

Konzeption und Implementierung einer Videodigitalisierung und Videoausgabe unter Embedded Linux

Diplomarbeit, vorgelegt von Frank Schwanz

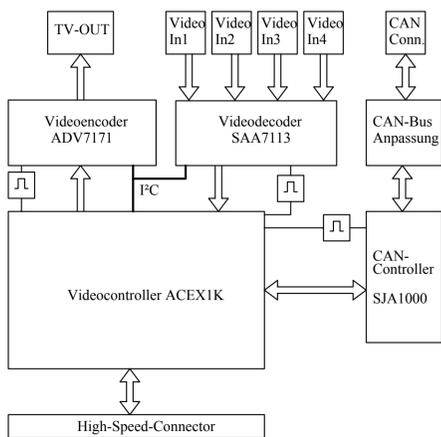


Aufgabenstellung

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 Bildverarbeitungs-karte mit StrongARM-Prozessor und ARM-Linux.

Zielstellung des Themas ist die Konzeption und Implementierung der Videodigitalisierung und Videoausgabe der Bildverarbeitungs-karte. 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.

Systemarchitektur



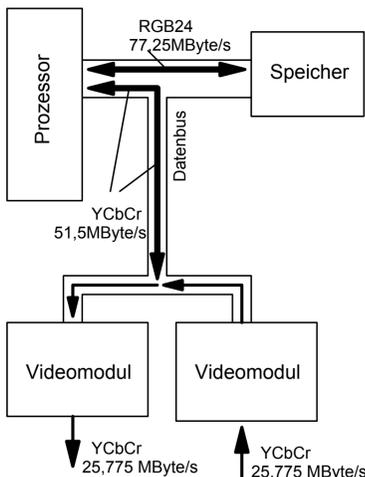
Blockschaltbild des LartVIO

Die Bildverarbeitungsplatte LartVIO erweitert das embedded Prozessorboard LART um eine Videoein- und -ausgabe. Das LartVIO besitzt 4 Videoeingänge und 1 Videoausgang.

Die Kommunikation zwischen den Grafikkchips und dem LART-Board erfolgt über den programmierbaren Logikbaustein ACEX1K.

Vorüberlegungen

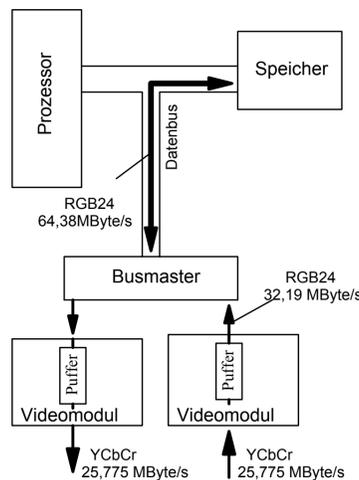
Die maximale Datentransferleistung auf dem Speicherbus des LART-Board liegt bei ca. 100 MByte/s. Der zu übertragene Videostrom benötigt eine maximale Transferleistung von ca. 128 MByte/s, wenn ein Treiber auf dem Prozessor die Bearbeitung und Übertragung des Videostroms übernimmt.



Übertragung der Videodaten durch den Gerätetreiber

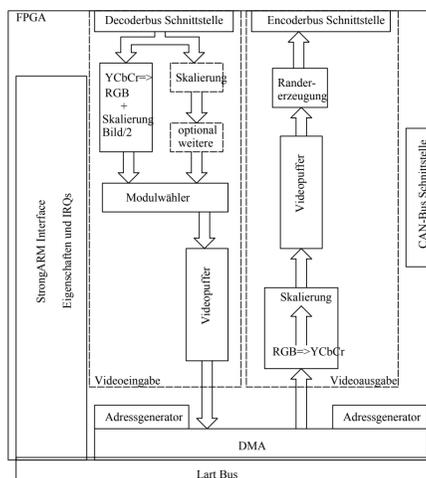
FPGA Design

Aufgrund der Ressourcenknappheit auf dem Hauptspeicherbus wurde Übertragung vollständig in den programmierbaren Logikbaustein integriert. Alle Videodaten werden im programmierbaren Logikbaustein bearbeitet und an den Hauptspeicher übertragen. Der Prozessor kann in den Busmastermode geschaltet werden. In diesem Modus gibt er den Speicherbus für externe Geräte frei.



Übertragung der Videodaten mittels Busmaster

Datenpuffer nutzen die Bildaustastzeiten um die notwendige Bandbreite zu reduzieren. Diese Lösung kommt mit einer maximalen Bandbreite von ca. 65 MByte/s aus.



Funktionseinheiten des programmierbaren Logikbausteins

Die Verarbeitung der Videodaten muss damit im programmierbaren Logikbaustein implementiert werden.

Die Videobarbeitung wurde modularisiert. Damit sind weitere Bearbeitungsmodulare, wie Farbraumkonverter, Filter etc. einfach zu integrieren.

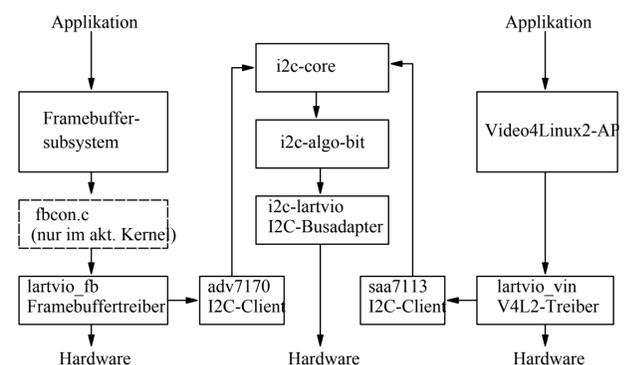
Software-Architektur

Das LART-Board läuft unter dem Betriebssystem Linux.

Für die Videoausgabe wurde das Subsystem für Grafikkarten, das Framebuffersubsystem, gewählt. Standardsoftware, wie grafische Oberflächen, sind ohne Anpassungen auf dem LART-Board lauffähig.

Der Gerätetreiber für die Videoeingabe nutzt die Video4Linux2-API, die im Linuxkernel 2.6 die alte Video4Linux-API ablösen soll.

Die Schnittstellen sind so gestaltet, dass der Einstieg in die Programmierung mit geringem Lernaufwand möglich ist. Für die Performanceoptimierung stehen zusätzliche Funktionen, wie Streamingtechnologien bei der Video4Linux2-API, bereit.



Gerätetreiberdesign

Fazit

Im Ergebnis stellt die Videoein- und ausgabekarte LartVIO in Verbindung mit dem Prozessorboard LART eine autonome Bildverarbeitungs-karte zur Verfügung. Durch verschiedene Auflösungen und Farbformate kann der Ressourcenverbrauch an die Anforderungen der Bildverarbeitungsanwendung angepasst werden.

Aufgrund der standardisierten Schnittstellen konnte die grafische Oberfläche QT durch ein einfaches Kompilieren für den StrongARM-Prozessor auf das LART-Board migriert werden.