

Inhalt

Einleitung.....	2
Digitale Bildverarbeitung.....	3
Funktionsweise Darstellung von Pixeln.....	3
Farbunterschiede erkennen	4
Übersicht Ist-Zustand	5
Modul: SDRAM_Pixelbuffer.....	6
Aufbau Intern: SDRAM_Pixelbuffer(Vereinfachte Version)	7
Modull: Debay	8

Einleitung

Das Regeln/die Regelung ist ein Vorgang bei dem eine Größe, die zu regelnde Größe (Regelgröße), fortlaufend erfasst, mit einer anderen Größe, der Führungsgröße verglichen und im Sinne einer Angleichung an die Führungsgröße beeinflusst wird. Kennzeichen für das Regeln ist der geschlossene Wirkungsablauf, bei dem die Regelgröße im Wirkungsweg des Regelkreises fortlaufend sich selbst beeinflusst. [Definition Regelung nach DIN 19226]

Diese Arbeit beschäftigt sich mit der Realisierung eines Messglied, bei denen geringe Totzeiten im Fokus stehen. Totzeiten können allgemein dafür sorgen, dass ein Regelkreis instabil oder träge wird und somit einer definierten Anforderung nicht mehr entsprechen könnte.

Die Messgröße soll die horizontale Position eines Objektes darstellen, die anhand seines Farbunterschied zum Rest einer Szene (Hintergrund) erkannt wird. Für die Erkennung eines solchen Objektes bietet sich eine Sensorlösung an, die Teile des elektromagnetischen Spektrums aufnimmt und diese in ein elektrisches Signal umwandelt.

Farbe ist, in dieser Arbeit durch die Auswahl des Sensors auf eine Kamera der Firma Semtech, durch einen Rot- Grün und Blaukanal (RGB) definiert. Diese drei Farbkanäle spiegeln die Grundfarben wieder, welcher der Sensor dem natürlichen Licht entnimmt.

Um die Totzeiten im Messglied gering zu halten, wurde bei der Auswahl der Kamera auf eine hohe Anzahl an Bildern die Sekunde geachtet. Die Auswertung der Bilder für die Erkennung eines Objektes und dessen Position erfolgt in einem Field Programmable Gate Array (FPGA-Board). Das FPGA-Board bietet die Möglichkeit die Signale von der Kamera in Echtzeit zu verarbeiten und auszuwerten und hat aufgrund seines Aufbaus weniger Limitationen als ein herkömmlicher Mikroprozessor, was die Bearbeitungsgeschwindigkeit angeht.

Zu Bearbeitungsbeginn der Studienarbeit war es möglich, die Daten der Kamerabilder mittels Camera Link von der Kamera auf das FPGA-Board zu übertragen und diese dort zwischenspeichern. Die zwischengespeicherten Daten konnten über den VGA-Ausgang auf einem angeschlossenen Monitor ausgegeben werden. [Lukas Herbst, Bilddatenvorverarbeitung in einem FPGA] Zudem war es möglich anhand von Graustufen (Schwarzweißbild) und dessen Wechsel in der Intensität bis zu drei Objekte zu erkennen.

Auf Basis der zuvor beschreibenden Funktionen werden neue Funktionen implementiert. Die Funktionsweise der bisherigen Objekterkennung im Graustufenbereich soll für die Erkennung im Farbbereich als Vorlage verwendet werden. So sollte es einfacher sein ein farbiges Objekt von einem anders gefärbten Hintergrund zu unterscheiden und somit auch dessen Position zu erkennen.

Wichtig bei der Erfassung ist das Herausfinden der relevanten Parameter, die eine robuste Objekterkennung erst möglich machen. So spielen beispielsweise die Beleuchtungszeit, Umgebungslicht und die Farbwahl von Objekt und Hintergrund eine große Rolle.

...Bezug auf den Aufbau der Studienarbeit

Digitale Bildverarbeitung

Diese Studienarbeit stützt sich in vielen Abschnitten aus Erkenntnissen der vorangegangenen Studienarbeit „Bild datenvorverarbeitung in einem FPGA, 25. Januar 2020“ von Lukas Herbst. Dessen Arbeit bezieht sich in vielen Teilen auf die Literatur von Rafael C. Gonzalez und Richard E. Woods, die zusammen das Buch „Digital Image Processing“ verfasst haben. Eine detaillierte Beschreibung zu diesem Abschnitt finden Sie in der Arbeit von Herrn Herbst. Folgend die Details die für diese Ausarbeitung am wichtigsten waren.

Funktionsweise Darstellung von Pixeln

Ein digitales Bild kann aus einer endlichen Anzahl aus Pixeln bestehen, so wird meist mit x die horizontale Breite und mit y die vertikale Höhe definiert. Die Gesamtzahl der Pixel in einem Bild ergibt sich aus dem Produkt von x und y .

So ergibt sich ein Raster aus Pixeln. Jeder Pixel kann mit der Funktion $f(x,y)$ einzeln ausgelesen werden.

					x
	f(0,0)	f(1,0)	f(2,0)	f(3,0)	f(4,0)
	f(0,1)	f(1,1)	f(2,1)	f(3,1)	f(4,1)
	f(0,2)	f(1,2)	f(2,2)	f(3,2)	f(4,2)
	f(0,3)	f(1,3)	f(2,3)	f(3,3)	f(4,3)
	f(0,4)	f(1,4)	f(2,4)	f(3,4)	f(4,4)
y					

Abbildung Bildverarbeitung 1: Digitales Bild Beispiel

Der Rückgabewert der Funktion $f(x,y)$ ist abhängig von der Definition des Pixels. So wird beispielsweise für ein Bild im Graustufenbereich nur ein einziger Wert definiert oder wie in Abbildung Bilderverarbeitung 1 ein RGB Farbwert. Ein RGB Farbwert verfügt über drei Informationen, diese definieren wie stark ein rot, grün oder blau Anteil ausgeprägt ist.

Beispiel zu Abbildung Bildverarbeitung 1: {rot = 255, grün = 255, blau = 0} = $f(0,0)$

Das Beispiel ergibt einen starken Gelbton. Die Zahl 255 stellt hier den maximal Wert für die Darstellung mit 8-Bit pro Farbkanal dar und wird auch Farbtiefe genannt. Je mehr Bits für die Farbtiefe zur Verfügung stehen, desto feiner können Farbunterschiede realisiert werden. Für diese Arbeit wird von einer Farbtiefe von 8-Bit ausgegangen.

Farbunterschiede erkennen

In [2.3 Mathematische Operationen zur Bilddatenverarbeitung, Bilddatenvorverarbeitung in einem FPGA, Lukas Herbst] beschrieben, können Unterschiede von Pixeln in eine Reihe wie folgt erfasst werden.

$$1. \text{Ableitung Definition von Gonzalles: } \frac{\partial f}{\partial x} = f(x+1) - f(x)$$

Das Ergebnis der ersten Ableitung beschreibt den Unterschied von einem zum nächsten Pixel.

Folgend eine Veranschaulichung zu einer Reihe die aus 5 Pixeln besteht. Jeder Pixel wird durch eine Graustufe repräsentiert.

	x				
$f(x):$	$f(0)$ = 100	$f(1)$ = 100	$f(2)$ = 200	$f(3)$ = 100	$f(4)$ = 100
	x				
$\frac{\partial f}{\partial x}:$	$f'(0)$ = 0	$f'(1)$ = 100	$f'(2)$ = -100	$f'(3)$ = 0	$f'(4)$ = ?

Abbildung Bildverarbeitung 2: Pixelreihe $f(x)$ und 1. Ableitung $f'(x)$, Graustufen in 8 Bit Darstellung

An der Stelle $f(0)$ ergibt die erste Ableitung null, da der Nachbar Pixel keinen Unterschied aufweist. Ein Sprung ist an einem hohen Betrag der ersten Ableitung zu sehen, so zum Beispiel an $f(1)$ oder $f(2)$. Hier ändert sich der Wert sehr stark im Vergleich zum Nachbarn. Eine Besonderheit stellen die Ränder da, hier kann man mit der Formel der ersten Ableitung keinen Wert berechnen und kann daher vernachlässigt werden.

Übersicht Ist-Zustand

Eine kurze Übersicht bietet folgende Grafik und beschreibt den vorgefundenen Zustand, der bei Beginn des Projektes übernommen wurde.

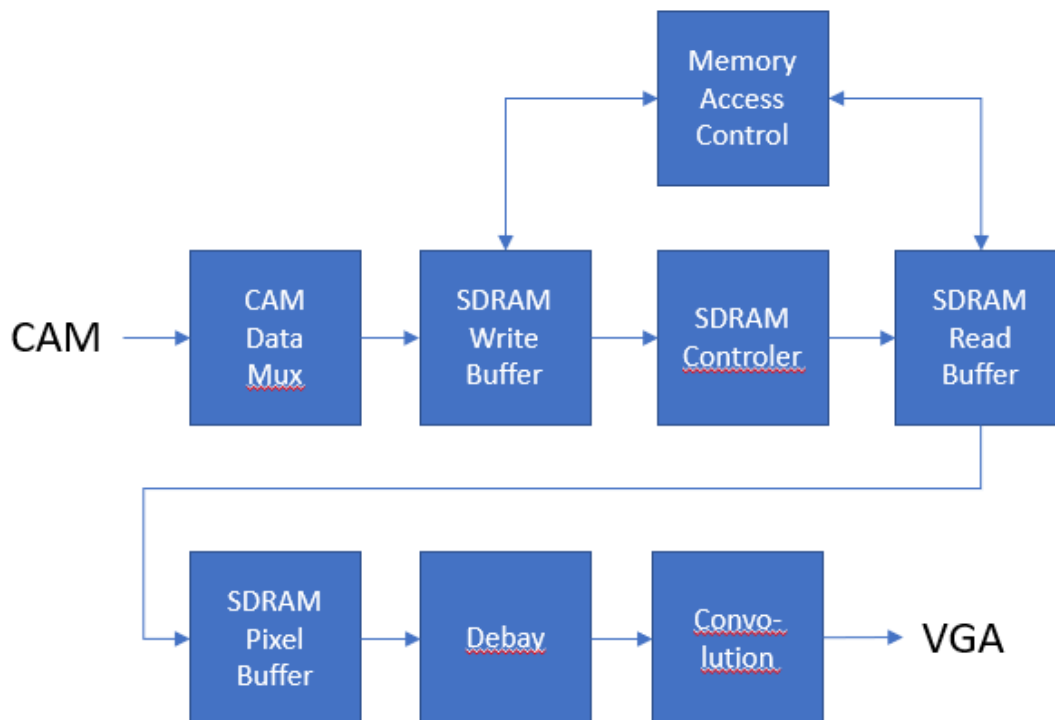


Abbildung Projekt 1: Übersicht der Module innerhalb des bisherigen Quartusprojekts

In der Ausarbeitung der Studienarbeit „Bildatenvorverarbeitung in einem FPGA“ von Herrn Herbst, sind die Module im Detail beschrieben. Daher werden im Folgenden die Module beschrieben an denen Änderungen oder direkter Gebrauch von gemacht wird erläutert und beschrieben.

Modul: SDRAM_Pixelbuffer

Durch vorherige Schritte liegen Bilddaten im SDRAM vor. Die einzelnen Pixel werden jeweils als Byte dargestellt und sind einzeln aus dem SDRAM lesbar. Für die weitere Bildverarbeitung und die Ausgabe auf einer VGA Schnittstelle werden immer 5x5 Pixelblöcke im Modul „SDRAM_Pixelbuffer“ gebuffert.

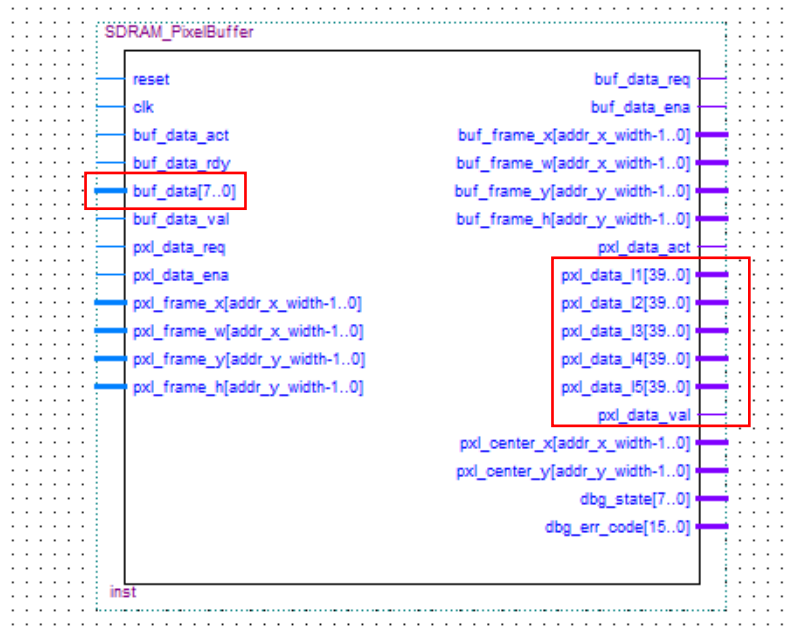


Abbildung SDRAM_Pixelbuffer 1: Blockschaltbild, rot markiert die wichtigsten IOs

Der 5x5 Byte Block ist zeilenweise auf den Ausgängen „pxl_data_l1[39 .. 0]“ bis „pxl_data_l5[39 .. 0]“ lesbar. Die Ausgänge dürfen erst gelesen werden, wenn der Ausgang „pxl_data_val“ einen logischen High-Pegel vorweist.

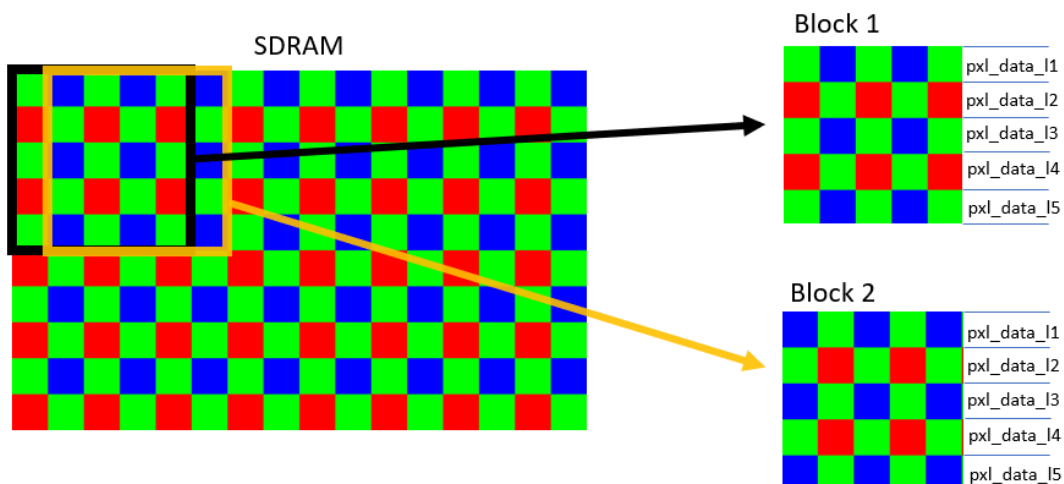


Abbildung SDRAM_Pixelbuffer 2: Schaubild Funktionsweise Pixelbuffer

Die fünf Ausgänge, für jeweils eine Zeile, sind auf den Eingang „pxl_data1 ... pxl_data5“ des Modul „debay“ gelegt für die Auswertung des Bayer Pattern.

Wenn ein Block komplett erfasst und bearbeitet wurde, wird der Block um eine Pixelspalte im Bild verschoben und der nächste Block wird gepuffert. Ist das Ende der aktuellen Spalte erreicht, dann wird der Buffer mit dem nächsten Block, am Spaltenbeginn um eine Zeile versetzt, befüllt. Siehe dafür Abbildung SDRAM_Pixelbuffer 3.

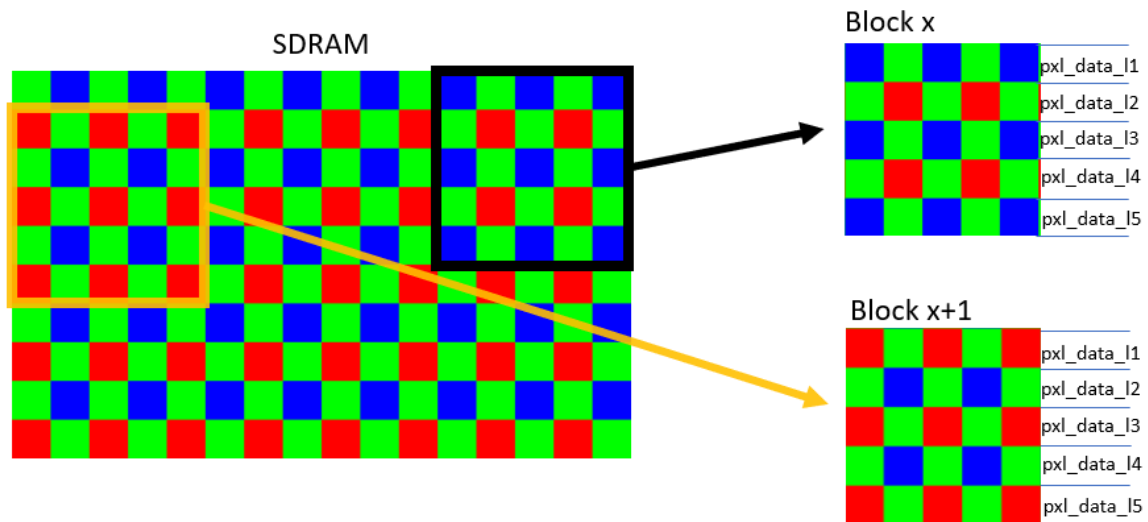


Abbildung SDRAM_Pixelbuffer 3: Schaubild Pixelbuffer Zeilenende

Aufbau Intern: SDRAM_Pixelbuffer(Vereinfachte Version)

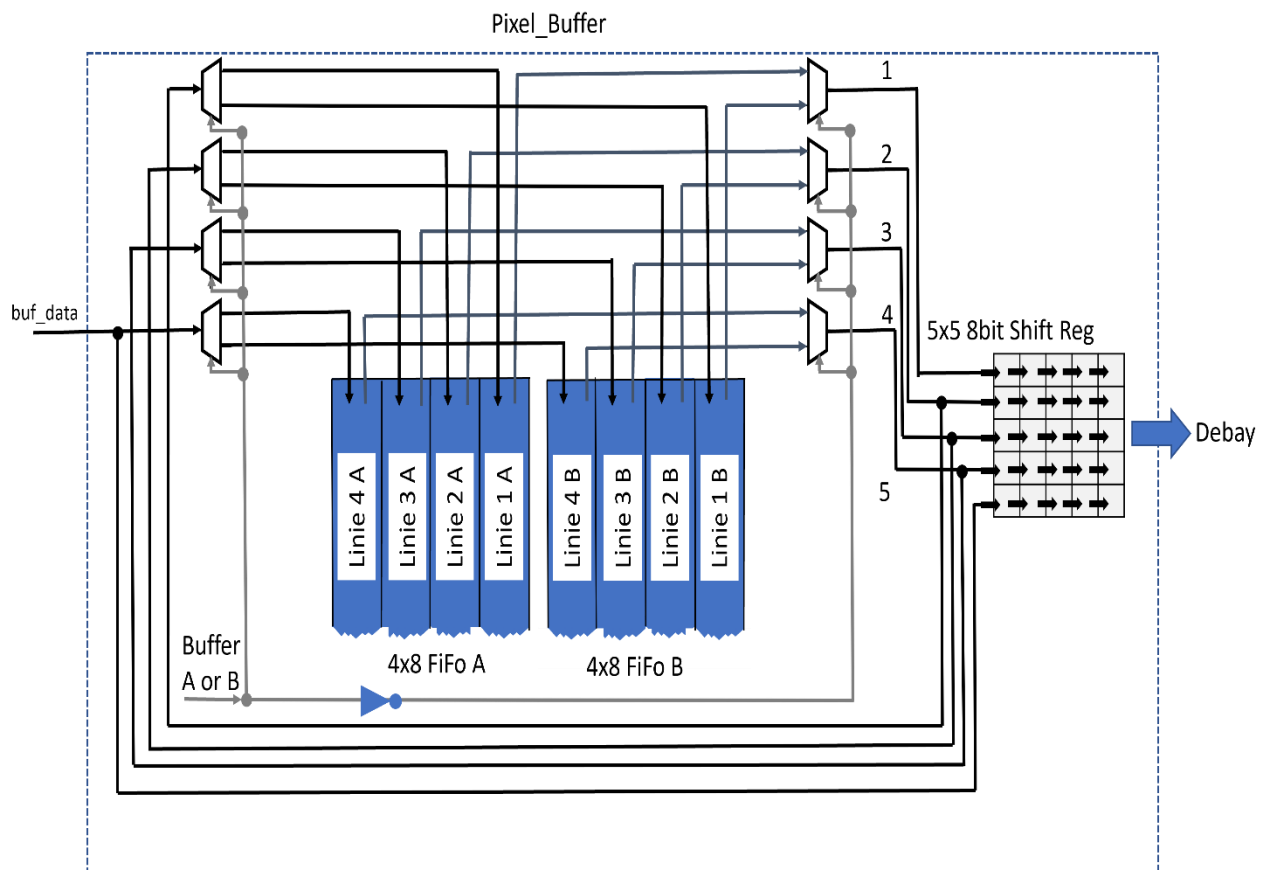


Abbildung SDRAM_Pixelbuffer 4: Schaubild interner Aufbau SDRAM_Pixelbuffer

Modul: Debay

Die Pixeldaten (5x5 Byte Block) des Modul „sdram_pixelbuffer“ liegen an den Eingängen „pxl_data_l1[39 .. 0]“ bis „pxl_data_l5[39 .. 0]“ an. Jeder Eingang steht für eine Zeile im Block.

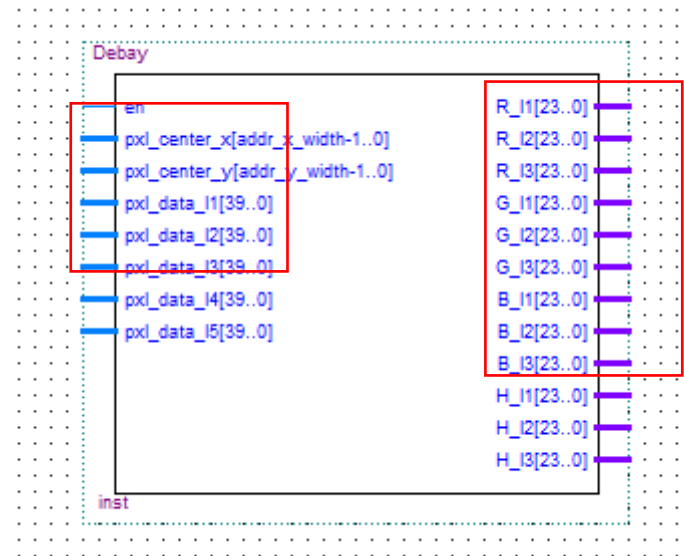


Abbildung Debay 1: Blockschaltbild

Der 5x5 Byte Block ist nach dem Bayer Muster formatiert. Das Bayer Muster reduziert die Anzahl der Farbkanäle pro Pixel auf einen einzigen Farbkanal. Für die Ausgabe des Bildes über eine VGA-Schnittstelle sollen drei Kanäle pro Pixel zur Verfügung stehen. Im Modul „Debay“ wird aus dem 5x5 Byte Block für jeden der drei Farbkanäle (R, G, B), ein 3x3 Byte Block gewonnen. Um jeweils 3x3 Bytes pro Kanal zu bekommen, wird der 5x5 Byte Block wie in Abbildung Debay 2 aufgeteilt.

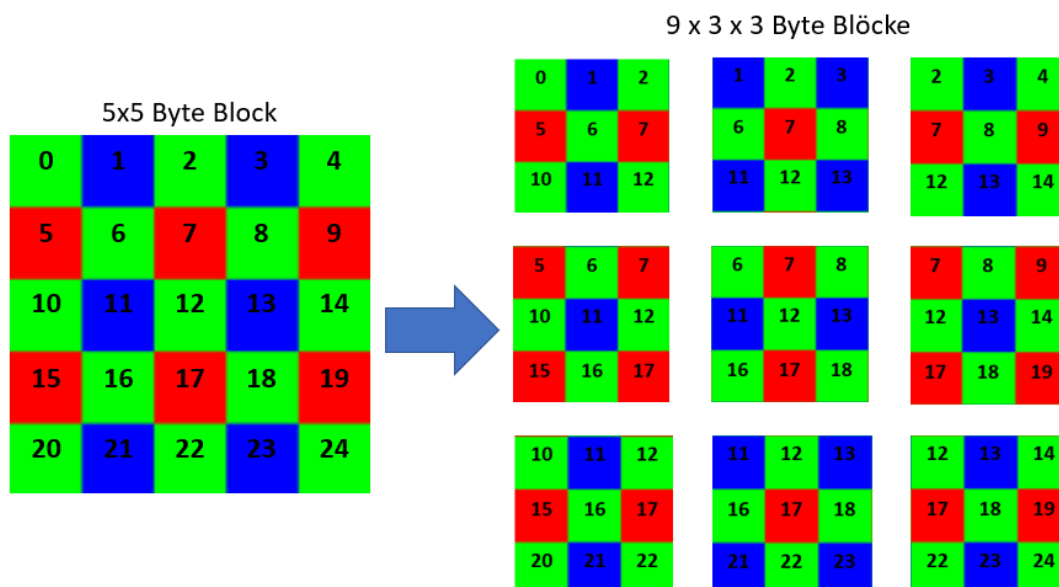


Abbildung Debay 2: Aufteilung 5x5 Byte Block in kleinere 3x3 Blöcke

Durch die Aufteilung erhalten wir neun 3x3 Byte Blöcke. Aus jedem Block wird für alle drei Farbkkanäle ein Byte extrahiert. Wie in Abbildung Debay 2 zu sehen, beinhaltet ein 3x3 Block mehr als nur eine Farbinformation für einen Kanal, so ist beispielsweise im ersten Block an Stelle 5 und 7 die Farbinformation für rot doppelt. Um aus zwei Byte ein zu bekommen, nehmen wir das arithmetische Mittel der zwei Farbinformationen aus diesem Block. Zum besseren Verständnis dieser Vorgehensweise siehe Abbildung Debay 3.

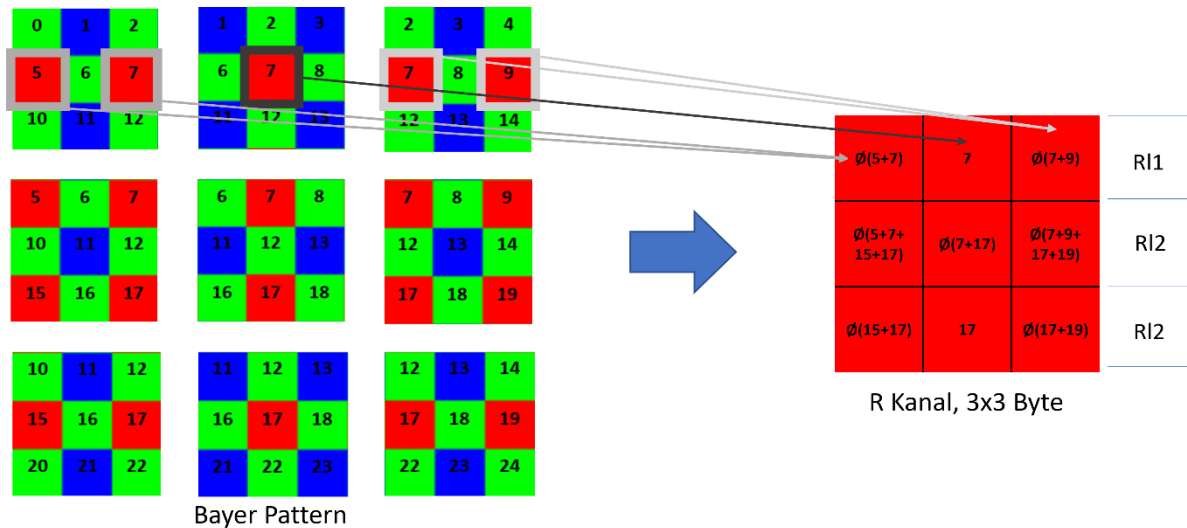


Abbildung Debay 3: Zusammensetzung 3x3, Beispiel am Rotkanal, R1 = Output der ersten Zeile des Rotkanals

Nach dem Auswerten der kleineren Blöcke, werden die Informationen der 3x3 Blöcke pro Farbkkanal, auf den, in Abbildung Debay 1 rot markierten, Ausgängen gelegt.

Für die reine Ausgabe der Pixel über eine VGA Schnittstelle, würde ein Overhead bei der Anzahl der gelesenen Pixel entstehen, da 3x3 große Blöcke als Eingabe für das Decodieren des Bayer Pattern reichen würde. Jedoch ist die Größe von 5x5 als Eingabe gewünscht, da die zusätzlichen Pixel ein Maß an extra Information über Farbänderung und Farbbewegung liefert.

Übersicht Zustand (nach Studienarbeit Oster)

Eine Übersicht über das Projekt bietet folgende Grafik und beschreibt den Zustand nach Beendigung dieser Ausarbeitung.

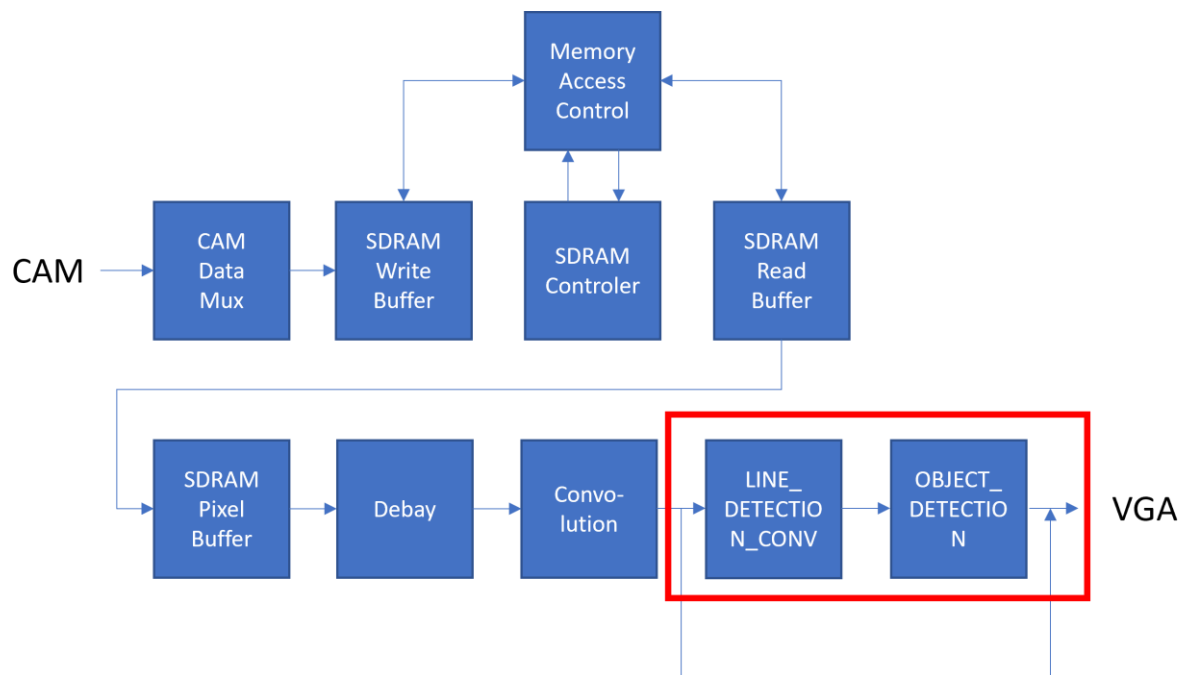


Abbildung Zustand Jetzt 1: Vereinfachte Übersicht Module im Projekt

In dem neusten Stand haben sich zwei neue Module in das Gesamtbild eingefügt die in den folgenden Kapiteln beschrieben werden.