

# ASICs und CSICs

Mit der Flexibilität moderner integrierter Schaltkreise nimmt auch die Typenvielfalt und der Abkürzungsdschungel zu.

Klaus P. Eckl

LIT-96

## 1. Allgemeines

Die Abkürzung ASIC (*Application Specific Integrated Circuit*) oder CSIC (*Customer Specific IC*) beschreibt nur ansatzweise die große Gruppe von Bauteilen, welche speziell für den Kunden anwendungsorientiert programmiert werden können. Micro-Controller und EPROMs muß man genauso unter diesen Begriff einordnen, wie Gate Arrays oder PLDs.

Um mit Abkürzungen sorglos umgehen zu können, folgt nun eine Zusammenstellung der wichtigsten Begriffe:

<b>AIM</b>	Avalanche Induced Migration
<b>CLB</b>	Configurable Logic Block
<b>CPLD</b>	Complex Programmable Logic Device
<b>EEPLED</b>	Electrically Erasable Programmable Logic Device
<b>EPLD</b>	(UV) Erasable Programmable Logic Device
<b>FPAD</b>	Field Programmable Address Decoder
<b>FPGA</b>	Field Programmable Gate Array
<b>FSM</b>	Finite State Machine
<b>PLD</b>	Programmable Logic Device
<b>LCA</b>	Logic Cell Array
<b>MPGA</b>	Mask Programmable Gate Array
<b>OTP</b>	One Time Programmable
<b>PCB</b>	Printed Circuit Board
<b>PLE</b>	Programmable Logic Element
<b>GAL</b>	Generic Array Logic
<b>PAL</b>	Programmable Array Logic
<b>PLA</b>	Programmable Logic Array
<b>PLS</b>	Programmable Logic Sequencer
<b>PSD</b>	Programmable System Device
<b>PML</b>	Programmable Macro Logic
<b>UPLD</b>	User Programmable Logic Device
<b>VHDL</b>	VHSIC Hardware Description Language
<b>VHSIC</b>	Very High Speed Integrated Circuit
<b>VLSI</b>	Very Large Scale Integration

## 2. Einteilung der ASIC's

Unter ASIC wollen wir nun Bausteine verstehen, welche weitgehend die Standard Logik wie TTL- oder CMOS-Bausteine ersetzen können.

ASIC	
MPGA	Gate Array Sea of Gates Cell Array Standard Cell
UPLD	FPGA (RAM based, Antifuses) LCA CPLD (EPROM,EEPROM) GAL PLD (Lateral Fuse) PAL

Ein wesentliches Unterscheidungsmerkmal bei den UPLD ist die Art der Programmierung, wie diese auch in Punkt 5 beschrieben wird. MPGAs können nur in Fabriken außer Haus gefertigt oder finalisiert werden. Bei den UPLDs finden wir fertige Bausteine, die aus Katalogen ausgesucht werden können und teilweise oder gänzlich geprüft wurden. Durch die Komplexität des ASIC wird ein Großteil der Konstruktionsarbeit vom PCB (Leiterplatte) in den IC verlagert. Dementsprechend wird der Kunde je nach Komplexität des ASIC mit seiner Eigenverantwortung immer mehr in den Designprozeß einbezogen. Die Zeiten, wo eine Logikschaltung am Papier entworfen, aufgebaut und dann getestet wurde, sind vorbei. Besonders bei MPGAs können zu spät entdeckte Fehler sehr teuer werden, denn eine Korrektur mit Fädeldrähten oder anderen Hilfsmitteln ist nicht mehr möglich. Vielmehr sind mächtige Simulationswerkzeuge erforderlich, und zu mehreren Zeiten der Entwicklung ist eine Überprüfung der Funktion angesagt.

## 3. Welchen ASIC soll ich einsetzen?

Wer möchte nicht gerne seinen eigenen IC bauen? Doch bevor man ans Werk geht, sollte man sich über ein paar Fragen und Tatsachen Klarheit verschaffen:

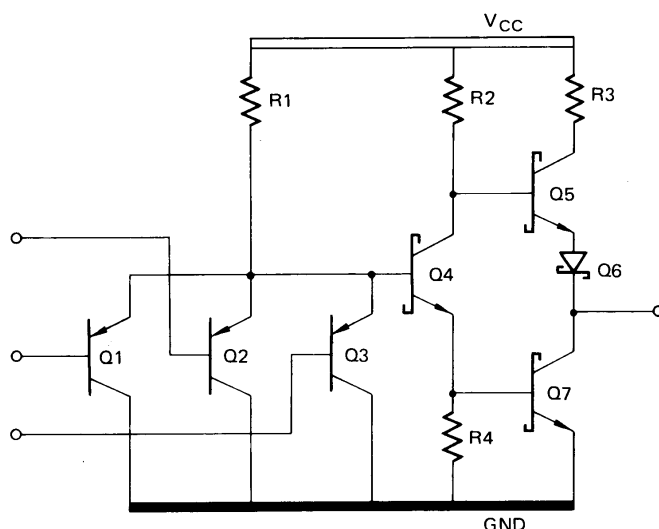
Wo ist meine verlässliche Klagemauer, bei der ich in kritischen Momenten wirklich Hilfe erhalte? Sind die Bausteine, mit denen ich arbeiten will, auch verfügbar?

Warum will ich ASIC einsetzen? Primär muß man immer den Druck des Marktes, also die Kosten als 1. Gebot setzen. Die Kosten können durch die steigende Qualität und Sicherheit, die Platzreduktion, der geringere Leistungsverbrauch, fallende Print- und Gehäusekosten gesenkt werden. Weiters sind durch den Einsatz von ASIC eine größere Nachbausicherheit, eine Komplexität der Logikfunktionen die ohne Integration nicht möglich wäre und besonders bei den UPLDs eine kurze Time-to-Market und eventuell Designänderungen ohne ein Redesign des PCB möglich. Negativ für die Kostenseite wirken sich die Preise für Entwicklungssoftware, Workstations und Programmier-einrichtungen, sowie die Mannkosten der höher qualifizierten Entwickler aus. Der Preis für die Chips selbst spielt meist erst bei sehr großen Stückzahlen eine wesentliche Rolle. Bei Stückzahlen unter 1000 wird es in den meisten Fällen nicht lohnen, ein Redesign durchzuführen, um mit einem billigeren Baustein sparen zu können. Der Spieltrieb des Technikers kann bei der ASIC Entwicklung enorme Kosten verursachen.

MPGAs wird man erst ab Stückzahlen von 1000 einsetzen. Die Wahl der Komplexität der Struktur wird von der Anzahl der erforderlichen „Grundzellen“ abhängen. Diese sind heute von 1000 bis 2.000.000 Grundgattern fertigbar. UPLDs kommen mit mehr als 20.000 Gatterfunktionen bereits weit in den Bereich der MPGAs und werden wegen der geringeren Designkosten sicher bei vielen Anwendungen vorzuziehen sein.

## 4. Mask Programmable Gate Array

Prinzipiell bestehen alle Gate Arrays aus Basic Cells, welche je nach Technologie unterschiedlich plaziert, verdrahtet oder sogar in der Größe (wegen der erforderlichen Stromdichten) variiert werden können. **Bild 1** zeigt eine Basic Cell in Low Power Schottky TTL Technologie.



**Bild 1:** Basic Cell Equivalent Circuit

Der Einsatz von Basic Cells in CMOS bringt die Vorteile dieser Technologie auch für die ASIC. **Bild 2** zeigt eine solche Basiszelle und das Photo der Struktur am Silizium. In **Bild 3** ist die Innenschaltung eines D-Flip-Flops dargestellt und das Photo zeigt den Verbrauch an Siliziumfläche.

Im einfachsten Fall sind im MPGA bereits alle Basic Cells fix plaziert und vorgegeben und der Designer muß nur mehr die Verdrahtung in den zwei letzten metallischen Schichten realisieren.

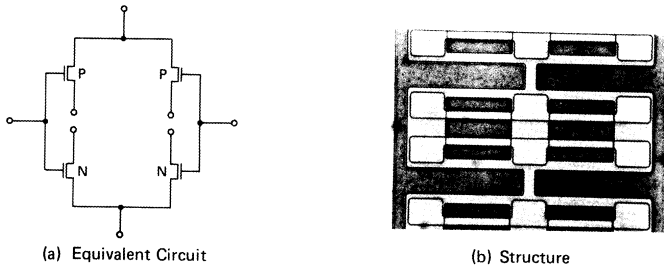
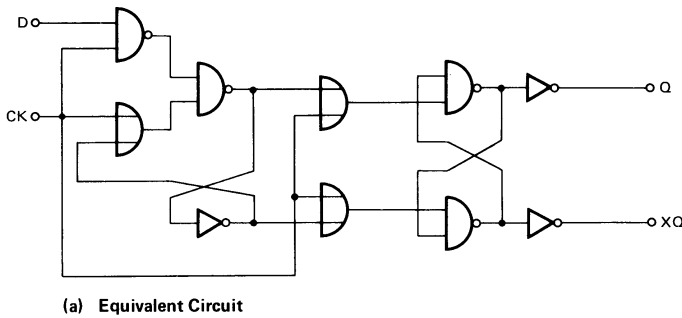
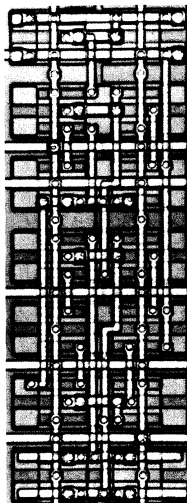


Bild 2: Basic Cell in CMOS Technologie



(a) Equivalent Circuit

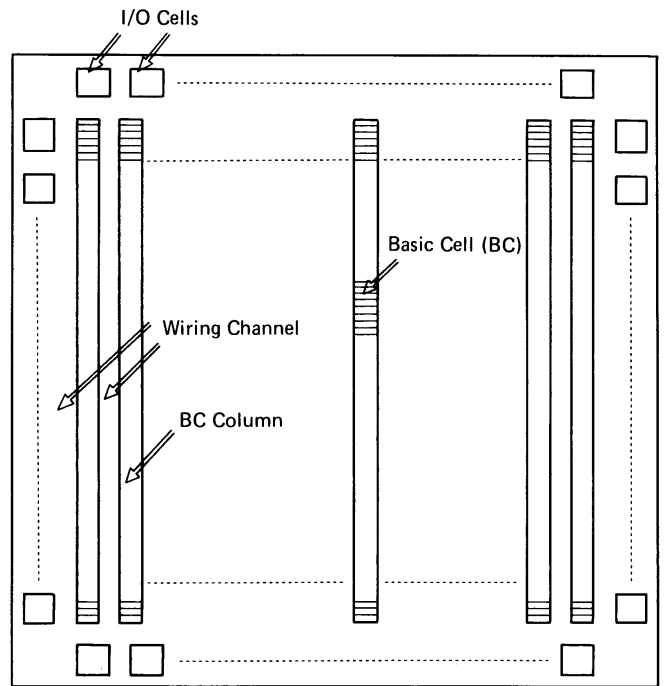
(b) Cell Symbol



(c) Structure

Bild 3: D Flip-Flop eines Gate Array

Zum Design von **Sea of Gates** benötigt man Softwarewerkzeuge ähnlich den CAD Tools für PCBs. Es handelt sich um eine Master Slice Technik. Der Chip ist als Halbfabrikat aus Transistoren mit einer fixen geometrischen Struktur vorgegeben. Von den Firmen werden Bauteilbibliotheken mit den Funktionen von Standard ICs, aber auch komplexeren Strukturen geliefert. Es gibt sogenannte Hardmacros, welche durch ihre vorgegebene fixe Platzierung und Teilverdrahtung auch kurze Laufzeiten garantieren, und Softmacros, welche noch eine beliebige Platzierung zulassen, aber Laufzeitprobleme nicht beherrschen. Der Designer braucht keine detaillierten Kenntnisse über die Technologie. Unter dem Mikroskop erkennt man ein Gate Array an seiner gleichmäßigen Streifenstruktur am Silizium. (Siehe auch Bild 5). Als Schnittstelle zwischen Hersteller und Designer verwendet man meist die simulierte Netzliste. Das Plazieren hat beim ASIC Design eine noch größere Bedeutung als beim PCB Layout. Die Platzierung und das Routen erfolgt daher meist beim Hersteller und der Kunde kontrolliert bzw. simuliert nochmals die nunmehr mit Timing-Information erweiterte Netzliste.



Each column contains 100 Basic Cells, and each chip has 20 columns, making 2,000 BCs. Around the BC matrix there are 72 I/O cells.

Bild 5: Anordnung der Basic Cells in einem Gate Array

**Gate Array** und **Sea of Gates** unterscheiden sich hauptsächlich durch ihre Verdrahtungstechnik. Sie werden manchmal auch als Channelled Gate Array bezeichnet, weil zwischen den in Form einer Matrix angeordneten Gates sogenannte Routing channels geführt werden. Bei den **Sea of Gates** geht Siliziumfläche für die Verdrahtung verloren, weil für diese auch Basic Cells herangezogen werden.

### Full Custom

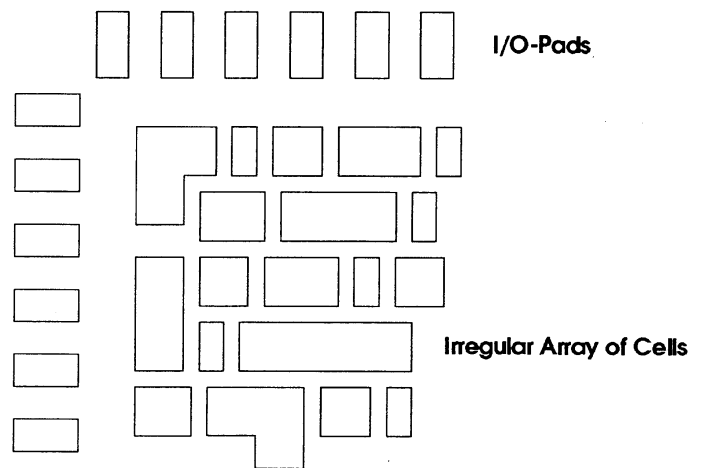


Bild 6: Siliziumstruktur von einem Full Custom Design

Für ein **Standard Cell Design** werden vom Hersteller Bibliotheken mit Hard Macros, welche nicht mehr aus Standardzellen sondern aus unterschiedlich großen Blöcken bestehen, angeboten. Diese fertigen, vom Hersteller getesteten Blöcke (wie RAM, Arithmetikelemente, Logikstrukturen etc.), benötigen minimale Siliziumflächen (Preis). Die Fertigung wird allerdings teurer, weil für den Kunden alle Prozeßschritte wie auch für die Maskenerstellung, kundenspezifisch durchgeführt werden müssen. So wie bei den Gate Arrays gibt es Routing Channels. Platzierung und Verdrahtung erfolgt meist beim Hersteller. Als Designschnittstelle dient wieder die Netzliste.

Beim Full Custom Design bleiben keine Wünsche offen. Der Kunde kann selbst die Größe der Transistoren wählen und alle Techniken, die bisher aufgezeigt wurden, kombinieren. Vielfach werden jedoch die SoftwareTools nicht ausreichen und man wird Arbeiten an die Hersteller vergeben.

In den HTLs wird das Entwickeln von MPGAs wahrscheinlich noch länger ein Traum bleiben. PCAD, das derzeit im Einsatz befindliche CAD Paket besitzt keine Schnittstelle oder Tools für solche Designs. Es besteht jedoch die Hoffnung, daß die Werkzeuge für die UPLD Entwicklung immer mächtiger werden und vielleicht auch das Design von einfachen Sea of Gates und deren Simulation zulassen. Der Traum aller Entwickler ist jedoch, daß ein wirklich einheitliches Netzlistenformat und eine genormte Sprache für die Beschreibung der Logikfunktionen sich durchsetzt.

### 5. User Programmable Logic Device

Im Unterschied zu MPGA werden nach der Fertigung, außerhalb der Fabrik die Bausteine noch modifiziert. MPGAs können vom Hersteller getestet werden. Bei den UPLDs gibt es große Unterschiede. OTP Chips können vom Hersteller überhaupt nicht geprüft werden. Der Hersteller kann nur Richtlinien für die Programmierung erstellen. Der Designer muß sich meist auf die Qualität seines Programmiergerätes verlassen und meist selbst Testmöglichkeiten überlegen. SRAMs oder ASICs mit „flüchtiger Programmierung“ kann der Hersteller wieder zu 100% testen. Besonders schwierig für den Kunden stellen sich löschbare ASICs wie EPROMs dar, weil sich die Eigenschaften des Chips mit jedem Löschvorgang verändern und man nur im Versuch feststellen kann, wann der Baustein unbrauchbar wird. Fuses und Antifuses (Schmelzsicherungen) im halbfertigen Baustein kann man nur bedingt testen. Bei schlechter Programmierung ist ein „wieder Zusammenwachsen“ der Sicherung möglich.

Besonders die Sicherungs- und Verbindungstechnologie der ASICs ist für die unterschiedlichsten Anforderungen wie große Geschwindigkeit, hohe Komplexität, geringer Stromverbrauch oder große Flexibilität verantwortlich:

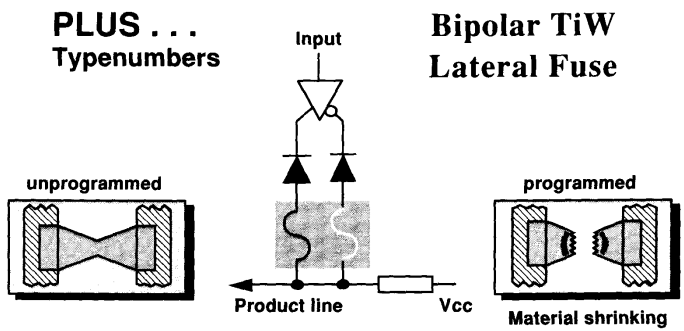


Bild 7: Verbindungstechniken bei UPLDs

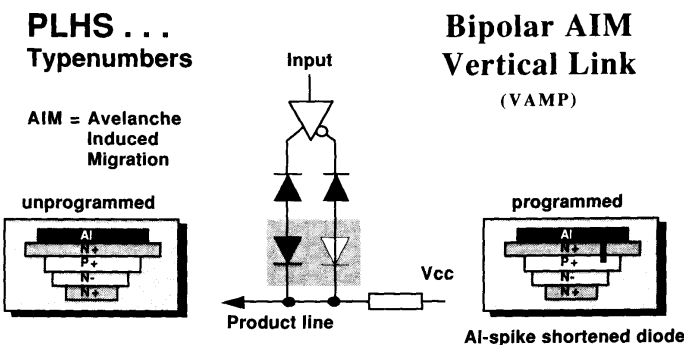


Bild 8: Verbindungstechniken bei UPLDs

### PLC ... Typennumbers

### CMOS - EPROM Floating Gate

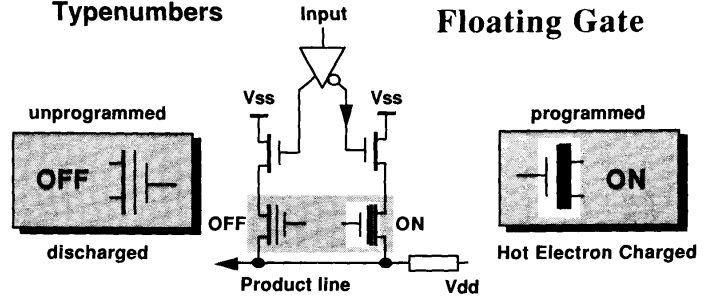


Bild 9: Verbindungstechniken bei UPLDs

Die älteste Technologie stellt das schrumpfen von NiCr Lateral Fuses dar. Heute verwendet man TiW Fuses und erreicht damit Geschwindigkeiten < 5ns (siehe auch Bild 7). Bei der AIM Technik läßt man einen Aluminiumdorn durch das N+ Material in das P+ Material wachsen und schließt so eine Diodenstrecke kurz (siehe auch Bild 8). Wegen der einfachen Meß- und Testbarkeit ist diese Technologie für Military PROMs vorgeschrieben. Die höchste Packungsdichte und den geringsten Stromverbrauch erreicht man wie auch bei den CMOS EPROMs angewendet, mit dem Floating Gate, wo eine eingeschlossene Ladung einen FET Transistor durchsteuert. (siehe auch Bild 9) Durch Energiezufuhr (Bestrahlung mit UV etc.) kann die Gateladung wieder abgeleitet werden. Es gibt aber auch flüchtige Verbindungspunkte: PIP (Programmable Interconnect Points). Der ASIC besitzt für seine Konfiguration ein RAM, welches mit einem BitStream geladen wird. In dem RAM wird die Information für die FET-Schalter solange gespeichert, wie der ASIC mit Betriebsspannung versorgt wird. In der folgenden Tabelle sind einige charakteristische Eigenschaften der Programmiermöglichkeiten zusammengestellt:

	AntiFuse	EPROM	EEPROM	SRAM
Bootdevice	nein	nein	nein	ja
Reprogrammable	nein	ja	ja	ja
Reconfigurable	nein	nein	ja	ja
Volatility	nein	nein	nein	ja
PhysikalischeGröße	klein	mittel	mittel	groß
R on (Ohm)	100	1000	1000	1000
C off (fF)	5	15	15	50
Testability	wenig	gut	100 %	100 %
Erase Time	-	20min	200ms	0
Programmierzeit	langsam	langsam	schnell	schnell
Program.spannung	10-20V	12V	5V	-

Die Aufzählung der UPLDs in Punkt 2 Einleitung ist sehr lückenhaft. Einen besseren Überblick und Einblick in die Strukturen erhält man vielleicht, wenn man die historische Entwicklung verfolgt. Nachdem man begonnen hatte, die Aufgaben der Digitaltechnik universell und flexibel mit programmgesteuerter Logik (Micro-Controller) zu lösen, kam sehr bald der Ruf nach mächtigen Peripheriebausteinen, welche auch universell einsetzbar und wenn es möglich wäre programmierbar wären. Die Stunde der PLE (siehe Bild 10) war gekommen. An den Eingängen angeschlossen befindet sich eine fix verdrahtete Und-Matrix, welche als Adressdecoder dient. Das benötigte Datenmuster wird im Ausgangsteil durch die programmierbare Oder-Matrix realisiert.

### PLE - Architecture

...field programmable memories (PROM, EPROM)

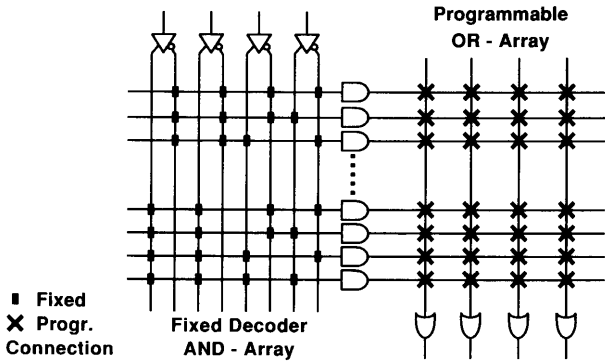


Bild 10: Grundstruktur von ROMs.

Macht man das Eingangsfeld programmierbar und führt die Und-Gatter an den Ausgang, dann erhält man sehr schnelle Adressdecoder. (siehe [Bild 11](#))

### PSD - Architecture

...ultra fast address decoders

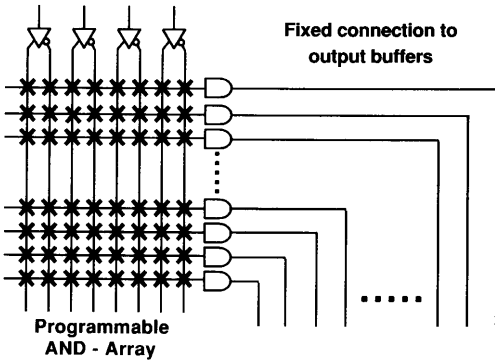


Bild 11: Programmierbarer Adressdecoder

Setzt man an den Ausgang eines PSD noch eine in der Fabrik fix vorverdrahtete Oder-Matrix, dann ist das klassische PAL geboren. (siehe [Bild 12](#)). Die Ausgänge kann man dann noch als Buffer oder Multiplexer (Three State Möglichkeit) oder über gemeinsam getaktete Latches führen. Man spricht dann auch von programmierbaren Macro Cells. ([Bild 13](#))

### PAL - Architecture

...combinatorial and registered fast PLDs

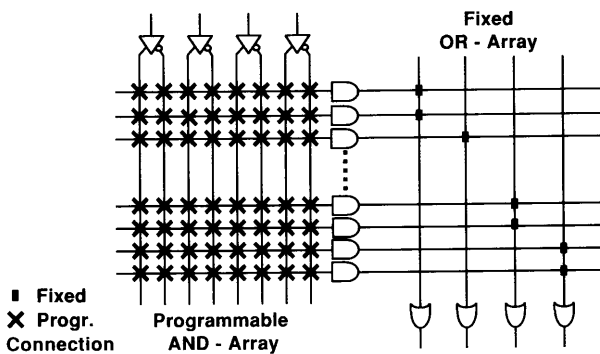


Bild 12: Innenschaltung eines PAL

### Output Macro Cell (OMC)

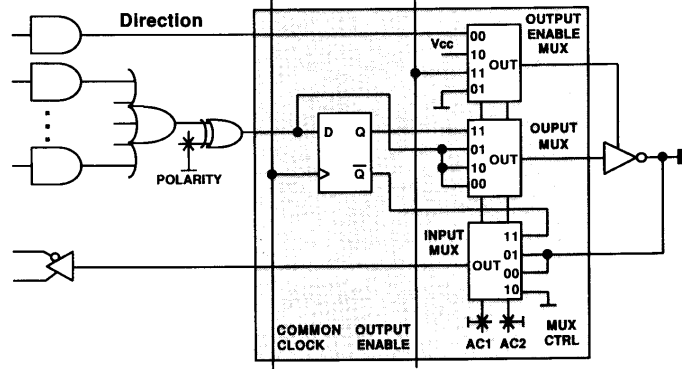


Bild 13: Macro Cell eines PAL

Die weiteren Entwicklungsschritte liegen auf der Hand. Die Ein- und Ausgangsmatrix ausgeführt (PLA), die Flip-Flop Ausgänge werden in die Eingangsmatrix eingebunden (PLS, siehe [Bild 14](#))

### PLS - Architecture

...advanced sequencer circuits

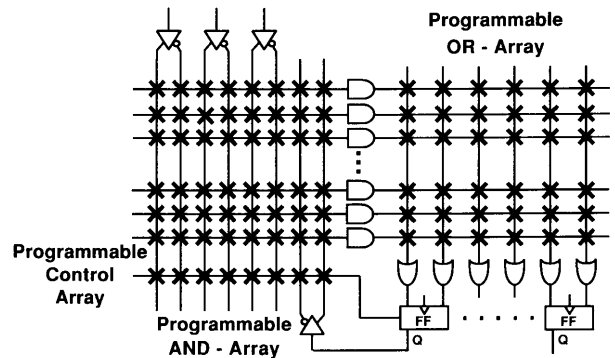


Bild 14: Struktur eines PLD mit rückgekoppelten FlipFlop Ausgängen

In der dritten Generation der UPLDs werden die programmierbaren Matrizen gewaltig vergrößert (PML) oder mehrere PAL Strukturen auf einem Kristall integriert und über spezielle Verdrahtungskanäle verbunden. Für die Bedürfnisse des Micro-Controller-Designs werden spezielle programmierbare Bausteine entwickelt, welche EPROM, RAM und PAL Strukturen beinhalten ([Bild 15](#)).

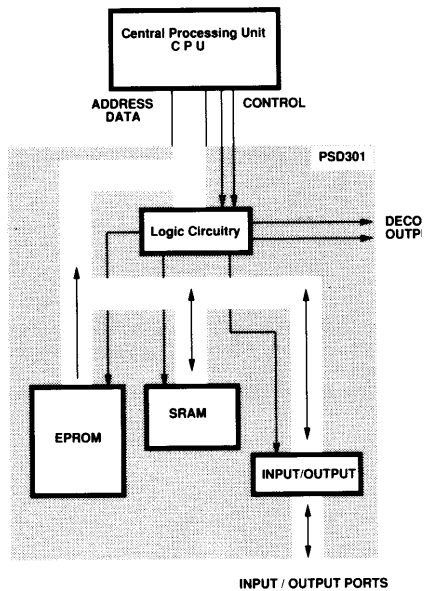


Bild 15: Blockschaltbild eines PDS Devices

Eine besondere Stellung in der Entwicklung der UPLDs nehmen die GALs ein. Am Ausgang sitzen z.B. 8 mächtige Macro Cell mit beliebig beschaltbaren Flip-Flops. Diese können mit vielen (z.B. bis zu 8) Produkttermen einer gewaltigen programmierbaren Eingangsmatrix gesteuert werden. Die Bausteine sind zu 100 % testbar und in einer „low power, electrically reprogrammable E2CMOS“ Technologie ausgeführt. Weil sie kein OTP Device sind, schnell programmiert werden können und sich relativ gutmütig verhalten, sind sie für den Übungsbetrieb von Schulen zu empfehlen.

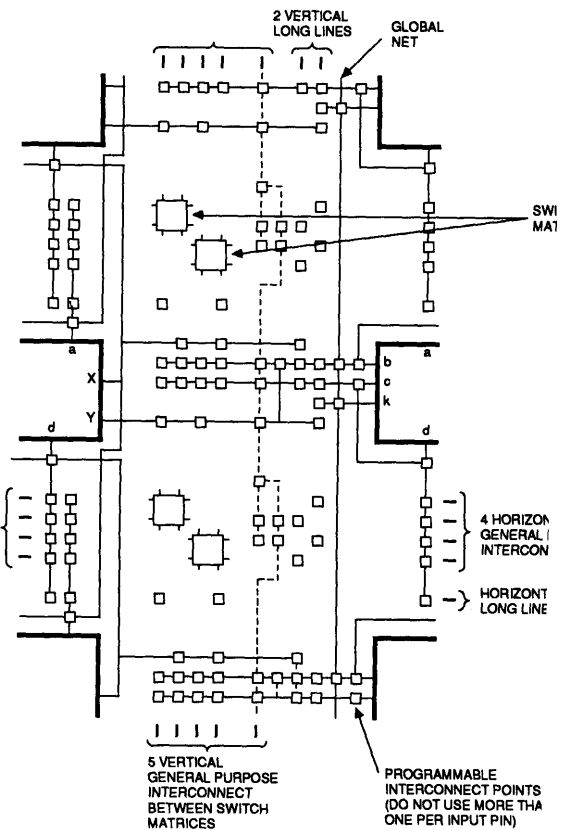


Bild 17: Verbindungstechnik beim LCA

Auf Stichleitungen von den CLB Ein- und Ausgängen sitzen die PIPs. Manche Stichleitungen enden bei den benachbarten CLBs und erlauben so eine sehr kurze und schnelle Verbindung von Blöcken (direct interconnect). In Bild 17 nicht eingezeichnet sind die Leitungen des „General Purpose Interconnect“ welche durch die Gassen der CLBs, über die PIPs führen und in Switch Boxes enden. Diese wieder erlauben eingeschränkt das Umlenken des Signals von horizontaler auf vertikaler Verdrahtung und umgekehrt. Zusätzlich gibt es in der Gassen noch wenige, durchgehende „Long Lines“, welche von kräftigen Bufferverstärkern, die am Rand plaziert sind, getrieben werden. Ihre Hauptaufgabe ist es mehrere Flip-Flops mit einem synchronen Takt zu versorgen, wie dies bei Zählern oder Registern erforderlich ist.

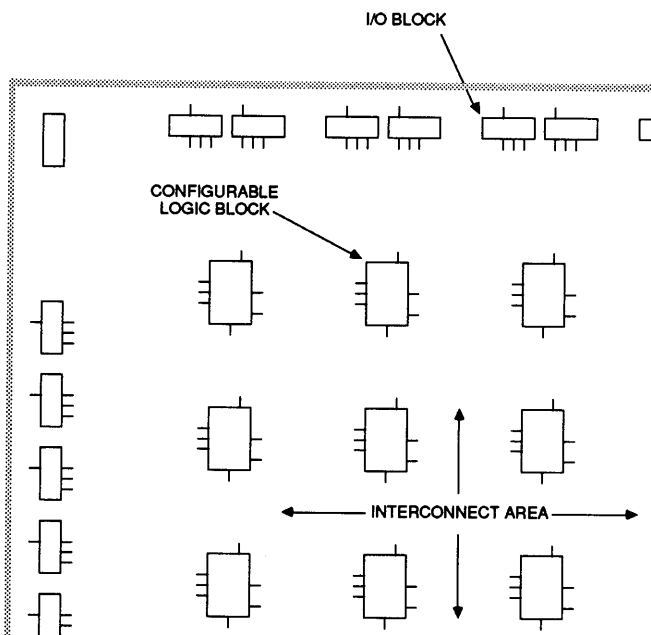


Bild 16: Struktur eines LCA

Die Entwickler der LCAs haben den Designer der MPGA über die Schultern geschaut. Ähnlich der Struktur der Sea of Gates wurden als Macro Cell die CLB in Matrixform angeordnet und I/O Cells an den Rand plaziert. Für die Konfiguration werden die Daten in ein internes RAM geladen und mit der Information FET Schalter der sogenannten PIPs (Programmable Interconnect Points) gesteuert. Mit Hilfe der PIPs und einer Vorverdrahtung zwischen den Blöcken läßt sich ein Verbindungssystem aufbauen. (Bild 17)

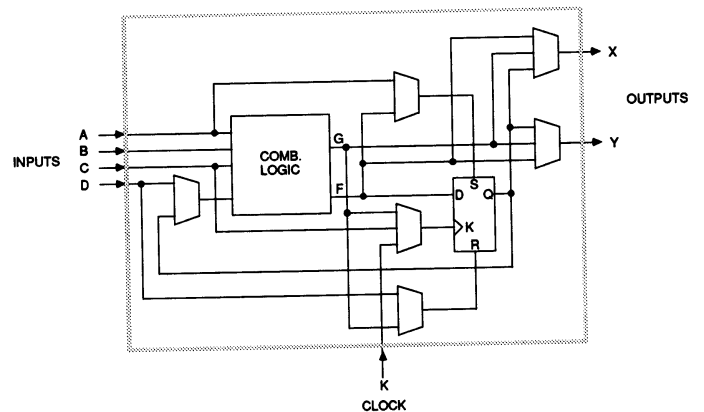
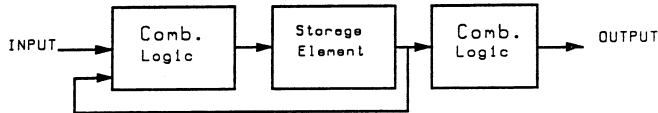


Bild 18: Blockschaltbild eines CLB

Die Mächtigkeit des ASIC ist in den CLB begründet. Der Combinatoric Logic Block ist als Look up Table ausgeführt (RAM) und erlaubt jede beliebige logische Verknüpfung der Eingangsvariablen. Bei wahlweise Verwendung des D-Flip-Flops wird diese Funktion etwas eingeschränkt. Die Trapeze stellen programmierbare Multiplexer dar. Manche Bausteine haben den Bufferverstärker für einen Quarzoszillator bereits integriert. Somit lassen sich sehr komplexe Logic Designs wie auch FSM problemlos realisieren.

## 6. Logic Design an HTLs

Die Digitaltechnik läßt sich in die Kapitel Kombinatorische und Sequentielle Logik unterteilen. In der historischen Entwicklung wurden daher speziell für die Bedürfnisse dieser beiden Gebiete noch kleinere ASICs gebaut. Mit wachsender Größe konnte dann in den FPGA beides großzügig vereint werden. Obwohl man beim Logic Design wie FSM auch asynchrone Ausgänge möglich sind, sollte heute besonders bei komplexeren Applikationen nur eine synchrone Logik mit einem einzigen Taktsignal angestrebt werden. Die Programmierbarkeit von vielen ASICs, wie auch die mächtigen Software Tools erlauben die Wahl der Synchronität mit der sogenannten Moore oder Mealy Machine. (Siehe Bild 19 und 20)



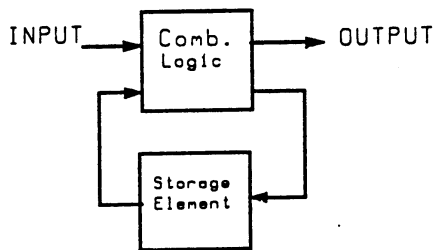
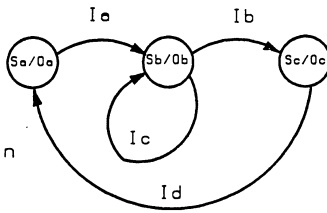
WHERE :

I : Input Condition

O : Output

S : State

(a, b, c, and d are different values assigned to the corresponding variables).



WHERE :

I : Input Condition

O : Output

S : State

(a, b, c, and d are different values assigned to the corresponding variables).

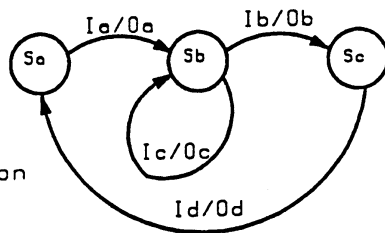


Bild 19 und 20: Moore und Mealy Machine

Bei der Mealy Machine ist der Ausgang eine Funktion der Eingänge und des Registerausgangs, und kann sich somit mit jeder Eingangsvariablen sofort asynchron ändern. Bei der Moore Machine ist der Aus-

gang an ein synchron getaktetes Register angeschlossen. Nun noch ein paar Gedanken zur Zustandskodierung. Binary benötigt wenige Flip-Flops mit vielen vorgeschalteten Produkttermen und eignet sich nur gut für PALs. Gray Codierung ändert zwischen den Zuständen immer nur ein Bit und vermindert die Störanfälligkeit (Metastabilität bei Flip-Flops bei hohen Taktfrequenzen und zu kurzer Set Time). One Hot Encoding (1 of N) verwendet für jeden Zustand ein Flip-Flop. Damit ist diese Codierung sehr aufwendig, aber auch sehr schnell, ermöglicht eine einfache Erkennung von illegalen Zuständen und ist für FPGA gut geeignet.

Und nun zur Schule. Geld und Übungszeit sind Mangelware. Welche Möglichkeiten gibt es, ASIC kennen zu lernen und in Projekten einzusetzen. Der Weg führt sicher über das Erlernen der Grundlagen der Kombinatorischen Logik, mit dem Erstellen von Wahrheitstabellen, dem Entwickeln von Produkttermen und der Konjunktiven Normalform mit Hilfe der Karnaugh Tafeln und letztlich dem Umsetzen in eine Schaltung mit Normsymbolen. Aber an diesem Punkt stellt sich zum ersten Mal schon der Wunsch nach Überprüfbarkeit (Simulation) oder praktischer Realisierung. Nachdem an fast allen Schulen das CAD Paket PCAD standardmäßig vorhanden ist, bietet sich für eine praktische Implementation das LCA-Protoboard von Xilinx an. Es sind Schalter und LED Anzeigen vorhanden und für PCAD gibt es eine Bauteilbibliothek für die LCAs. Nachdem das Erstellen von Schematic für die Schüler keine Schwierigkeiten sein sollte, steht der graphischen Eingabe des Designs nichts im Wege. Als zusätzliches Wissen ist nur die Kenntnis einiger Sondersymbole und Attribute doch von Bedeutung. Wenn und nur wenn keine Fehler gemacht wurden, geht das dreimalige Übersetzen der Netzlisten problemlos. Aus der PCAD Netzliste muß in das Standardformat EDIF übersetzt werden und daraus wieder kann man je eine Netzliste für den Simulator, und eine für die LCA Software gewinnen. Die Übersetzung in die Programmierdaten mit der Xilinx Software gelingt mit den heute verfügbaren Rechner schnell und problemlos. Innerhalb von wenigstens 6 Stunden Labor kann den Schülern ein vorbereiteter Designfluß demonstriert und anschließend von den Schülern ein zu hause ausgearbeitetes einfaches Design mit z.B. einem Zähler und Decoder entwickelt und implementiert werden.

Zum Kennenlernen von CPLDs oder PMLs etc. eignen sich besonders gut die GALs. Voraussetzung ist das Vorhandensein eines einfachen Programmiergerätes. Als Software Tools sind an die Schulen ABEL oder Logic vorhanden. (Easy Abel ist als Share Ware über den PCC-TGM erhältlich). Beide Programme sind sehr mächtig und erlauben eine textuelle Eingabe der Designs. Das mühevollen erarbeiten der Terme mit den Karnaugh Tafeln entfällt und mit wenigstens 4 Stunden Labor läßt sich das Erstellen einfacher Designs wie z.B. Decoder erlernen.

Für die Zukunft zum Designen von komplexeren LCAs wird ein Umstieg auf neuere Software Tools erforderlich sein. Für ein Schematic Entry wäre insbesondere auch ein integrierter Simulator und die Möglichkeit für iterative Designs wünschenswert. Bei der textuellen Beschreibung sollte eine einfacher Weg zur Verwendung von VHDL gefunden werden.

### Literatur

Firmenunterlagen von

Actel; Algotronix; Altera; AT&T; AMD; Fujitsu; NEC; Lattice; MMI; Philips; Signetics; Texas Instruments; WSI; Xilinx

PIB Seminar: ASIC: Balog, Prasky, Thorwartl □