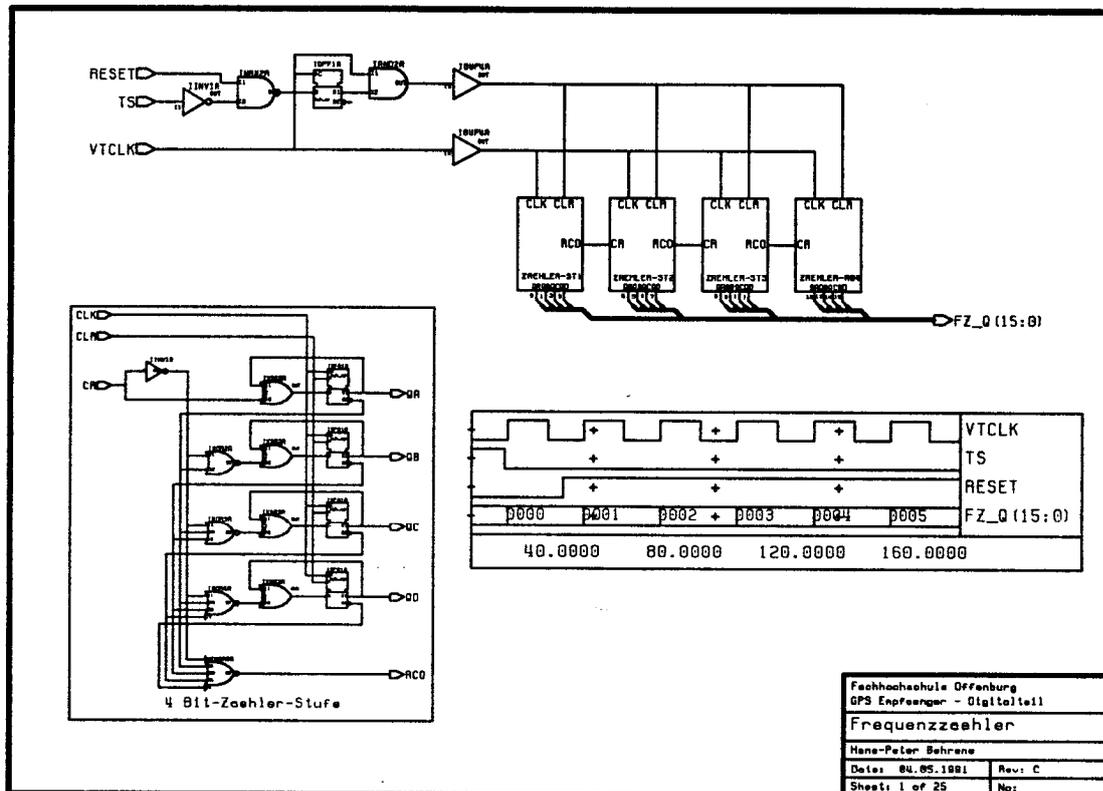


Bild 6: Frequenzzähler aufgebaut aus 4-Bit-Zählerstufen

Bild 7 zeigt die Ausgabeeinheit des *Frequenzzählers* auf den Datenbus. Eine Ausgabestufe besteht aus einem Multiplexer, einem D-Flip-Flop und einer Tristateausgangsstufe. Diese Einheit wurde von mir als Block definiert und ist für alle Ausgabestufen auf den Datenbus benutzbar, so zum Beispiel im *Codeoffset* und *Codeepochezähler*.

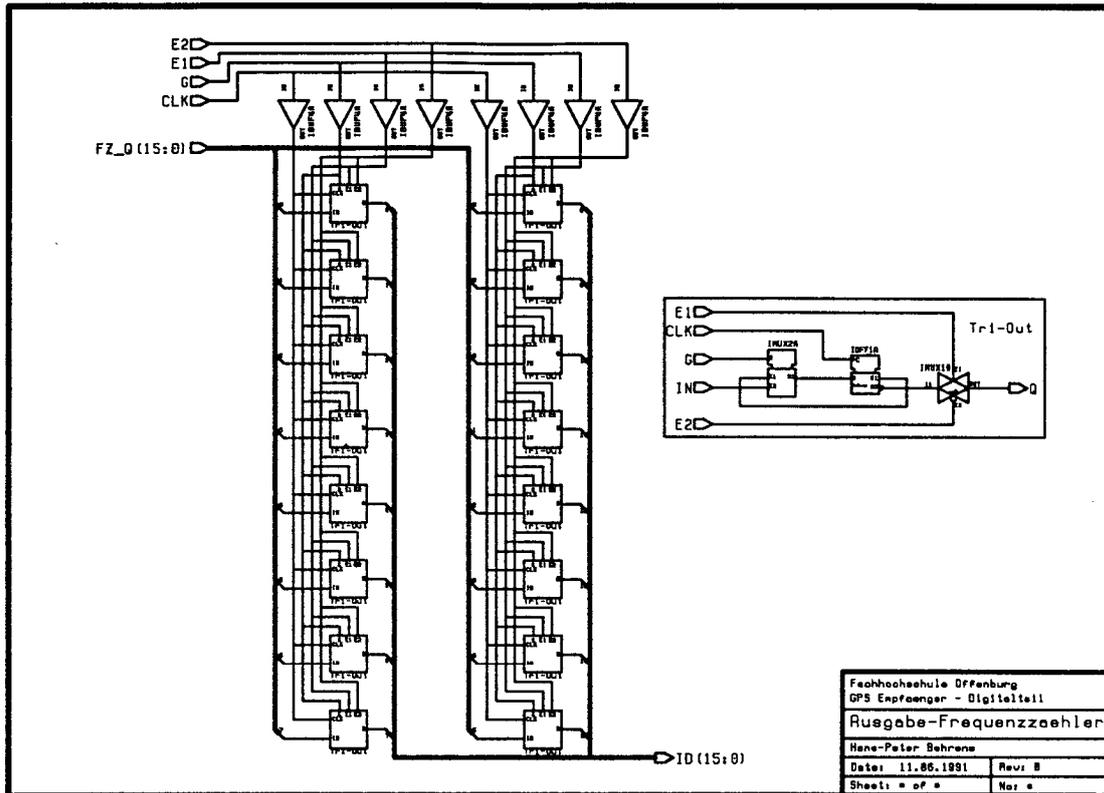


Bild 7: Ausgabeeinheit auf den Datenbus

Berechnung der vorläufigen Chipkomplexität:

Folgende Standardzellen werden zum derzeitigen Stand der Schaltung benötigt:

Gatterbezeichnung	n * 17u Breite	Anzahl	Summe Sites
IINV1A	2	27	54
ICIN1B	5	55	275
IBUF1A	3	17	34
IBUF4A	7	55	385
INAN2A	3	19	57
INAN3A	4	11	44
INAN4A	6	14	84
INAN5A	7	5	35
INOR2A	3	32	96
INOR3A	5	17	85
INOR4A	6	14	84
INOR5A	7	11	77
IAND2A	5	35	175
IORO2A	5	23	115
IMUX2A	6	76	456
IXOR2A	5	103	515
IDFF1A	13	158	2054
IDFS1A	16	22	352
IDFR1A	17	75	1275
IPBF2A	15	11	165
ITPB2A	16	31	496

			6913

Würde man einen IMS-Gate-Forrest des Typs GFXX2 verwenden, so entspräche dies einem Ausnutzungsgrad von 47%. Hierbei ist jedoch der Platzbedarf für die Verdrahtungskanäle nicht miteinbezogen. Sollte die Integrierung der Schaltung auf dem GFXX2 nicht ohne Probleme möglich sein, wird erwogen den neuen GF-Baustein mit 1,2u Prozess zu benutzen.

Als Grundlage für den Entwurf der Schaltung stand mir die Diplomarbeit von H. Fiesel zur Verfügung. Außerdem möchte ich mich für die Unterstützung von Prof. Dr. Jansen und dem Laboringenieur B. Reinke bedanken.

Literaturverzeichnis:

1. Navstar GPS Joint Program Office, Los Angeles Air Force Station, US Air Force Space Division/CWNI, Introduction to Navstar GPS User Equipment 1988
2. Spilker, J. J., Global Positioning System: Signal Structure and Performance Characteristics, Stanford Telecommunications Inc., 1979
3. Summ, Patrik, Entwicklung eines GPS-Empfängers, FHO Forschungsbericht 1/90
4. Fiesel, Hans, Digitalteil eines GPS-Empfängers, Diplomarbeit SS90