

Disposition

Marc de Bever

June 30, 2020

Contents

1	Einleitung	2
2	System Varian	3
2.1	Bootvorgang Imager	4
2.2	Kommunikation Sensor - XI Computer	4
3	Unser System	4
4	Hardware	4
5	Software	4
5.1	ARM	5
5.2	FPGA	5
5.3	PC Applikation	5
6	Konfiguration	5
7	Entwicklungsumgebung	5
8	Theorie	5
8.1	Enclustra Mercury PE5 Modul	5
8.2	Xilinx SoC	5
8.3	Bootvorgang	5
8.4	Standart Konfiguration der CPU	5
8.5	AXI	5
8.6	ARM R5	6
8.7	USB Standart	6
8.8	USB Treiber	6
8.9	SerDes	6
8.10	MGT	6
8.11	Debugging	6
9	Erläuterungen	6

1 Einleitung

Dieses Dokument ist ein Teil der Dokumentation des Projektes Imager-Emulator. Dieses Projekt besteht aus drei Teilprojekten. Genauer gesagt besteht es aus zwei Bachelor Thesen und einem Projekt 5 (P5). Das P5 und eine Thesis laufen parallel und die zweite Thesis wird ein Semester später durchgeführt. Diese Dokumentation ist diejenige von der ersten Thesis, welche parallel mit dem Projekt 5 durchgeführt wird. Das Projekt 5 und die zweite Thesis wird von Fabio Nardo geschrieben und diese Dokumentation ist von mir, Marc de Bever, geschrieben.

Das Projekt Imager-Emulator wurde von der Firma Varian Medical Systems ausgeschrieben. Das Ziel ist es, einen Sensor zu emulieren. Der Sensor gibt Bilder über eine nicht standardisierte LWL Schnittstelle an einen Computer weiter. Der Computer, welcher die Bilder des Sensors entgegen nimmt, rechnet diverse Algorithmen, die unter anderem Pixelfehler erkennen und korrigieren. Jedoch kann dem Sensor nicht gesagt werden, *mach mal 'nen Pixelfehler*. Daher braucht es ein Gerät, um Bilder mit Pixelfehlern zu generieren und an den Computer zu senden. Und dieses Gerät soll in diesem Projekt entwickelt werden.

Da dieses Projekt eine spezielle Konstellation hat, ist die Aufteilung der Teilprojekte wie folgt. Im P5 soll die Hardware entwickelt werden. In der ersten Thesis, also dieser Arbeit, sollen die Entwicklungsumgebungen und das FPGA-Modul in Betrieb genommen werden. Sowie sollen auch die Schnittstellen programmiert werden. Die zweite Thesis soll schlussendlich die Kommunikation mit dem Computer implementieren, die Pixelfehler in die Bilder einbauen und das ganze Projekt abschliessen.

Dieser Bericht dient zum einen dazu, dass im weiterführenden Projekt verstanden werden kann, wie die Entwicklungsumgebung und Schnittstellen funktioniert, deren Möglichkeiten und Grenzen. Zum anderen dient er um zu dieser Arbeit als Thesis zu dokumentieren. Der Bericht ist folgendermassen aufgebaut.

Das zweite Kapitel dieses Berichtes beschreibt wie das Sensorsystem der Varian aufgebaut ist. Zuerst gibt es einen groben Überblick und danach geht es genauer auf die Teilbereiche ein. Das dritte Kapitel beschreibt wie unser System aussehen soll. Es gibt einen top-down Blick auf das Projekt. Das heisst es beinhaltet die Blockschaltbilder. Zusätzlich beschreibt es die Wahl der wichtigsten Komponenten, wie das FPGA-Modul.

Das vierte Kapitel beschreibt die Hardware, da dies die Arbeit des Projektes 5 ist, werden in diesem Kapitel nur die wichtigsten Punkte erwähnt, welche für das Programmieren relevant sind.

Das fünfte Kapitel beschreibt die Software. Diese besteht aus drei Teilen, dem Code für den Mikrocontroller, dem Code für den FPGA und dem Code für die PC-Applikation. Das sechste Kapitel beschreibt wie zusätzliche Konfigurationen nachträglich hinzugefügt werden können. Das siebte Kapitel beschreibt die Entwicklungsumgebung.

Und das letzte Kapitel beschreibt Funktionalitäten, welche nicht von uns entwickelt worden sind, sondern schon auf dem Modul vorhanden sind.

Table 1: Eckdaten der verschiedenen Sensorversionen der Varian

Imager	Auflösung	Pattern¹	FPGA	Schnittstelle
DMI ²	1280x1280	b10'1010'1010'1010'1010 XI: 0xAAA	Spartan 3 XC3S200	externer SerDes
RTI 1.0 ² (RTI4343L)	1280x1280	b10'1010'1010'1010'1010 XI: 0xAAA	Spartan 3 XC3S200	externer SerDes
RTI 2.0 (RTI4343iL)	3072x3072	b10'1010'1001'0101'0101 XI: 0x955	Artix 7 XC7A100T	aktuell: externer SerDes zukünftig: MGT(GTP)
RTIXL 1.0 (RTI8643L)	6144x3072	b10'1010'1001'1001'0101 XI: 0x995	Artix 7 XC7A200T	externer SerDes und MGT(GTP)

¹Das XI wertet nur die letzten 12 bit des Pattern aus. Um das DC-Balancing zu erreichen, müssen trotzdem alle 18 bits korrekt gesendet werden.

²DMI und RTI 1.0 besitzen exakt die gleiche Elektronik.

2 System Varian

Dieses Kapitel beschreibt, wie das System der Varian aufgebaut ist. Zuerst gibt es einen groben Überblick und danach wird in den Unterkapiteln genauer auf die einzelnen Komponenten eingegangen. Die Unterkapitel sind noch nicht vollständig aufgelistet

Der Sensor ist Teil eines medizinischen Röntgengeräts. Genauer gesagt ist er ein Bildsensor für Röntgenstrahlung. Der Sensor ist über eine LWL Schnittstelle an einen Computer angeschlossen. Diesen Computer nennen wir XI-Computer. Der Sensor besteht grob gesagt aus dem Bildsensor, welcher die Röntgenstrahlung in ein elektrisches Signal umwandelt und einem FPGA, welcher die Bild daten des Bildsensors entgegen nimmt und diese, auf Befehl, an den XI-Computer weiterleitet. Der XI-Computer steuert den Sensor. Beim einschalten Konfiguriert er den FPGA auf dem Sensor und danach gibt er Befehle an den Sensor. Es gibt verschiedene Versionen des Sensors. Die Unterschiede liegen darin, das die Bildübertragung der älteren Versionen über einen dem FPGA externen Serialisierer/Deserialisierer(SerDes) läuft und die der neueren die internen multi-gigabit Transceiver(MGT) des FPGA benützen. Der Vorteil der MGTs ist, das sie schneller sind. Jedoch geschieht die Konfiguration des FPGAs immer noch durch den externen SerDes, da der FPGA zuerst konfiguriert sein muss, damit die MGTs laufen. Damit der XI-Computer weiss, welche Sensorversion angehängt ist, liegt mittels pull-up und pull-down Widerständen ein Code am SerDes. Anhand von diesem Code weiss der XI-Computer, welche Firmware er auf den FPGA laden muss und welche Befehle er unterstützt. Die Tabelle 1 zeigt die verschiedenen Sensorversionen und deren Eckdaten.

wieso XI?

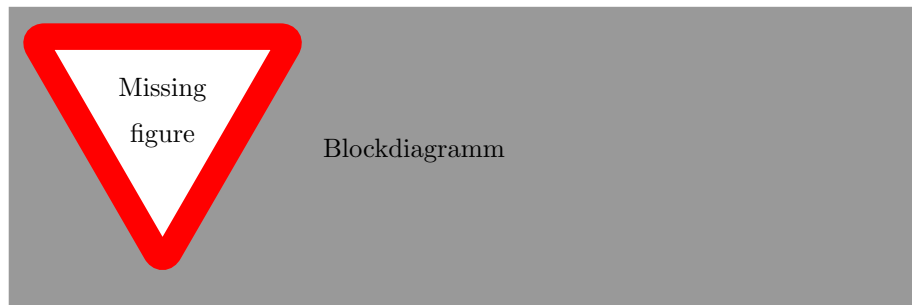


Figure 1: Blockdiagramm

2.1 Bootvorgang Imager

2.2 Kommunikation Sensor - XI Computer

Dieses Kapitel beschreibt, wie die Sensoren mit dem XI-Computer kommunizieren. Dies beinhaltet die physikalische Ebene mit dem SerDes, den MGTs und dem SFP; den Gleichspannungsfreien Leitungscode, einer abgeänderten 8b10b-Code; dem Bootprotokoll; und dem Protokoll um Daten zu übertragen.

3 Unser System

Dieses Kapitel soll einen Top-Down Blick auf unser System geben. Dies beinhaltet die Blockschaltbilder und die wichtigsten Entscheidungen der Komponenten, wie des FPGA. Zudem definiert es die Schnittstellen zwischen der PC Software und dem ARM Core, sowie die zwischen dem ARM Core und dem FPGA

4 Hardware

Dieses Kapitel beschreibt die Hardware des Projektes. Es gibt einen kurzen Überblick und verweist auf die Dokumentation von Fabio und die anderen Dokumente für genauere Angaben. Es definiert die Angaben, welche für das Programmieren des SoCs wichtig sind. Und beschreibt die Inbetriebnahme des SoC Teils der Hardware.

5 Software

Dieses Kapitel ist der Hauptteil der Arbeit. Es beschreibt den FPGA und Mikrocontroller Code, welcher auf dem SoC läuft und wie dieser getestet wurde. Es ist in die Unterkapitel ARM, FPGA und PC Software aufgeteilt.

5.1 ARM

Dieses Kapitel beschreibt, welche Komponenten des ARM Cores gebraucht werden und wie sie konfiguriert sind, wie die Module des ARM Cores aufgebaut sind und auf welchem Core sie laufen.

5.2 FPGA

Dieses Kapitel beschreibt die Module, welche auf dem FPGA laufen.

5.3 PC Applikation

Dieses Kapitel beschreibt die Software, welche auf dem PC läuft, welche den Imager konfiguriert.

6 Konfiguration

Dieses Kapitel beschreibt, wie zusätzliche Konfigurationen hinzugefügt werden können.

7 Entwicklungsumgebung

Dieses Kapitel beschreibt, wie der Code für den SoC entwickelt werden kann und worden ist. Dies beinhaltet Vitis, Vivado und wie das Programm auf den SoC geladen werden kann.

8 Theorie

Dieses Kapitel beschreibt Funktionalitäten, welche nicht von mir entwickelt worden sind, sondern schon auf dem Modul vorhanden sind. In den einzelnen Unterkapiteln listet es die wichtigsten Konzepte und referenziert auf die Dokumentationen um genauere Informationen zu erhalten. Diese Unterkapitel sollen dazu dienen, dass in den anderen Kapiteln auf diese referenziert werden kann.

8.1 Enclustra Mercury PE5 Modul

8.2 Xilinx SoC

8.3 Bootvorgang

8.4 Standart Konfiguration der CPU

8.5 AXI

Eventuell werden noch folgende weitere Spezifikationen und Komponenten beschrieben.

8.6 ARM R5

8.7 USB Standart

8.8 USB Treiber

8.9 SerDes

8.10 MGT

8.11 Debugging

9 Erläuterungen

Dieses Kapitel listet alle Abkürzungen mit den Bedeutungen und einer kurzen Erklärung auf. Zudem erklärt es, wie die verschiedenen Fachwörter zu verstehen sind.

ARM Core, Core, FPGA, Sensor, XI-Computer, SoC, Imager,
Imager-Emulator
P5 LWL FPGA SerDes Imager