

Digitaltechnik

Kapitel 9, Programmierbare Logik

Prof. Dr.-Ing. M. Winzker

*Nutzung nur für Studierende der Hochschule Bonn-Rhein-Sieg gestattet.
(Stand: 21.03.2022)*

9.1. Physikalische Programmierung

- Physikalische Programmierung kann über drei verschiedene Mechanismen erfolgen:
 - Brennen von Sicherungen
 - Programmierung von EEPROM-Zellen
 - Laden eines SRAM-Speichers

Brennen von Sicherungen

- Bei dieser Methode kann ein Baustein nur einmal programmiert werden. Dies bezeichnet man als **OTP** (engl. „One-Time-Programmable“).
- Ursprünglich wurden dünne Metallbahnen durch hohe Ströme durchgebrannt.
 - **Nachteile:** Hoher Widerstand im unprogrammierten Zustand sowie hohe erforderliche Ströme.
- Heute werden **Antifuses** eingesetzt, also Verbindungen, die durch Programmierung geschlossen werden.
- Eine Antifuse wird z.B. gebildet, durch eine spezielle Schicht (u.a. mit Wolfram, engl. „Tungsten“) zwischen zwei Metalllagen eines CMOS-Prozesses.

Physikalische Programmierung (II)

Programmierung von EEPROM-Zellen

- Grundzelle ist ein Feldeffekttransistor mit „**Floating-Gate**“.
- Das Floating-Gate ist elektrisch isoliert und kann über einen Tunnel-Effekt vom Control-Gate elektrisch geladen werden.
- Die Ladung im Floating-Gate steuert dann den Kanal des Feldeffekttransistors an.
- Die Programmierung kann ca. 1000 bis 100.000 mal geändert werden.

Laden eines SRAM-Speichers (Statisches RAM)

- Die Konfiguration wird in SRAM-Zellen (oder Flip-Flops) des FPGAs gespeichert.
- Die Konfiguration ist „**flüchtig**“ (engl. „volatile“), also nach Abschalten der Versorgungsspannung verloren.
- Das FPGA muss nach Anlegen der Versorgungsspannung von einem externen EEPROM oder einer CPU geladen werden.
- Die Programmierung kann beliebig oft, auch im Betrieb, geändert werden.
 - Das externe EEPROM kann auf z.B. 100.000 Programmierungen begrenzt sein.

Physikalische Programmierung - Vorteile und Nachteile

Antifuse

- ± Nur einmal programmierbar (kann Vorteil und Nachteil sein)
- + Geringer Platzbedarf
- Spezielle CMOS-Technologie erforderlich

EEPROM

- + Wiederprogrammierbar
- Spezielle CMOS-Technologie erforderlich

SRAM-basiert

- + Kein spezieller CMOS-Prozess erforderlich. Durch Verwendung der neusten CMOS-Technologie schnell, kostengünstig und hohe Kapazität
- + Im Betrieb umprogrammierbar
- Externes EEPROM erforderlich (oder Programmierung über CPU)
- Benötigt nach Einschalten Zeit bis zur Funktionsfähigkeit (ab ca. 1-10 Sek.)

Marktanteile: 1) SRAM-basiert 2) Antifuse 3) EEPROM

9.2 Programmable Logic Device

PLD- und FPGA-Begriffsvielfalt

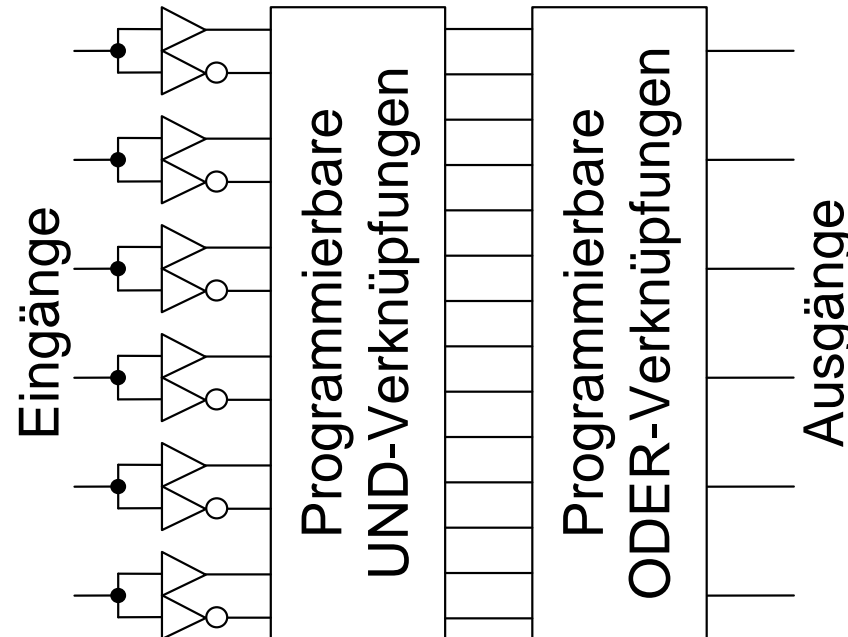
- **PLD** steht für „Programmable Logic Device“ und ist der Oberbegriff für programmierbare Schaltungen mit einer **UND-ODER-Struktur**
- Hier wird im Folgenden unterschieden zwischen:
 - **SPLD**: „Simple PLD“, also PLDs kleiner Komplexität
 - **CPLD**: „Complex PLD“, also PLDs höherer Komplexität
- **FPGA** steht für „Field-Programmable-Gate-Array“

SPLDs

SPLDs („**Simple Programmable Logic Device**“) haben eine zweistufige **UND-ODER-Struktur** (plus Inverter)

- Die Eingangssignale werden invertiert und nicht invertiert bereit gestellt
- Eine programmierbare UND-Verknüpfung erzeugt Produktterme
- Eine programmierbare ODER-Verknüpfung erzeugt Ausgangswerte

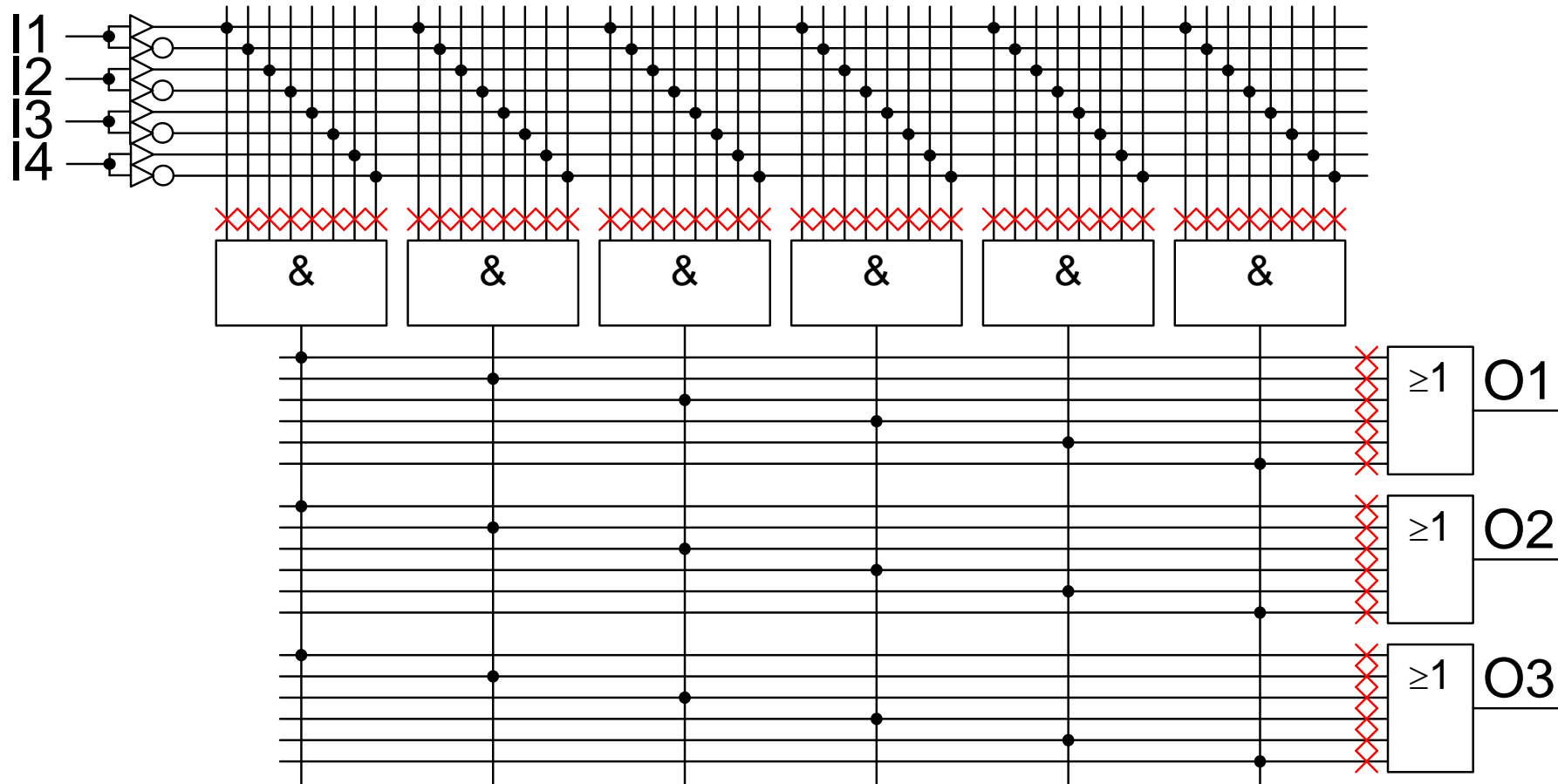
- Jede Logikfunktion kann durch ihre Disjunktive Normalform und Logikminimierung programmiert werden



SPLD-Struktur

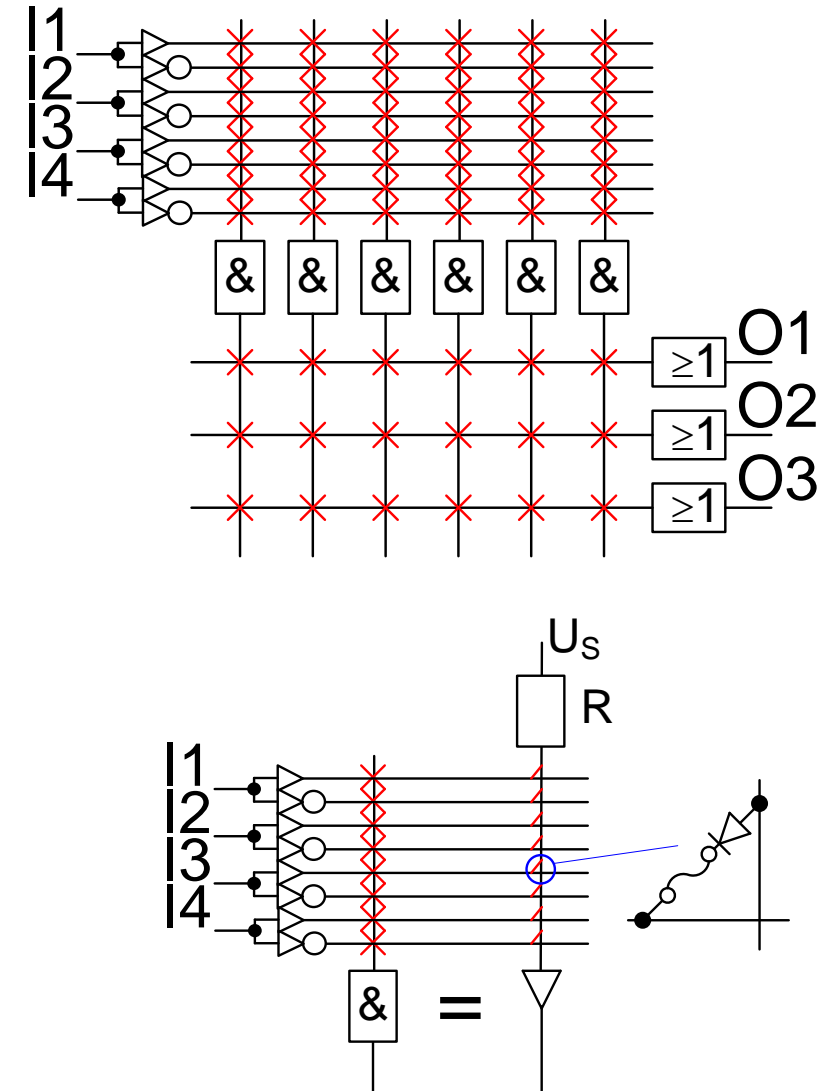
Blockschaltbild für (beispielsweise) 4 Eingänge, 6 UND-Terme, 3 Ausgänge

✗ kennzeichnen programmierbare Schalter, • kennzeichnen feste Verbindungen



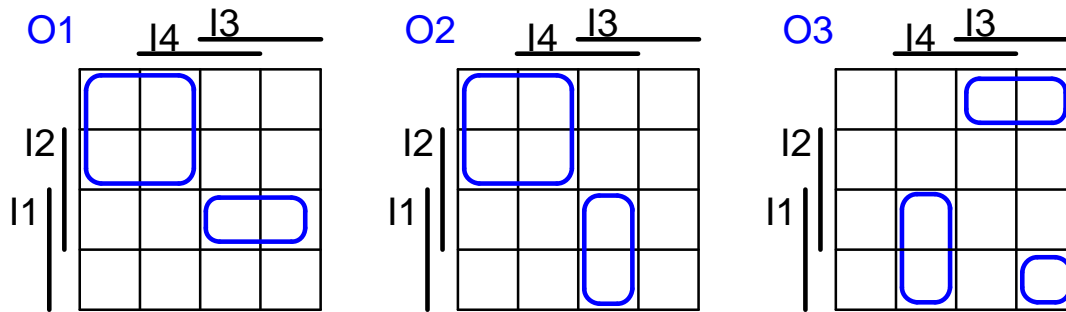
SPLD-Struktur (II)

- Zur besseren Übersichtlichkeit werden die Eingänge der UND-, ODER-Verknüpfungen zusammengefasst
- Die Darstellung entspricht der physikalischen Anordnung
 - Die UND-Verknüpfung wird durch einen Verstärker oder Inverter realisiert
 - Der Verstärker-Eingang liegt über einen Widerstand an Versorgungsspannung
 - Eine Verbindung zum Eingang (Fuse/ Antifuse) kann den Verstärker-Eingang nach Masse ziehen
 - Eine Diode verhindert eine Rückwirkung
- Die Ausgabe der UND-ODER-Stufe kann in **Flip-Flops** gespeichert werden
 - Implementierung von Automaten

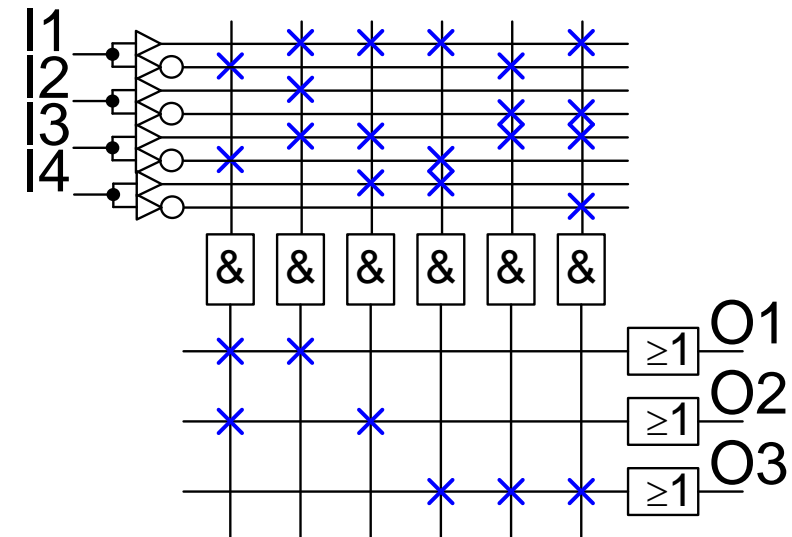


Beispiel: Programmierung eines SPLDs

- Drei Ausgänge abhängig von vier Eingangssignalen
 - Die Bestimmung der Terme erfolgt über Logikminimierung
 - Produktterme können für mehrere Ausgänge verwendet werden

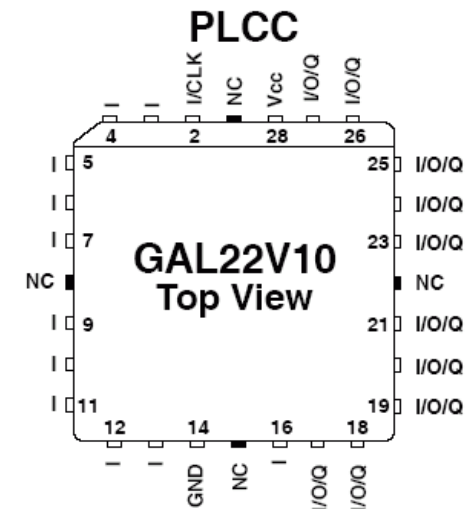
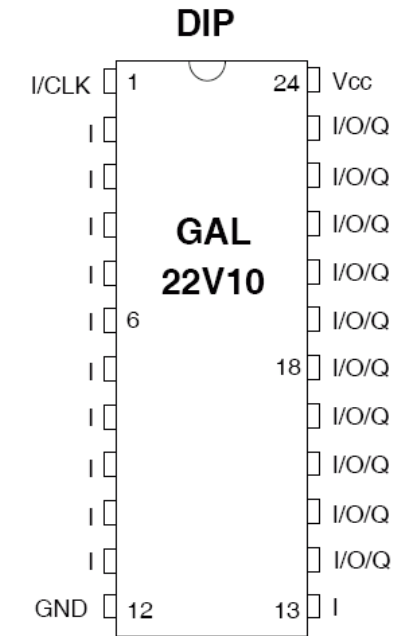


$$\begin{aligned}
 O1 &= (\overline{I1} \& \overline{I3}) \vee (I1 \& I2 \& I3) \\
 O2 &= (\overline{I1} \& \overline{I3}) \vee (I1 \& I3 \& I4) \\
 O3 &= (I1 \& \overline{I3} \& I4) \vee \\
 &\quad (\overline{I1} \& \overline{I2} \& I3) \vee \\
 &\quad (I1 \& \overline{I2} \& I3 \& \overline{I4})
 \end{aligned}$$

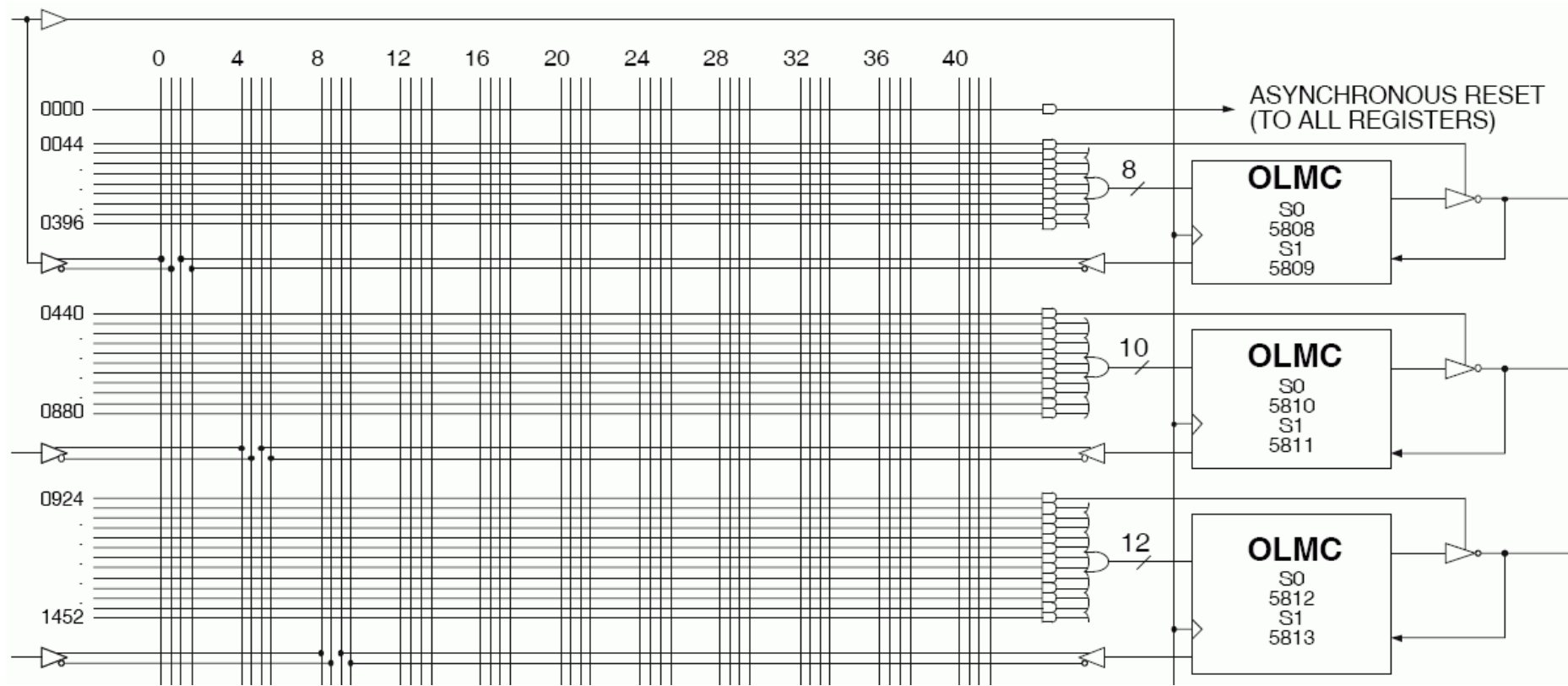


Produkt: GAL22V10 von Lattice

- Datenblatt unter <http://www.latticesemi.com>
- 22 Ein-, Ausgänge (plus VCC, GND)
- 10 UND-ODER-Verknüpfungen
 - Je zwei Verknüpfungen mit 8, 10, 12, 14, 16 Termen
- Ein-/ Ausgänge:
 - I: Eingang
 - I/O/Q: Eingang, Ausgang, FF-Ausgang
 - I/CLK: Eingang oder Takt
 - NC: „Not Connected“
 - VCC, GND: Versorgungsspannung
- Maximale Verzögerungszeit: 4 ns
 - ➔ 250 MHz Taktgeschwindigkeit
- Wiederprogrammierbar, aber nur 100 Lösch-/ Schreibzyklen spezifiziert
- Kompatible Produkte z.B. von Atmel verfügbar
- Preis: wenige Euro



Produkt: GAL22V10 von Lattice (II)

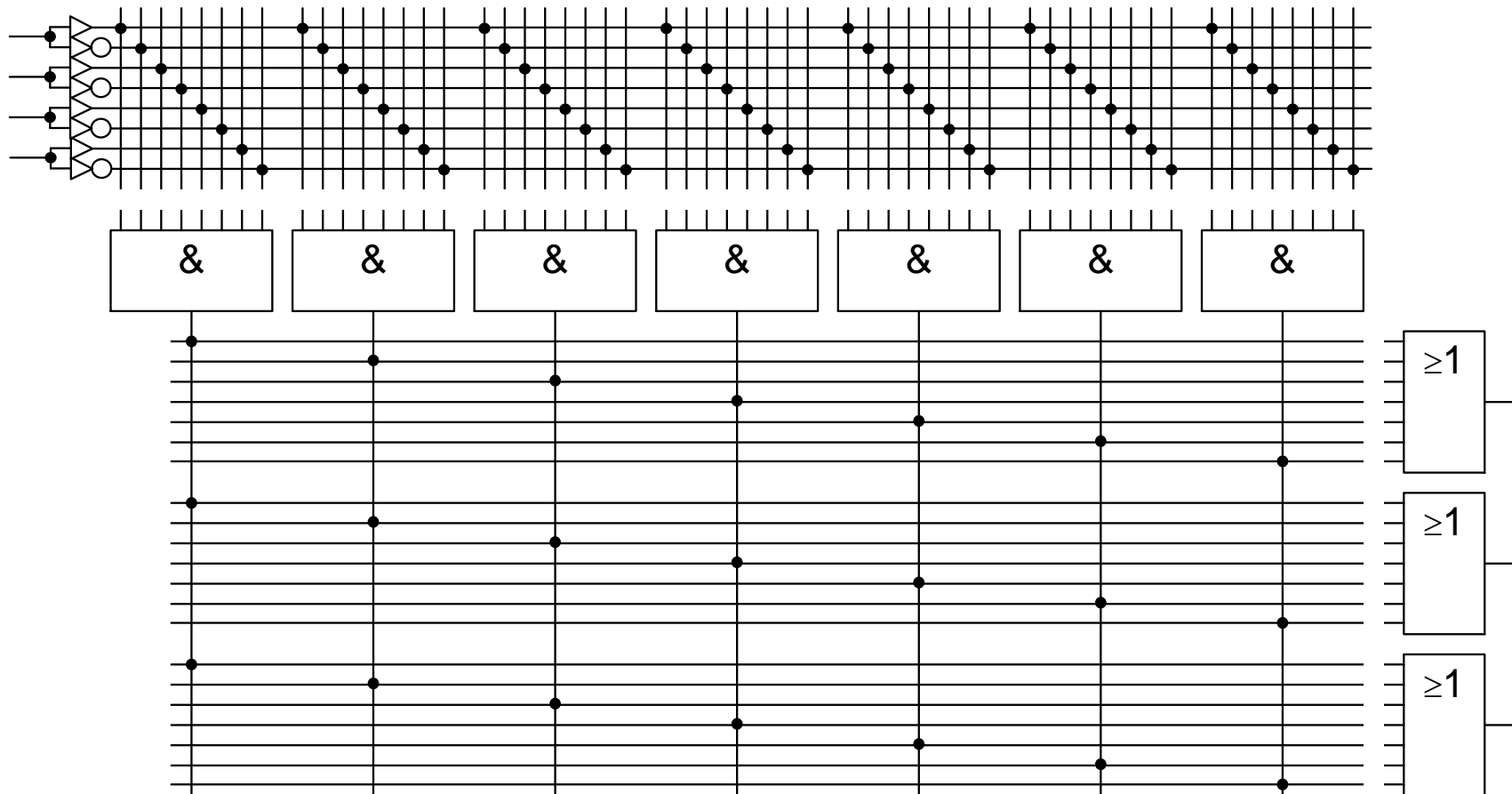


- OLMC = Output Logic Macrocell (FF plus Multiplexer)
- Ziffern bezeichnen Programmiersicherungen
- Neben dem GAL22V10 gibt es noch GAL16V8, GAL20V8, GAL26V12

Übungsaufgaben

Aufgabe 9-1

Programmieren Sie in dem gezeigten PLD die Rechenoperation „ $Y = A$ modulo 5“ für den Zahlenbereich 0 bis 9.



CPLD

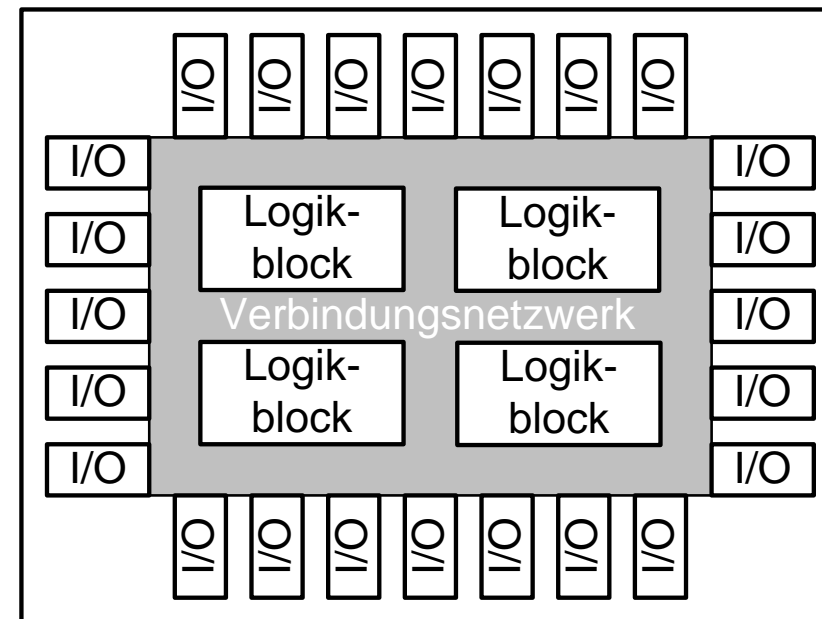
- Für komplexere Funktionen müssen sowohl Größe der Produktterme, Anzahl der Produktterme und Anzahl der Ausgangsfunktionen steigen
 - Dadurch steigt der Aufwand stark an und die Geschwindigkeit sinkt
 - Es wird aber nur ein immer kleinerer Teil der Funktionalität genutzt
- ➔ Die SPLD-Struktur ist nur für relativ kleine Schaltungen geeignet
 - Das größte verfügbare SPLD ist das GAL26V12

Ein CPLD („**Complex Programmable Logic Device**“) enthält

- mehrere SPLD-Blöcke und
- ein flexibles Verbindungsnetzwerk auf einem IC

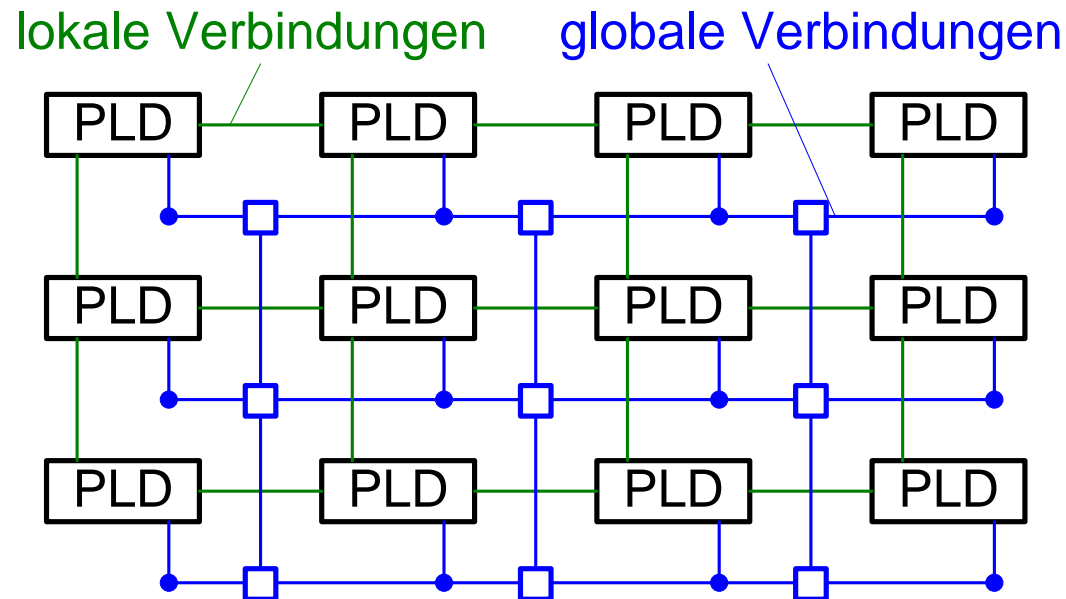
Typische Größen:

- Ausgänge je Logikblock: etwa 16
- Anzahl an Logikblöcken: 2 - 32
- Signalpins: 30 - 200



Verbindungsnetzwerk

- Bei kleinen CPLDs mit 2 oder 4 Logikblöcken (PLD) können alle PLDs miteinander verbunden werden
- Bei großen CPLDs mit 16 oder 32 Logikblöcken ist eine komplette Verknüpfung nicht mehr effizient
- Implementiert sind üblicherweise:
 - Lokale Verbindungen zu benachbarten PLDs
 - Globale Verbindungen zu weiter entfernten PLDs
 - Spezielle globale Verbindungen für 2-4 Takte und Reset



Produkt: Altera MAX 3000A Familie

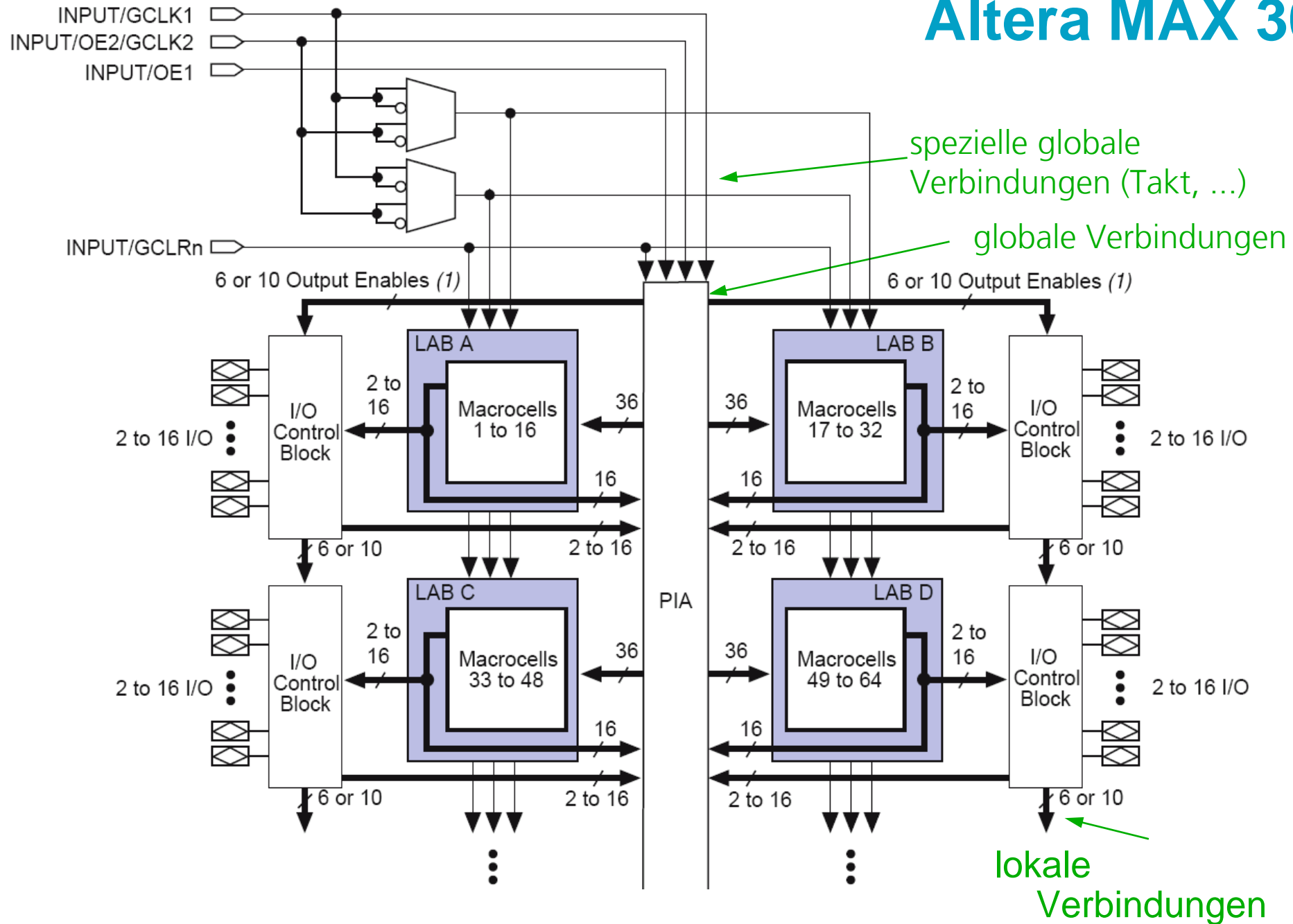
- Datenblatt unter <http://www.altera.com>
- 5 verschiedene Bausteine in 6 verschiedenen Gehäusen
 - Jeder Baustein ist in 2 oder 3 verschiedenen Gehäusen verfügbar, pin-kompatibel zu anderen Bausteinen mit demselben Gehäuse
 - Verschiedene Geschwindigkeiten verfügbar

Achtung: Angaben zu „Usable gates“ und „ f_{CNT} “ sind **Marketingwerte**, also formal korrekt, aber für reale Schaltungen nicht direkt aussagekräftig

- Als Frequenz ist (natürlich) stets die schnellste (also teuerste) Variante angegeben

Feature	EPM3032A	EPM3064A	EPM3128A	EPM3256A	EPM3512A
Usable gates	600	1,250	2,500	5,000	10,000
Macrocells	32	64	128	256	512
Logic array blocks	2	4	8	16	32
Maximum user I/O pins	34	66	98	161	208
...					
f_{CNT} (MHz)	227.3	222.2	192.3	126.6	116.3

Altera MAX 3000A (II)

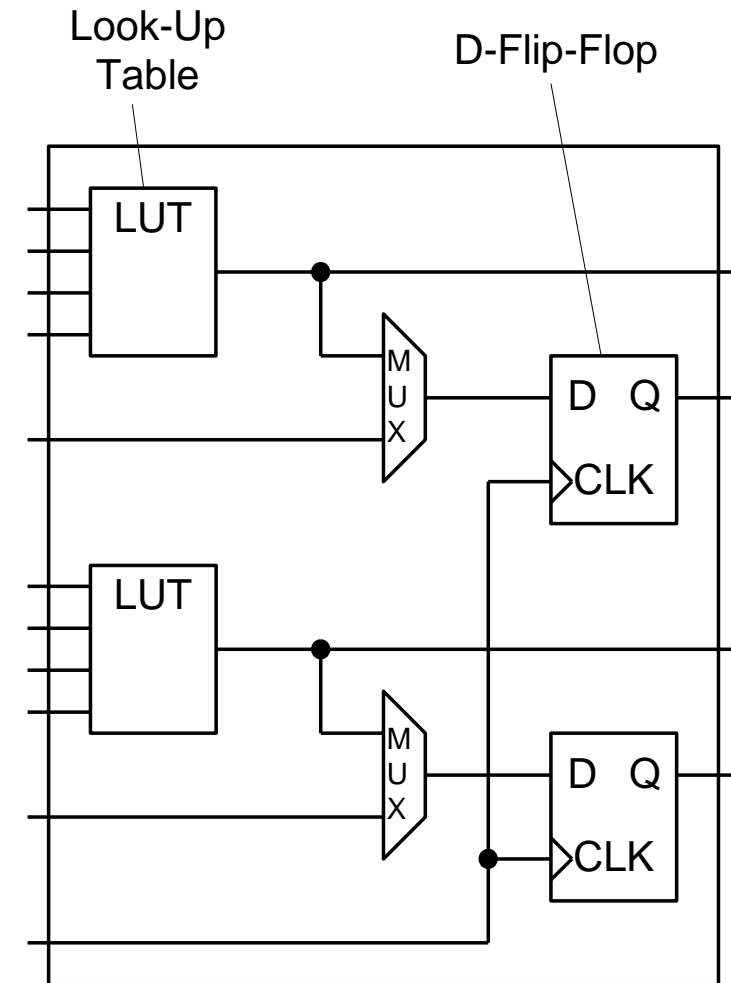


9.3 FPGAs

- FPGAs („**Field-Programmable-Gate-Array**“) haben eine höhere Komplexität als PLDs.
- Gegenüber PLDs ergeben sich wesentliche Unterschiede:
 - Die Logikblöcke sind deutlich kleiner.
 - Es gibt sehr viele Logikblöcke (200 bis über 100.000).
 - Moderne FPGAs enthalten spezielle Funktionsblöcke, z.B.:
 - Speicherblöcke.
 - Schnelle Multiplizierer für große Integer-Zahlen (z.B. 18 bit * 18 bit)
 - CPU.
 - Taktbehandlung: PLL („Phase Locked Loop“), DLL („Delay Locked Loop“).
- SPLDs und CPLDs verbinden meist andere Bausteine („**Glue Logic**“).
- FPGAs können dagegen komplette **digitale Systeme**, z.B. aus CPU, Speicher und Peripherie bilden („**System-on-a-Chip**“).
 - Dies kann sich kommerziell lohnen, muss aber nicht.

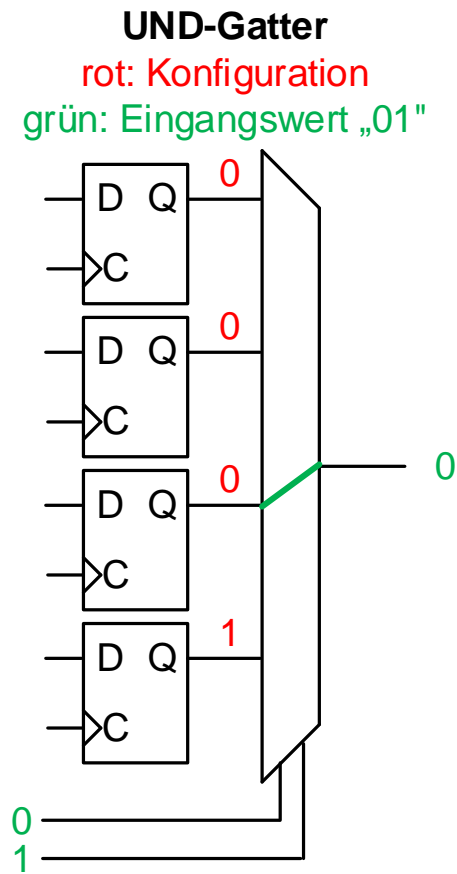
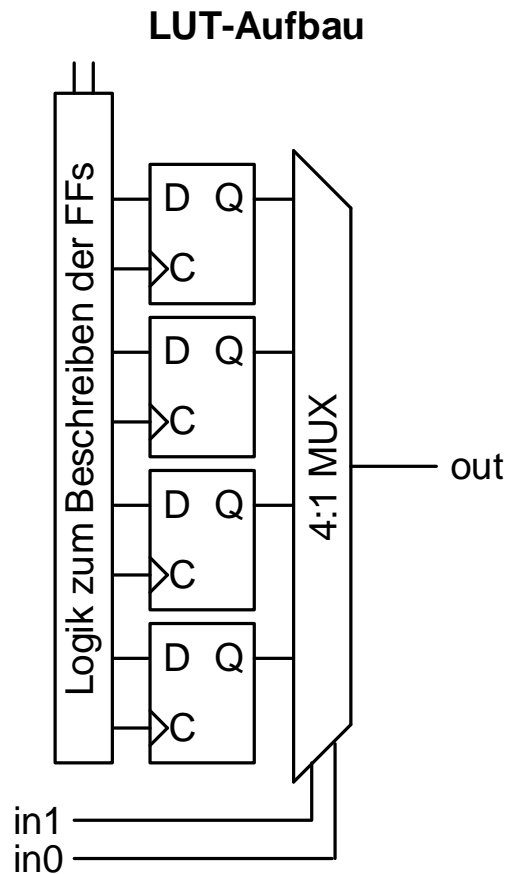
FPGA-Logikblöcke

- Logikblöcke eines FPGAs enthalten kombinatorische Logik sowie 1 oder 2 FFs.
- Die kombinatorische Logik ist oft als Look-Up-Table (LUT) aufgebaut.
 - Eine LUT ist ein Speicher, in dem die Eingänge (meist 4) den Ausgangswert, wie in einer Funktionstabelle auswählen.
- Die Flip-Flops können die Ergebnisse der LUT speichern oder unabhängig benutzt werden.



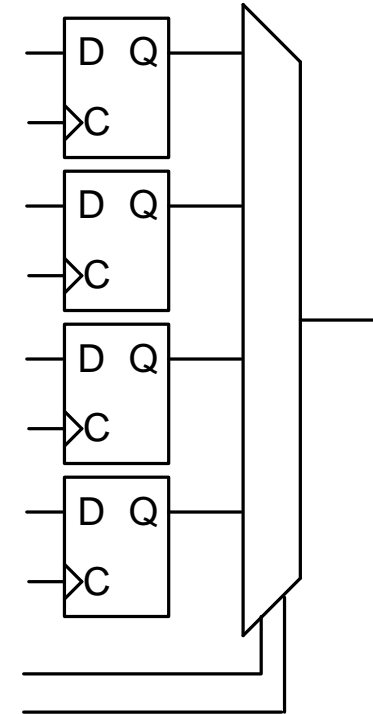
Look-Up-Table (LUT)

- In der Look-Up-Table (LUT) werden die möglichen Ausgangswerte gespeichert
- Typische Größen sind 4 bis 6 Eingangswerte und somit 16 bis 64 Speicherplätze
 - Bild: 2 Eingangswerte, 4 Speicherplätze



Aufgabe

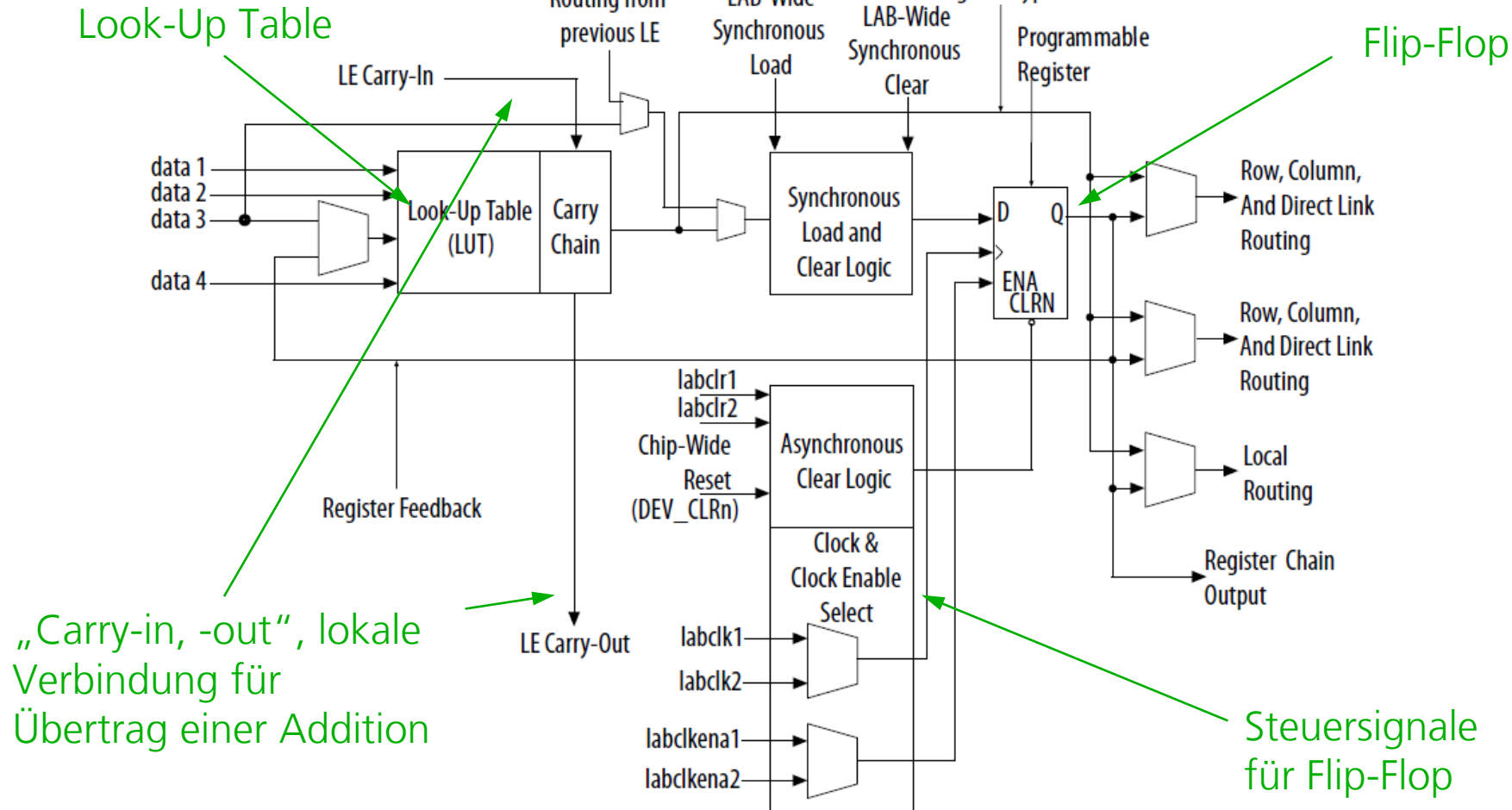
Wie lautet die Konfiguration
für ein XOR-Gatter?



Produkt: Intel Max10 Familie - Logikblock

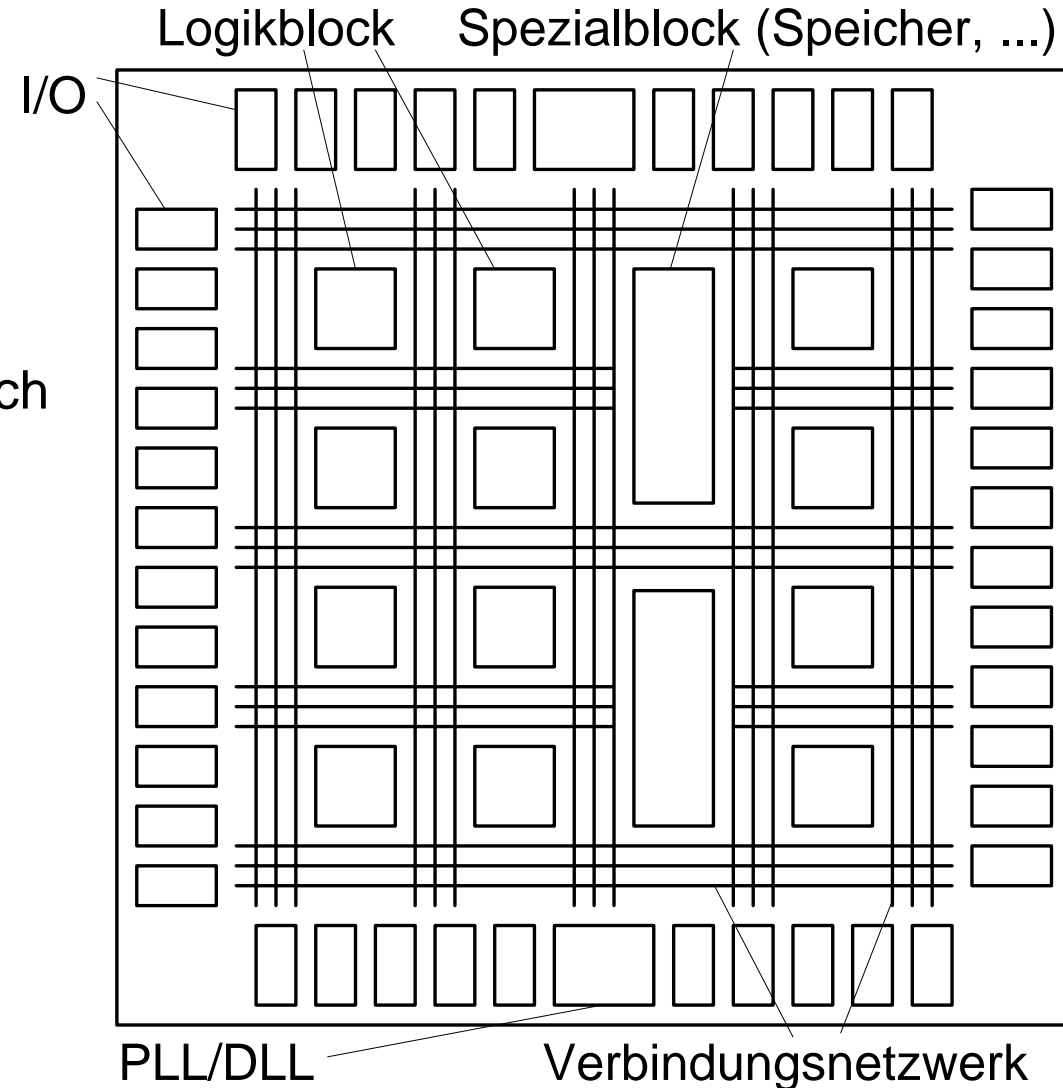
Auf DE10-Lite Board, Praktikum 2. Semester

LE = Logic Element



Verbindungsnetzwerk

- Logikblöcke sowie spezielle Funktionsblöcke sind in einem **Array** angeordnet.
 - FPGA heißt ja „Field-Programmable-Gate-Array“.
 - Anders als im Bild hat ein FPGA jedoch tausende Logikblöcke.
- Das Verbindungsnetzwerk enthält, unterschiedliche Verbindungen.
 - Lokale Verbindungen.
 - Globale Verbindungen, bei großen FPGAs über verschieden lange Distanzen.
 - Globale Verbindungen für Takte und Reset.

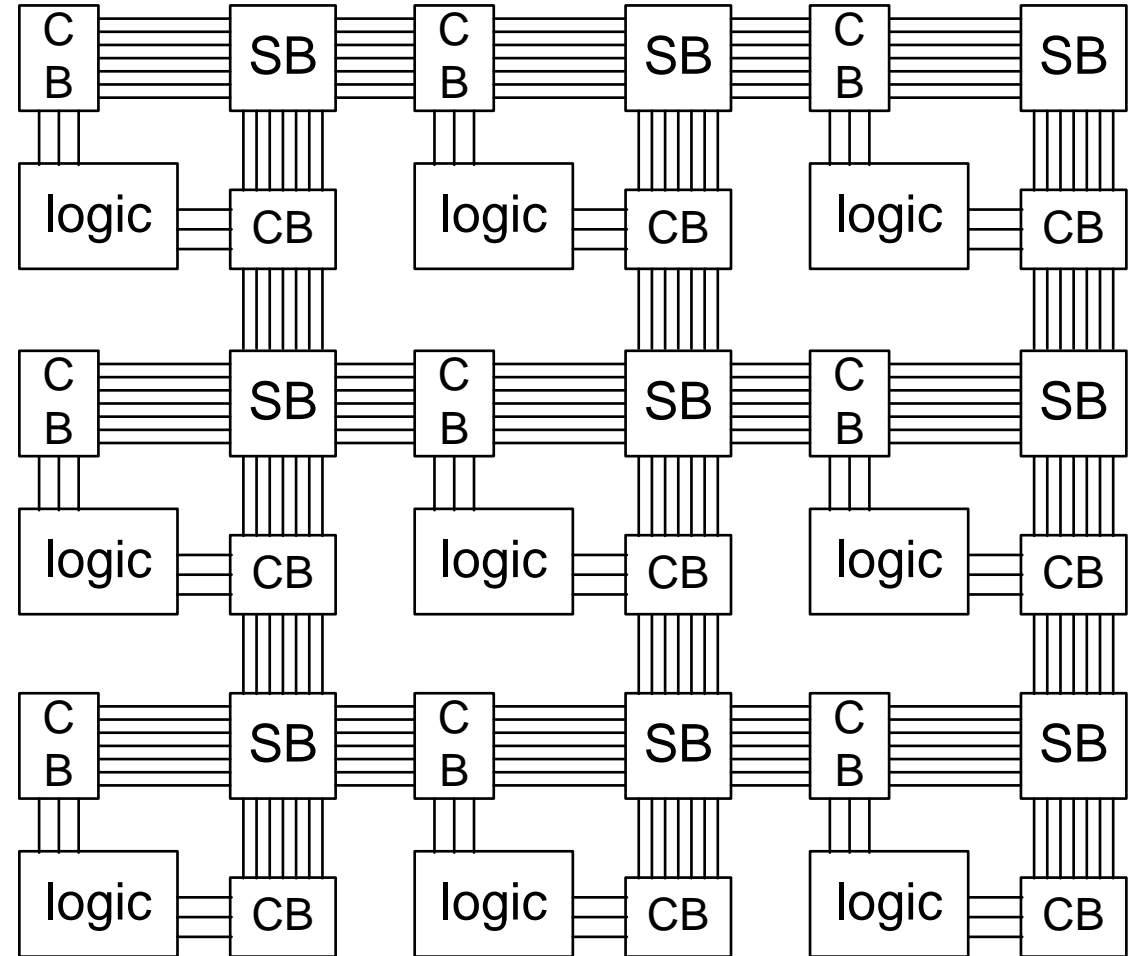


Verbindungsnetzwerk (II)

- Verbindungen machen einen großen Teil des FPGA aus
 - etwa 3/4 der Fläche
 - etwa 3/4 der Verzögerung
 - etwa 2/3 des Stromverbrauchs(grobe Schätzungen, je nach Anbieter und Technologie)

Die Verbindungsstruktur besteht aus

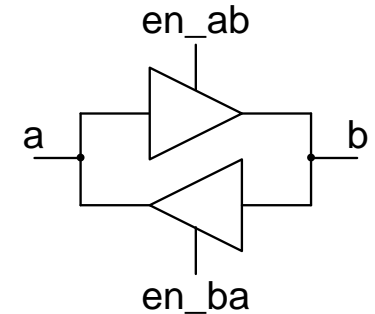
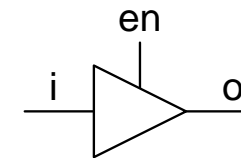
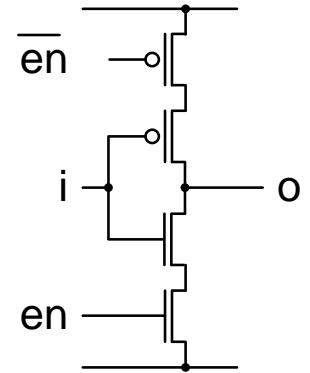
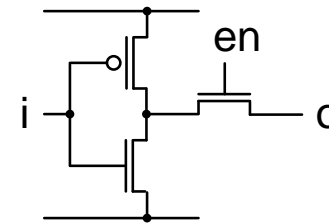
- Den eigentlichen Logikblöcken (logic)
- Connection Blocks (CB) zum Verbinden der Logik mit dem Verbindungsnetzwerk
- Switch Blocks (SB) zum Schalten der Verbindungsleitungen



Verbindungsnetzwerk (III)

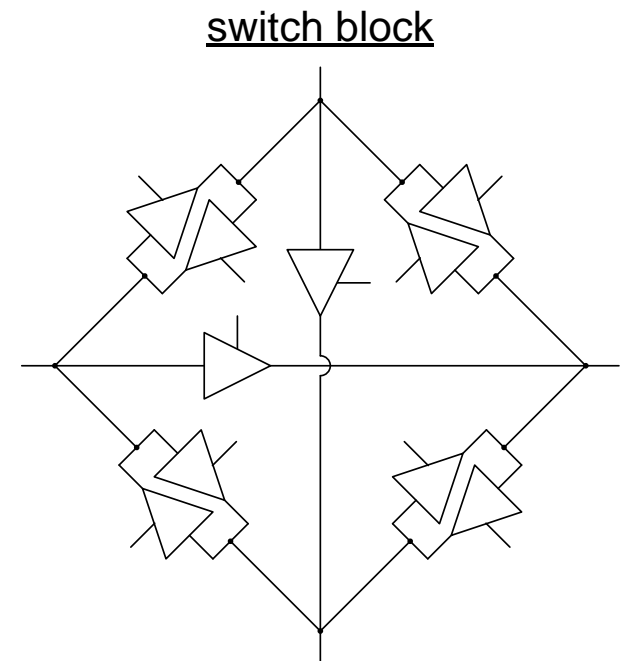
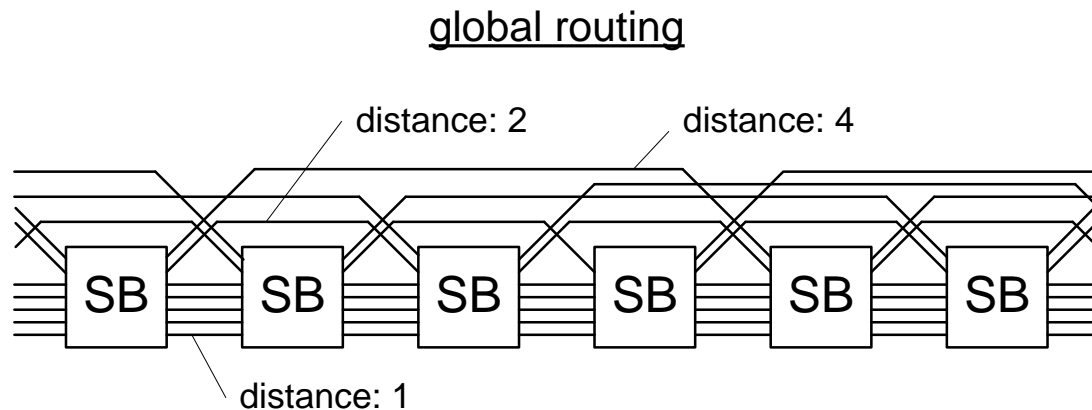
Detaillierte Informationen zum Verbindungsnetzwerk eines FPGAs sind vertraulich und nicht verfügbar

- Grundsätzlich kann die programmierbare Verbindung realisiert werden durch
 - Pass Transistor (3 Transistoren)
 - Tri-State-Puffer (6 Transistoren; 4 im Schaltplan plus 2 für den Inverter von "en")
- ➔ Siehe Kapitel 10
- Abhängig von der Position im Netzwerk kann die Verbindung unidirektional oder bidirektional sein
- Das Steuersignal 'en' wird von einem Konfigurationsbit angesteuert (und erfordert zusätzliche Logik)

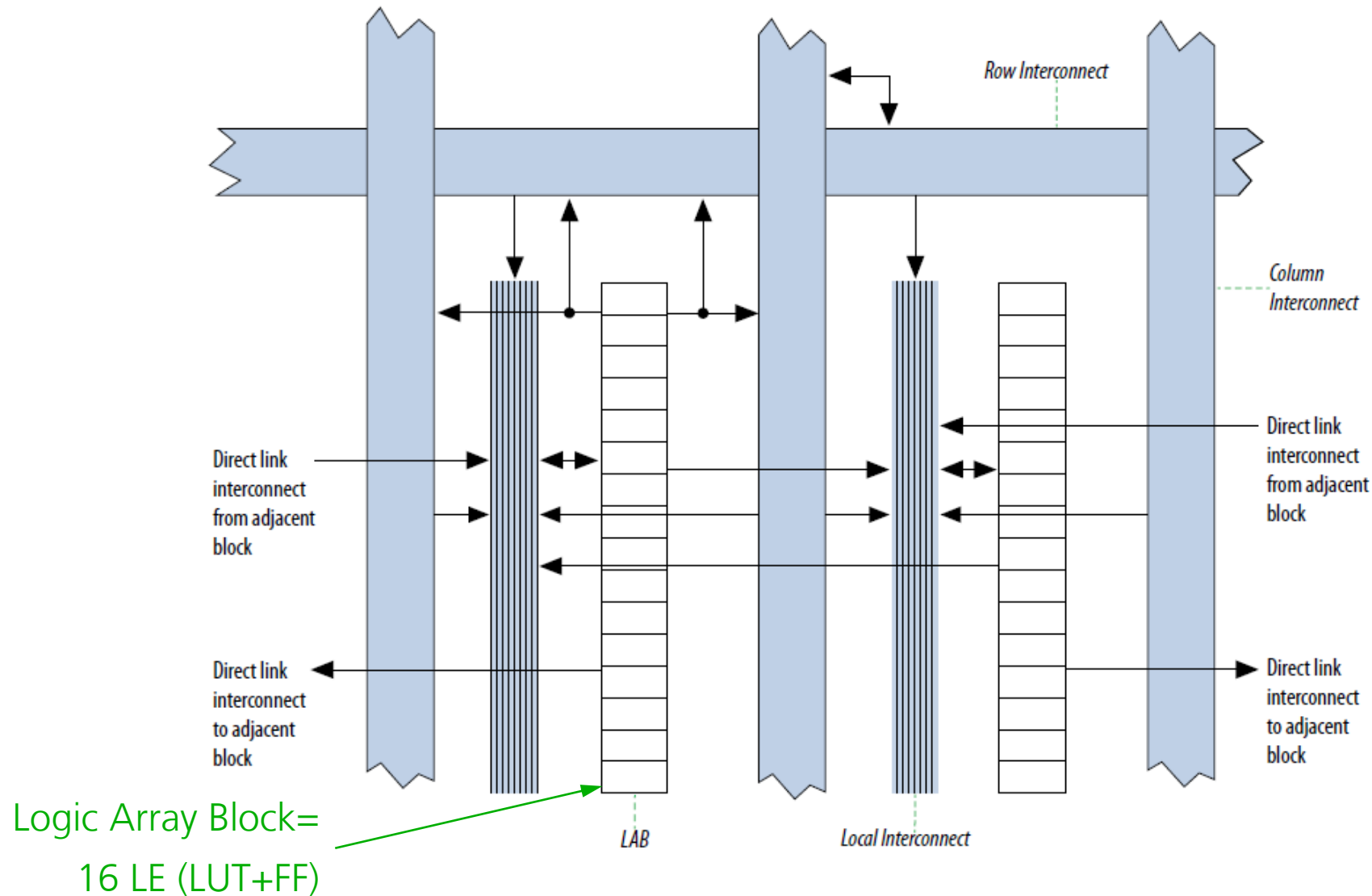


Lokale und Globale Verbindungen

- Eine typische Digitalschaltung hat viele kurze, einige mittlere und wenige lange Verbindungen
 - Dies wird durch die “Rent’s Rule” vorhergesagt
- Daher hat auch ein FPGA viele kurze und einige lange Verbindungen
 - Direkte Verbindungen der Slices oder LABs entsprechen den lokalen Verbindungen
 - Verbindungen unterschiedlicher Abstände zwischen Switch Blocks (Bild zeigt vereinfacht eindimensionale Struktur)
- Switch Blocks sind ähnlich wie Connection Blocks implementiert
- Konfigurationssignale benötigen entsprechenden Speicher auf dem FPGA und sind im Bitfile enthalten



Produkt: Intel Max10 Familie - Verbindungsnetzwerk



Produkt: Altera/Intel Max10 Familie

- 7 verschiedene Bausteine, Datenblatt unter <http://www.altera.com>

DE10-Lite
Board

Resource		Device						
		10M02	10M04	10M08	10M16	10M25	10M40	10M50
1 LUT + 1 FF	Logic Elements (LE) (K)	2	4	8	16	25	40	50
Speicher- blöcke	M9K Memory (Kb)	108	189	378	549	675	1,260	1,638
	User Flash Memory (Kb) ⁽¹⁾	96	1,248	1,376	2,368	3,200	5,888	5,888
	18 × 18 Multiplier	16	20	24	45	55	125	144
Takt- aufbe- reitung	PLL	2	2	2	4	4	4	4
	GPIO	160	246	250	320	380	500	500
	LVDS	Dedicated Transmitter	10	15	15	22	26	30
		Emulated Transmitter	73	114	116	151	181	241
		Dedicated Receiver	73	114	116	151	181	241
differen- tielle Daten- leitungen	Internal Configuration Image	1	2	2	2	2	2	2
	ADC	—	1	1	1	2	2	2

(Bild: m10_handbook.pdf)

Produkt: Altera/Intel Max10 Familie (II)

- 9 verschiedene Gehäuse erhältlich
- Überlappungen mit Pincompatibilität
 - Wechsel der FPGA-Größe bei gleichem Platinenlayout möglich

Device	Package								
	V36	V81	M153	U169	U324	F256	E144	F484	F672
10M02			↑	↑	↑				
10M04			↓	↓	↓	↑	↑		
10M08			↑	↑	↑	↑	↑	↑	
10M16				↓	↓	↑	↑	↑	
10M25						↑	↓	↑	↑
10M40						↑	↑	↑	↑
10M50						↓	↓	↓	↓

(Bild: m10_handbook.pdf)

Produkt: Lattice iCE40

- FPGAs mit sehr kleiner Bauform (z.B. 16 Pins) und sehr geringer statischer Stromaufnahme (30 bis 70 μA)
- Anwendung z.B. als „Co-Prozessor“ für CPU
 - CPU geht in Sleep-Mode und FPGA führt einfache Operationen durch
- Konfiguration in RAM oder durch Einmalprogrammierung

Part Number		LP384	LP640	...	HX8K
Logic Cells (LUT + Flip-Flop)		384	640		7,680
RAM4K Memory Blocks		0	8	...	32
RAM4K RAM bits		0	32K		128K
Phase-Locked Loops (PLLs)		0	0	...	2
Maximum Programmable I/O Pins		63	25		206
Maximum Differential Input Pairs		8	3	...	26
High Current LED Drivers		0	3		0
Package	Code			...	
16 WLCSP (1.40 mm x 1.48 mm, 0.35 mm)	SWG16		10(0) ¹		—
32 QFN (5 mm x 5 mm, 0.5 mm)	SG32	21(3)	—		—

⋮

⋮

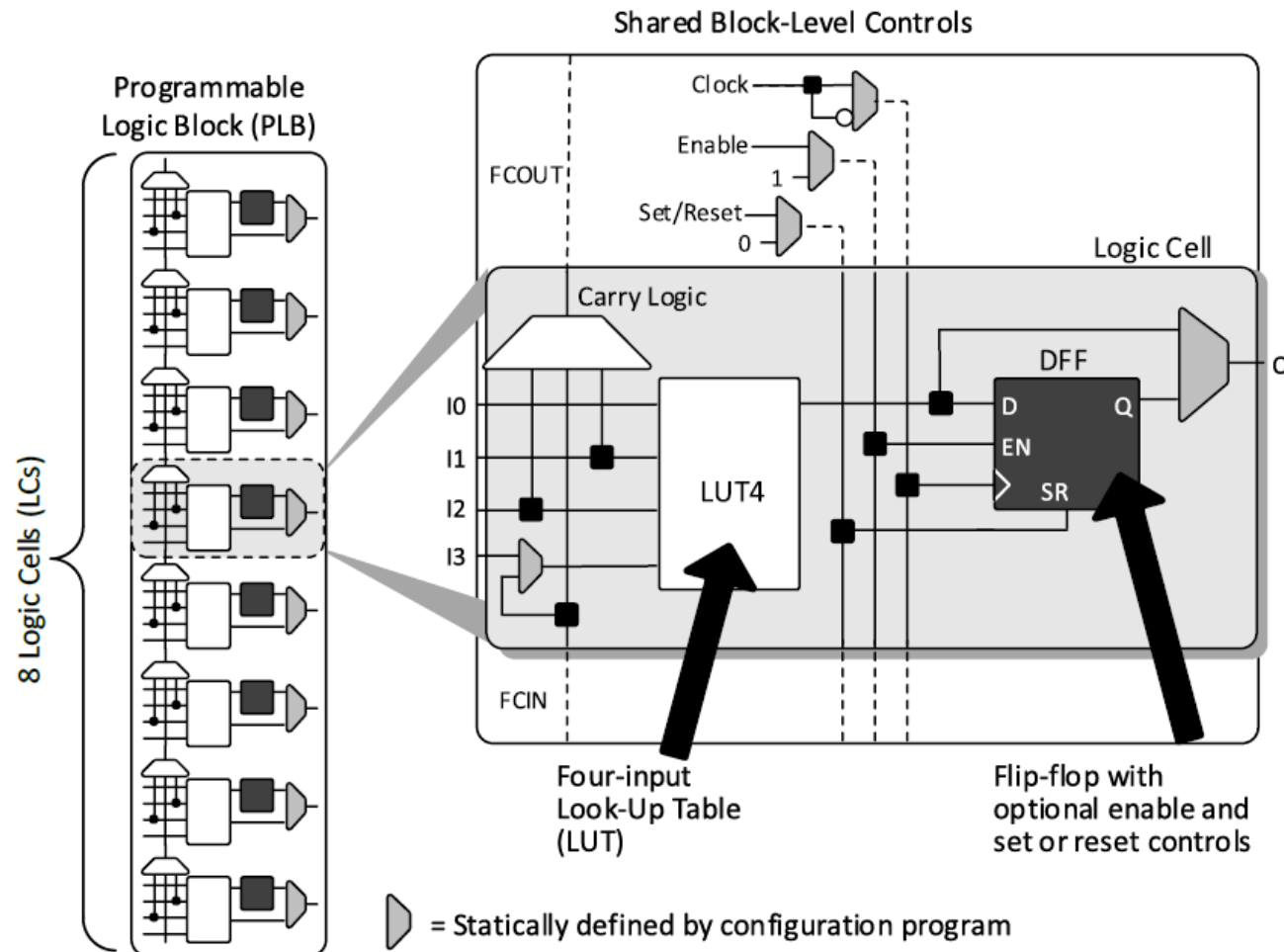
I/O-Pins
(LVDS Pairs)

(Bild: Lattice Datenblatt iCE40 LP/HX Family)

Produkt: Lattice iCE40 (II)

- Logic Cells ähnlich Max10, aber kleine Unterschiede

➔ **Frage:** Welche Gemeinsamkeiten und Unterschiede erkennen Sie?



Anmerkung:

Diagramme sind teilweise etwas vereinfacht. Für vollen Vergleich daher genauer Blick ins Datenblatt sinnvoll.

(Bild: Lattice Datenblatt iCE40 LP/HX Family)

9.4 SoC FPGAs

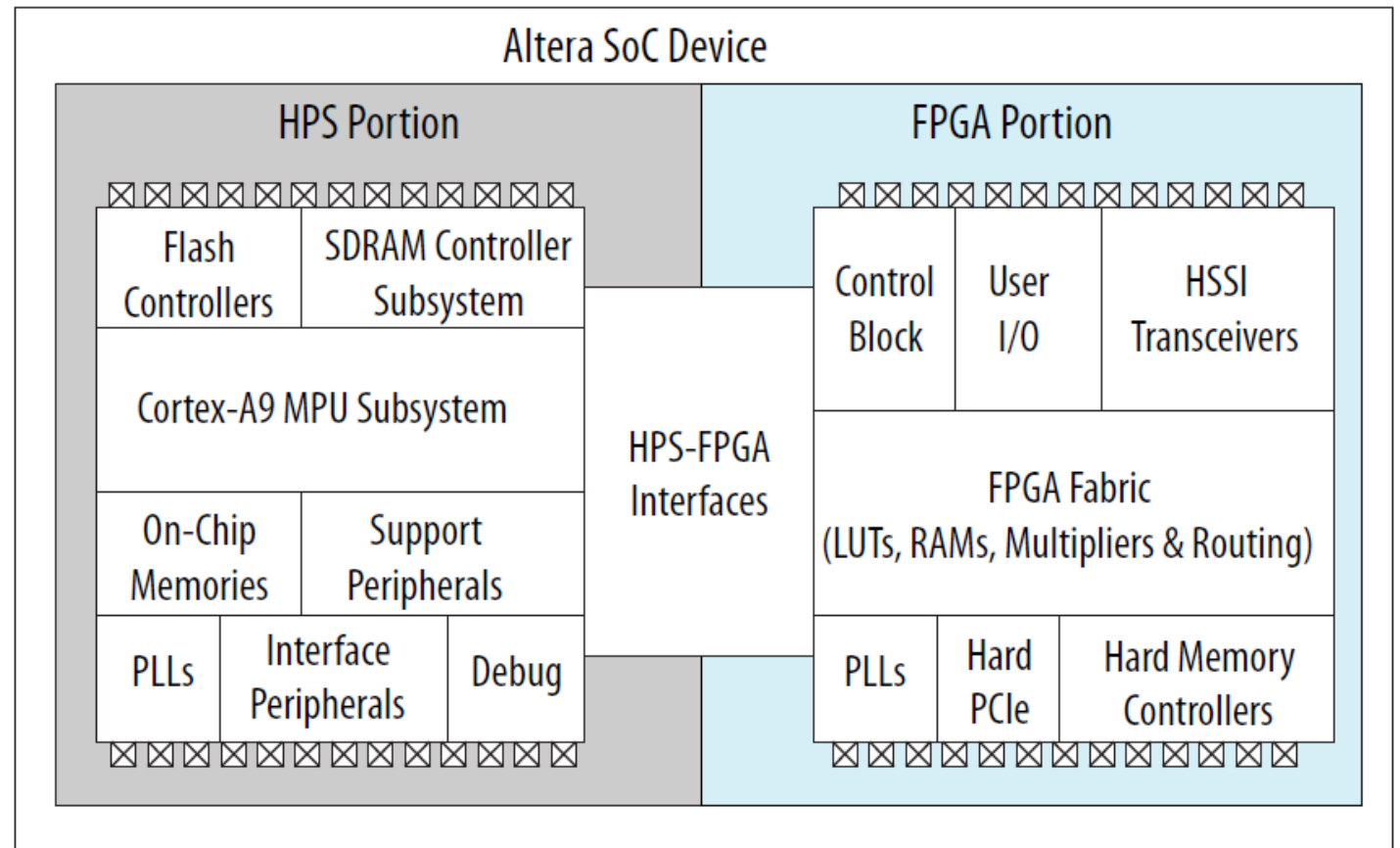
- FPGAs werden oft in Systemen mit einer CPU eingesetzt
- Oft wird externer DRAM-Speicher verwendet und ein Speicher-Controller benötigt
- CPU und Speicher-Controller können aus FPGA-Elementen aufgebaut werden (LUT, FF, ...)
 - Das wird bei kleineren, langsamen Systemen auch so gemacht
 - Z.B. DRAM-Interface mit 100 MHz
- Größere FPGAs haben spezielle Module für diese Funktion
- Damit kann ein ganzes Rechnersystem auf dem FPGA aufgebaut werden
 - Bezeichnung: System-on-Chip (SoC)
- **Vorteil:**
 - Vorgefertigte Module sind kleiner und schneller
 - Funktionen mit hoher Geschwindigkeit sind nur so umsetzbar
 - Z.B. DRAM-Interface mit 500 MHz
- **Nachteil:**
 - Wird ein Modul nicht verwendet, ist diese Teilschaltung nicht anders nutzbar

Beispiel: Intel/Altera Cyclone V SX

- Variante der Cyclone V Bausteine mit CPU-System verfügbar
 - HPS (Hard Processor System) mit ARM-CPU
 - „Übliche“ FPGA Struktur
- Jedes System hat eigene Peripherie
- Anbindung des FPGA-Bereichs über internes Interface

Abkürzungen im Bild:

- PLL: Phase-Locked-Loop, zur Taktbehandlung
- HSSI: High-Speed Serial Interface, z.B. RapidIO
- PCIe: PCI Express Bus



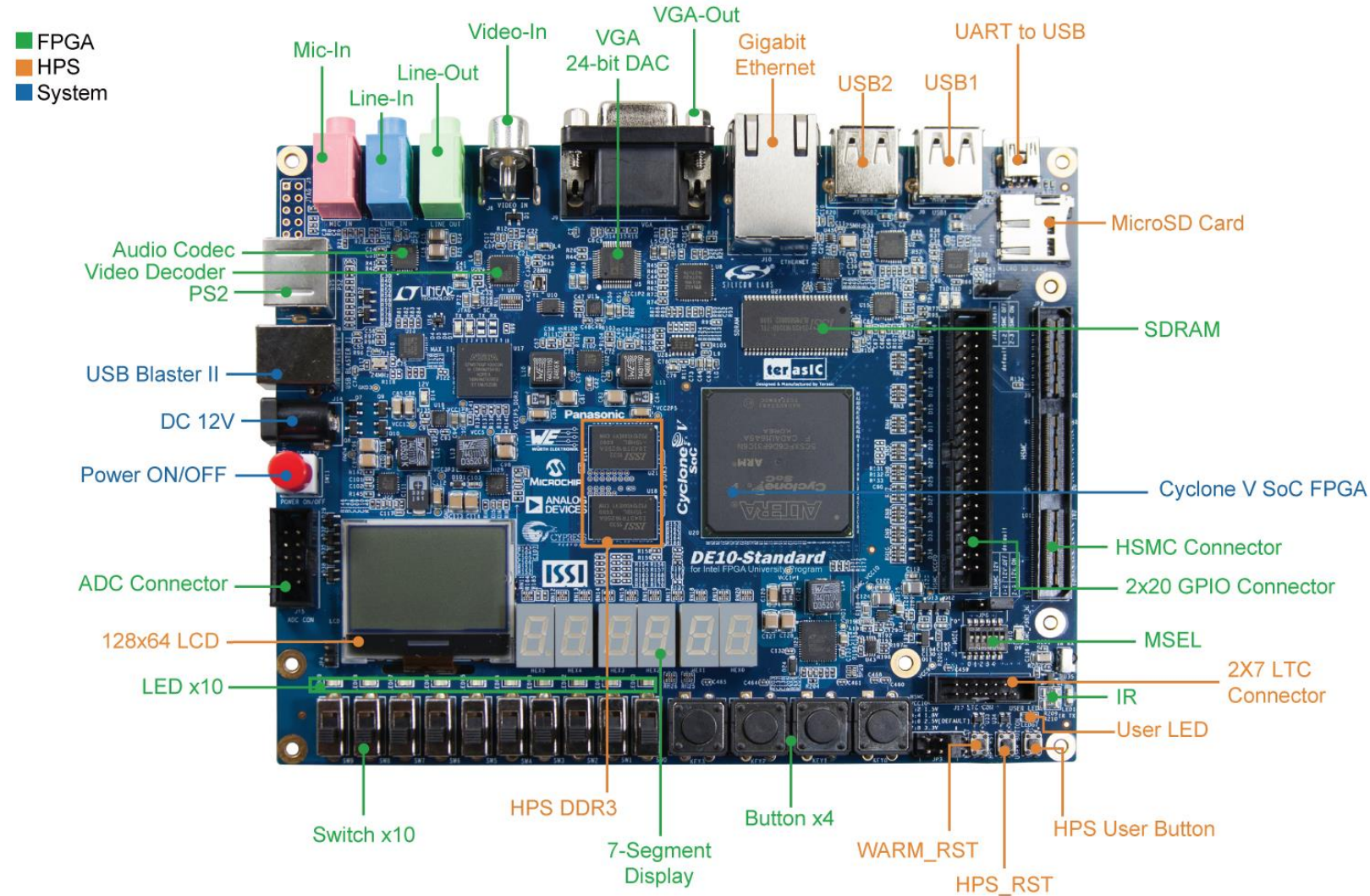
Beispiel: FPGA-Board mit Cyclone V SX

Terasic DE10-Standard

- Cyclone V SX SoC, 5CSXFC6D6F31C6N
- 925 MHz, Dual-Core ARM Cortex-A9 MPCore Processor
- Produktbild zeigt Aufteilung der externen Peripherie an HPS und FPGA
- Preis: 259\$ (Academic, 2/2020)

Frage:

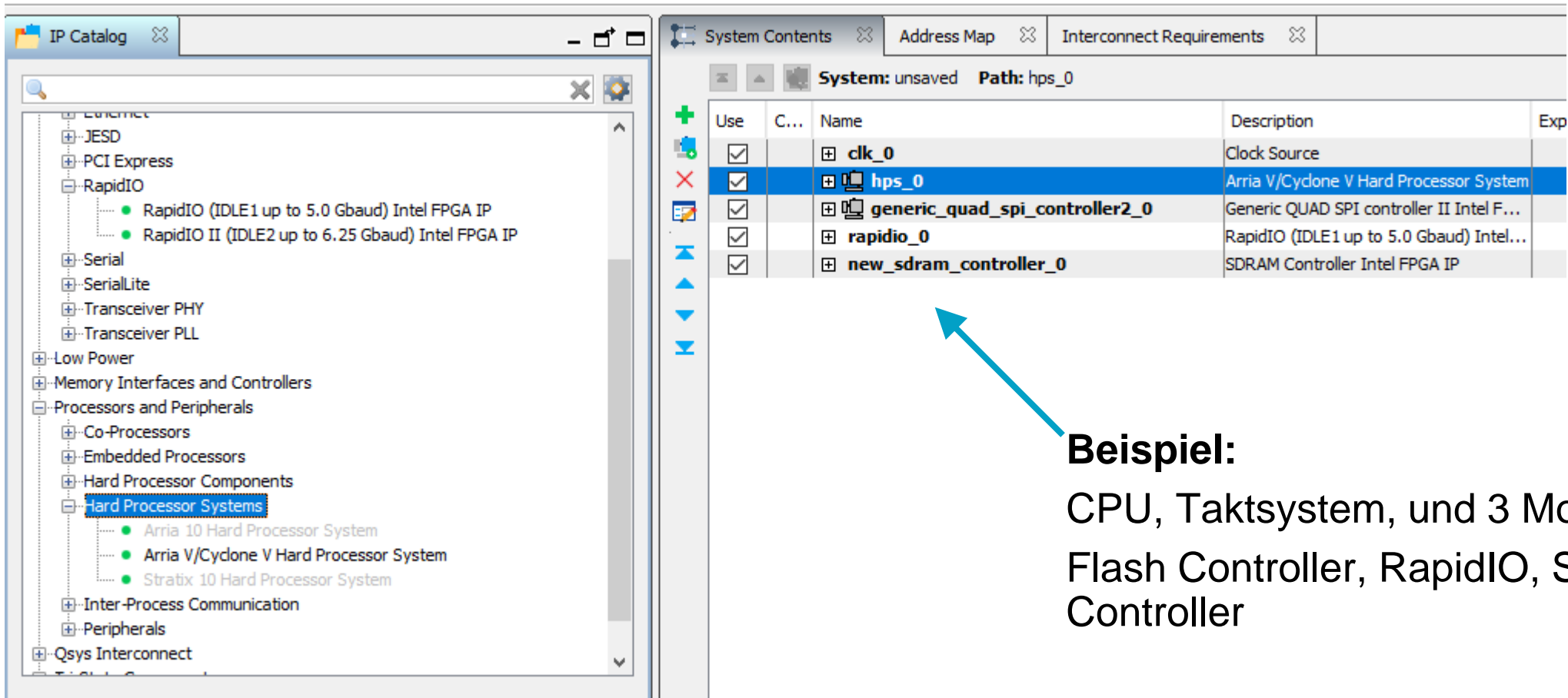
Wie unterscheiden sich Peripherie an HPS und FPGA?



SoC Entwurf mit Plattform Designer

Entwurf von Hardware und Software erforderlich

- Plattform Designer erlaubt Zusammenstellung des CPU Systems
- IP Catalog (Intellectual Property) mit verfügbaren, konfigurierbaren Komponenten



The screenshot displays the Intel Platform Designer interface. On the left, the 'IP Catalog' window shows a hierarchical tree of components. Under 'Processors and Peripherals', 'Hard Processor Systems' is selected, showing options like 'Arria 10 Hard Processor System', 'Arria V/Cyclone V Hard Processor System', and 'Stratix 10 Hard Processor System'. On the right, the 'System Contents' window shows a table of components added to the system. A blue arrow points from the text 'Beispiel:' to the 'hps_0' entry in the table.

Use	C...	Name	Description	Exp
<input checked="" type="checkbox"/>		clk_0	Clock Source	
<input checked="" type="checkbox"/>		hps_0	Arria V/Cyclone V Hard Processor System	
<input checked="" type="checkbox"/>		generic_quad_spi_controller2_0	Generic QUAD SPI controller II Intel F...	
<input checked="" type="checkbox"/>		rapidio_0	RapidIO (IDLE1 up to 5.0 Gbaud) Intel...	
<input checked="" type="checkbox"/>		new_sdram_controller_0	SDRAM Controller Intel FPGA IP	

Beispiel:
CPU, Taktsystem, und 3 Module:
Flash Controller, RapidIO, SDRAM
Controller

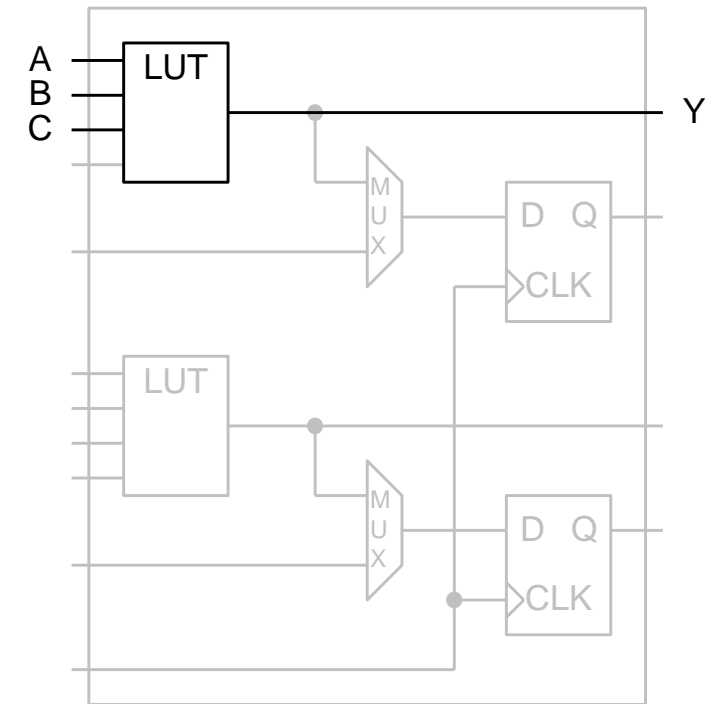
9.5 Schaltungsentwurf für FPGAs

- Die Umsetzung einer VHDL-Beschreibung in eine FPGA-Schaltung erfolgt mittels EDA-Tools („Electronic Design Automation“).
- Der Anwender muss sich normalerweise nicht mit der Aufteilung der Schaltung auf verschiedene LUTs befassen.

Beispiel:

- Die logische Funktion $Y = (A \& B) \vee C$ soll in ein FPGA umgesetzt werden.
- Der VHDL-Code lautet:

```
y <= (a and b) or c;
```
- Ein EDA-Tool kann die Funktion in eine LUT umsetzen.
- Der Rest des CLB bleibt ungenutzt oder kann von anderen Schaltungsteilen verwendet werden.



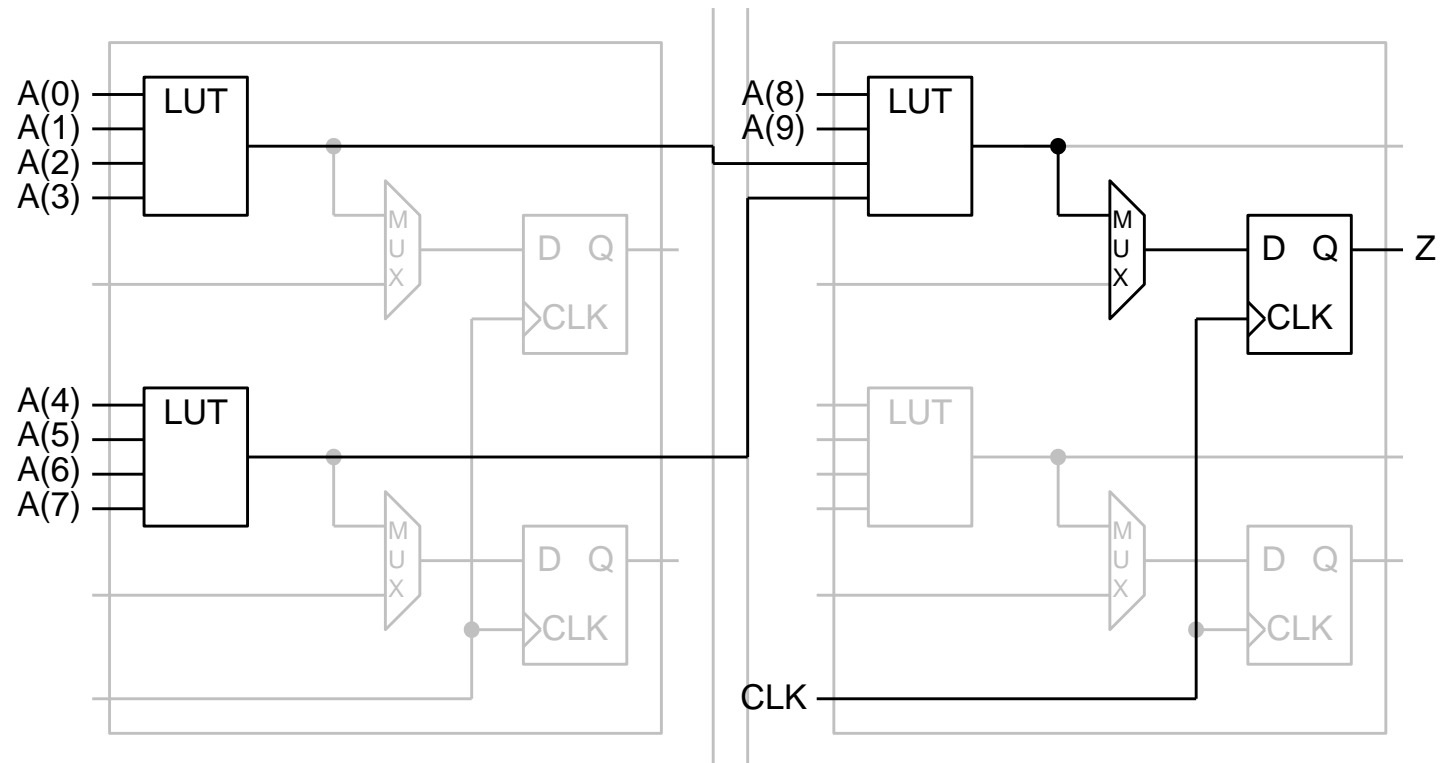
Prinzipieller Schaltungsentwurf

- Komplexere Logikfunktionen werden **automatisch** auf mehrere LUTs und CLBs aufgeteilt.

Beispiel:

- Es soll überprüft werden, ob die 10-bit Zahl A gleich Null ist. Das Ergebnis soll in einem Flip-Flop gespeichert werden.
- Die Funktion der LUT entspricht einen 10-Input NOR.

```
process (clk)
begin
  if rising_edge(clk) then
    if (a="0000000000") then
      z <= '1';
    else
      z <= '0';
    end if;
  end if;
end process;
```



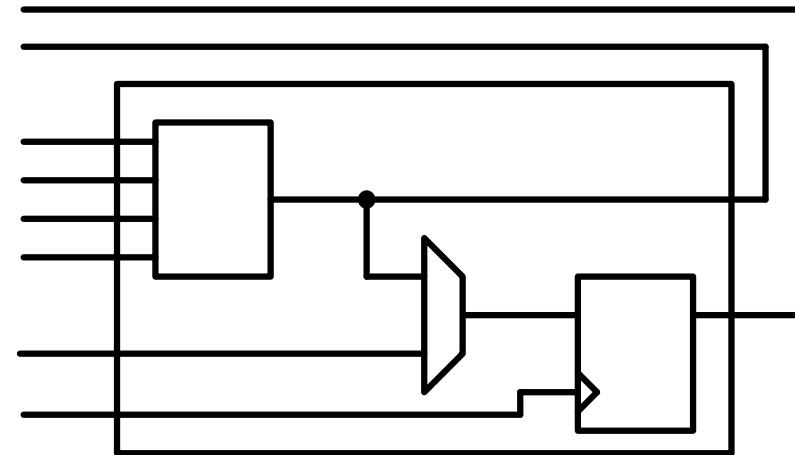
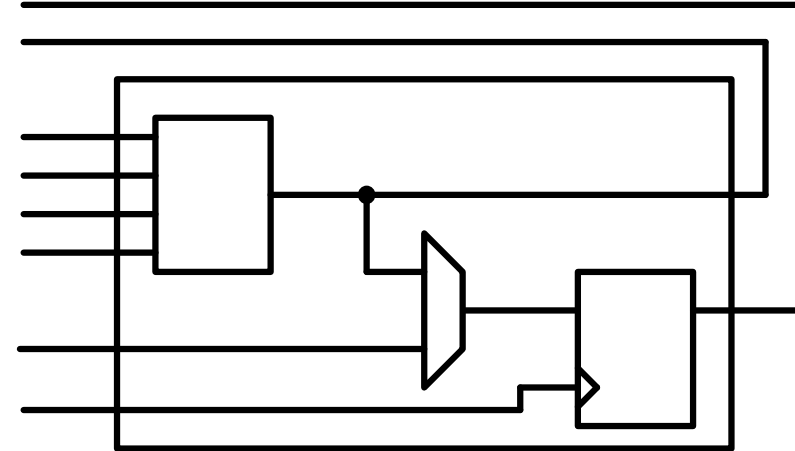
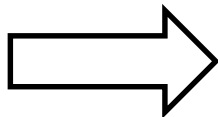
Übungsaufgaben

Aufgabe 9-2

In einem FPGA sollen die beiden 3 Bit-Zahlen A und B verglichen werden. A und B sind Dualzahlen.

Wenn A größer als B ist, soll der Ausgang Y = ,1' sein, ansonsten ist Y = ,0'. Der Ausgang Y soll sich erst bei einer steigenden Taktflanke ändern.

Wie wird das FPGA konfiguriert?



Beispiel für FPGA-Implementierung

Schaltungsbeispiel:

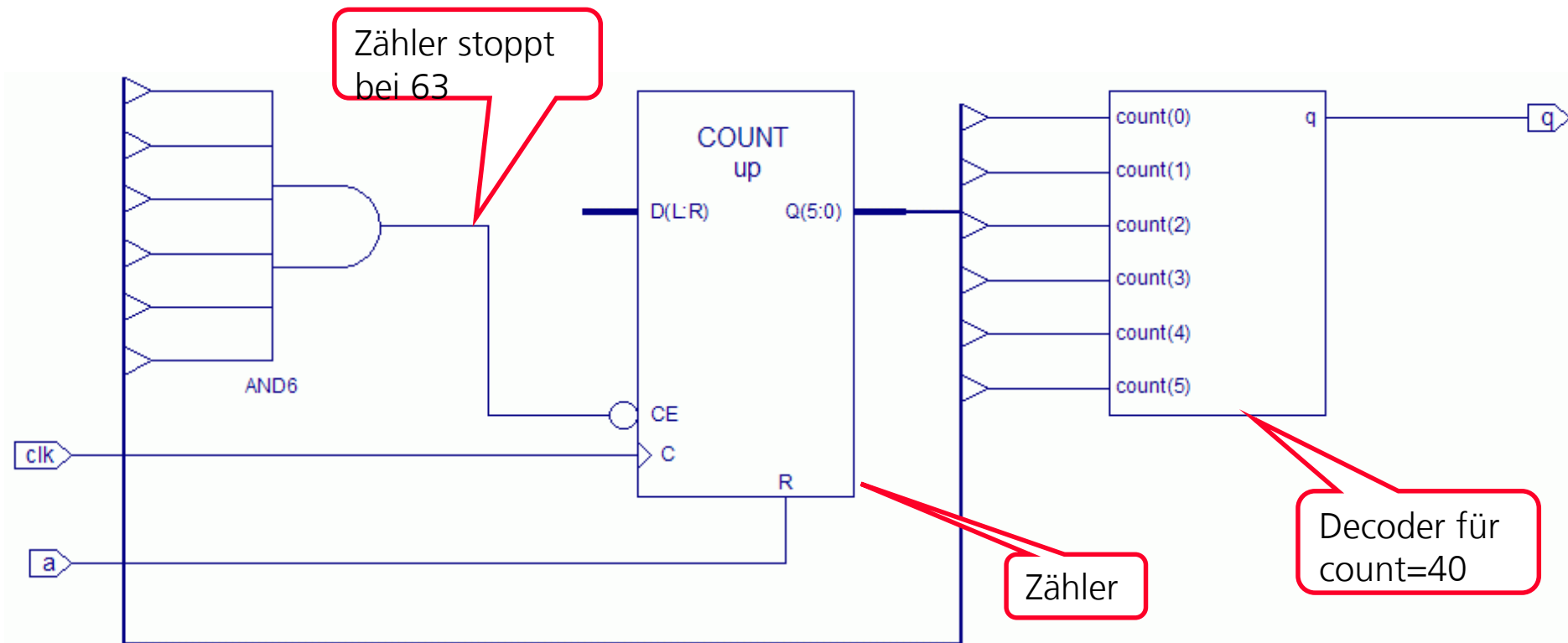
- Erkennung, ob ein Signal A von ,1' auf ,0' wechselt und 40 Takte lang ,0' bleibt.
- Dann soll für einen Takt der Ausgang Q auf ,1' gehen.
- Ansonsten ist Q gleich ,0'.

```
architecture rtl of x_delay is
    signal count : std_logic_vector(5 downto 0);
begin
    process(clk)
    begin
        if rising_edge(clk) then
            if (a = '1') then                -- restart
                count <= (others => '0');
            elsif (count = "111111") then    -- limit
                count <= count;
            else                             -- increment
                count <= count + "000001";
            end if;
        end if;
    end process;

    q <= '1' when (count = "101000") else '0'; -- 40 = 32+8
end rtl;
```

Synthese

- Die VHDL-Beschreibung wird interpretiert und in eine Schaltung umgesetzt
 - In der Synthese wird die Beschreibung zunächst in allgemeine Funktionsmodule (Zähler, Decoder, Speicher, ...) übersetzt
- Die Ergebnisse können graphisch angezeigt werden (hier mit XILINX-Tools)



(Screenshots, auch auf Folgeseiten, aus Xilinx ISE)

Technologie-Mapping

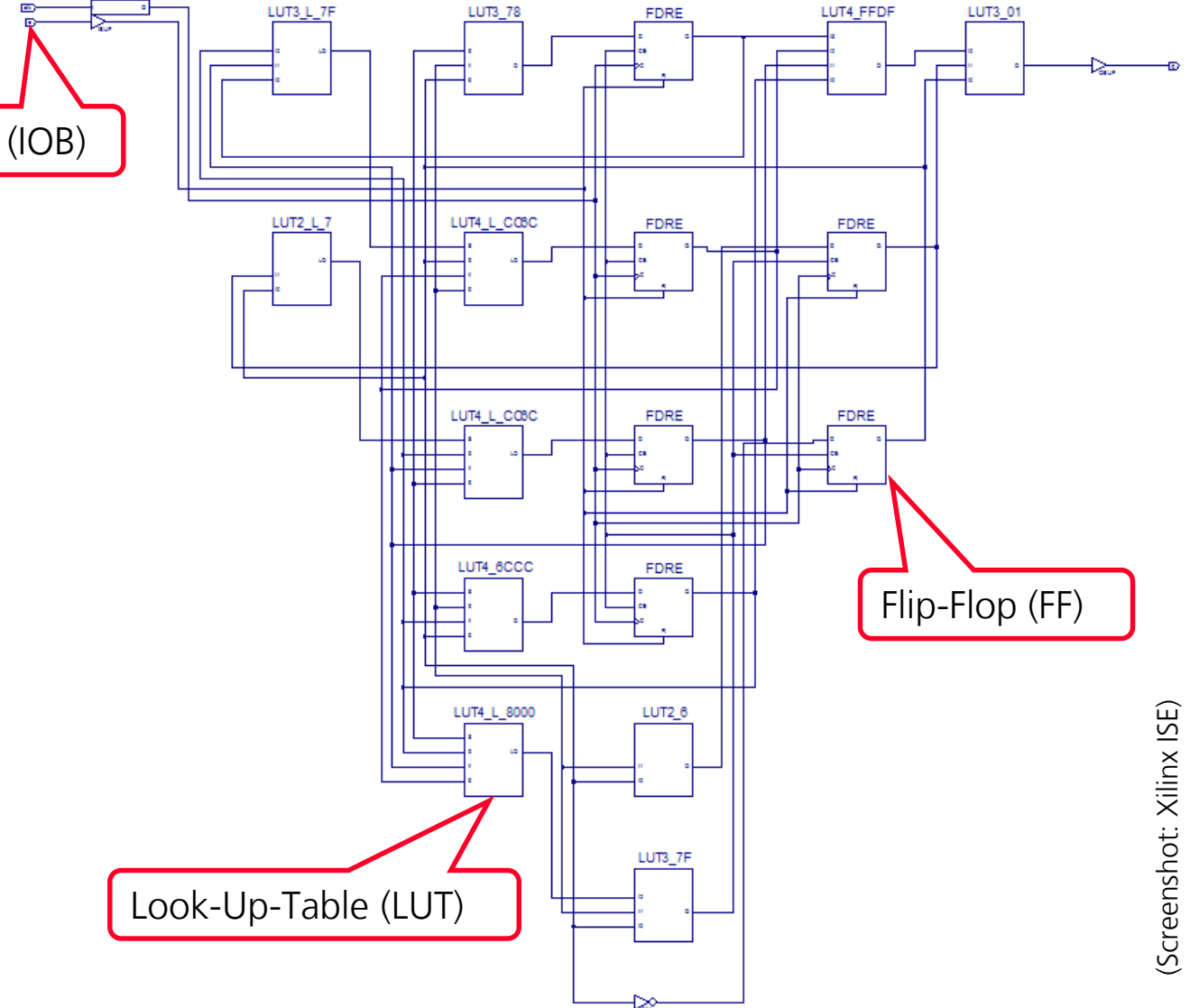
- Beim „Technologie-Mapping“ erfolgt eine Übersetzung in FPGA-Elemente, also Look-Up-Tables (LUT), Flip-Flops, ...
- Dabei werden die erforderlichen Elemente und damit der ungefähre Platzbedarf bestimmt
- Hier:
 - LUT: 11
 - FF: 6
 - IOB: 3
 - GCLK: 1

Takttreiber (GCLK)

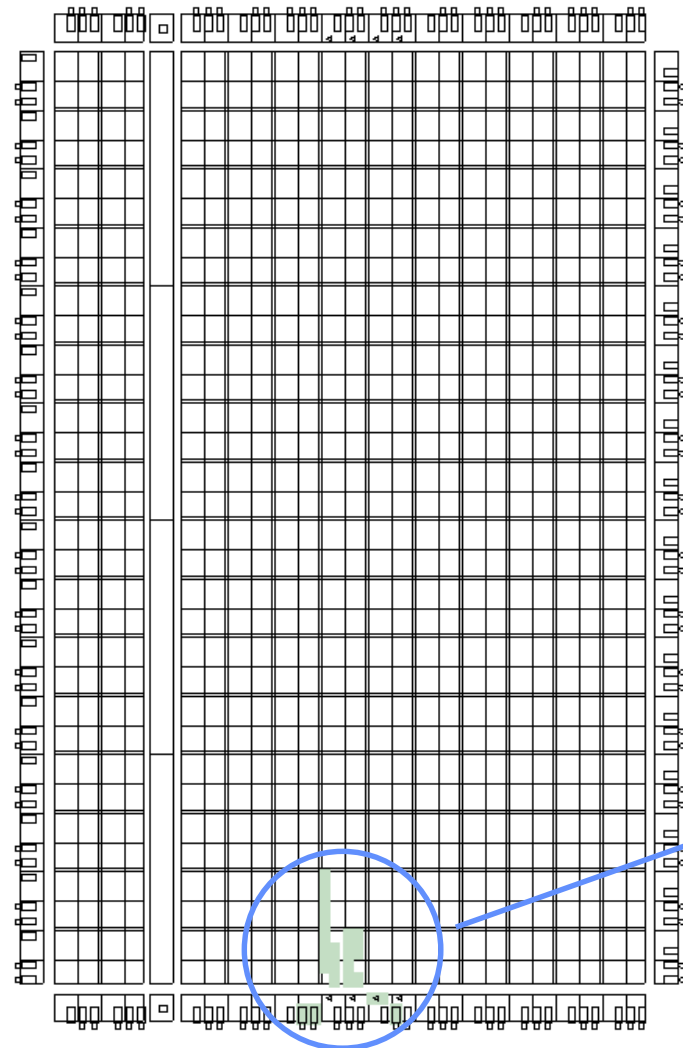
IO-Pin (IOB)

Flip-Flop (FF)

Look-Up-Table (LUT)

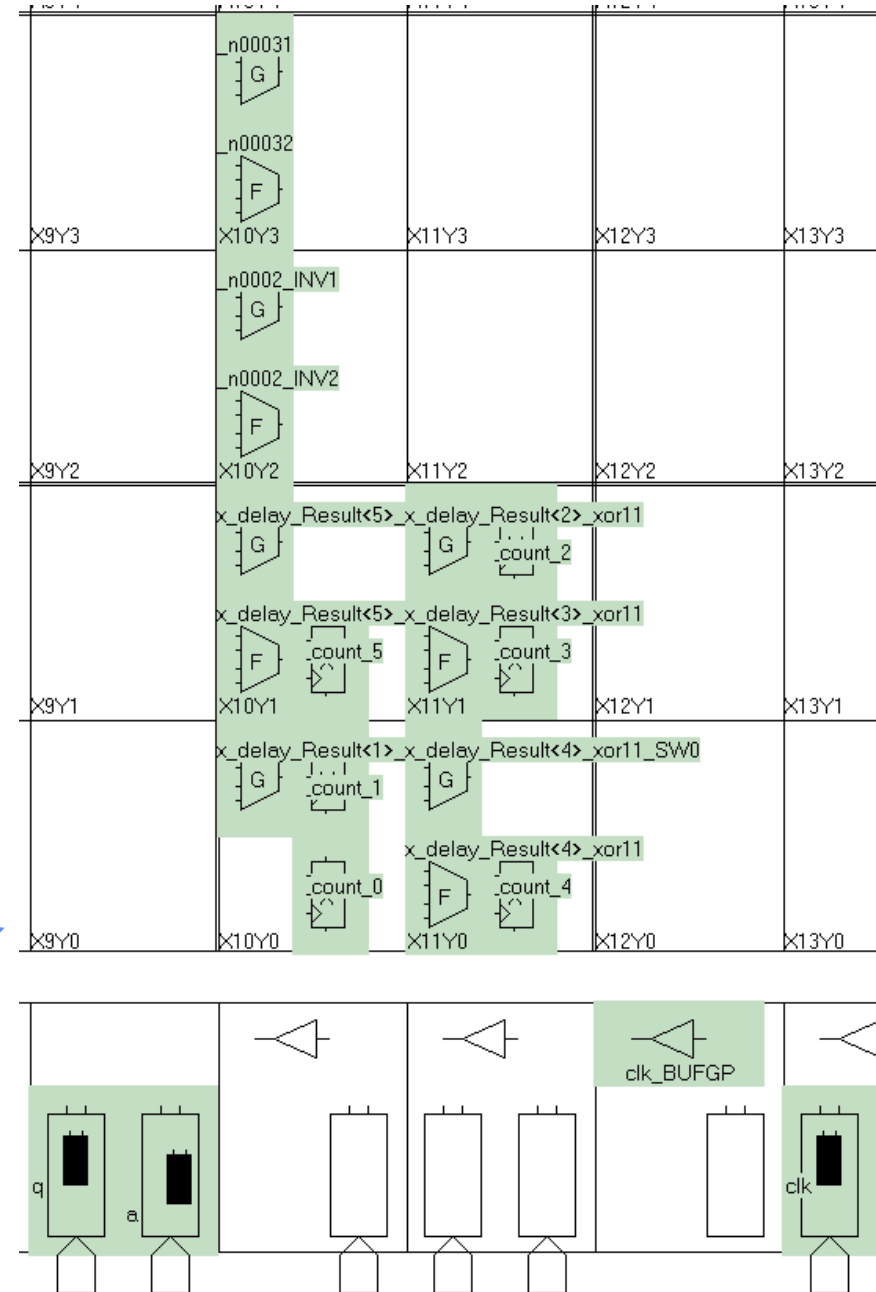


Beispiel: Placement

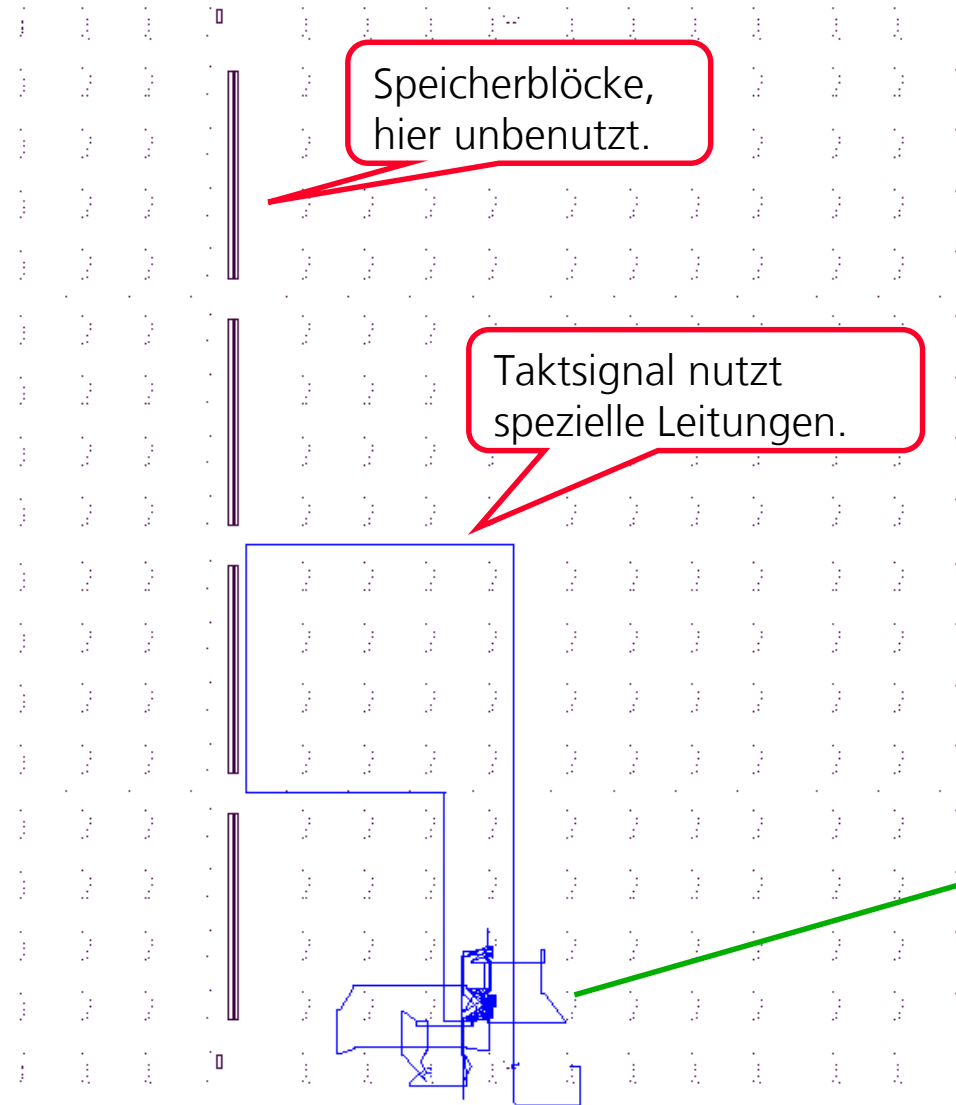


Gesamtansicht

Zoom



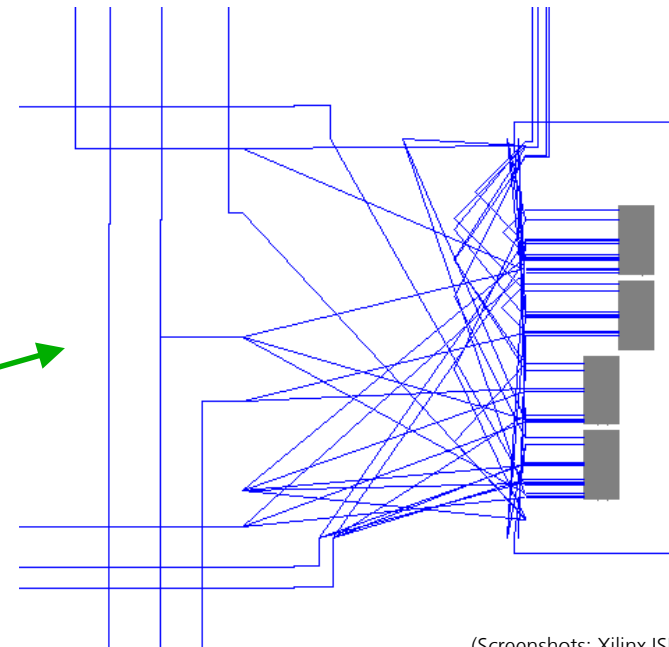
Beispiel: Routing



Links: Gesamtansicht.

Rechts: Zoom in Logikblöcke.

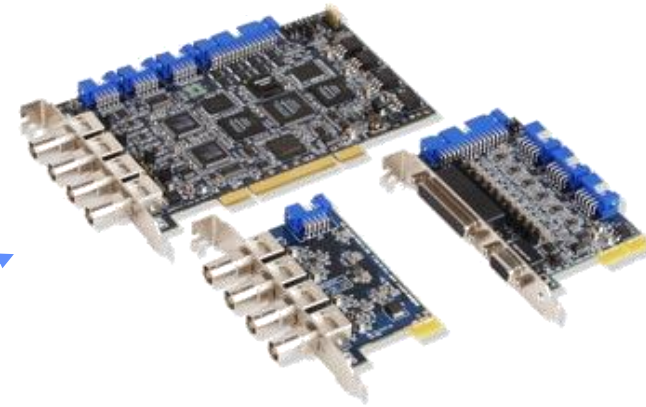
- Navigation durch Suche nach Signalnamen möglich, aber mühsam.



(Screenshots: Xilinx ISE)

9.6 Systeme mit FPGAs

- FPGAs werden in Produkten aus verschiedenen Bereichen eingesetzt
- Beispiele, Bilder von Intel/Altera
 - Matrox: „Morphis“ - JPEG 2000 Video Capture Board
 - Panasonic: AG-DVX100A Camcorder
 - AVM FritzBox 7490, 7580 und weitere
- Beispiele Xilinx auf Webseite:
<http://www.xilinx.com/about/xilinx-go/powered-by-xilinx.html>



Bulat LLC

BM1000 Series & BM100 Series Virtual CPE,
Powered by Zynq



ZF

ZF ProAI Gen 3 for Automated Driving to Level
4: Powered by Xilinx



DELTACAST

DELTA-Series Products for Professional Digital
Video, Powered by Kintex

FPGA im Speedport W 701V

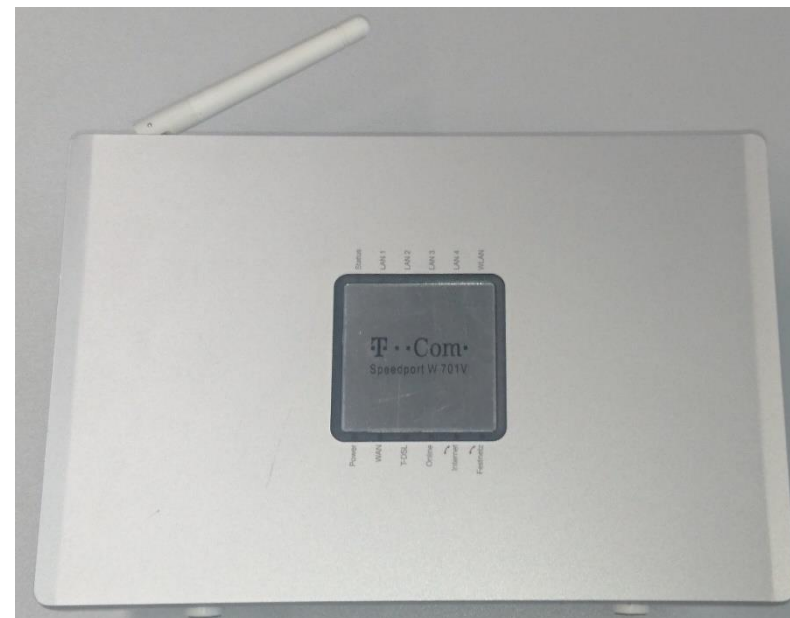
WLAN Router Speedport W 701V

- Einführung ca. 2007
- DSL-Anbindung, WLAN, 4*LAN, 2*TAE-Stecker

Mittlerweile obsolet:

“die Firmware für dieses Produkt steht nicht mehr zum Download zur Verfügung.
Sie entspricht inzwischen nicht mehr den aktuellen Sicherheitsanforderungen.
Einen Ersatz wird es nicht geben.“

<https://www.telekom.de/hilfe/geraete-zubehoer/router/weitere-router/speedport-w-7xx-serie/speedport-w-701v>

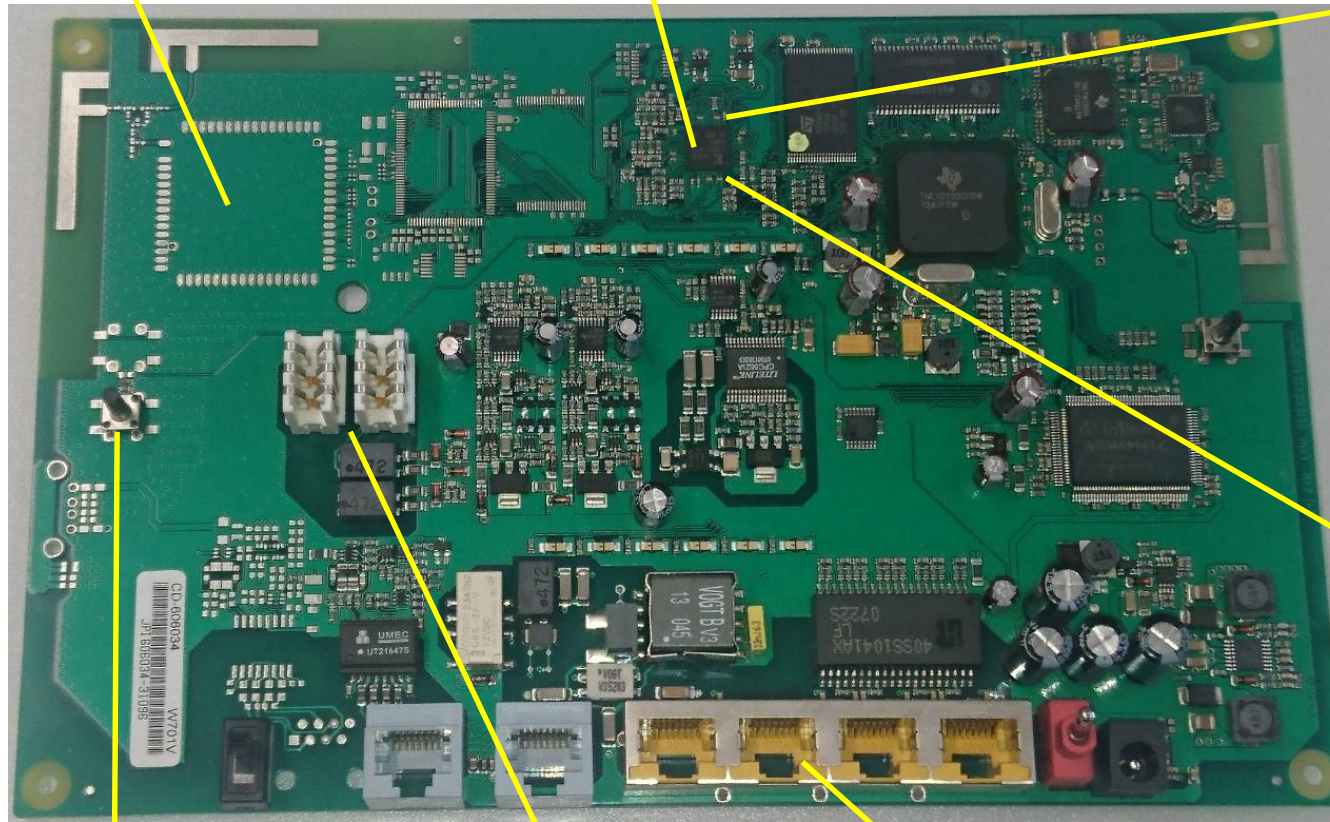


Artikel über moderne FritzBox7490:
Router-Funktion und Innenleben, c't 17/2016, Seite 162.

Platine des Speedport W 701V

unbestückt für
Produktvariante

FPGA Xilinx XC3S250E



Taster

TAE-Buchsen

LAN-Buchsen