FPGA system for

RF Camera transmitter-receiver

System

Written by: וUdi Edny

# Contents

## Table of Contents

1

1 Contents 2

**1.1 Table of Contents 2**

**1.2 List of Tables 2**

**1.3 List of Figures 2**

**1.4 Public Version History 3**

2 General 4

**2.1 Terminology 4**

**2.1.1 Terms 4**

**2.2 Conventions 4**

**2.3 Bibliography 5**

**2.3.1 Standards 5**

**2.3.2 External Documents 5**

**2.3.3 Internal Documents 5**

**2.4 Architecture assumptions/restrictions 5**

3 Top level overview 6

## List of Tables

Table ‎5‑1: VCU118 FPGA Interface signals 15

Table ‎5‑2: ZC706 FPGA Interface signals 16

Table ‎5‑3: FPGA registers 17

Table ‎5‑4: OCP-DRA mapping 19

Table ‎5‑6: Radar core Integration table 20

Table ‎7‑1 dynamic configurations 30

## List of Figures

Figure ‎3‑1: Top level diagram 7

Figure ‎5‑1: Alpha+Beta block diagram 10

Figure ‎5‑2: System transmit path 11

Figure ‎5‑3: System receive path 12

Figure ‎5‑4: FPGA clock scheme 13

Figure ‎5‑5: FPGA reset scheme 14

Figure ‎5‑6: AXI->APB->OCP bridge 19

Figure ‎5‑7: ADC data Mux logic 20

Figure ‎6‑1: Gamma block diagram 27

## Public Version History

|  |  |  |  |
| --- | --- | --- | --- |
| version No. | Issue Date | Prepared By | Details of Change |
| **0.1** | 25/3/2018 | Elik Haran | First release |
| **0.2** | 25/3/2018 | Elik Haran | Post PDR comments |
| **0.3** | 3/4/2018 | Elik Haran | Added integration guidelines for radar core |

# General

This document discusses the functionality and architecture details of the FPGA platform which is used for Arbe Robotics ASIC’s Range block evaluation board.

## Terminology

### Terms

| Term | Definition |
| --- | --- |
| b | bit |
| B | Byte = 8 bits |
| TBD | To Be Defined |
| I/F | interface |
| I/O | Inputs/Outputs |
| N/A | Not applicable |
| MOD | Modulo |
| RD | Read |
| WR | Write |
| uP / uC | The external software entity controlling the block over the control path |
| ADC or A/D | Analog to Digital converter |
| DAC or D/A | Digital to analog converter |
| EAD | External ADC device |
| IAD | Internal ADC device |
| FSM / SM | Finite state machine |
| WD | Watchdog |
| SOC / SoC | uP & sub-system |
| Tx | Transmit |
| Rx | Receive |

## Conventions

1. Bits in a bus / register are numbered from n to 0, where bit n is the leftmost, most significant bit, while bit 0 is the rightmost, least significant and last transmitted bit.
2. Note that wherever pseudo‑code appears in the text it is only in order to detail the tasks performed by the sub‑block in a semiformal manner. The names of wires and registers in the various pseudo‑code snippets will obviously not be the names used in the code itself.
3. Hexadecimal values are expressed using the ‘0x’ prefix.

## Bibliography

### Standards

* AMBA APB:

<http://web.eecs.umich.edu/~prabal/teaching/eecs373-f12/readings/ARM_AMBA3_APB.pdf>

### External Documents

* VCU118 user guide:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**Repository**](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp) | / | [Everest](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest) | / | [CARMEL](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL) | / | [EVEREST](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST) | / | [FPGA](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FFPGA) | / | [Datasheets](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FFPGA%2FDatasheets) | / | [ug1224-vcu118-eval-bd.pdf](http://185.220.207.15:8082/polarion/svnwebclient/fileContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FFPGA%2FDatasheets%2Fug1224-vcu118-eval-bd.pdf) |

* VCU118 board schematics:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | [**Repository**](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp) | / | [Everest](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest) | / | [CARMEL](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL) | / | [EVEREST](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST) | / | [HW](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FHW) | / | [Schematics](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FHW%2FSchematics) | / | [0381739\_HW-U1-VCU118\_REV1\_1\_SCHEMATIC\_20161201\_133938.pdf](http://185.220.207.15:8082/polarion/svnwebclient/fileContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FHW%2FSchematics%2F0381739_HW-U1-VCU118_REV1_1_SCHEMATIC_20161201_133938.pdf) |  | |

* JESD204 Xilinx core product guide:

<https://www.xilinx.com/support/documentation/ip_documentation/jesd204/v7_2/pg066-jesd204.pdf>

* ZC706 board evaluation board:

<https://www.xilinx.com/support/documentation/boards_and_kits/zc706/ug954-zc706-eval-board-xc7z045-ap-soc.pdf>

* ZC706 board schematics:

<https://www.xilinx.com/support/documentation/boards_and_kits/zynq-7000/zc706-schematic-xtp215-rev1-1.pdf>

* TI ADC

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**Repository**](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp) | / | [Everest](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest) | / | [CARMEL](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL) | / | [EVEREST](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST) | / | [HW](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FHW) | / | [External ADC](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FHW%2FExternal%20ADC) | / | [TI ads52j90.pdf](http://185.220.207.15:8082/polarion/svnwebclient/fileContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FHW%2FExternal%20ADC%2FTI%20ads52j90.pdf) |

* Clock distributor

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**Repository**](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp) | / | [Everest](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest) | / | [CARMEL](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL) | / | [EVEREST](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST) | / | [HW](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FHW) | / | [Clock distributor](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FHW%2FClock%20distributor) | / | [LMK04832\_SNAS688, 2018-01-13 (2).pdf](http://185.220.207.15:8082/polarion/svnwebclient/fileContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FHW%2FClock%20distributor%2FLMK04832_SNAS688%2C%202018-01-13%20%282%29.pdf) |

* OCP standard

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**Repository**](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp) | / | [Everest](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest) | / | [CARMEL](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL) | / | [EVEREST](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST) | / | [ASIC](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FASIC) | / | [ARCHITECTURE](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FASIC%2FARCHITECTURE) | / | [standards and general documents](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FASIC%2FARCHITECTURE%2Fstandards%20and%20general%20documents) | / | [OCP](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FASIC%2FARCHITECTURE%2Fstandards%20and%20general%20documents%2FOCP) |

### Internal Documents

* Everest architecture specification

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**Repository**](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp) | / | [Everest](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest) | / | [CARMEL](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL) | / | [EVEREST](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST) | / | [ASIC](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FASIC) | / | [ARCHITECTURE](http://185.220.207.15:8082/polarion/svnwebclient/directoryContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FASIC%2FARCHITECTURE) | / | [Everest\_system\_architecture.docx](http://185.220.207.15:8082/polarion/svnwebclient/fileContent.jsp?url=Everest%2FCARMEL%2FEVEREST%2FASIC%2FARCHITECTURE%2FEverest_system_architecture.docx) |

## Architecture assumptions/restrictions

* ADC drives the samples at **unsigned** format.

# Top level overview

This system is designed to transmit a video image over 4 RF channels. The system has two modules a transmit module (Tx) and a receive module (Rx). Each has an FPGA to capture/display the video image and 4 CC1200 chips to transmit/receive the data between the Tx & Rx. See Figure ‎3‑1.

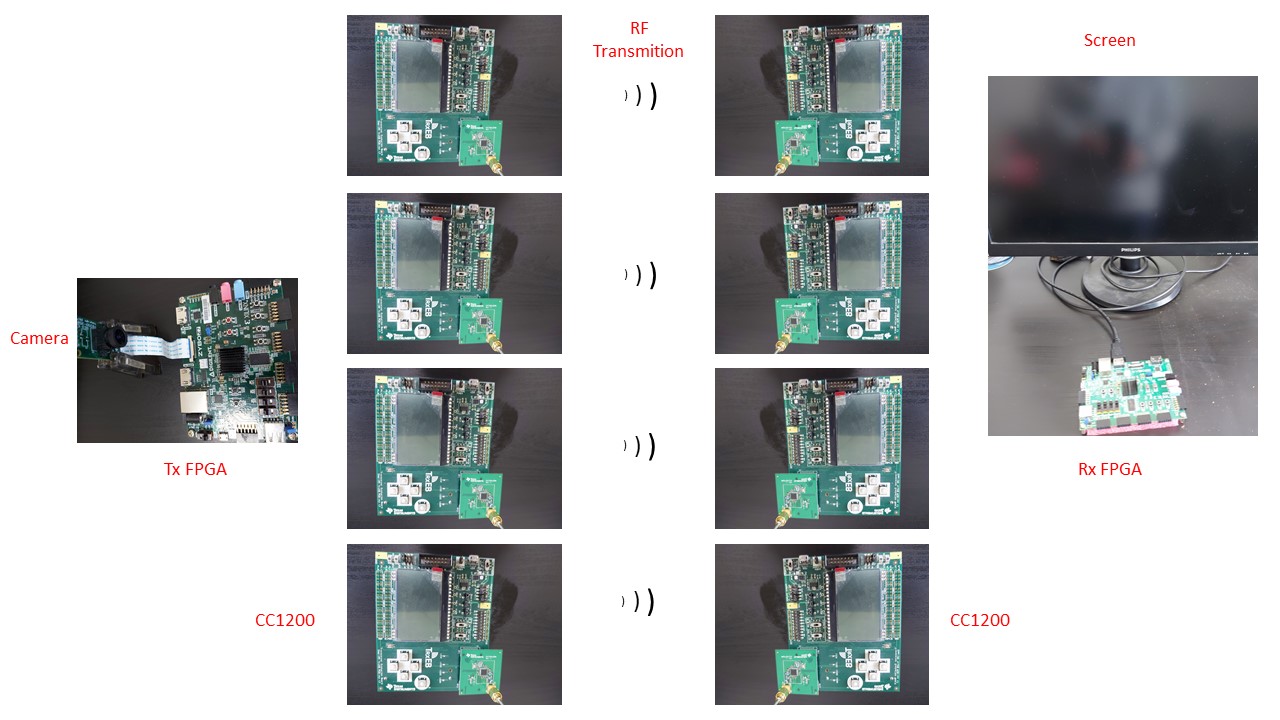


Figure ‎3‑1: RF system Top level

## The camera sensor

The camera captures the video image for the system.

## Tx FPGA

The Tx FPGA controls the Tx part of the system. On boot it is responsible for

The Camera & 4 CC1200 configurations. After boot the data from the camera is stored in 4 memories each hold one slant of the picture. After one frame is stored in the memory, the FPGA start transferring the thought the CC1200 chip. The data is divided into packets of 120 bytes. Each packet is transferred to the CC1200 and the FPGA waits for the CC1200 finishes this packet transmission. Only then another packet can be sent.

## CC1200

The CC1200 is the RF chip. In the Tx side it is configured for transmitting the data stored in the FPGA memory. In the Rx side the CC1200 is configured to receive mode and wait for a packet detection. Once the packet was detected the CC1200 indicate the FPGA that a packet was received and is waiting in the CC1200 FIFO. Then the FPGA start reading the data.

## Rx FPGA

The Rx FPGA controls the Rx part of the system. The Rx FPGA start controlling the screen after reset. The screen control module reads the FPGA memories and send the data to the screen in 60 frames for second rate. Whatever data is stored in the memories it is displayed on the screen. This insures, that the picture is stable, no matter what the RF link is. In the boot section the FPGA configure the CC1200 to receive mode. After boot the FPGA waits for the packets to arrive from the Tx. Once an indication that a packet was received in the CC1200, the FPGA reads the data and store it in the FPGA memories. As mentioned before the screen control module reads and display the data from the memories to the screen all the time.

# Detailed description

## Tx FPGA

The Tx FPGA operate the Tx system. It has Zynq CPU with a SCCB & CC1200 control units. It has the image data path and the memory to hold the data and transfer it to the CC1200 for transmission.

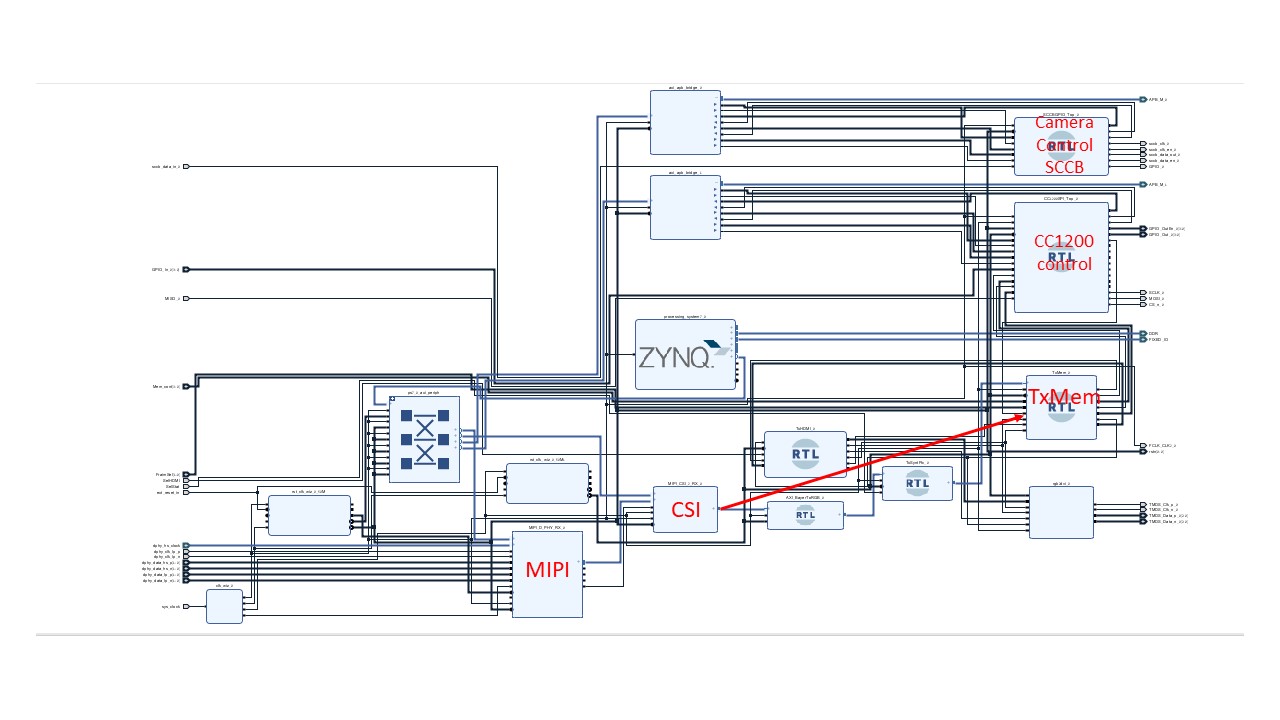


Figure 4‑1: Tx FPGA

### Tx FPGA Zynq CPU

In the Tx FPGA there is a Zynq CPU. This CPU is a Xilinx module based on the ARM cortex 9 CPU. The CPU is used to configure the Camera & the CC1200 chips through the Camera & CC1200 control modules. The last configuration for the CC1200 is to transfer the CC1200 control to the TxMem block, and from this point on the system works independently (without CPU control).

### Tx image data path

The camera is connected to the FPGA using MIPI protocol. The MIPI protocol drives the data on high-speed lines, which the MIPI block transfer to parallel data inside the FPGA. After the data was received in the FPGA the CSI block transfer it to an FPGA video bus which generate the start frame and end of line signals. This video bus is then transferred to the memory in the TxMem module.

### TxMem

The TxMem module is the main part in the image data path. It receives the data from the CSI module and save it in 4 memories each memory saves one slant from the picture.

After one frame is saved in the memories, the TxMem module stop receiving any more data from the camera and start transmitting the data to the CC1200 chip. The CC1200 can sends 120 bytes in a single packet so the TxMem sends a 120 bytes packet and wait for the CC1200 chip to indicate that the packet was sent and anther packet can be sent. This process is done in all memories & CC1200 chips in parallel. There for the time to sent a full picture (4 slants) is the time it takes to send one slant. After the 4 memories were transferred the TxMem module can now receive the next frame.

### Camera and CC1200 control blocks.

The camera receives commands through a SCCB protocol. The SCCB protocol is a protocol similar the I2C protocol. The Camera control block transfer the Zynq AXI (APB) bus commands to a SCCB commands for the camera.

The CC1200 control block is the same block in the Tx FPGA and in the Rx FPGA. This block has control IOs for the Rx which in the Tx are left unconnected and vice versa. This block gets the first commands from the Zynq processor (like the SCCB block). But in the last command the Zynq transfer the control to the TxMem block (RxMem block in the Rx FPGA) and the system keeps working without CPU control.

### Screen control

In the prototype system there is a TxHDMI & rbg2dvi modules to enable screen connection in the Tx module. In the final system those modules will be left unconnected.

## Rx FPGA

The Rx FPGA transfer the data received by the CC1200 receiver and display it through the HDMI to the screen.

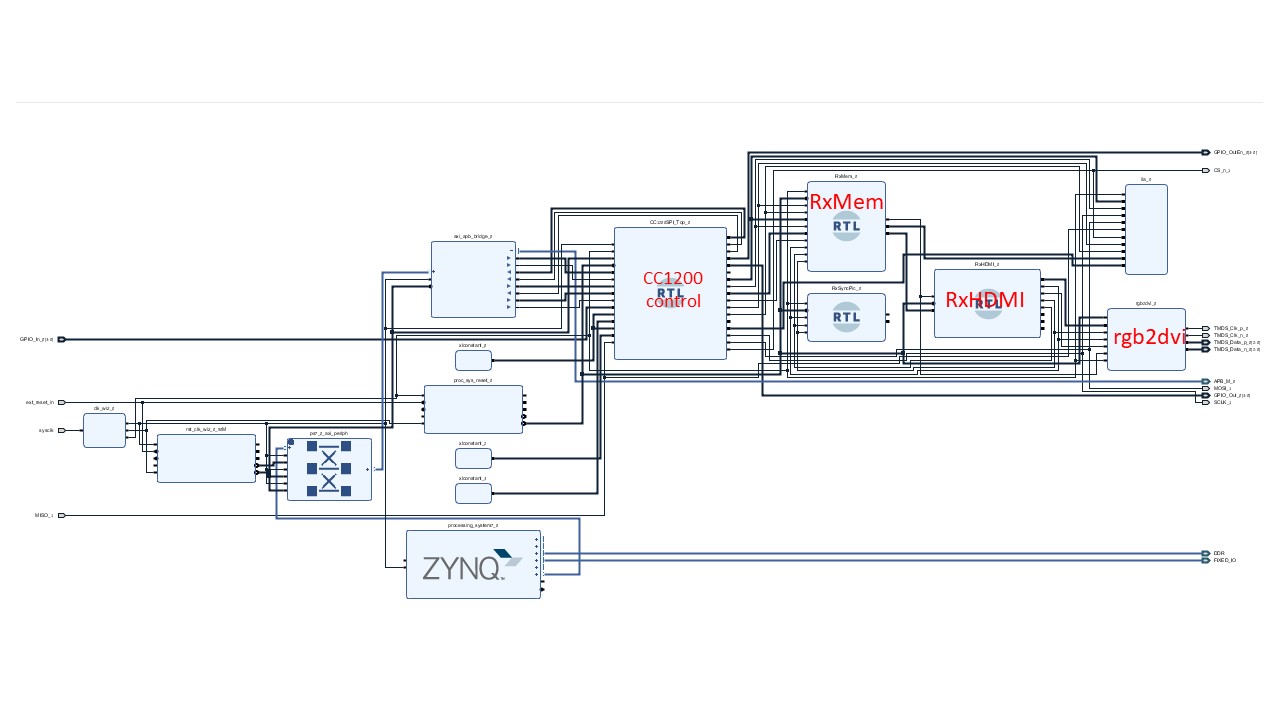


Figure 4‑2: Rx FPGA

### Rx FPGA Zynq CPU

This is the same module as the Zynq CPU in the Tx module. Only here there is no need for the camera control and the only thing the Zynq does is to configure the CC1200 and transfer its control to the RxMem.

### CC1200 control

In this module the CC1200 control block is the start of the data path. After configuration this block get the indication from the CC1200 chip that there is a packet to read. It reads it and transfer it to the RxMem module.

### RxMem

In this module the RxMem is mostly a storage for the data from the recever for the HDMI display to read and display on the screen.

### RxHDMI

This module controls the screen signal it generates the reads from the RxMem memories and send it to the rgb2dvi for the screen to display.

### Rgb2dvi

This is an interface to the HDMI signal. It also transfers the parallel data inside the FPGA to fast serial data to be send to the screen.