

Entwicklung eines 16-Bit Mikroprozessor-Kernels mit Hilfe von VHDL

Fritz Zimpfer, Thomas Gieringer

Prof. Dr.-Ing Dirk Jansen

Mitteilung aus dem ASIC-Design-Center der FH Offenburg

Für die Implementation in ASIC's wurde ein kompakter Mikroprozessor-Kernel als Standardzellen-Makro entworfen. Durch konsequenten Einsatz von Hochsprachen und CAE-Werkzeugen (VHDL, Synthese) konnte ein vollständiges Design in nur vier Monaten durchgeführt werden. Der Prozessor wird in einem Testchip erprobt.

Einführung

Mit der Verfügbarkeit moderner CAE-Werkzeuge können heute komplexe Entwürfe angegangen werden. Um diese Leistungsfähigkeit zu demonstrieren und gleichzeitig den inneren Aufbau moderner Prozessoren für die Lehre zu erschließen, wurde am ASIC-Design-Center der FH Offenburg das Studentenprojekt FHOP ins Leben gerufen.

FHOP steht für First Homemade Operational Processor. Im Rahmen dieses Projekts wird ein Mikroprozessor-Kernel entwickelt, der zur Implementation als Standardzellen-Makro in verschiedenen Steuer- und Kontrollanwendungs-ASIC's geeignet ist. Durch Platzieren des Kernels kann Prozessorleistung auf einem ASIC zur Verfügung gestellt werden. Der Anwender fügt dann die für seine Anwendung spezifischen Komponenten (z.B. I/O-Ports, Interruptcontroller, Timer usw.) hinzu. Den typischen Aufbau eines solchen ASIC's mit Prozessor-Kernel zeigt Bild 1.

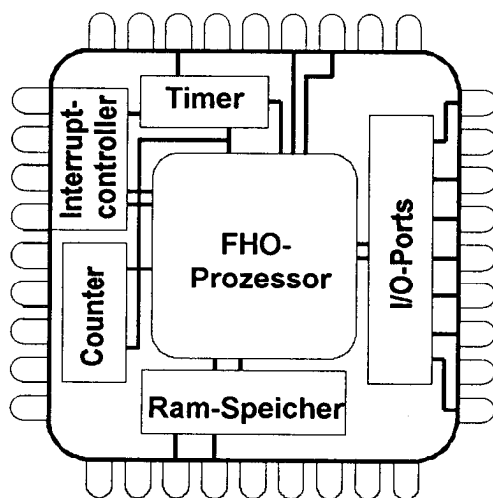


Bild 1: Einsatz des Mikroprozessor-Kernels

Entwicklungsumgebung

Im folgenden sollen die wichtigsten Werkzeuge der Entwicklungsumgebung kurz erläutert werden.

Die Entwicklungsumgebung besteht aus drei HP-Workstations mit MENTOR GRAPHICS CAE-Tools in der Version 8.2. Der Chip wird auf Grundlage der ES 2 - Bibliothek in 1,0 µm-Technologie über EUROCHIP gefertigt.

VHDL

Die Entwicklungsumgebung erlaubt Beschreibungsmodelle eines Designs bzw. Chips zu entwerfen und zu simulieren. VHDL ist eine Beschreibungssprache für digitale Hardware, die anderen Programmiersprachen ähnlich ist. Sie enthält Komponenten der Portbeschreibung (engl. entity) und der Verhaltensbeschreibung (engl. architecture). Somit kann auf abstrakter Ebene ein Teilmodul oder die komplette Architektur einer Schaltung untersucht werden. Aus der Beschreibung kann anschließend schrittweise die Gatterlogik abgeleitet, zum Teil automatisch synthetisiert werden. Damit kann die Chipentwicklung digitaler Schaltkreise wie hier beim FHOP wesentlich beschleunigt werden.

Design Architect

Mit Hilfe des DESIGN ARCHITECT können Symbole entworfen und mit VHDL-Modellen hinterlegt werden. Zusammen mit Bibliothekselementen werden die so kreierte Symbole auf einem gemeinsamen Schaltplan platziert. Der Gesamtentwurf präsentiert sich als hierarchisch gegliederte, übersichtliche Struktur.

Quicksim II

Mit dem Digitalsimulator läßt sich diese heterogene Architektur aus realen Gattern und VHDL-Modellen simulieren. Designfehler können damit früh erkannt und behoben werden.

AutoLogic

Dieses Software-Paket beinhaltet ein Syntheseprogramm, das ein korrektes VHDL-Modell automatisch in eine Gatterlogik umsetzt. Weiterhin ist eine zielgerichtete Flächen- und Geschwindigkeitsoptimierung der Schaltung auf die Zieltechnologie - hier ES2 1,0 μm Technologie - möglich.

IC - Station

Mit Hilfe dieses CAE-Werkzeugs werden die Bibliothekselemente und Makros auf dem Silizium plaziert und geroutet. Hieraus werden die Layoutdaten für die Chipfertigung generiert.

Entwicklungsschritte

Der Entwicklungsprozeß wird konsequent nach dem Top-Down-Prinzip (vgl. Bild 2) durchgeführt. Zuerst wird die Gesamtstruktur des Prozessors und dessen Befehlssatz auf dem Papier festgelegt. Dann wird der FHO-Prozessor in logische Funktionsblöcke partitioniert, um das Design übersichtlich zu halten. Diese einzelnen Komponenten werden dann in VHDL-Modellen beschrieben und mit Simulation auf ihre Funktion hin überprüft.

Im nächsten Schritt wird der gesamte Kernel aus den einzelnen Modulen zusammengesetzt und im Simulator getestet. Alle Befehle und Funktionen des Kernels werden im Simulator überprüft. Damit steht ein komplettes in VHDL beschriebenes Modell des Prozessor-Kernels zur Verfügung. Dieses ist auch für einen späteren Einsatz des Kernels in Applikationen wichtig, da hiermit eine effiziente Simulation auch komplexer Strukturen einschließlich der für den Kernel geschriebenen Software ermöglicht wird. Eine solche Simulation auf Gatterebene würde bald den Rahmen erträglicher Simulationszeiten sprengen.

Nach der Beseitigung von Entwurfsfehlern werden die VHDL-Verhaltensmodelle mit AUTOLOGIC synthetisiert. Die Synthese erlaubt durch steuernde Parameter eine zielgerichtete Optimierung auf gewünschte Eigenschaften. So konnte im vorliegenden Fall die Durchlaufzeit der ALU von zunächst über 70 ns auf unter 20 ns gedrückt werden, ohne daß die benötigte Fläche und Gatterzahl wesentlich anstieg. Hierfür war allerdings Rechenzeit von über 30 Stunden erforderlich.

Wegen Fehlens der Geometriedaten konnte das Layout bisher nicht generiert werden. Dies erfolgt, wenn von EUROCHIP das "Back - End" der ES2 1,0 μm Bibliothek vorliegt.

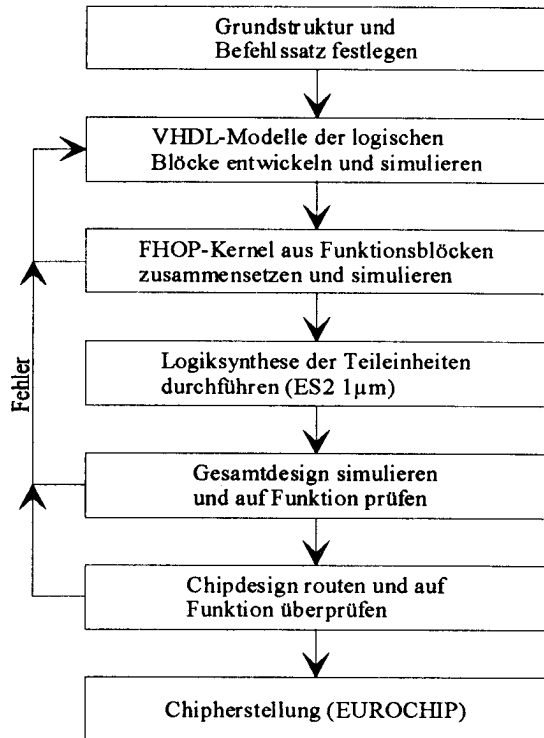


Bild 2: Top-Down-Design (verkürzt)

Der FHOP-Kernel

Für die vorgesehene Anwendung mußte eine Architektur gewählt werden, die sich durch folgende Eigenschaften auszeichnet:

- Regularität im Aufbau des Datenpfads,
- 16-Bit ALU mit integrierter Shiftoperation,
- byteorientierte Bussysteme,
- umfangreicher Befehlssatz (CISC-Konzept)
- einfaches Taktschema, kein Pipelining
- externe Schnittstelle mit Hold, Waitecycle-Steuerung und Interrupt,
- großer Adreßraum von 64 kByte,
- separater I/O-Raum.

Nicht zuletzt wurde die Architektur so gewählt, daß sie sich für eine Erläuterung der Prinzipien eines Mikroprozessors in der Lehre eignet. Das erzielte Resultat ist in der Leistungsfähigkeit am ehesten zwischen einem 8051-Controller und einem 8088-Mikroprozessor anzusiedeln. Auf eine Kompatibilität im Objektcode wie im Befehlssatz wurde bewußt verzichtet. Eine kommerzielle Auswertung wird nicht angestrebt, jedoch soll der Kernel der akademischen Welt lizenzfrei zugänglich gemacht werden.

Das Blockschaltbild (Bild 3) zeigt die Struktur des Mikroprozessors. Sämtliche Operationen sind intern auf ein 16-Bit Datenformat ausgelegt. Der externe Busanschluß ist byteorientiert, um den externen Speicher mit minimalem Aufwand adaptieren zu können. Es steht ein Adreßbereich von insgesamt

64 kByte und ein getrennter I/O-Bereich von 256 Byte zur Verfügung.

Der Prozessor besitzt eine 16-Bit ALU. Die ALU verfügt über insgesamt 16 Funktionen, wie sie in Tabelle 1 dargestellt sind. Neben den üblichen arithmetischen und logischen Funktionen ist auch eine Shiftoperation implementiert. Die Ergebnisse stehen nach 20 ns maximaler Laufzeit zur Verfügung. Die ALU wurde mit etwa 1000 Gattern realisiert. Neben den Rechenergebnissen werden die in Tabelle 2 dargestellten Flags generiert.

Tabelle 1: Funktionen der ALU

| Steuerleitungen | Logische Funktionen (mc = 0) | Arithmetische Operationen (mc = 1) |
|-----------------|---------------------------------|------------------------------------|
| 000 | $Y = 0$ | $Y = A$ |
| 001 | $Y = A \text{ and } B$ | $Y = A + 1$ |
| 010 | $Y = A \text{ or } B$ | $Y = A - 1$ |
| 011 | $Y = A \text{ exor } B$ | $Y = A + B$ |
| 100 | $Y = A \text{ shift left}$ | $Y = A + B + CY$ |
| 101 | $Y = A \text{ shift left } CY$ | $Y = A - B$ |
| 110 | $Y = A \text{ shift right}$ | $Y = A - B - CY$ |
| 111 | $Y = A \text{ shift right } CY$ | $Y = (-A) - 1$ |

Der Registersatz besteht aus sechs 16-Bit Registern. Vier dieser Register werden als Universalregister zur Verfügung gestellt (A ... D), eines der Register wird als Stackpointer, ein Register als Programm-Zähler genutzt. Weiter besitzt der Registersatz ein 8-Bit breites Flagregister und ein 8-Bit breites Zwischenregister.

Tabelle 2: Flags der ALU

| | |
|----|-----------------|
| M | Minus |
| AC | Auxiliary Carry |
| HC | Half Carry |
| CY | Carry |
| Z | Zero |
| OV | Overflow |
| P | Parity |
| HP | Half Parity |

Um in der Adressierung möglichst flexibel zu sein, wird die ALU zur Adreßberechnung wie auch zum Inkrementieren des Programm-Pointer-Registers verwendet. Ein eigentlicher Programm - Zähler existiert damit nicht.

Es sind insgesamt vier unterschiedliche Adressierungsarten realisiert:

- Registeradressierung (z.B. MOV r1,r2) die Adresse ist implizit im OP-Code enthalten
- direkte Adressierung (z.B. LDA adr) die Speicheradresse steht im zweiten und dritten Befehlsbyte
- indirekte Adressierung (z.B. LDA M) die 16-Bit Speicheradresse wird aus Register B entnommen
- Direktoperand-Adressierung (z.B. LDI konst) der Operand steht im zweiten und dritten Befehlsbyte

Bei dem Entwurf der Busschnittstelle des FHOP-Kernels zur externen Welt müssen die Gegebenheiten des ASIC-Entwurfs berücksichtigt

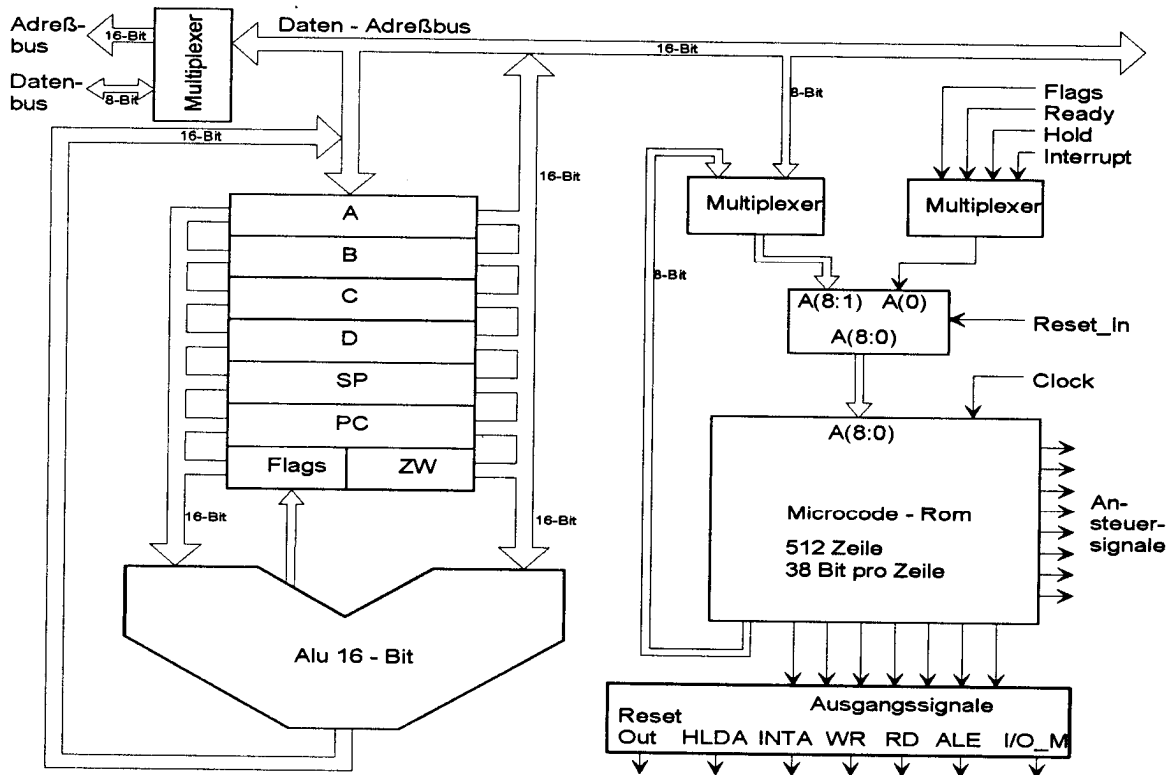


Bild 3: Blockschaltbild FHOP

werden. Da Speicher und Peripherieeinheiten sehr unterschiedlicher Zugriffszeit angeschlossen werden sollen, sind übliche Maßnahmen wie READY - Handshake zur Einfügung von Waitcyclen und ein hochohmig schalten der Prozessorsignale (HOLD) zu implementieren.

Die READY-Funktion wird im Prozessorzyklus berücksichtigt. Im Fetchzyklus sowie in allen Befehlen mit Peripheriezugriff ist dazu eine Abfrage der READY-Leitung integriert (Bild 4). Der Mikrocode wurde entsprechend gestaltet.

Ähnlich wie bei kommerziellen Prozessoren kann der Kernel über ein HOLD-Signal an der Schnittstelle hochohmig geschaltet werden. Auch dieser Zugriff erfolgt über einen Handshake und wurde durch die entsprechende Programmierung des Mikrocodes realisiert. Damit ist sowohl eine Multiprozessorfähigkeit als auch die Realisierung von selbständigen DMA-Controllern im Bussystem offen gehalten.

Die Steuerung besitzt eine Sammelinterruptleitung mit welcher, bei entsprechender Verwaltung durch einen externen Interruptcontroller, insgesamt 85 verschiedenen Interruptquellen unterschieden werden können.

Das Steuerwerk ist als klassische Mikroprogramm-Steuerung realisiert. Der Mikrocode, der speziell für den FHOP-Kernel entwickelt wurde, ist in einem auf dem Chip befindlichen ROM abgelegt. Gemäß der Intentionen bei der Entwicklung der Architektur wurde möglichst viel Steueraufwand in den Mikrocode verlagert, um ein kompaktes Design zu erreichen.

Der Mikrocode-ROM hat eine Größe von 512 Zeilen und 38 Bit pro Zeile. Er benötigt eine Chipfläche von etwa 1,4 mm² und wurde mit einem für die ES2 1,0µm - Bibliothek zur Verfügung stehenden ROM-Generator erzeugt.

Der Befehlssatz (vgl. Tabelle 3) umfaßt insgesamt 115 verschiedenen OP-Codes. Diese teilen sich in:

- 88 Ein-Byte Befehle,
- 3 Zwei-Byte Befehle und
- 24 Drei-Byte Befehle auf.

Faßt man verwandte Befehle zusammen (z.B. alle MOV-Befehle), bleiben noch 64 unterschiedliche Befehle übrig. Der Befehlssatz ist weitgehend verfügbaren Standardprozessoren ähnlich.

Das Zustandsdiagramm in Bild 4 zeigt den Fetchzyklus, wie er im Prozessor abgearbeitet wird. Dabei wird zu Beginn immer auf die Eingangssignale HOLD und INTERRUPT abgefragt. Dann beginnt der eigentliche Programmspeicherzugriff. Dieser Ablauf ist bei jedem Befehl gleich.

Der Fetchzyklus beginnt mit der Ausgabe des Program-Counters auf den Datenbus. Die

Peripherie, i.a. der Speicher wird adressiert und legt das erste Byte des Objektcodes auf den Bus. Das Byte wird eingelesen, dekodiert und gleichzeitig der Program-Counter inkrementiert. Bei Mehrbytebefehlen wird dieser Vorgang wiederholt. Anschließend wird der Befehl ausgeführt. Die Abfrage auf READY erfolgt beim Einlesen des OP-Codes. Der Zustand wird erst verlassen, wenn die READY-Bedingung logisch "1" ist.

Die HOLD-Bedingung wird in dem Zustand abgefragt, in dem der Program-Counter ausgegeben wird, ebenso der INTERRUPT. Erst anschließend werden beide Bearbeitungswege getrennt. Der HOLD-Zustand wird erst nach einem Wartetakt, der ein Umschalten der Treiber in dem hochohmigen Zustand ermöglicht, quittiert. Nach Beendigung der HOLD-Bedingung wird der Fetchzyklus neu begonnen.

Auf die INTERRUPT-Bearbeitung soll hier nicht weiter eingegangen werden.

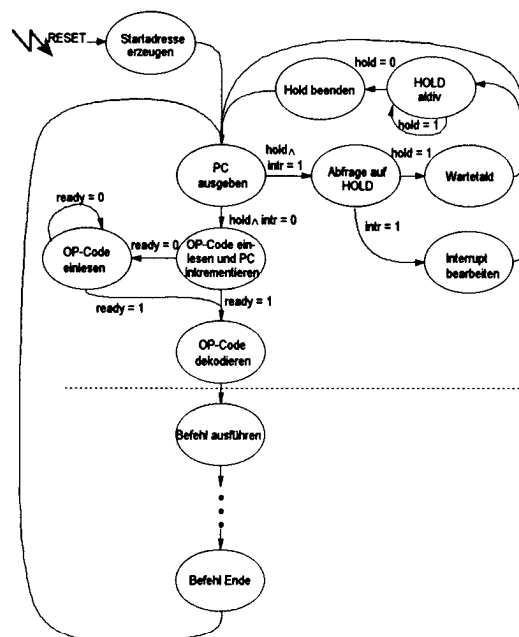


Bild 4: Zustandsdiagramm Fetchzyklus

Plazierung des Kernels auf einem Testchip

Um den Prozessor-Kernel auf seine Funktion hin überprüfen zu können, wird ein Testchip entwickelt und gefertigt. Er enthält außer dem FHOP-Kernel noch weitere Komponenten. Dazu zählen eine interne und externe Busschnittstelle, eine RAM-Makrozelle und ein Pad-Interface für die Ansteuerung der Anschlüsse des ASIC's.

An externen Komponenten werden nur noch ein EPROM oder sonstiger Speicher benötigt, ferner ein GAL oder PLD zur READY-Steuerung. Damit ist der Test des Kernels in allen Funktionen

durchführbar. Gleichzeitig liegt damit für den späteren Anwender ein übersichtliches Beispiel für die Integration des Kernels in eigene Anwendungen vor.

Bei der Entwicklung von ASICs mit Prozessorfähigkeiten platziert man den FHOP - Kernel als fertige Makrozelle auf dem zu entwickelnden Chip und baut die jeweils spezielle Anwendungsschaltung daneben auf. Hinzu kommen Speicher (RAM, ROM) nach Bedarf, sowie Schaltungen für Ports und Schnittstellen. Alle Einheiten werden miteinander verbunden. Damit ist eine anwendungsspezifische Systemintegration auf einem Chip möglich geworden (Application Specific System on Chip, ASSOC).

Der Prozessor-Kernel steht als getestetes VHDL - Modell zur Verfügung. Er kann als Symbol in Schaltpläne eingefügt und effizient auf Systemebene simuliert werden. Eine Abbildung auf Grundelemente einer Bibliothek (Netzliste) ist ebenfalls verfügbar, wodurch ein Softmakro mit definierten Eigenschaften gegeben ist. Mit relativ geringem Aufwand kann von diesen Grundelementen auf verfügbare Bibliotheken abgebildet werden (Hardmacro ES2 - 1,0 µm). Durch die Top - Down - Vorgehensweise wird damit eine relative Unabhängigkeit von der verwendeten Technologie sichergestellt und damit eine längere Lebensdauer des Designs erwartet.

Tabelle 3: Befehlssatz

| Transferbefehle | Arithm. Befehle | Logische Befehle | Shift - Befehle |
|-----------------------|-----------------|------------------|-------------------------|
| MOV r1,r2 | INC r1 | ANA r1 | RAL |
| MOV F,A | DEC r1 | ANI konst | RAR |
| SWP | ADD r1 | ANA M | RLC |
| LDI konst | ADC r1 | ANA adr | RRC |
| LDA M | ADI konst | ORA r1 | |
| LDA adr | ACI konst | ORI konst | |
| STA M | ADD M | ORA M | Sprung - Befehle |
| STA adr | ADC M | ORA adr | JMP adr |
| PSH r1 | ADD adr | XRA r1 | JMC adr |
| PSH F | ADC adr | XRI konst | JMZ adr |
| POP r1 | SUB r1 | XRA M | JMO adr |
| POP F | SUC r1 | XRA adr | |
| | SUI konst | INV | Unterp.- Befehle |
| Sonst. Befehle | SCI konst | CMP r1 | CAL adr |
| NOP | SUB M | CMI konst | RET |
| DIS | SUC M | CMP M | |
| ENI | SUB adr | CMP adr | I/O Befehle |
| HLT | SUC adr | | PIN port |
| SWI n | | | POT port |

Zusammenfassung

Es wurde der Kernel eines 16-Bit Mikroprozessors als VHDL-Modell entwickelt, simuliert und auf eine ES2 1,0 µm-Technologie synthetisiert. Der Prozessor-Kernel ist durch hohe Regularität und insgesamt einfachem Aufbau gekennzeichnet.

Der Datenpfad enthält die Busstruktur, den Registersatz und die ALU. Die Steuerbaugruppe schließt das Mikrocode-ROM und den Befehlssatz ein. Die ALU wurde spezifisch entworfen und aus der VHDL-Verhaltensbeschreibung effektiv synthetisiert. Das Verhalten wurde auf VHDL wie auf Gatterebene überprüft. Eine Taktfrequenz von 33 MHz wird erwartet. Damit steht nun neben dem VHDL-Modell die simulierte Netzliste des gesamten Prozessor-Kernels zur Verfügung. Die Leistungsdaten des FHOP - Kernels sind in Tabelle 4 zusammengestellt.

Dem späteren Einsatz des Mikroprozessor-Kernels als Standardzellen-Makro steht damit nichts mehr im Wege. Dies kann entweder als VHDL-Modell, als Gatter-Modell (Softmakro) oder als fertig platzierte Standardzellen (Hardmakro) erfolgen.

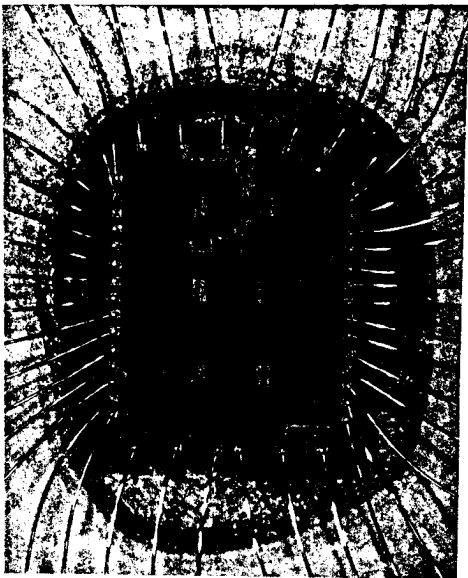
Der Prozessor-Kernel wird auf einem Testchip mit notwendigen Peripherieblöcken wie einem RAM und einer Busschnittstelle derzeit erprobt. Das Ziel dieser Entwicklung ist es, den Prozessorkernel als Standardzellen-Makro für weitere Forschungs- und Entwicklungsarbeiten an der FH Offenburg nutzbar zu machen. Damit können im ASIC-Design-Center der FH zukünftig sehr komplexe Logikschaltungen in einem Halbleiterchip integriert werden.

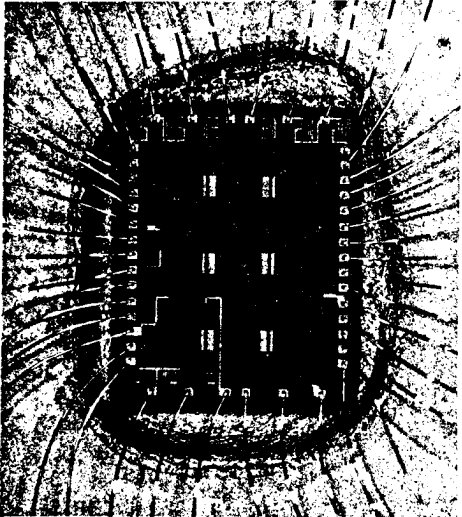
Tabelle 4: Leistungsmerkmale

| | |
|--------------------------|---|
| Busstruktur | - Daten/Adreßbus intern 16-Bit breit - Adreßbus extern 16-Bit breit - Datenbus extern 8-Bit breit |
| Adreßraum | - Memorybereich 64kByte - I/O-Bereich 256 Byte |
| ALU | - 16-Bit Operationen - 8 Flags - Durchlaufzeit \approx 20 ns |
| Registersatz | - 6 Register 16-Bit breit - 1 Flagregister 8-Bit breit - 1 Zwischenregister 8-Bit breit |
| geschätzte Chipfläche | 4 ... 5 mm ² |
| Geschwindigkeit | - Taktfrequenz 33 MHz - minimale Befehlszykluszeit \approx 8 MIPS |
| Peripheriesteuer-signale | - Unterbrechungs-Anforderung - READY-Eingang - HOLD-Anforderung |
| Befehlssatz | - 88-Einbyte Befehle - 3-Zweibyte Befehle - 24-Dreibyte Befehle |

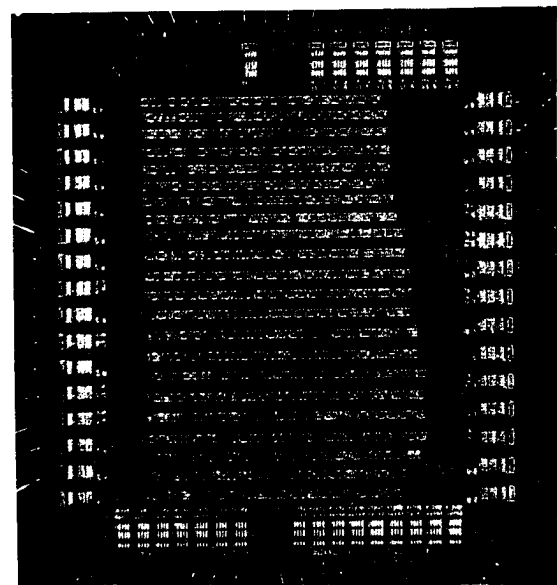
Literaturverzeichnis

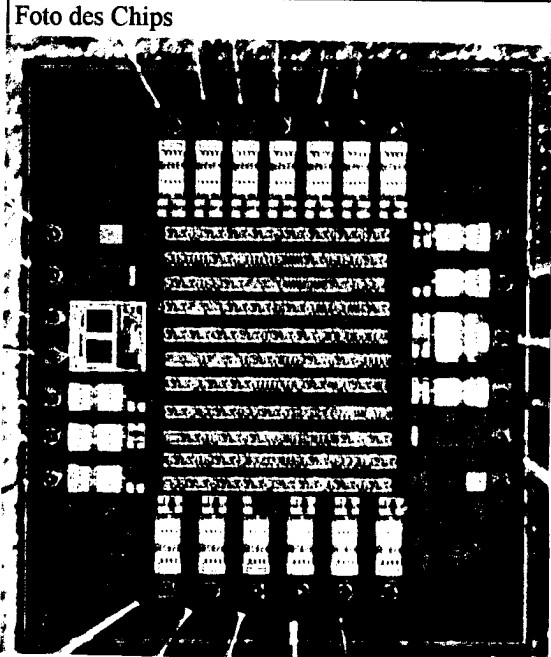
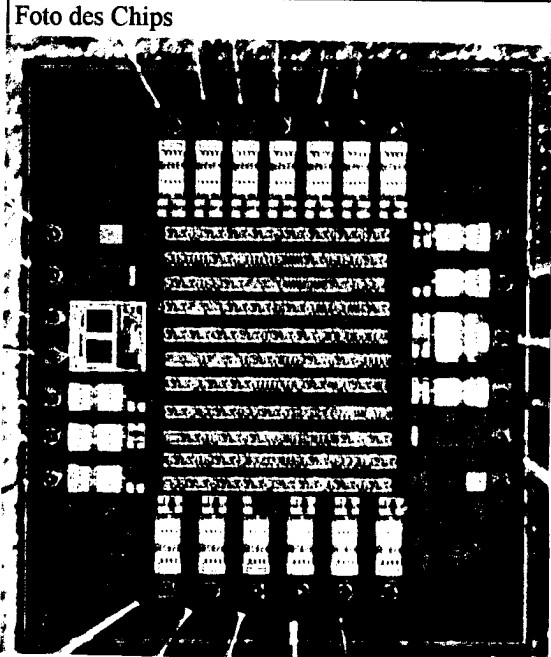
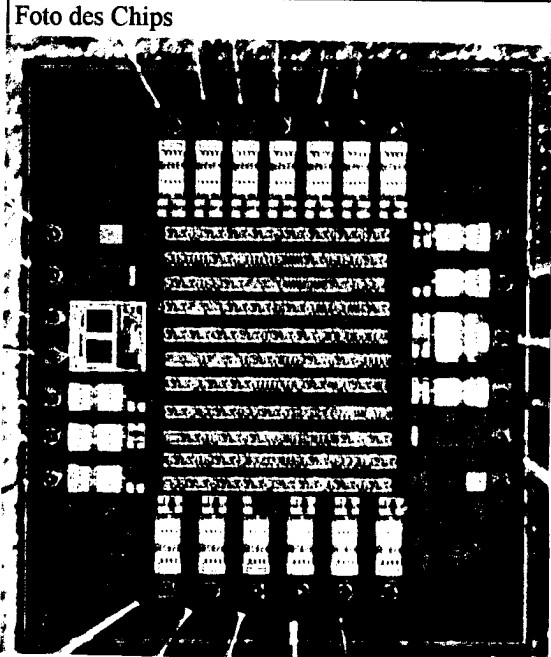
- [1] Zimpfer, Fritz: *Entwicklung des Datenpfads eines 16-Bit Mikroprozessor-Chips mit VHDL*, Diplomarbeit an der Fachhochschule Offenburg, 1994
- [2] Gieringer, Thomas: *Entwicklung der Steuerstruktur eines 16-Bit Mikroprozessor-Chips mit VHDL*, Diplomarbeit an Fachhochschule Offenburg, 1994
- [3] Jansen, Dirk: *Vorlesungsskript Schaltungstechnik*, Vorlesung an der Fachhochschule Offenburg, 1992
- [4] Mentor Graphics Corporation: *Dokumentation der CAE-Tools*, 1992
- [5] D.D. Gajski, et.al: *Computer Architecture, Tutorial. IEEE Catalog Nr. EH0294 - 3*, Computer Society Press, IEEE Washington D.C., 1991
- [6] S.B. Furber: *VLSI RISC Architecture and Organization*, Marcel Dekker, Inc New York, 1989
- [7] Spruth, William G.: *The Design of a Microprocessor*, Springer Verlag, Berlin, 1989

| | | |
|--|--|---|
| Bezeichnung | | |
| <h1>ZF-Verstärker/Costas Loop</h1> | | |
| Funktion: Der Bipolar - Array - Chip enthält einen mehrstufigen regelbaren Zwischenfrequenzverstärker für ZF bis etwa 20 MHz sowie 3 Multiplizierer zu Aufbau eines synchronen Costas - Loop Phasendemodulators. Die Schaltung ist vorgesehen zu Einbau in den FHO - GPS - Empfänger und enthält praktisch den gesamten analogen Schaltungsaufwand zur Demodulation des ZF - Signals. | | |
| Technologie: B 500 A Bipolar - Array (AEG), 1 Metallisierungsebene zu spezifizieren | | |
| Entwicklungsbeginn: Juli 1990 | Lieferung: November 1990 | |
| Anzahl: 30 | gut /schlecht: 24/6 | geprüft: Kern 12/90, Mackensen 1992 |
| Entwurfshistorie: <ol style="list-style-type: none"> 1. Schaltungsentwurf auf Spice - Basis (Michael Kern) Juli 1990 2. Plazierung und Routen von Hand mit CHIPGRAPH gemäß AEG Designrules (Aufwand ca 2 Wochen) 3. Prüfen des Layouts durch FH - Ulm 4. Designabgabe September 1990 (Reinke, Kern) 5. Lieferung November 1990 6. Erste Tests Dez 90 (Kern) : erfolgreich 7. Komplette Vermessung: im Rahmen der Integration in den GPS-Analogteil (Mackensen 1992/93) | | |
| Bemerkungen: Sehr komplexer Chip mit mehreren OPs, 3 Multiplizierern und einem mehrstufigen regelbaren ZF - Verstärker. Spezifizierte Daten wurden erreicht. Übereinstimmung mit Spice - Simulation ist gut. | | |
| Dimensionen: ca 3,5 x 4,5 mm ² | Komplexität: hoch, 90 % voll | |
| Package: CLCC 44 | | |
| Anschlußzahl: 44 Pin | | |
| | | Foto des Chips  |

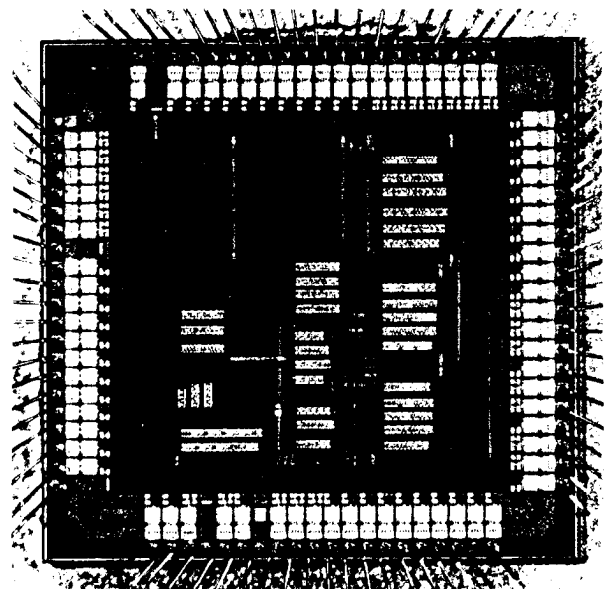
| | | |
|--|--|-----------------------------------|
| Bezeichnung | | |
| <h1>Synchrondemodulator</h1> | | |
| Funktion: Der Chip enthält alle Funktionen, die zum Betrieb eines induktiven Weg - Aufnehmers erforderlich sind, u.a. einen Sinus - Oszillator, einen Treiberverstärker für die Erregerspule (12 V), einen empfindlichen Vorverstärker sowie einen 4 Quadranten - Multiplizierer und einen Tiefpaß. Zum Aufbau eines Wegaufnehmers sind außer der Spule nur noch wenige externe Elemente wie Widerstände und Kondensatoren erforderlich. Die Schaltung arbeitet im NF Bereich bis etwa 40 kHz. | | |
| Technologie: B 500 a Analog Bipolar Array (AEG), entworfen mit SPICE und CHIPGRAPH auf MENTOR - CAE | | |
| Entwicklungsbeginn: Juli 1990 | Lieferung: November 1990 | |
| Anzahl: 30 | gut /schlecht: 25/5 | geprüft: Keßner 12/1990 |
| Entwurfshistorie: 1. Klassischer analoger Entwurf unter Verwendung von SPICE (Keßner) 2. Platzierung und Routen von Hand in einer Ebene mit CHIPGRAPH (Keßner) 3. DRC bei FH - Ulm 4. Designabgabe September 1990 (Keßner, Reinke) 5. Lieferung November 1990 6. Test Dez 1990 (Keßner, Reinke) | | |
| Bemerkungen: Chip arbeitete auf Antrieb mit guten elektrischen Eigenschaften. Übereinstimmung mit SPICE - Simulation gut. Chip auf SMD - Platine montiert und in Aufnehmer integriert . Veröffentlicht in der ELEKTRONIK 14/92 und ELEKTRONIK - CAE Sonderheft 94. Erster Analog - Chip an der FHO ! | | |
| Dimensionen: 3.5 x 4.5 mm ² | Komplexität: hoch, 90 % voll | |
| Package: CLCC 44 | | |
| Anschlußzahl: 44 Pin | | |
| Foto des Chips | | |
|  | | |

| | | | | | | | | | | | | | | | | |
|--|--|------------------------------------|------------|------------------------------------|---------------------------|--------------|-------------------------------|-------------------------------|-----------------|------------------|-----------------------------|-----------|------------------|------------------------|-----------|-----------|
| Bezeichnung | | | | | | | | | | | | | | | | |
| <h1>GPS - Kodegenerator</h1> | | | | | | | | | | | | | | | | |
| Funktion: Der Chip vereinigt alle notwendigen Schaltkreise zur Generierung der im GPS - System verwendeten Gold - Codes . Die Codes können selektiert, zurückgesetzt und gestartet werden. Der Kode kann einem Synchronsignal in der Phase nachgeführt werden, wobei die zeitliche Verschiebung bestimmt werden kann. Der Kode kann ferner gedithert, d.h. um +/- 1/2 Chip - Länge relativ zu einem vorgegebenen Phasenwert verschoben werden. Der Kodegenerator enthält mehrere größere Zählerketten und rückgekoppelte Schieberegister sowie umfangreiche Random - Logik. Die Schaltungen sind nur in geringem Maße regulär. Der Chip ersetzt 3 FPGA - Bausteine Xilinx 3200 /3400 und weist eine Gatterkomplexität von ca 7000 Gattern auf. | | | | | | | | | | | | | | | | |
| Technologie: ES 2 1.5 um CMOS - Prozeß via EUROCHIP auf MENTOR 7.05 CAE | | | | | | | | | | | | | | | | |
| Entwicklungsbeginn: 1991 | Lieferung: 1. EUROCHIP # 33 August 92 2.EUROCHIP #48 Febr. 93 | | | | | | | | | | | | | | | |
| Anzahl: 10 | gut /schlecht: 9/1 geprüft: Wagner Juli 93 | | | | | | | | | | | | | | | |
| Entwurfshistorie: <table border="0" style="width: 100%;"> <tr> <td style="width: 33%;">1. Schaltungsentwurf</td> <td style="width: 33%;">Fiesel, H.</td> <td style="width: 33%;">2 x Xilinx 3200 1 x Xilinx 3400</td> </tr> <tr> <td>2. Überarbeitung zum ASIC</td> <td>Behrens, H.P</td> <td>IMS - Lib, später ES 2 1.5 um</td> </tr> <tr> <td>3. Erstellung Fertigungsfiles</td> <td>Feißt/Mackensen</td> <td>EUROCHIP Run #33</td> </tr> <tr> <td>4. Überarbeitung Vss - Netz</td> <td>Mackensen</td> <td>EUROCHIP Run #48</td> </tr> <tr> <td>5. Test der Prototypen</td> <td>M. Wagner</td> <td>Juli 1993</td> </tr> </table> | | 1. Schaltungsentwurf | Fiesel, H. | 2 x Xilinx 3200 1 x Xilinx 3400 | 2. Überarbeitung zum ASIC | Behrens, H.P | IMS - Lib, später ES 2 1.5 um | 3. Erstellung Fertigungsfiles | Feißt/Mackensen | EUROCHIP Run #33 | 4. Überarbeitung Vss - Netz | Mackensen | EUROCHIP Run #48 | 5. Test der Prototypen | M. Wagner | Juli 1993 |
| 1. Schaltungsentwurf | Fiesel, H. | 2 x Xilinx 3200 1 x Xilinx 3400 | | | | | | | | | | | | | | |
| 2. Überarbeitung zum ASIC | Behrens, H.P | IMS - Lib, später ES 2 1.5 um | | | | | | | | | | | | | | |
| 3. Erstellung Fertigungsfiles | Feißt/Mackensen | EUROCHIP Run #33 | | | | | | | | | | | | | | |
| 4. Überarbeitung Vss - Netz | Mackensen | EUROCHIP Run #48 | | | | | | | | | | | | | | |
| 5. Test der Prototypen | M. Wagner | Juli 1993 | | | | | | | | | | | | | | |
| Bemerkungen: Der erste Fertigungslauf war wegen eines Fehlers im ES 2 - Entwurfskit, durch den die Spannungsversorgung nicht geroutet wurde, nicht erfolgreich. Der Fehler wurde für den 2. Lauf umgangen, sodaß jetzt ein voll funktionsfähiger Chip vorliegt. | | | | | | | | | | | | | | | | |
| Dimensionen: 4,1 x 4,7 mm ² | Komplexität: ca 7000 Gater | | | | | | | | | | | | | | | |
| Package: CLCC 68 | | | | | | | | | | | | | | | | |
| Anschlußzahl: 68 Pin | | | | | | | | | | | | | | | | |



| | | | | |
|---|--|---------------------------------------|-----------------------|--|
| Bezeichnung | | | | |
| <h1>Elektronischer Würfel 2</h1> | | | | |
| Funktion: Elektronischer Würfel mit Anzeige der Augenzahl über 7 LEDs, die direkt angesteuert werden. Der Chip wird durch die Start - Taste erst aktiviert und schaltet sich automatisch nach etwa 10 Sekunden wieder ab. Ein Uhrenquarz dient als Zeitbasis. Der Würfel "rollt" langsam aus, d.h. die Augenzahl wird nicht sofort erreicht, sondern erst nach einer Ausrollzeit von etwa 2 Sekunden. Mit dem Rollen wird ein "Klick" generiert, der durch einen Piezo- Summer hörbar gemacht wird. Das Ergebnis wird mit einem aus 2 Klängen bestehenden Ton signalisiert. Bei den Ziffern "6" und "1" wird jeweils eine kurze Melodie gespielt. Der Chip arbeitet mit 3 V aus einer Lithiumbatterie und nimmt extrem wenig Strom auf. Es werden nur eine Taste, der Uhrenquarz, der Piezo -Speaker und die LEDs als externe Bauteile benötigt. | | | | |
| Technologie: ES 2 1.5 um CMOS Standardzellendesign auf Mentor 7.05 CAE - Tools mit EUROCHIP - ES2 Design Kit | | | | |
| Entwicklungsbeginn: Juli 1993 | Lieferung: EUROCHIP Run #66 Sept. 93 | | | |
| Anzahl: 50 | gut /schlecht: 46/4 | geprüft: Schweiker, Dez. 93 | | |
| Entwurfshistorie: 1. Basiert auf ursprünglichem diskreten Aufbau und IMS - Version (Bernd Reinke, 1991) 2. Neuentwurf im Rahmen einer Studienarbeit Juni 1993 (Schweiker) 3. Designabgabe Juli 1993 (Feißt) 4. Lieferung Muster durch EUROCHIP September 1993 5. Geprüft und auf SMD - Platinen montiert: Dez 93/Jan 1994 | | | | |
| Bemerkungen: Entwurfszeit betrug nach funktioneller Vorsimulation in Hochsprache nur 2 Wochen! Chip war auf Anhieb voll funktionsfähig. | | | | |
| Dimensionen: 2.6 x 2.6 mm ² | Komplexität: ca 4000 Gatter | | | |
| Package: CLCC 64 | | | | |
| Anschlußzahl: 64 Pin | | | | |
| <table border="1"> <tr> <td style="text-align: center;">Foto des Chips</td> </tr> <tr> <td style="text-align: center;">  </td> </tr> </table> | | | Foto des Chips |  |
| Foto des Chips | | | | |
|  | | | | |

| | | |
|---|---|--------------------------------|
| Bezeichnung | | |
| <h1>Hardmacro - Chip</h1> | | |
| Funktion: Der Chip enthält eine Sammlung von Hardmacros, die jeweils für sich geroutet und als eigene Macros platziert wurden. Damit wurden die hierarchischen Routemöglichkeiten der Mentor - Software ausprobiert. Als Zellen wurden realisiert: ALU 8 bit, Schieberegister 8 bit, Zählermodule 4bit und 8 bit, Multiplexer und Decoder. Der Chip wurde so konfiguriert, daß alle einzelnen Macros für sich getestet werden können. | | |
| Technologie: ES 2 1.5 um Standardzellen, hierarchisch geroutet, von Hand platziert auf Mentor 7.05 CAE | | |
| Entwicklungsbeginn: Oktober 1992 | Lieferung: EUROCHIP RUN #62 Juli 93 | |
| Anzahl: 10 Muster | gut /schlecht: noch nicht geprüft | geprüft: siehe unten |
| Entwurfshistorie: 1. Entwurf im Rahmen einer Diplomarbeit (Detlef Pouw) 2. Große Probleme mit der Software, deshalb Beschränkung auf einfache Funktionen 3. Teile des Chips müssen Full Custom realisiert werden, DRC erforderlich 4. Design endgültig abgegeben März 1993 (Feißt, Mackensen) 5. Lieferung im Juni 1993 | | |
| Bemerkungen: Test konnte noch nicht durchgeführt werden wegen Problemen mit der Pin - Belegung und dem Sockel. Verhalten einzelner Macros noch ungeklärt. Stromaufnahme normal. Bearbeiter nicht mehr verfügbar. | | |
| Dimensionen: 4.9 x 4.7 mm ² | Komplexität: ca 3500 Gatter | |
| Package: CLCC 68 | | |
| Anschlußzahl: 68 Pin | | |



| | | |
|--|---------------------------|-----------------|
| Bezeichnung | | |
| <h1>Elektronischer Würfel 1</h1> | | |
| Funktion: | | |
| <p>Elektronischer Würfel mit Anzeige der Würfelaugen durch 7 LEDs. Enthält Taktgenerator mit Uhrenquarz, Start/Stop - Logik und eine Ausrolllogik, d.h die Augen werden nicht sofort angezeigt, sondern erst nach einem Ausrollvorgang von etwa 2 sek.</p> <p>So konfiguriert, daß nur minimale externe Bauteile benötigt werden.</p> <p>Versorgung mit 3 V aus Lithiumbatterie, sehr geringer Stromverbrauch.</p> | | |
| Technologie: | | |
| IMS Gatearray , Gate - Forrest 2.5 um | | |
| Entwicklungsbeginn: | Lieferung: | |
| 1990 | Sommer 1991 | |
| Anzahl: | gut /schlecht: | geprüft: |
| 20 | 16/4 | Reinke |
| Entwurfshistorie: | | |
| <ol style="list-style-type: none"> 1. Basiert auf diskretem Aufbau mit SMD - Logik (Reinke) 2. Integriert alle Einzelbausteine und zusätzlich Ausrollverhalten (Reinke) 3. Entwurf als Gatearray mit IMS - Designkit (Reinke) 4. Fertigung beim IMS (Individualisierung), erste Version wies Kurzschluß in der oberen Zellreihe auf 5. Durch Korektur an 2. Metallisierungsmaske konnte Kurzschluß behoben werden (Nacharbeit IMS) 6. Chips schließlich einwandfrei in Funktion und auf SMD - Platinen montiert (Reinke) | | |
| Bemerkungen: | | |
| <p>Fehler im Designkit führte zu Kurzschluß in der obersten Plazierungsreihe. Konnte durch Nacharbeit im IMS an noch verfügbaren Wafern beseitigt werden. Schaltung arbeitete danach einwandfrei.</p> <p>Erster an der FHO entwickelte CMOS - Chip!</p> | | |
| Dimensionen: | Komplexität: | |
| 4,5 x 4,5 mm ² | ca 2000 Gatter, 50 % voll | |
| Package: | | |
| CLCC 44 | | |
| Anschlußzahl: | | |
| 44 Pins | | |

