

# Programmierbare Logik für die Displayansteuerung

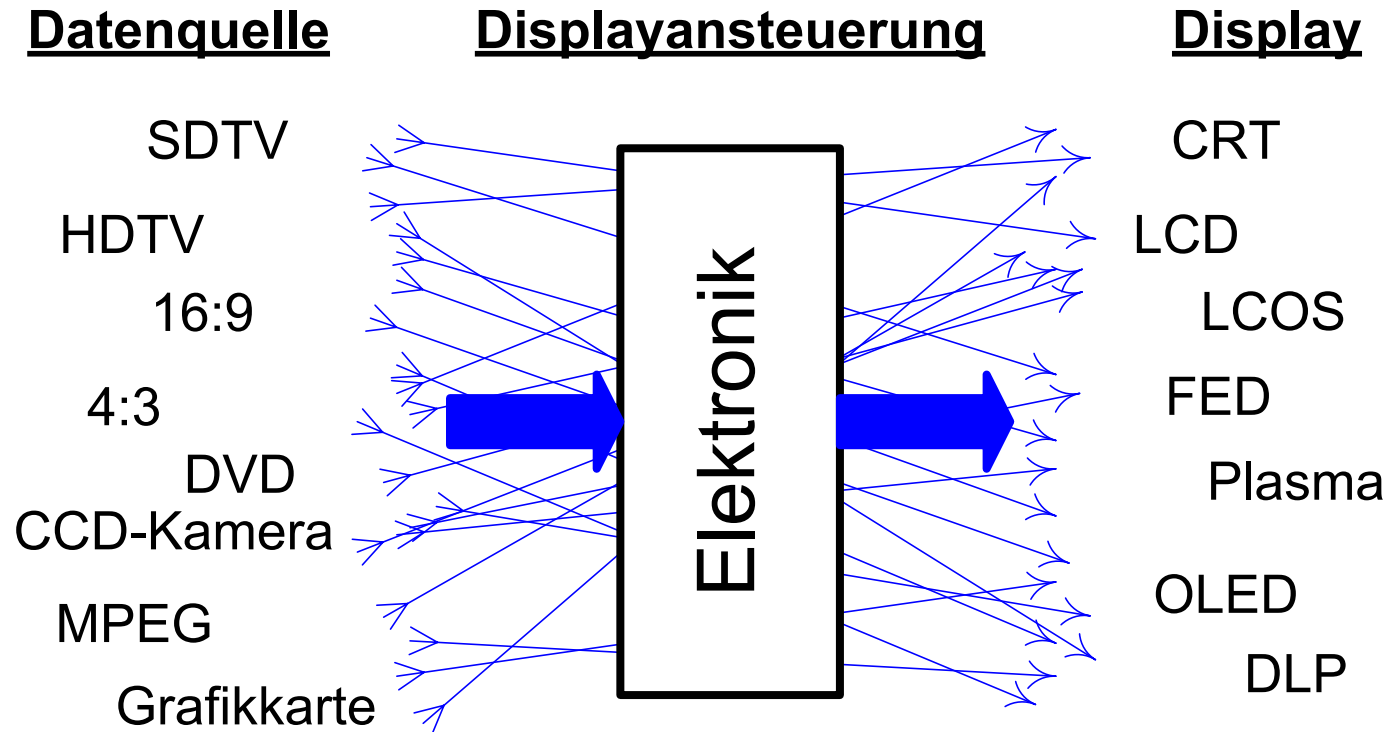
Electronic Displays 2005, Wiesbaden

Prof. Dr.-Ing. M. Winzker

Fachhochschule Bonn-Rhein-Sieg, Sankt Augustin

- Einleitung
- Algorithmen
- Anforderungen
- Programmierbare Logik
- Erfüllung der Anforderungen
- Konsequenzen und Zusammenfassung

# Einleitung



Informationen einer Datenquelle sollen auf einem Display dargestellt werden.

- Im Idealfall sind Datenquelle und Display aufeinander abgestimmt.
- Durch die Vielfalt an Formaten für Datenquelle und Display ist jedoch häufig eine mehr oder weniger komplexe Anpassung erforderlich.

# Algorithmen zur Datenanpassung

Die zur Datenanpassung erforderlichen Algorithmen lassen sich grob (!) in drei Kategorien einteilen:

- Gedächtnislose Algorithmen
- Lokale Algorithmen
- Globale Algorithmen

## Gedächtnislose Algorithmen

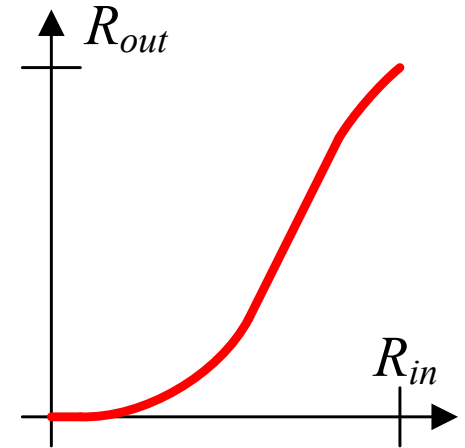
- Jeder Bildpunkt wird unabhängig von anderen Bildpunkten bearbeitet.
- Beispiele:
  - Kontrastanpassung
  - Gammakorrektur
  - Farbraumkonvertierung



# Algorithmen zur Datenanpassung (II)

## Kontrastanpassung:

- Die Farbkanäle werden einzeln bearbeitet.
  - Meist drei Farbkanäle, z.B. Rot, Grün, Blau (RGB) oder Luminanz und zweimal Chrominanz (YUV).
- Am flexibelsten ist eine Tabelle:
  - Für 8 Bit Eingabe und Ausgabe sind  $2^8=256$  Tabelleneinträge zu je 8 Bit erforderlich.
  - ➔ 3 Tabellen = 6 kBit Speicherbedarf



## Farbraumkonvertierung:

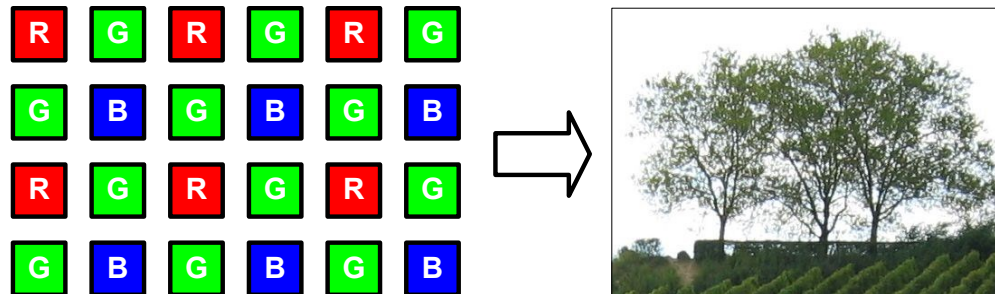
- Alle drei Farbkanäle werden gemeinsam ausgewertet.
- Eine Tabelle würde für 8 Bit Daten  $2^{8+8+8}=16M$  Einträge erfordern.
- Nur arithmetische Funktionen und Fallunterscheidungen sind möglich.
- Beispiel: YUV-nach-RGB Wandlung

$$R = Y + 1,140 \cdot V; \quad G = Y - 0,395 \cdot U - 0,581 \cdot V; \quad B = Y + 2,032 \cdot U$$

# Algorithmen zur Datenanpassung (III)

## Lokale Algorithmen

- Für die Bearbeitung eines Bildpunktes wird die nähere Umgebung eines Bildpunktes berücksichtigt.
- Beispiele:
  - Aufbereitung von Rohdaten einer CCD-Kamera.
    - Rote, grüne und blaue Bildinformation wird aus zwei oder mehr Zeilen zusammengefügt.



- Filter.
  - Anzahl an Bildpunkten typischerweise 3 bis etwa 10~20.

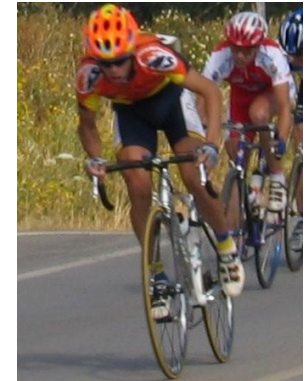
→ Arithmetik und Speicher für Bildzeilen erforderlich.

Beispiel für Speicherbedarf: 4 Zeilen eines VGA-Bildes (640\*480 Pixel) = 60 kBit.

# Algorithmen zur Datenanpassung (IV)

## Globale Algorithmen

- Für die Bearbeitung eines Bildpunktes werden Bildpunkte aus anderen Teilen des Bildes oder von vorherigen Bildern benötigt.
- Beispiele:
  - Bilddrehung.
  - Deinterlacing.
  - Änderung der Bildwiederholfrequenz.
  - Rauschreduktion.



➔ Speicher für ganze Bilder erforderlich.

Beispiel für Speicherbedarf: Ein VGA-Bild benötigt 7 Mbit Speicher.

### Anmerkung:

Die Einteilung in Algorithmenklassen ist kein festes Raster, sondern dient zur Analyse der Anforderungen.

# Anforderungen an Elektronik zur Datenanpassung

## Arithmetik

- Additionen, Multiplikationen, Vergleiche.
- Wortbreite meist 8 bis 12 bit, für Zwischenwerte auch mehr.

## Speicher

- Tabellen und Zeilenspeicher mit z.B. 6 bis 60 kBit Speicherkapazität.
  - Bei größeren Bildformaten auch über 100 kBit.
- Bildspeicher mit mehreren Mbit Speicherkapazität.
  - Effizient nur durch spezielle Speicherbausteine zu realisieren, z.B. DRAM.

## Eingangs- und Ausgangsinterface

- Ca. 60 Signalleitungen für einfache Datenkonvertierung.
  - 24 Eingangssignale, 24 Ausgangssignale, Steuerleitungen.
- Über 200 Signalleitungen für komplexe Verarbeitung möglich.
  - Zwei Dateneingänge, Datenausgang, CPU-Anbindung, externer Speicher.



# Elektronik zur Datenanpassung

- Etablierte Displays:
    - Üblicherweise sind ASICs zur Displayansteuerung verfügbar.
    - Für große Stückzahlen können spezielle Lösungen als ASIC entwickelt werden.
  - ASICs jedoch zu teuer oder unflexibel für:
    - Neue Displaytechnologien.
    - Prototypen.
    - Spezialanwendungen mit kleiner Stückzahl.
- ➔ Programmierbare Digitalschaltungen.



# Programmierbare Logik

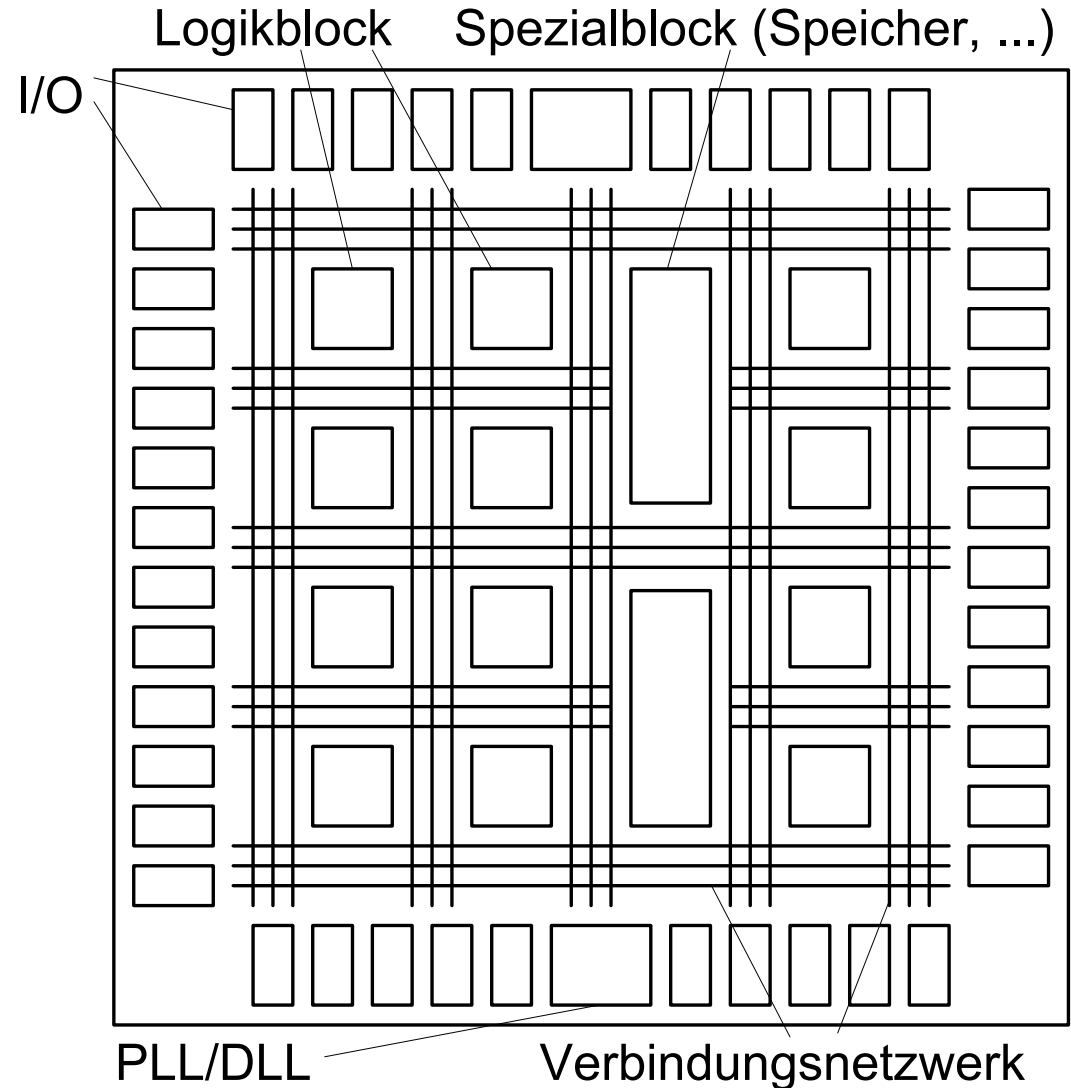
## **FPGAs** („Field-Programmable-Gate-Array“):

- Standardbaustein zur Konfiguration (Programmierung) durch Anwender.
- Verschiedene Hersteller mit ähnlichen aber nicht kompatiblen Produkten.
- Herstellerunabhängige Schaltungsentwicklung durch Hardwarebeschreibungssprache.
  - Sprachen: VHDL, Verilog, neuerdings auch „C“ und „C“-Varianten.
  - Gewisse Anpassung an Hersteller trotzdem erforderlich.
- Ursprünglich verbanden FPGAs andere Bausteine („Glue Logic“).
- Heute umfassen FPGAs komplette digitale Systeme („System-on-a-Chip“).
- Kosten in der Größenordnung von unter 10 bis einigen 100 Euro (je nach Komplexität, Stückzahl, ...).

# Programmierbare Logik (II)

## Aufbau eines FPGAs:

- „Logikblöcke“:  
Programmierbar mit  
UND/ODER-Funktion und  
Flip-Flops.
- Spezielle Funktionsblöcke,  
insbesondere:
  - Speicherblöcke.
  - Schnelle Multiplizierer  
für große Integer-Zahlen  
(z.B. 18 Bit \* 18 Bit).
  - Taktbehandlung:  
PLL, DLL.
- Programmierbares  
Verbindungsnetzwerk.



# FPGA: Xilinx Spartan-3

- Fa. Xilinx bietet mehrere „FPGA-Familien“ an.
- Die Spartan-Bausteine sind auf Kosten optimiert.
- Die Familie „Spartan-3“ umfasst 8 Bausteine unterschiedlicher Komplexität.

## Arithmetik

- Additionen und Vergleiche erfolgen durch Logikblöcke.
- Zwischen 8 und 104 Multiplizierer für 18 Bit Zahlen verfügbar.

## Speicher

- Konfigurierbare Speicherblöcke mit insgesamt 72 kBit bis 1,8 Mbit Kapazität.
  - **Achtung:** Je nach Anforderung kann nur ein Teil des Speicher genutzt werden.

## Eingangs- und Ausgangsinterface

- Zwischen 63 und über 700 I/O-Pins
- Verschiedene I/O-Standards verfügbar, auch für DDR-DRAM und LVDS.

# FPGA: Altera / Lattice

- Kostenoptimierte FPGAs anderer Hersteller haben ähnliche Eigenschaften.

## Altera Cyclone-II

- Arithmetik: Multiplizierer können für 18 Bit oder 9 Bit konfiguriert werden.
- Speicher: Andere Größe der Speicherblöcke.
- I/O-Interface: Verschiedene Standards verfügbar.

## Lattice ECP

- Arithmetik: Multiplizierer für 36 Bit, 18 Bit oder 9 Bit.
  - Speicher: Andere Größe der Speicherblöcke.
  - I/O-Interface: Verschiedene Standards verfügbar.
- ➔ Prinzipiell sind alle betrachteten FPGAs für die Displayansteuerung geeignet.
- ➔ Einschränkungen bei besonderen Anforderungen möglich:
- Z.B. Frequenzbereich für PLL.

# Ausnutzung von Speicherblöcken

- Bei der Abschätzung des Speicherbedarfs einer Anwendung muss die Struktur der verfügbaren Speicherblöcke berücksichtigt werden.

Beispiel: Zeilenspeicher für VGA, d.h. 640 Pixel, je 24 Bit = 15 kBit.

- Xilinx: Speicherblöcke mit 18 kBit.
  - Mögliche Konfigurationen 1K\*18Bit, 2K\*9Bit, 4K\*4Bit, 8K\*2Bit, 16K\*1Bit.
  - Zwei Speicherblöcke erforderlich.
  - 36 kBit Speicherbelegung (entspricht 42% Ausnutzung).
- Altera: Speicherblöcke mit 4,5 kBit.
  - 22,5 kBit Speicherbelegung.
- Lattice: Speicherblöcke mit 9 kBit.
  - 27 kBit Speicherbelegung.

**Anmerkung:**

Dieses Beispiel zur Speicherbelegung soll keine Bewertung der verschiedenen Anbieter darstellen.

# Konsequenzen und Zusammenfassung

- Mit FPGAs können flexible Displayansteuerung auch in kleinen Stückzahlen wirtschaftlich realisiert werden
- Für die Signalverarbeitung sind spezielle Funktionsblöcke vorhanden:
  - Multiplizierer.
  - Speicherblöcke.

Bei der Anwendung von FPGAs muss jedoch beachtet werden:

- Die Spezialblöcke sind nur in begrenzter Anzahl vorhanden.
- Der Funktionsumfang einer Displayansteuerung sollte darum bei Projektstart möglichst konkret spezifiziert werden.
  - Meist ist der nächst größere Baustein der FPGA-Familie pinkompatibel verfügbar, aber zu einem deutlich höheren Preis.
- Durch Anpassung an die Spezialblöcke kann der Aufwand (und damit der Preis) möglicherweise deutlich reduziert werden.
  - Beispiel: Durch eine Wortbreite von 9 Bit können eventuell mehr Multiplizierer genutzt werden als bei 10 Bit Wortbreite.

