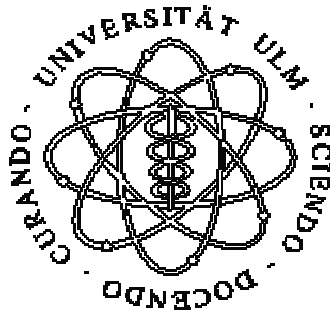


Geschichte und Einführung in Aufbau und Arbeitsweise von FPGA

**Proseminar FPGAs
SS 2003**



Matthias Fuchs

INHALT

1. Einführung

1.1 Geschichte von Logikbausteinen

1.2 Programmierbare Logik

1.2.1 PALs & PLAs

1.2.1.1 Die PAL Familie

1.2.1.2 Eigenschaften von PALs

2. ASICs

2.1 Entwurstile

2.1.1 Full-Custom-Entwurf

2.1.2 Semi-Custom-Entwurf

2.1.2.1 Standardzellen-Entwurf

2.1.2.2 Makrozellen-Entwurf

2.1.2.3 MPGAs & LPGAs

2.2 Kosten von ASICs

3. Aufbau und Funktionsweise

3.1 Logikblöcke

3.1.1 Granularität

3.1.2 Gatteräquivalente

3.1.3 Aufbau von Logikblöcken

3.2 Architektur und Verdrahtung

3.2.1 Kanalorientierte Struktur

3.2.2 Symmetrische Arrays

3.3 Auswahl eines geeigneten FPGA

3.4 Was ist der Vorteil von FPGAs

3.5 FPGA vs. Mikroprozessor

4. Ausblick

5. Literatur

1. Einführung

1.1 Geschichte von Logikbausteinen

In den letzten Jahren konnten programmierbare Logikbausteine enorme Zuwachsraten verzeichnen, heute sind sie unverzichtbare Bauteile beim Design von elektronischen Systemen. Es lassen sich nun Logikfunktionen realisieren ohne auf Standardbauteile zurückgreifen zu müssen. Hiermit lassen sich Bauelemente sowie Platz auf den Leiterplatten einsparen. Da in der Elektronik immer komplexere Schaltungen auf immer weniger Fläche realisiert werden müssen und das bei immer kürzeren Entwicklungszeiten, sind die Entwickler gezwungen sich mit programmierbaren Logikbausteinen zu beschäftigen. Hohe Integrationsdichte, große Flexibilität, geringe Entwicklungszeiten und die damit verbundenen geringen Entwicklungskosten machen programmierbare Logik für fast alle Bereiche der Elektronik attraktiv.

Auch das Militär förderte die Entwicklung dieser Bausteine, um sichere und kleine Bauelemente als Steuerungen von so genannten Intelligenten Waffen einzusetzen.

Zunächst wurden programmierbare Logikbausteine in bipolarer Technik hergestellt, die so genannten PALs. Diese waren vor allem in der Industrie weit verbreitet und wurden von zahlreichen Herstellern in einer fast unübersehbarer Anzahl von Typen hergestellt. Ein erheblicher Nachteil dieser Bauelemente ist die Tatsache, daß diese nur einmal programmiert werden können. Ist eine Verbindung eines in bipolarer Technik hergestellten PALs einmal durchgeschmolzen, ist diese nicht wieder herzustellen.

Die aus dem TI-Praktikum bekannten GALs wurden Mitte der achtziger Jahre von der Firma Lattice entwickelt. Die eigens für diese Bausteine entwickelte EECMOS-Technik kombinierte den hoch entwickelten CMOS-Prozess mit elektrisch löschbarer Speichertechnologie.

1.2 Programmierbare Logik

Unter Programmierbarer Logik versteht man Bausteine, die zwar von ihren physikalischen Eigenschaften von vornherein festgelegt sind, jedoch von den Entwicklern an ihre jeweiligen Bedürfnisse angepasst werden können.

Bei der Entwicklung und Programmierung von Logikbausteinen geht man ähnlich wie bei der Softwareentwicklung vor. Man beginnt mit dem Design, indem man schematische Darstellungen oder Tabellen erstellt. Diese werden in eine Quellcode-Datei umgewandelt, welche wiederum von einem Compiler auf Syntaxfehler überprüft werden. Erkennt der Compiler Fehler, ändert man lediglich den Quellcode und kompiliert ihn neu.

Jetzt kann der Chip programmiert werden. Daraufhin erfolgt wiederum eine Testphase, die wiederum Änderungen im Quellcode nach sich ziehen kann. Arbeitet der Chip korrekt, wird er eingebaut und dient evtl. als Vorlage für Masken.

Hierzu gibt folgende Bausteine :

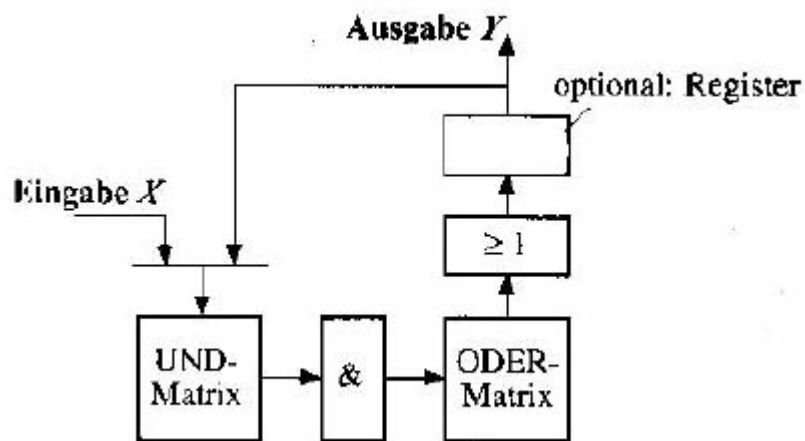
- PLDs (Programmable Logic Devices)

Diese Bausteine sind durch eine zweistufige UND/ODER-Matrix programmierbar und entstanden aus den Technologien für PROMs (Programmable Read Only Memory) und EPROMs (Eraseable Programmable Read Only Memory). Durch die „UNDs“ können aus den Eingangssignalen Produktterme gebildet werden, die wiederum durch die ODER-Matrix disjunkt miteinander verknüpft werden können. Man erhält also eine DNF der Eingangssignale. Nachgeschaltet werden sog. Ausgangs-Makrozellen, die z.B.

Flip Flops, Ausgangsinverter oder ähnliches enthalten. Diese Ausgänge können wiederum auf die Eingänge zurückgeschaltet werden, so daß diese Bausteine gut zur Realisierung von Zählern oder Zustandsmaschinen geeignet sind. Sie haben drei wesentliche Vorteile :

- Sie können günstig als Standardbausteine in großen Stückzahlen gefertigt werden.
- Sie erreichen hohe Gatterkomplexität durch eine speicherähnliche, regelmäßige Matrixstruktur.
- Bewahrung von Firmengeheimnissen, da der Entwurf nicht an die Halbleiterfirmen weitergegeben werden muß.

Zweistufige Schaltnetze, die durch Rückführung und Flipflops ergänzt werden dienen meist als Grundstrukturen (Abb. 1)



(Abb. 1 Vorherrschende Grundstruktur bei PLDs)

Unterscheidet man zwischen festen und programmierbaren Matrizen so ergeben sich folgende Bausteintypen (Tab. 1) :

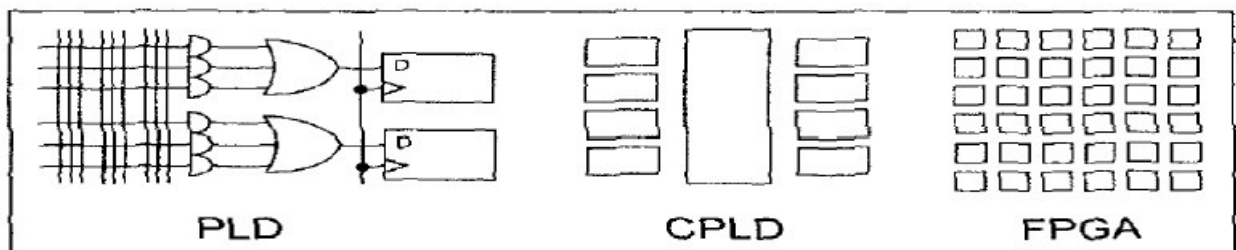
Hauptbausteintyp	UND-Matrix	ODER-Matrix
PROM (<i>Programmable Read Only Memory</i>)	fest	programmierbar
PLA (<i>Programmable Logic Array</i>)	programmierbar	programmierbar
PAL (<i>Programmable Array Logic</i>)	programmierbar	fest

(Tab. 1)

Es existieren zahlreiche Typen von PLDs der bekannteste ist wohl der PAL (1.3).

- CPLDs (Complex PLD)
Hier werden mehrere PLD-Blöcke mittels einer Schaltmatrix miteinander verbunden. Auch diese Bausteine eignen sich besonders für den Aufbau von Steuerwerken, Zustandsmaschinen etc. .
- FPGA (Field Programmable Gate Array)
Dieser Name kommt zu einem wegen seines Aufbaus als Array von Gattern und zum anderen wegen seiner Programmierbarkeit beim Entwickler bzw. Anwender „out in the field“. Hierdurch ist auch schon ein wichtiger Unterschied zu anderen Bausteinen gegeben. Die Funktion eines FPGA ist im voraus nicht festgelegt und somit kann der Entwickler den Baustein seinen individuellen Bedürfnissen anpassen. Ein FPGA besteht aus vielen kleinen Funktionsblöcken. In diesen Funktionsblöcken ist jeweils eine kleine logische Funktion realisiert. Diese einzelnen Blöcke können über ein Netzwerk von Verbindungen miteinander verknüpft werden. Durch diese Verknüpfungen entsteht dann eine personalisierte Schaltung, das fertige Endprodukt. Neben logischen Funktionen stehen auch noch Register und Flipflops auf den Chips zur Verfügung. Somit sind sie auch für speicherintensive Anwendungen geeignet.
Man unterscheidet rekonfigurierbare und nichtrekonfigurierbare Typen. Bei rekonfigurierbaren wird SRAM, EPROM,... verwendet also wiederbeschreibbare Speichertechnologien, in denen die Konfiguration des FPGA gespeichert wird. Bei den nicht konfigurierbaren Typen kann man nur einmal eine Konfiguration in den Baustein schreiben. Hierzu werden sog. Antifuses verwendet.

In der Folgenden Abbildung sind diese Ansätze gegenübergestellt.



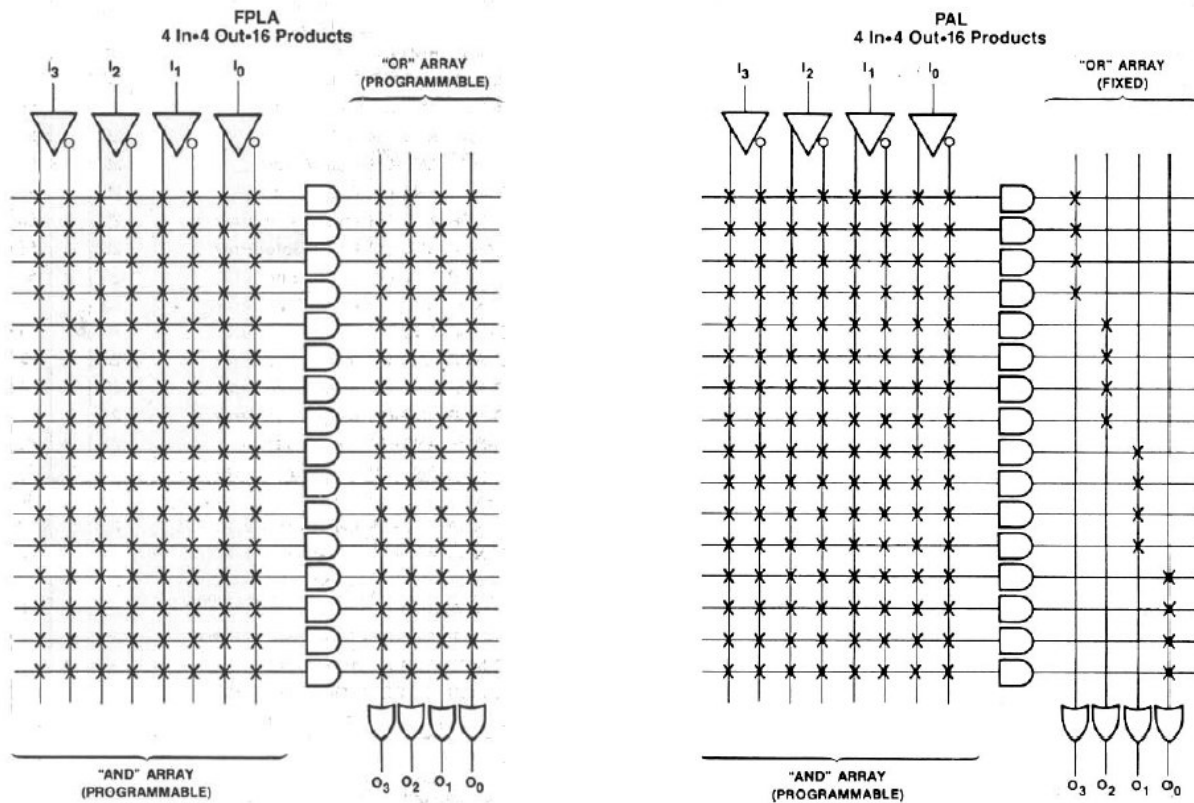
(Abb. 2 Typen programmierbarer Bausteine)

1.2.1 PALs & PLAs

PALs (Programmable Array Logic) sind eine Weiterentwicklung von PROMs und lassen sich genau wie diese nur einmal programmieren. Sie wurden von John Birkner (Monolithic Memories, heute zu AMD gehörend) in der Mitte der siebziger Jahre entwickelt und stellen eine Realisierung von ASIC (siehe Kapitel 2) dar. Im Unterschied zu PROMs bestehen PALs nicht aus einer festen UND- und einer variablen ODER-Matrix sondern aus einer programmierbaren UND- und einer festen ODER-Matrix; Hierdurch kann die Programmierung einzig über das UND-Array erfolgen. Hieraus ergibt sich eine hohe Anzahl von Eingängen durch die UND-Matrix während die ODER-Matrix eine kostengünstige, kompakte Baugröße und aufgrund kurzer Signalwege eine gute Geschwindigkeit erzielt.

Bei den PLAs sind beide Matrizen programmierbar (Abb 3), was eine größere Flexibilität ermöglicht. Jedoch ist dieses Verfahren mit einem höheren Aufwand und größerem

Energieverbrauch verbunden so das in der Praxis meist zu den einfacheren und zuverlässigeren PALs zurückgegriffen wird.



(Abb. 3 UND-/ODER-Matrizen von FPLA und PAL)

1.2.1.1 Die PAL Familie

- PAL20

In dieser ersten Generation existierten 15 verschiedene PAL-Typen, die alle in 20 poligen DIPs integriert waren. Sie waren durch sog. Sicherungs-Matrizen programmierbar (fuse Programmable). Von diesen 15 PAL-Typen gab es 9 in kombinatorischer Logik und 6 für sequentielle Operationen in Registerausführung. Die kombinatorischen PALs wurden häufig als Adressdekodierer in Mikro-Prozessorsystemen verwendet. Jeder PAL besteht aus einer oder mehreren der Folgenden Zellen :

- AND-OR/NOR-Zelle für kombinatorische Logik.
- Registerzelle mit internem Feedback um sequentielle Logik zu implementieren.
- Programmierbare I/O-Zelle um dynamisch die Pinbelegung von Eingabe in Ausgabe und umgekehrt zu wechseln.

- PAL24

Die PAL24 sind eine Weiterentwicklung der PAL20 und bestehen deshalb ebenfalls aus kombinatorischen und Register-Typen. Die kombinatorischen Zellen können nun 20 Eingaben parallel verarbeiten und die Registerzellen wurden auf 10 Ausgänge erweitert. Diese PALs besitzen nun 24 Pins. Der größte Unterschied ist jedoch das Ersetzen der bisherigen Registerzelle durch eine AND-OR-EXOR Zelle, mit der nun eine recht einfache Realisierung von Zählwerken ermöglicht wurde.

1.2.1.2 Eigenschaften von PALs

- Programmierung

Die PALs werden über Schmelzsicherungen (fuses) programmiert. Durch einen kurzen Spannungsimpuls wird an dem gewünschten Knotenpunkten die Verbindung dauerhaft getrennt. Dadurch sind PALs zwar nur einmalig programmierbar, was beim Testen und Entwickeln enorme Kosten verursachen kann. Sie haben aber auch den Vorteil gegenüber (E)EPROMs und GALs, daß sie fast unempfindlich gegenüber Strahlungen und elektrischen Feldern sind. Diese Eigenschaften machen PALs vor allem für Militär und Raumfahrt interessant.

- Geschwindigkeit

Es existieren zwei wichtige Parameter für PALs von denen ihre Geschwindigkeit maßgeblich abhängen. Diese sind die Signalverzögerung bei den kombinatorischen Zellen und die Set-Up-Zeit bei den Registerzellen. Diese zwei Parameter lassen sich in einer Zeitfunktion darstellen und sind gleichzusetzen. Dies nahm folgende Entwicklung :

- 1979 : 40ns
- 1981 : 35ns
- 1982 : 25ns
- 1983 : 15ns
- 1998 : 7.5ns (1032E von Lattice)
- 2002 : 5ns (PALCE16V8H von Lattice)

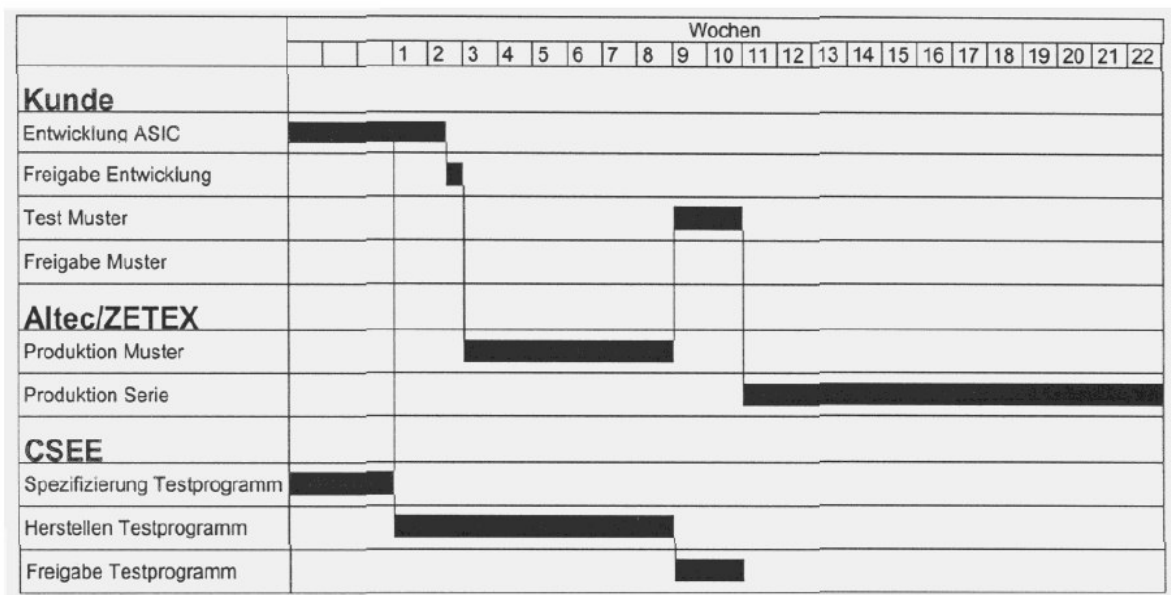
Nahezu alle großen TTL-Hersteller (z.B Texas Instrument, National Semiconductor) akzeptierten den TTL (Transistor Transistor Logik) schnell als neuen Industriestandard und begannen mit großen Lizenzfertigungen. Das JEDEC Komitee erklärte PAL20 (später PAL24) als einen de facto Standard.

2. ASICs

ASICs (Application Specific Integrated Circuit) sind Schaltungen, die für spezielle Anwendungen entworfen und gefertigt werden. Sie werden auch als semi-custom ICs bezeichnet, da der Entwurf zum Teil vom Kunden (Auftraggeber) erfolgen kann und nur noch die Fertigung vom Halbleiterhersteller durchzuführen ist.

Da die Frage nach speziellen Bausteinen für alle möglichen Arten von Anwendungen stetig gestiegen ist, wurde es schnell nicht mehr möglich für jede Anwendung einen eigenen Standardbaustein zu entwickeln. Also sollte der Entwickler selbst in der Lage sein, ICs zur Lösung seiner Aufgaben zu erstellen. Um dies zu vereinfachen wurden CAE-Werkzeuge (Computer Aided Engineering) entwickelt, die mit einer Auswahl von vorgegebenen, einfachen Funktionseinheiten einer bestimmten Bibliothek und deren Verschaltung automatisch die Platzierung und Verdrahtung der Elemente auf dem Silizium durchführt. Dadurch wird eine schnelle, kostengünstige, einfache und vor allem fehlerfreie Produktion ermöglicht.

Der Auftraggeber stellt also den ASIC-Entwurf selbst her. Parallel dazu wird eine Spezifikation eines Testprogramms von einer Firma (CSEE, Abb. 4) entwickelt. Der fertige Systementwurf wird dann an eine Halbleiterfirma (ZETEX, Abb. 4) übermittelt, welche ein erstes Muster erstellt. Dieses Muster wird dann vom Kunden oder der Testfirma mit dem eigens für diesen Baustein entwickelten Testprogramm getestet. Wenn das Muster dann fehlerfrei arbeitet, geht der Entwurf beim Halbleiterhersteller in Serie.



(Abb. 4 Beispiel einer typischen ASIC-Entwicklung)

2.1 Entwurststile

2.1.1 Full-Custom-Entwurf

Hier werden die Bausteine bis zum physischen Layout des einzelnen Transistors auf dem Silizium individuell und vollständig entworfen. Es existieren also keinerlei Vorgaben.

Hierdurch wird beste Ausnutzung der Chipfläche, geringe Verlustleistung und höchste Schaltgeschwindigkeiten ermöglicht. Hierfür muß der Entwickler jedoch sehr erfahren sein. Desweiteren sind intensive Tests des Chips notwendig, um Fehler möglichst früh zu beseitigen. Da dies sehr zeitaufwendig ist treten relativ hohe Kosten auf. Deshalb sind Full-Custom-Entwürfe nur für hohe Stückzahlen rentabel. Im analogen Bereich ist diese Methode jedoch oft unumgänglich.

2.1.2 Semi-Custom-Entwurf

Hier sind einige Teile fest vorgegeben, dies minimiert zwar die Entwicklungsfreiheit senkt jedoch, vor allem bei geringen Stückzahlen, die Kosten. Ein nicht zu vernachlässigender Vorteil hierbei ist, daß die vorgegebenen Teile über längere Zeiträume optimiert und von Fehlern befreit werden können. Hierdurch verringern sich Entwurfsfehler und Aufwand und Entwicklungskosten werden erheblich reduziert.

Bei dieser Technik werden folgende vier Arten unterschieden :

- Standardzellen-Entwurf
- Makrozellen-Entwurf
- programmierbare Gate Arrays (MPGAs und LPGAs)
- feldprogrammierbare Bausteine (FPDs)

2.1.2.1 Standardzellen-Entwurf

Hier kommen (wie der Name schon sagt) Standardzellen zum Einsatz. Diese wurden incl. Layout vom Halbleiterhersteller in langen Prozessen entwickelt, optimiert und getestet. So haben die Zellen meist die gleiche Länge und Breite und lassen sich deshalb lückenlos aneinanderreihen. Es werden jedoch kleine Kanäle zwischen den Bausteinen für die Verdrahtung freigehalten. Je nach Anforderung variieren Anzahl, Größe und Verteilung dieser Kanäle. Die Standardzellen stehen in Form von Bibliotheken zur Verfügung. Sie enthalten alle Zellen mit Angaben zu ihren Funktionen, ihrer Größe und elektrischen Parametern. Kosten und Fehler werden durch das fertige Layout der Zellen reduziert.

2.1.2.2 Makrozellen-Entwurf

Dies stellt eine Weiterentwicklung der Standardzellen-Technik dar. Hier kommen zusätzlich zu den Standardzellen noch sog. Makrozellen zum Einsatz. Diese Makrozellen bestehen aus Standardzellen und können entweder von einem Entwickler oder automatisch von einem Makrozellengenerator zusammengesetzt werden. Besonders Funktionsblöcke mit regelmäßigen Strukturen eignen sich zur automatischen Generierung. Der Entwickler muß hierfür nur die erforderlichen Parameter (z.B. Speichertiefe und Wortbreite) vorgeben. Die fertigen Makrozellen können nun frei auf der Chipfläche platziert werden hierbei kann der Signalfluß berücksichtigt werden, um Packungsdichte und Signallaufzeiten zu optimieren.

2.1.2.3 MPGAs & LPGAs

Bei diesen sog. Gate Arrays sind die Transistor- oder Gaterzellen in Reihen oder Zeilen auf dem Silizium angeordnet. Die bis auf die Verdrahtung vorgefertigte Siliziumscheibe (engl. Wafer) wird dann erst auf Kundenwunsch programmiert. Deshalb kann man Gate Arrays bis zum Fertigungsschritt auch als Standardbausteine bezeichnen. Auch hier findet die Verdrahtung über freigelassene Kanäle zwischen den Reihen bzw. Zeilen statt. Durch große Vorproduktion können Kosten und Lieferzeiten gegenüber den Zellenentwürfen deutlich reduziert werden. MPGAs (Mask Programmable Gate Array) und LPGAs (Laser Programmable Gate Array) unterscheiden sich nur in ihrer Verbindungsart. Bei MPGAs findet die Verdrahtung mittels Belichtungsmasken statt während bei LPGAs schon vorhandene Verbindungen mit einem Laserstrahl durchtrennt werden. Die Programmierung von LPGAs kann in kürzerer Zeit erfolgen, da die aufwendige Herstellung von Belichtungsmasken entfällt.

2.1.2.4 FPDs (Field Programmable Devices) (keine ASICs !!!)

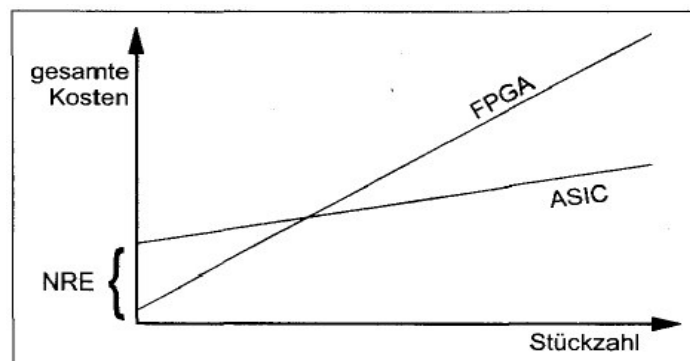
Als UPL (User Programmable Logic) wird diese zweite große Gruppe bezeichnet, zu denen auch FPGAs gehören. Der Aufbau ist mit dem der MPGAs weitgehend identisch. Bei den FPGAs sind jedoch die Funktionen der einzelnen Zellen und die Verbindung zwischen diesen frei programmierbar.

	Standard- baustein	Full-Custom- ASIC	Semi-Custom- ASIC	programmier- bare Logik
IC-Preis	klein	hoch	mittel	klein
Preis/Gatter	gering bis mittel	gering	mittel	mittel
Entwick- lungszeit	–	Monate	Wochen	Stunden bis Tage
Fertigungs- zeit	direkt verfügbar	Wochen	Tage bis Wo- chen	Sekunden bis Minuten
NRE-Kosten	–	sehr hoch	hoch	gering
Silizium- Ausnutzung	sehr gut	sehr gut	schlecht bis gut	schlecht
Entwurfs- änderungen	–	sehr auf- wendig	aufwendig	sehr einfach
Lieferanten	viele	einer	zumeist einer	viele

(Tab. 2 Vergleich der Entwurststile)

2.2 Kosten von ASICs

Als NRE-Kosten (Non recovering Engineering) werden die Kosten für die Entwicklung und Vorbereitung zur Fertigung von ASICs bezeichnet. Wie in Abb. 5 zusehen sind diese Kosten bei geringen Stückzahlen für ASICs sehr hoch. Der endgültige preis pro IC hängt also wesentlich von der benötigten Stückzahl ab. Es ist deswegen stets zu prüfen, welche Realisierungsform im Einzelfall günstiger ist. Wie in der Grafik vereinfacht dargestellt kann man also mit ASICs nur bei großen Stückzahlen eine Kostenersparnis gegenüber FPGAs erzielen.



(Abb. 5 Kosten von ASICs und FPGAs)

3. Aufbau und Funktionsweise

3.1 Logikblöcke

Ein FPGA besteht aus vielen kleinen logischen Einheiten den sog. „Logic Blocks“. Sie enthalten einfache logische Schaltungen, die konfiguriert werden können. Hinsichtlich der Logikblöcke lassen sich folgende zwei Eigenschaften unterscheiden :

- Granularität
- Gatteräquivalente

3.1.1 Granularität

Unter Granularität eines FPGA versteht man die Größe und Komplexität seiner Logikblöcke. Als feinkörnig bezeichnet man ein FPGA wenn die Logikblöcke nur wenig Logik implementieren und daher nur wenige Ein- und Ausgänge besitzen. Die gewünschte Schaltung wird also aus vielen kleinen Einheiten zusammengebaut. Das Zusammenbauen kann von automatischen Werkzeugen übernommen werden, dies bezeichnet man als synthesefreundlich. Die Fläche eines FPGA kann unter Verwendung kleiner Logikblöcke besser ausgenutzt werden, jedoch werden mehrere Verbindungen benötigt, was sich teilweise nachteilig auswirkt.

Als grobkörnig bezeichnet man ein FPGA, wenn die Logikblöcke des FPGA viel Logik implementieren. Dieser Aufbau ist zwar nicht so synthesefreundlich und die Flächenausnutzung ist nicht so hoch, jedoch birgt diese Variante Geschwindigkeitsvorteile.

3.1.2 Gatteräquivalente

Die Gatteräquivalente ist ein Maß für die Anzahl logischer Einheiten auf einem Baustein. Ein Gatteräquivalent entspricht einem NAND mit zwei Eingängen. Dieses Maß wird fast von jedem Hersteller verwendet, jedoch sind nur Bausteine eines Herstellers über die Gatteräquivalente miteinander vergleichbar, da die Hersteller sie oft zu ihren Gunsten verschieden berechnen.

3.1.3 Aufbau von Logikblöcken

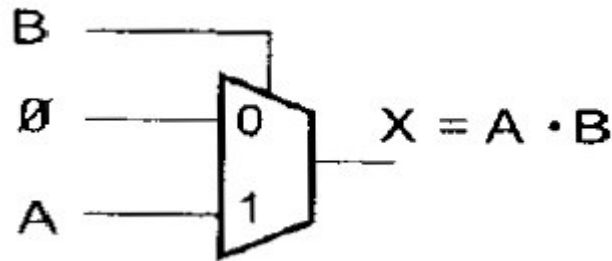
Beim internen Aufbau von Logikblöcken unterscheidet man folgende drei Konzepte :

- Lookup Tables

Diese Technik findet oft in auf SRAM basierten FPGAs ihre Anwendung. Der Speicher wird durch die Eingangssignale adressiert und der Wert der adressierten Speicherzelle entspricht dann dem Ergebnis, welches zu den Eingangssignalen ausgegeben werden soll. Der Speicher wird also derart beschrieben, daß zu jeder Eingangskombination der entsprechende Wert der Logikfunktion ausgegeben wird. Der Speicher kann sowohl beim konfigurieren beschrieben werden als auch von der Anwendung selbst; die Schaltung kann sich dann selbst anpassen.

- Multiplexer

Diese Technik wird oft in Antifuse-FPGAs verwendet. Das Ausgangssignal wird aus den Eingangssignalen mittels eines Multiplexers ausgewählt. Obwohl diese Technik sehr einfach ist lassen sich mit ihr sehr komplexe Schaltungen entwickeln. Der Ansteuereingang des Multiplexers wird hierbei von einem anderen Signal der Anwenderschaltung angesteuert. Um etwa eine UND-Funktion zu erzeugen, kann man den Aufbau aus Abbildung 6 verwenden: Wenn $B=1$ wird A durchgeschaltet. Sonst wird der Ausgang auf 0 gesetzt.



(Abb. 6 Multiplexer stellt AND dar)

- Sea of Gates

Diese Technologie wird derzeit nur von dem Hersteller GateField eingesetzt. Dabei besteht ein Logikblock aus bis zu acht Transistoren, die über programmierbare Verbindungen miteinander verbunden werden können, was die entsprechende Logikschaltung ergibt.

FPGAs enthalten zu den Logikblöcken auch Ein- und Ausgabeblocke, die am Rand des Chips angesiedelt sind und sich um den Signalaustausch zur Außenwelt kümmern. Diese IO-Blöcke können sowohl Schutzschaltungen beinhalten als auch Signalaufbereitung durchführen. Sie können auch Ausgangstreiber zur Aufbereitung der Ausgangssignale haben und Register zur Zwischenspeicherung besitzen.

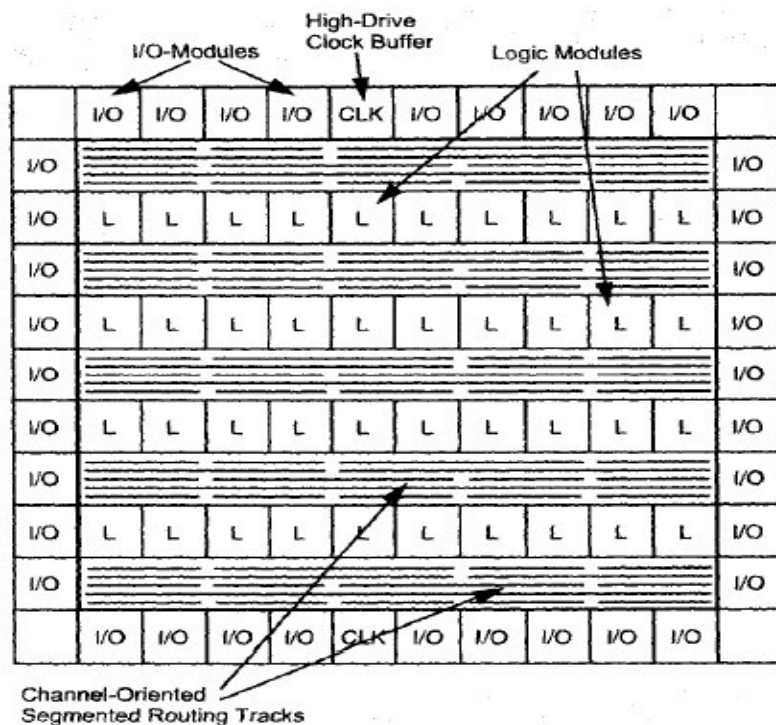
3.2 Architektur und Verdrahtung

Ein wichtiges Merkmal eines FPGA ist seine interne Struktur und die Möglichkeit der Verdrahtung der Logikblöcke.

3.2.1 Kanalorientierte Struktur

Bei dieser Struktur sind die Logikblöcke in horizontalen Reihen angeordnet. Die Verdrahtungsressourcen befinden sich dazwischen. Diese untereinander konfigurierbare Verbindungen bestehen aus verschiedenen langen Verbindungssegmenten. Als Longlines bezeichnet man diejenigen Verbindungssegmente, die über die Gesamte Breite des Chips laufen. Verschieden lange Verbindungsleitungen gewährleisten, daß man nie eine „zu lange“ Leitung benutzen muß. Auch steigt die Geschwindigkeit bei der Verwendung von langen Leitungen im Vergleich mit vielen kurzen miteinander verbundenen Leitungen. Dies kommt dadurch zustande, daß die Leitungen mit Transistoren miteinander verbunden werden und diese die Signallaufzeit erhöhen.

Die Logikblöcke sind im Vergleich mit den Verdrahtungskanälen klein. Ihre Anschlüsse ragen nach oben und unten in die Verbindungskanäle heraus. Diese Ein- und Ausgänge können mittels programmierbarer Verbindungen an die Leitungen im Verdrahtungskanal angeschlossen werden. Es sind auch Verbindungen von oben nach unten möglich, da auf einer anderen Ebene des Chips vertikale Verbindungsleitungen laufen (Abb. 7)

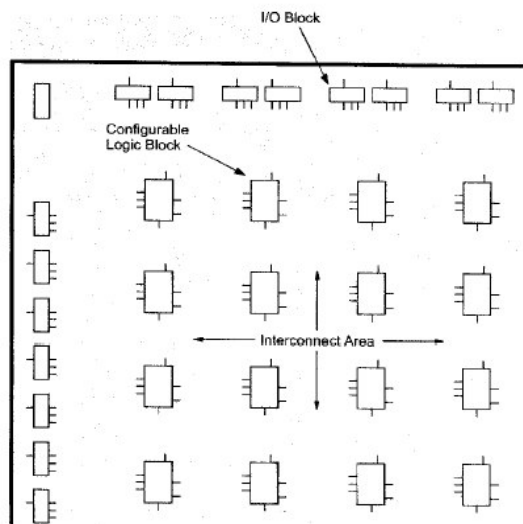


(Abb 7 Kanalorientierte Struktur)

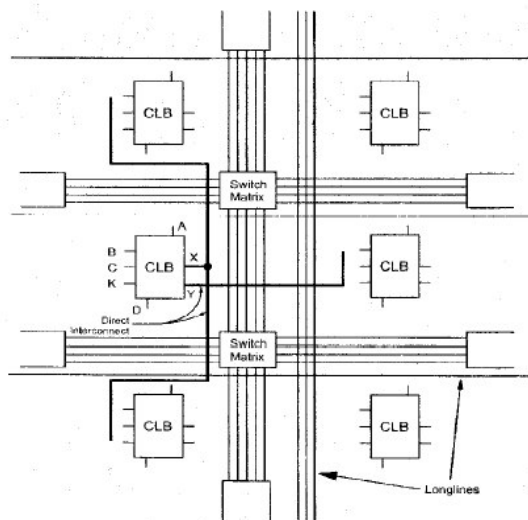
3.2.2 Symmetrische Arrays

Bei diesem Aufbau sind die Logikblöcke regelmäßig auf dem Chip verteilt. Dazwischen befindet sich eine sog. Interconnect-Area, in der Leitungen angeordnet sind. Dieser Aufbau wird von der Firma Xilinx verwendet (Abb. 8).

Es sind verschiedene Arten von Leitungen zu unterscheiden. Zunächst gibt es direkte Verbindungen zu den Nachbarzellen. Genau wie bei der Kanalorientierten Struktur gibt es auch Longlines, welche über den ganzen Chip laufen. Switches gibt es an den Kreuzungspunkten der Interconnect Area. Diese Switches sind mit einer dritten Klasse von Verbindungen, welche sich auf einer anderen Ebene befinden, miteinander verbunden. Der Switch kann ebenfalls konfiguriert werden und ermöglicht es, Verbindungen von Switch zu Switch zu schalten (Abb. 9)



(Abb. 8 Interconnect Area)



(Abb. 9 Leitungen in einem Xilinx-FPGA)

3.3 Auswahl eines geeigneten FPGA

Je nach Einsatzgebiet sind folgende Faktoren zur Auswahl eines geeigneten FPGA zu berücksichtigen :

- Granularität: Anzahl und Größe der Logikblöcke
- Aufbau der Logikblöcke
- Art und Anzahl der Verbindungsleitungen
- Anordnung der Zellen und Verbindungsleitungen
- Programmier Technologie

Vor allem sollte das Verhältnis zwischen Aufbau der Logikblöcke und der Anzahl der Verbindungen stimmen. Ein FPGA mit kleinen Logikblöcken benötigt viele Verbindungen, wenn komplexe Funktionen berechnet werden sollen. Große Logikblöcke wirken sich unter Umständen nachteilig auf den Ausnutzungsgrad aus :

Ausnutzungsgrad = (Anzahl der genutzten Gatter/Anzahl der verfügbaren Gatter).

Heutzutage ist ein Ausnutzungsgrad von 80-95% erreichbar. Anhand von folgenden Leistungsdaten kann die Effektivität von FPGAs bestimmt werden :

- Wie viele Schaltfunktionen kann man auf eine vorgegebene Fläche bringen ?
- Welche Geschwindigkeit kann mit dem System erreicht werden ?
- Wie leicht kann die Schaltung implementiert werden ?
- Ist die Schaltung mit dem Baustein gut zu testen ?
- Wie hoch ist der Stromverbrauch ?

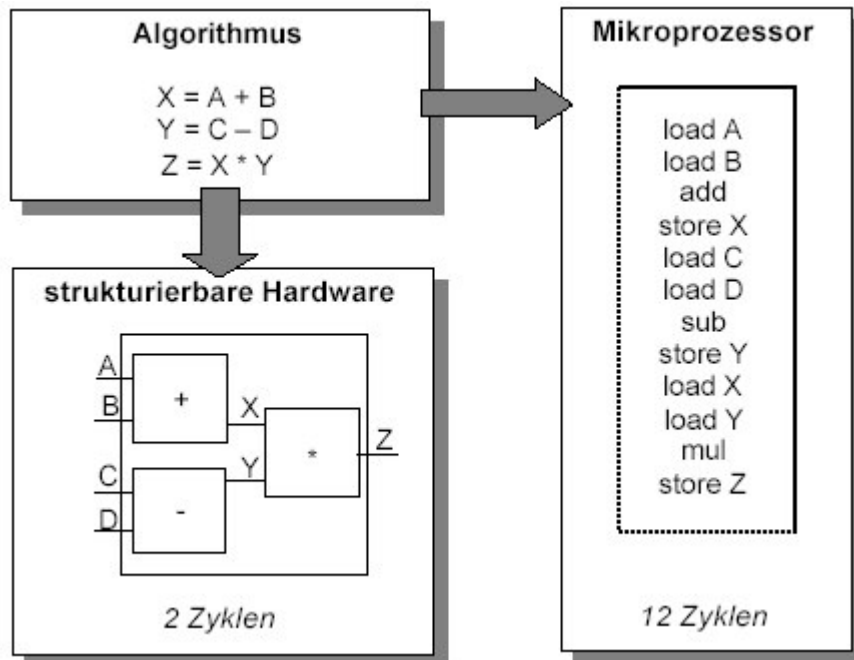
Daneben enthalten viele Chips noch Besonderheiten wie Oszillatoren oder Integrierte RAM-Bereiche. Auch dies kann in die Entscheidung für oder gegen ein bestimmtes FPGA einfließen.

3.4 Was ist der Vorteil von FPGAs

FPGAs schließen die Lücke von Standardbausteinen wie Mikroprozessoren auf der einen und ASICs auf der andere Seite. Der Vorteil von Mikroprozessoren liegt in der preiswerten Massenproduktion und der Entwickler kann den Baustein softwareseitig programmieren. Wie wir später sehen werden, eignen sich Mikroprozessoren nicht für unverzögerte parallele Berechnungen. Hierfür wurden bislang ASICs herangezogen darunter versteht man anwendungsspezifische, integrierte Schaltungen, die für jede Anwendung speziell entwickelt werden müssen. Da diese jedoch von großen Halbleiterherstellern gefertigt werden müssen, rentieren sich nur große Stückzahlen und die Entwicklungszeit für einen Prototypen kann bis zu 18 Monaten dauern. Also sind Standardbausteine für kleine Serien kostengünstig einzusetzen während sich ASICs nur für große Serien eignen. FPGAs schließen genau diese Lücke, da einerseits der physikalische Aufbau immer der gleiche ist (günstige Massenproduktion) andererseits kann der Entwickler die Bausteine genau auf seine Bedürfnisse abstimmen, indem er eine Konfiguration in den FPGA lädt. Dies kann er beliebig oft wiederholen. Nachdem die Konfiguration feststeht, ist auch das Zeitverhalten berechenbar, man kann demnach auch zeitkritische Anwendungen realisieren. Auch ist das Laden der Konfiguration in den Baustein relativ schnell. Bei aktuellen gebräuchlichen FPGAs dauert die Konfiguration ca. 10-250 ms. Die genaue Dauer ist dabei von der Länge des Bitstreams und der maximalen Taktrate, mit welcher der Bitstream eingeschrieben werden kann, abhängig.

3.5 FPGA vs. Mikroprozessor

Anhand der Abbildung 4 wird deutlich, daß die Leistungsfähigkeit von strukturierbaren Logikschaltungen um ein vielfaches über der liegen kann, was mit herkömmlichen Mikroprozessoren erreichbar ist :



(Abb. 10)

Unter der Annahme, daß das System nur den in der Abbildung 4 gezeigten Algorithmus auf einem kontinuierlichen Datenstrom ausführen muß, müßte der Mikroprozessor die zwölffache Taktfrequenz erreichen, um die gleiche Rechenleistung aufzuweisen.

4. Ausblick

Der Trend geht eindeutig weg von den Antifuse-Chips hin zu den auf SRAM basierten Bausteinen. Für die Hersteller von FPGAs sind dabei folgende Gesichtspunkte ausschlaggebend:

- Mit auf SRAM basierten FPGAs lassen sich höhere Gewinne erzielen
- Der Herstellungsprozess von SRAM basierten FPGAs wird gut beherrscht, während die Herstellung von Antifuse-FPGAs nicht zu den gängigen Herstellungsprozessen gehören.
- Auch werden FPGAs als Bestandteile rekonfigurierbarer Logik eingesetzt, was mit Antifuse-Technologie nicht möglich ist.

Dies gilt jedoch nur für die Hersteller von FPGAs und sind für die Auswahlkriterien eines Entwicklers nicht von großer Bedeutung.

5. Literatur

[1] Wannemacher, Markus : Das FPGA Kochbuch; Thompson Publishing, 1998

[2] Oldfield, Dorf : Field-Programmable Gate Arrays; John Wiley & Sons, 1995