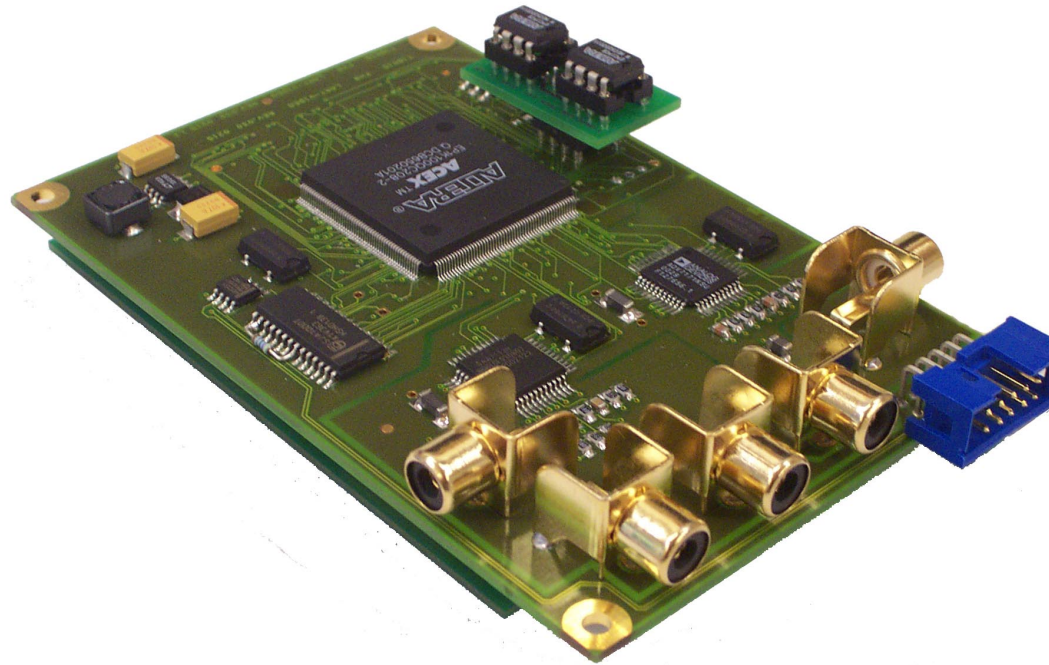


Konzeption und Implementierung einer Videodigitalisierung und Videoausgabe unter Embedded Linux



Kolloquium von Frank Schwanz

Aufgabenstellung

Im Rahmen des Projektes "Initiative Intelligente Autonome Systeme" wird an der FH Brandenburg ein Kernel für intelligente autonome Systeme entwickelt. Eine Teilkomponente des R-Cube-Systems ist eine eigenständige Bildverarbeitungskarte mit StrongARM-Prozessor und ARM-Linux.

Zielstellung des Themas ist die Konzeption und Implementierung der Videodigitalisierung und Videoausgabe der Bildverarbeitungskarte. Hierbei sind Schnittstellen zum Auslesen der Bilddaten (vorzugsweise Video4Linux) und Ausgabe von Bilddaten (vorzugsweise Framebuffer-Device) zu entwickeln. Die Umsetzung soll besonderen Wert auf die mögliche Nutzung in kommenden Linux-Kernelversionen sowie die Performance der Lösung legen. Die Funktionsfähigkeit des Systems ist durch geeignete Teststellungen zu evaluieren.

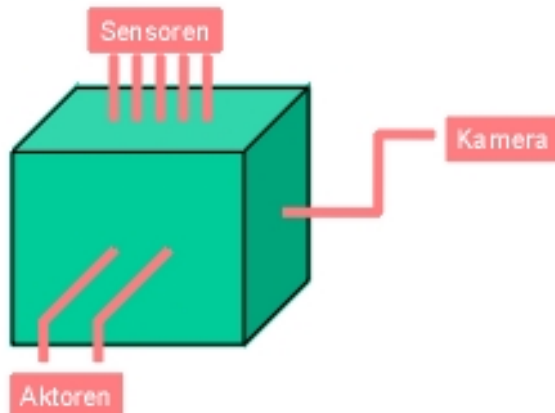
Überblick

- Was ist das RCube-Projekt
- Anforderungen an das BV-Board
- Die Hardware des BV-Board
- Ressourcenverbrauch
- Hardware-Architektur
- Software-Architektur
- Ergebnisse

Das RCube-Projekt

 R-Cube Vision

Autonome mobile sehende Plattform



CPU-Board

- LART-Board mit 32-Bit-StrongARM-Prozessor
- Betriebssystem Linux

Aksen-Board

- Analoge und digitale Sensoren
- Motor- und Servoausgänge
- 8Bit Mikrocontroller

BV-Board

- LART-Board
- Adapterkarte LartVIO für Videodigitalisierung und -ausgabe
- Entwicklung der Firma Tigris Elektronik in Berlin

Anforderungen an das BV-Board

Videodigitalisierung

- 720x576, sowie 360x288 Bildpunkte bei 25 Bildern/s
- RGB24, andere Formate optional
- Helligkeit, Kontrast, Sättigung einstellbar
- einfache Methoden Bilder einzulesen, Möglichkeiten zur Performanceoptimierung

Videoausgabe

- maximale Auflösung der Videodigitalisierung, optional kleine Auflösungen
- RGB24, optional andere Formate
- einfache Methoden Bilder darzustellen

allgemeine Anforderungen

- Dem Prozessor muss genügend Leistungsreserven für die Verarbeitung der Bilder bleiben.
- Powermanagementmethoden

Die Videoerweiterung LartVIO

Videodigitalisierung

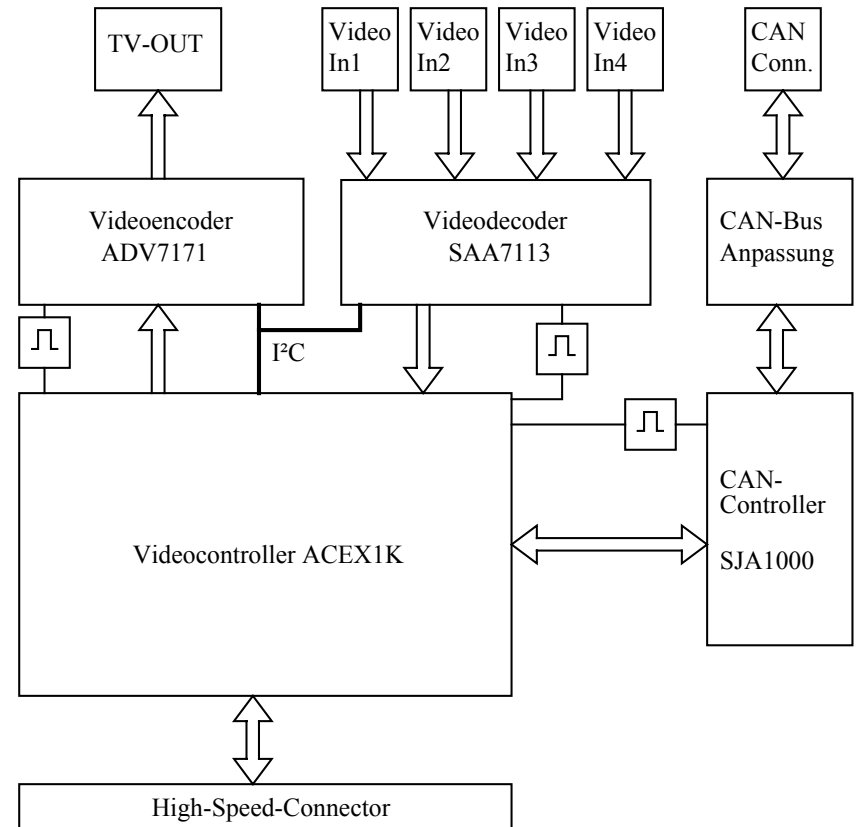
- Videodecoder SAA7113
- 4-Videoeingänge (umschaltbar)
- Ausgabe CCIR 601 Videostrom (PAL 720x576, YUV)

Videoausgabe

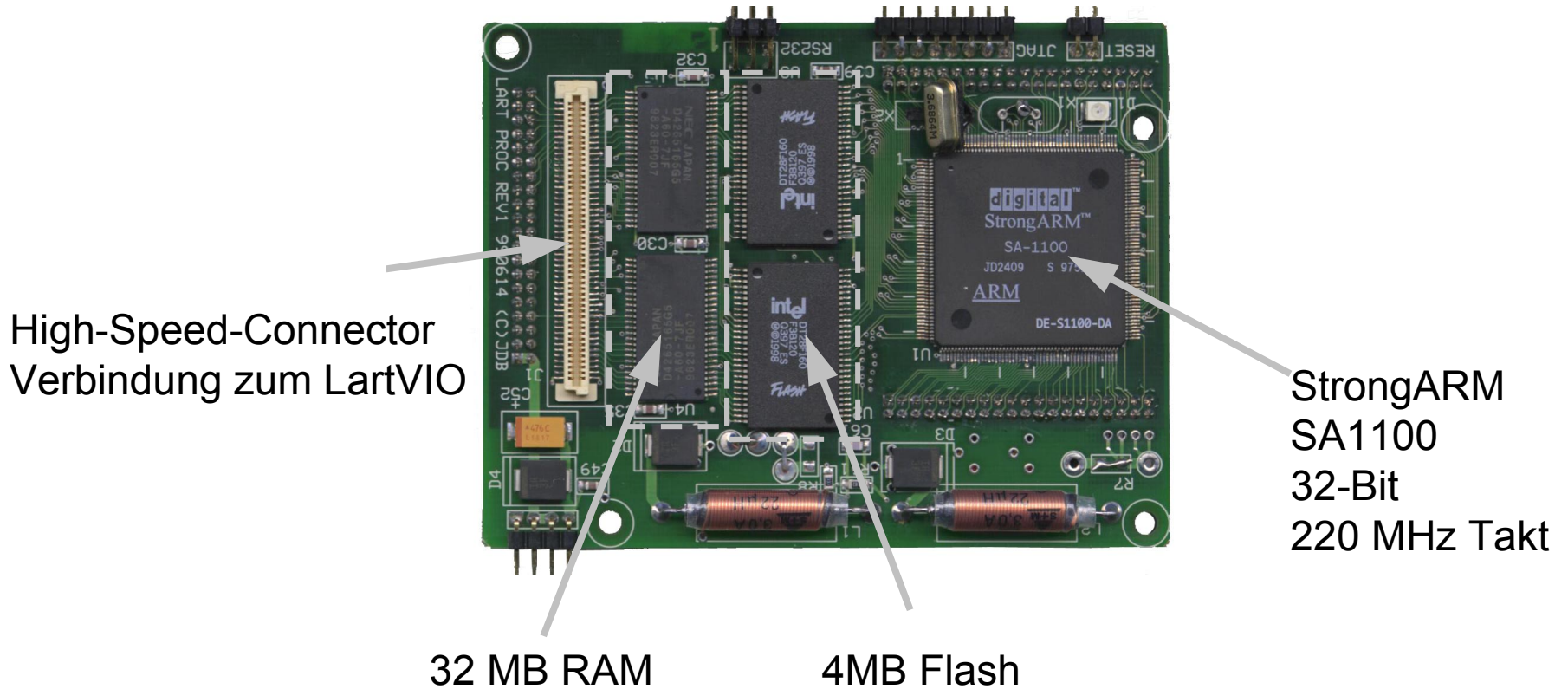
- Videoencoder ADV7171
- Übergabe YUV Videostrom

Videocontroller

- FPGA der Firma Altera
- bis zu 4992 Logikelemente
- ca. 6 KB RAM



Das Prozessorboard LART



Benötigte Ressourcen für die Übertragung in den Hauptspeicher

Bild 720x576 Pixel, Farbtiefe 16 Bit (2 Byte) YUV, YCbCr

$$720 * 576 * 2 \text{ Byte} = 810 \text{ KByte/Bild}$$

$$810 \text{ KByte/Bild} * 25 \text{ Bilder/s} = 20 \text{ MByte/s}$$

Bild 720x576 Pixel, Farbtiefe 24 Bit (3 Byte) RGB24

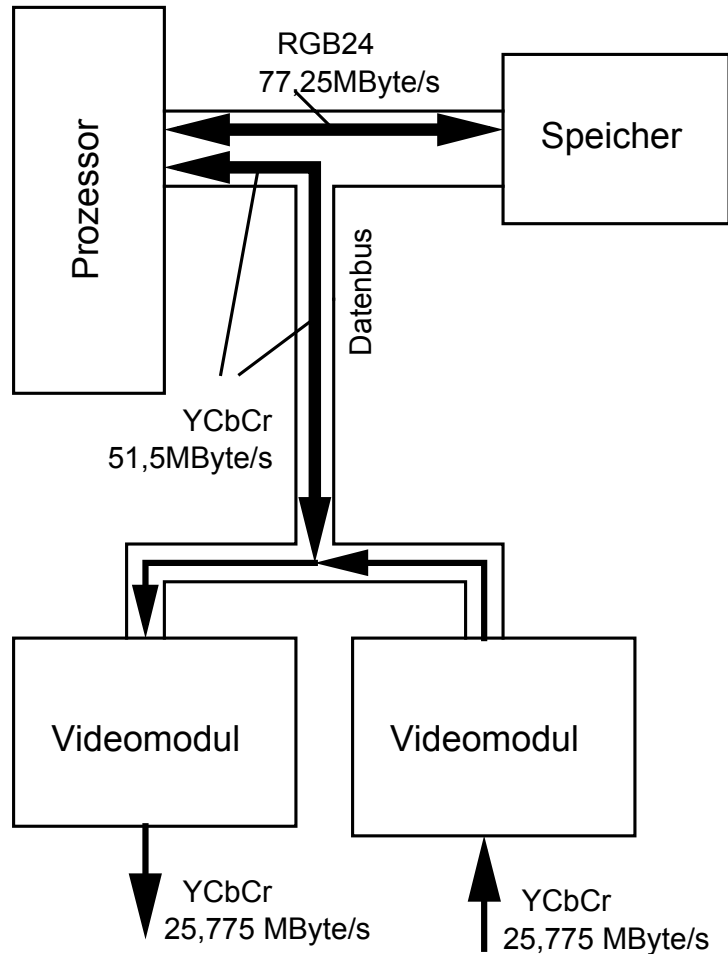
$$720 * 576 * 3 \text{ Byte} = 1,2 \text{ MByte/Bild}$$

$$1,2 \text{ MByte/Bild} * 25 \text{ Bilder/s} = 30 \text{ MByte/s}$$

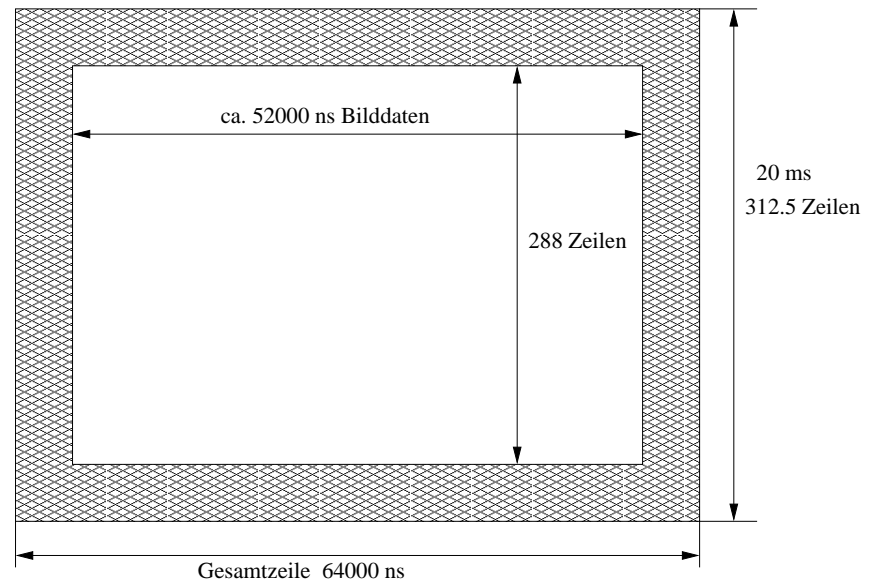
2 Videoströme je 20 MByte zwischen LartVIO und Prozessor

2 Videoströme je 30 MByte zwischen Prozessor und Hauptspeicher

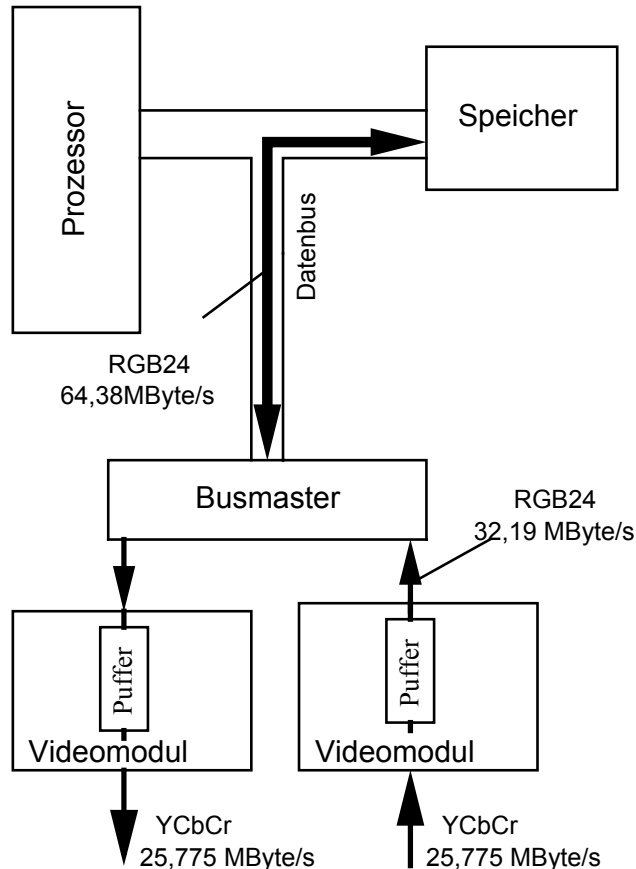
Speicherbus des LART



- max. Bandbreite zwischen Prozessor und Hauptspeicher ca. 100 MByte/s
- notwendige Bandbreite ca. 128 MByte/s

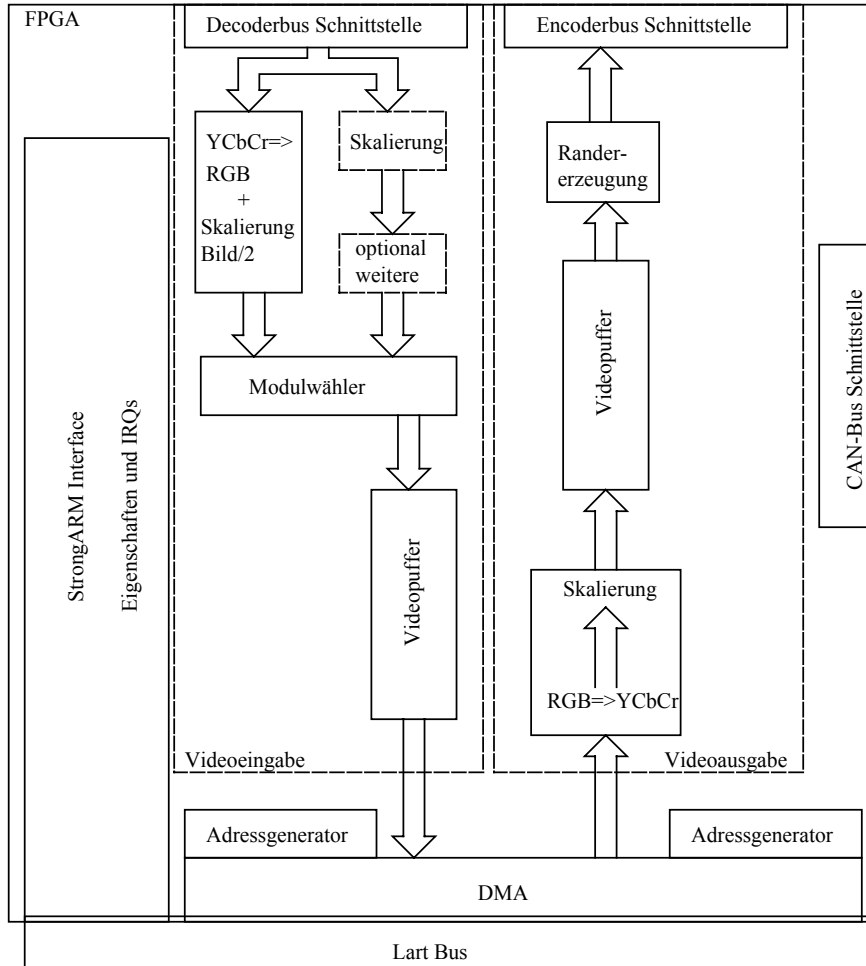


Busmaster



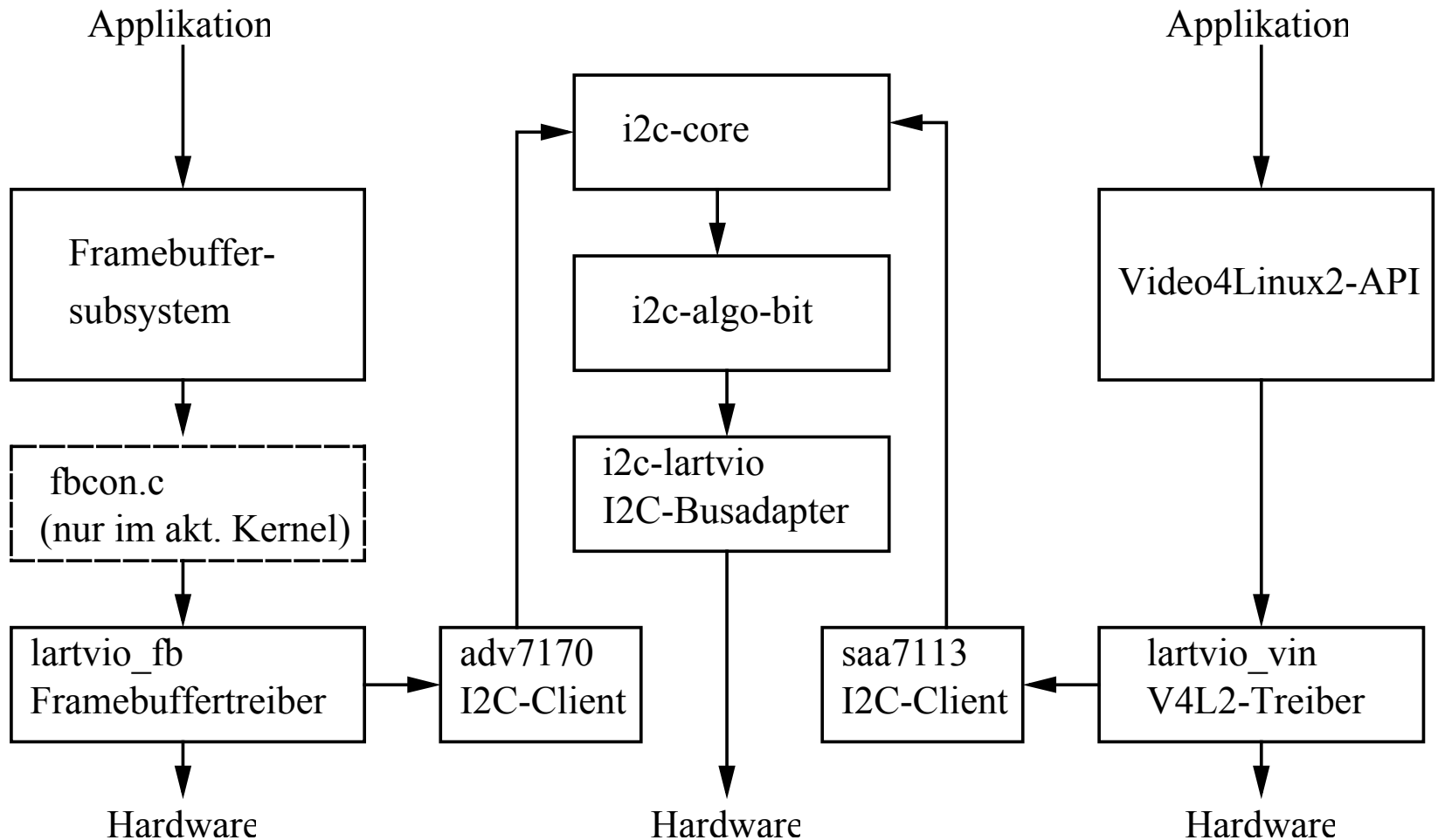
- Nutzen der Bildaustastzeiten durch Pufferung
- Übertragung der Videodaten direkt in den Hauptspeicher
- keine Beteiligung des Prozessors, daher Bearbeitung des Videostroms im Videocontroller notwendig.

FPGA-Design

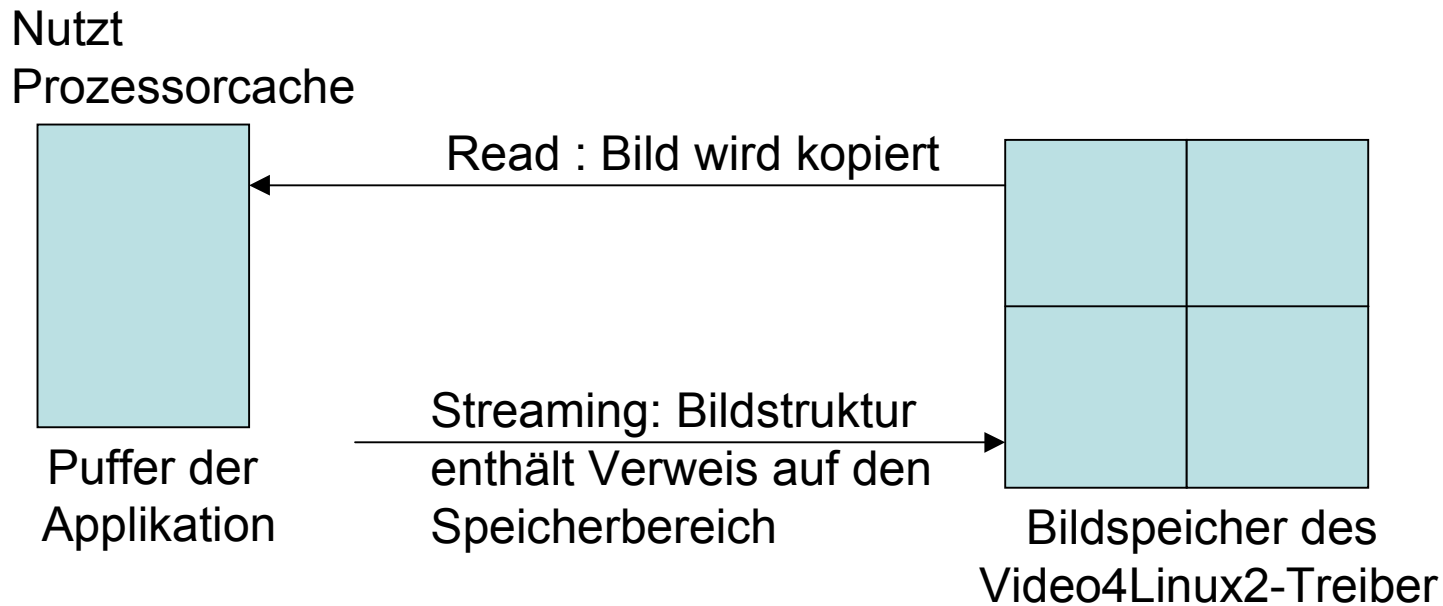


- erweiterbar durch Module
- verschiedene Farbformate und Filter möglich
- portierbar durch Anpassung von DMA-Modul und StrongARM-Interface
- Skalierung 720x576 und 360x288
- versch. Auflösungen bei Videoausgabe durch Randerzeugung

Softwarearchitektur



Zugriff auf Bilder (Video4Linux2)



Die Treiberschnittstelle

```
#include <linux/videodev.h>

struct v4l2_format fmt;          /* Videoformat */
int fd;                          /* Filedescriptor */
int bytes, err;                 /* Rückgabewerte */
char data[720*576];             /* Array für Bilddaten */

/* Treiber öffnen */
fd = open("/dev/video", O_RDWR);

/* aktuelle Formateinstellungen lesen */
fmt.type = V4L2_BUF_TYPE_CAPTURE;
err = ioctl(vid, VIDIOC_G_FMT, &fmt);

/* Setzen der gewünschten Auflösung und Farbtiefe */
fmt.fmt.pix.height = 576;
fmt.fmt.pix.width = 720;
fmt.fmt.pix.pixelformat = V4L2_PIX_FMT_RGB24;

/* Neue Auflösung dem Treiber übergeben */
err = ioctl(vid, VIDIOC_S_FMT, &fmt);

/* Lesen eines Bildes */
bytes = read(fd, data, 720*576*24 / 8);
```

Das fertige BV-Board

Videodigitalisierung

Auflösungen

- 720x576 u. 360x288

Farbformate

- RGB24, YUV
- leicht erweiterbar

Powermanagement

- nur aktive bei Zugriff auf die Treiber

Treiberschnittstelle

- Standardisiert
- Zugriff auf Bilder mit wenigen Zeilen Code
- versch. Methoden zur Performanceoptimierung

Performance

- 12,5 Bilder/s 720x576
- 25 Bilder/s 360x288
- 25 Bilder/s 720x576 bei Anzeige im Framebuffer

Videoausgabe

Auflösungen

- 86x120 bis 720x576 in Schritten von 4x4

Farbformate

- RGB24

Powermanagement

- extern abschaltbar
- automatische Abschaltung über Consoletreiber

Treiberschnittstelle

- Standardisiert
- Bilder mit wenigen Zeilen Code
- Standardsoftware verwendbar
- Integrierte Console