sensors

MDPI

Article

Enhanced Channel Calibration for the Image Sensor of the TuMag Instrument

Eduardo Magdaleno [1,*], Manuel Rodríguez Valido [1], David Hernández [2,3], María Balaguer [4], Basilio Ruiz Cobo [2,3], David Orozco Suárez [4], Daniel Álvarez García [4] and Argelio Mauro González [1]

[1] Department of Industrial Engineering, University of La Laguna, 38200 San Cristóbal de La Laguna, Spain; mrvalido@ull.edu.es (M.R.V.); alu0100070378@ull.edu.es (A.M.G.)
[2] Institute of Astrophysics of Canary Islands, 38205 San Cristóbal de La Laguna, Spain; dhdez@iac.es (D.H.); brc@iac.es (B.R.C.)
[3] Department of Astrophysics, University of La Laguna, 38200 San Cristóbal de La Laguna, Spain
[4] Institute of Astrophysics of Andalusia, 18008 Granada, Spain; balaguer@iaa.es (M.B.); orozco@iaa.es (D.O.S.); dalvarez@iaa.es (D.Á.G.)
* Correspondence: emagcas@ull.edu.es

Abstract: The Sunrise missions consist of observing the magnetic field of the sun continuously for a few days from the stratosphere. In these missions, a balloon supporting a telescope and associated instrumentation, including a Tunable Magnetograph (TuMag), is lifted into the stratosphere. In the camera of this instrument, the image sensor sends its data to a Field Programmable Gate Array (FPGA) using eight transmission channels. These channels must be previously calibrated for a correct delivery of the image. For this mission, the FPGA has been exchanged for a newer and larger one, so the firmware has been adapted to the new device. In addition, the calibration algorithm has been parallelized as the main innovation of this work, taking advantage of the increase in logic resources of the new FPGA, in order to minimize the calibration time of the channels. The algorithm has been implemented specifically for this instrument without using the Input Serial Deserializer (ISERDES) Intellectual Property (IP), since this IP does not support the deserialization of the data sent by the image sensor to the FPGA.

Keywords: Sunrise mission; sensor calibration; FPGA; TuMag instrument; Low-Voltage Differential Signaling (LVDS) deserialization

check for updates

Citation: Magdaleno, E.; Rodríguez Valido, M.; Hernández, D.; Balaguer, M.; Ruiz Cobo, B.; Orozco Suárez, D.; Álvarez García, D.; González, A.M. Enhanced Channel Calibration for the Image Sensor of the TuMag Instrument. *Sensors* **2022**, *22*, 2078. https://doi.org/10.3390/s22062078

Academic Editors: Angel Diéguez, José Luis Lázaro-Galilea and Gonzalo Pajares Martinsanz

Received: 13 January 2022
Accepted: 4 March 2022
Published: 8 March 2022

1. Introduction

The Sunrise-3 mission pursues the uninterrupted observation of the sun's magnetic field for a few days. With this objective, a telescope with its associated instrumentation will be carried to the stratosphere with the help of a balloon [1]. The National Program approved the Sunrise/IMaX project in 2002 as it was a strategic step towards the Polarimetric and Helioseismic Imager (PHI) for Solar Orbiter mission [2]. Sunrise's first mission flew in 2009 and studied solar magnetism at minimal solar activity using the IMaX (Imaging Magnetograph eXperiment) instrument [3–5].

The second Sunrise mission took place in 2013. That year, there was a period of maximum solar activity, unlike in the first mission, in which there was a period of minimum solar activity [6]. The design of this mission did not have many changes: the telescope and the gondola were the same as in the first mission. The IMaX instrument had only a few minor changes and updates including the replacement of the FPGA (Field Programmable Gate Array). Only the filter system of one of the instruments was modified [7]. The scientific results of the Sunrise missions can be consulted at [6–11].

Now, Sunrise-3 is meant to observe the evolution of the magnetically coupled solar atmosphere at high spatial resolution rather than being a transient thought for a given phase of solar activity. A new, and enhanced version of the IMaX instrument has been designed, named TuMag (Tunable Magnetograph). The major conceptual change in the optics has been realized due to the tunable capability of the instrument filter. TuMag is capable of imaging the sun in several narrow bands. From these images, the information of the four polarization states can be extracted [12–14].

The design of the electronics is completely new in the TuMag instrument compared to the old IMaX [15]. The electronics update includes a miniaturization of the components, leaving more physical space for other instruments on board. In addition, electronic devices have more logical resources, allowing more complex tasks to be carried out [16–20].

The camera image sensor has been replaced compared to the previous instrument. The new image sensor is the GPIXEL GSENSE400-BSI [21]. The data acquired by the sensor are read using eight differential channels that supply 12-bit pixels at 300 Mbps. The management of the acquisition and reading and sending times of data are done through control signals that ships a 7-series Xilinx FPGA (Artix-7 device) [22]. All eight sensor channels must be pre-calibrated to obtain a correct data reading.

A Xilinx Spartan-6 FPGA is used in [23] to implement the image sensor driver including channel calibration [24]. The calibration is performed sequentially for each channel to save FPGA logic resources. In the TuMag camera, the control device uses a newer FPGA that supports the CoaxPress standard communication protocol for vision applications based on coaxial cables. Specifically, an Artix-7 FPGA containing a large number of logic resources has been used [25]. This allows channel calibration to be performed simultaneously, so the time taken to perform this task is greatly reduced. On the other hand, a component that introduces delays in the input signals must be used to perform the calibration. This component is substantially different from 6-series to 7-series Xilinx FPGA [26,27]. This implies that the calibration algorithm has been changed to adapt it to the new component with respect to the calibration method used so far in the literature [23,28–30], in which the algorithm detects two edges of the incoming signal to adjust the sampling time and, furthermore, the algorithm runs sequentially, one channel after another.

This work presents a new algorithm for calibrating the channels of the GSENSE image sensor as the main innovation of this work. FPGA devices have proven to be ideal for implementing algorithms for images coming from cameras for astrophysical applications [31–39]. We have had to design at a low level (structural level) because the data sent by the image sensor to the FPGA cannot be deserialized with the ISERDES IP available in the Vivado suite. That is why in this prototype, the algorithm has to be adapted to the Xilinx FPGAs of the 7-series, which presents new components in its architecture and a greater amount of logic resources compared to previous FPGAs. Taking this into account, improvements have been made at two levels: at a structural level, the algorithm has been modified to adjust the sampling time using a single edge of the incoming signal (instead two edges), adapting it to the new components of the Xilinx FPGA family 7. At the architectural level, the calibration algorithm trains all eight channels simultaneously rather than sequentially. This implies a significant decrease in the calibration time, at the cost of a greater use of FPGA hardware resources, but the new FPGA supports this increase as shown in the Results Section. This parameter could be important in environments and applications where channels need to be calibrated relatively frequently.

The rest of the present work is organized as follows: Section 2 briefly describes the TuMag camera, the prototype and the most relevant components. Section 3 details the firmware implemented in the FPGA to control the image sensor and send data to the frame grabber. Section 4 explains the three levels of the calibration task and the most relevant aspects of the new and enhanced channel calibration (using the new input delay component for Artix-7 and changing for a parallel calibration of channels). Then, in Section 5, we present the results. Finally, the conclusions are presented in Section 6.

2. The TuMag Camera

The camera of the TuMag instrument basically consists of a design of two PCBs, one of them contains the GSENSE400 image sensor and its power supply electronics and the other contains an Artix-7 FPGA in charge of controlling and receiving in the first instance the images acquired by the sensor.

Figure 1 shows a schematic of the system consisting of the sensor and the FPGA, as well as their connections. As can be seen, the image sensor is configured by writing to a series of control registers via an SPI bus. Parameters, such as the training pattern, the sensor gain, the internal PLL multiplication and dividing factors, and so on, are selected in the register map accessed through this interface. In addition, the FPGA sends a reference clock signal with which the sensor sends a 25 MHz pixel clock that is synchronized with the image data that it sends in LVDS format to the FPGA. The *decoder* and *timing* signals in Figure 1 are used to determine both the reset and the reading of the different rows of the pixel array. The adjustment of the reset and reading time of each row by the FPGA driver determines the integration time, capturing the image sensor more or less light in order to form an image [21]. Finally, the *train* signal is used to calibrate each of the 8 differential channels with which the images are sent to the FPGA. As each pixel sent consists of 12 bits, data are sent at 300 Mbps. The training system is explained in Section 4.

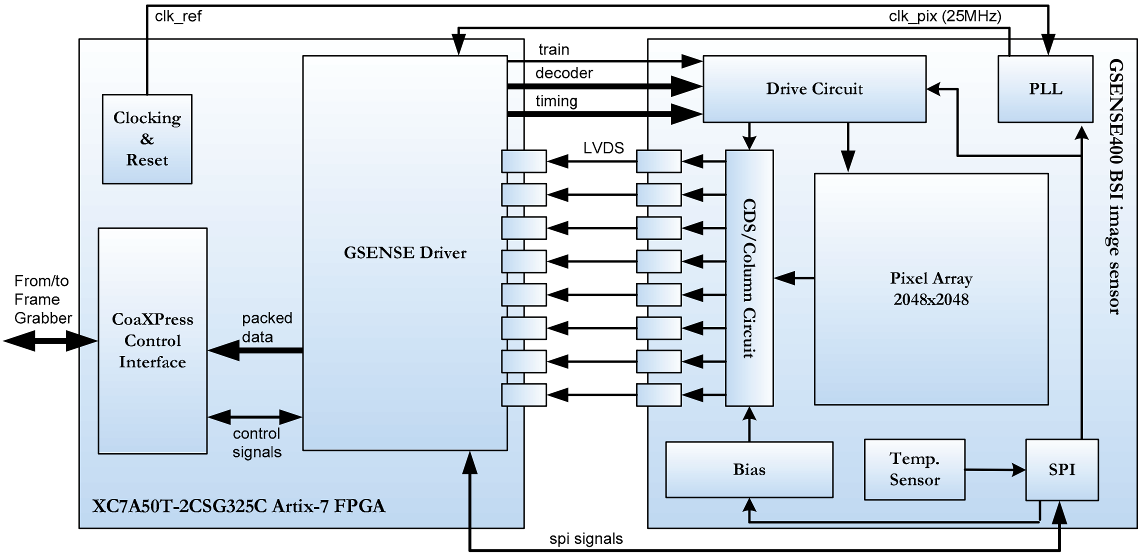


Figure 1. Diagram of the connection system between the image sensor and the FPGA device in the TuMag camera.

The prototype of the TuMag camera with which the development and debugging of the firmware was carried out consists of 2 PCBs (Printed Circuit Boards) as shown in Figure 2. The image sensor has been separated from the FPGA that communicates with this device. Thus, the vertically placed PCB includes the GPIXEL GSENSE400-BSI CMOS image sensor and the horizontally placed PCB includes an Artix-7 XC7A50T-2CSG325C FPGA [21]. The FPGA includes a CoaXPress communication IP in order to send the image data to the Data Processing Unit [22,40].

Basically, the image sensor has a resolution of 4 megapixel (2048 × 2048), 8 LVDS channels to send captured images and an SPI interface to configure the sensor operation. Each pixel has 12 bits and each pixel is sent at 25 MHz. Thus, the transmission speed is 300 Mbps through the 8 differential channels. These communication channels have to be calibrated before the image sensor operates in order for the data to be transmitted correctly. The calibration algorithm is implemented in the FPGA and includes controlled delays, bit rotations, and variable registers as shown below. In the standard mode operation, the image sensor works at 48 fps.

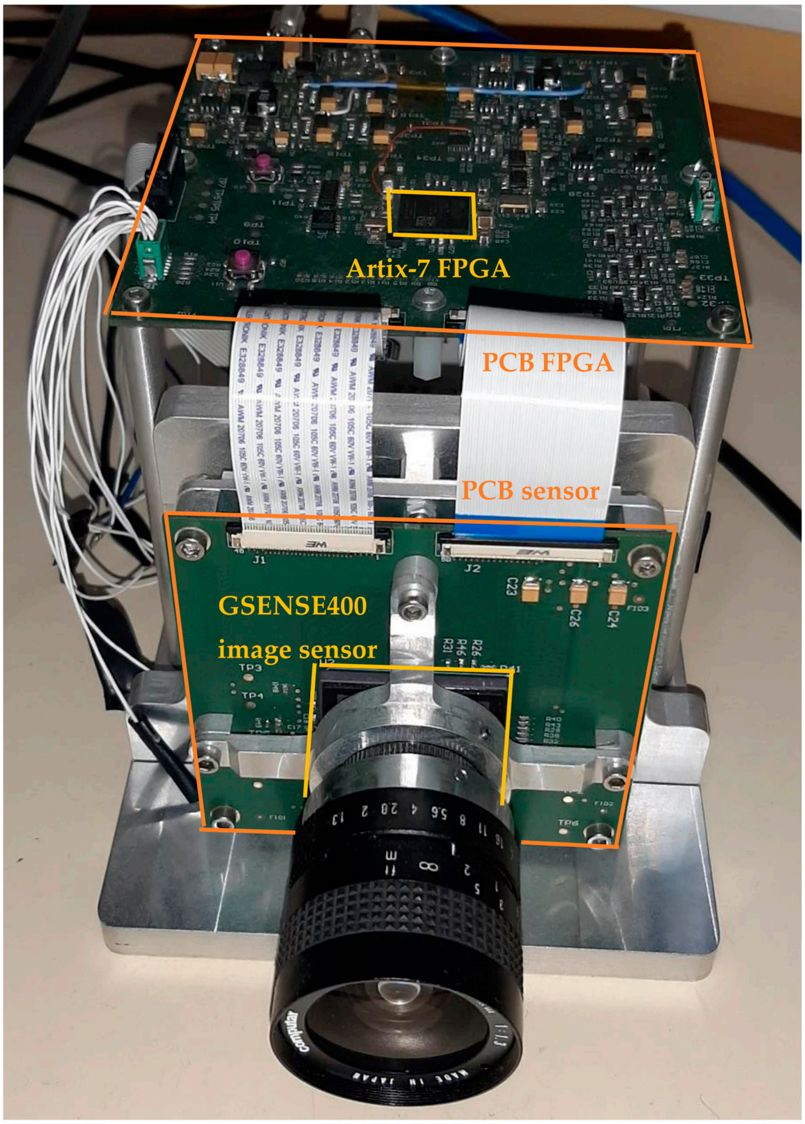


Figure 2. Picture of the first prototype of the TuMag instrument camera.

Artix-7 FPGA device consists of an array of Configurable Logic Blocks (CLBs) which are composed of two slices. These elements are connected to other similar blocks via programmable interconnects and switch matrices. Inside a slice, there are eight flip-flops and four Lookup Tables (6-LUT). Double data rate (DDR) is supported by all inputs and outputs. Any input and some outputs can be individually delayed by up to 32 increments of 78 ps, 52 ps, or 39 ps each. Such delays are implemented as IDELAY and ODELAY modules [22,26]. The number of delay steps has to be set by the calibration algorithm in order to receive the image data incoming from the sensor. These primitives make the design of serializer and deserializer circuits very straightforward and allows higher operation at speeds from 415 Mbps to 1200 Mbps per line [41].

The Artix-7 XC7A50T-2CSG325C FPGA has 32,600 6-LUT and 65,200 slice registers. The FPGA implementation makes the designed algorithm, flexible, customizable, reconfigurable or reprogrammable with advantages of well-customized, integration, accessibility and expandability. The system can be resized according to its needs taking advantages of the VHDL and verilog configurability. This device also has 150 input/output blocks (IOBs). Each IOB is configurable and can comply with a large number of I/O standards [24]. In this case, eight IOBs have been configured as LVDS inputs using DDR data reception with per bit-deskew.

3. TuMag Camera Firmware Overview

Two main blocks can be distinguished in Figure 1, namely, the GSENSE400 driver and the CoaXPress (CxP) control Interface. A detail of the implemented firmware in FPGA is depicted in Figure 3.

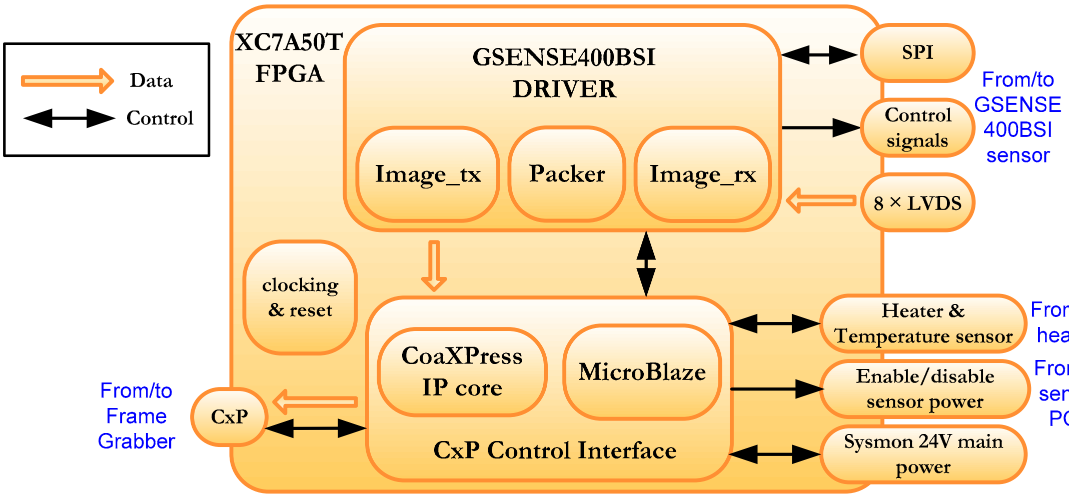


Figure 3. Camera FW device architecture.

The CxP Control Interface includes the Eurasys CoaXPress IP which allows communication with the Data Processing Unit using a coaxial cable [42]. The management of this IP is carried out using the embedded MicroBlaze soft-processor, which also monitors the main power and manages the sensor power and a heater [43–45]. Through this interface the data are sent at 3.125 Gbps.

The GSENSE400-BSI block is the other block of this system, responsible for the direct communication with the GESENSE400-BSI sensor. It manages several tasks such as:

1. Configuration of the sensor, through the SPI interface;
2. Generation of the control signals necessary to grab images from the sensor, such as the 19 control signals and the row address signal (decoder and timing signals in Figure 1);
3. Implementation of the image receiving interface (rx), based on 8 LVDS serial data channels from the camera that are converted to 8–12-bit parallel channels;
4. Channels calibration task;
5. Generation of the signals to arrange the received image;
6. Image ordering and packing;
7. Implementation of the image transmitting interface (tx) to CxP Control Interface.

Image_rx sub-module in Figure 3 implements the training algorithm to calibrate communication channels. Figure 4 depicts the *image_rx* sub-module. Red arrows are signals from/to the GPIXEL sensor.

The reception driver module of the sensor consists of two main modules. The first of these is the module called training, which receives data from the 8 differential channels. The second of them is the module called swapping, which performs the treatment of the channels to form an ordered image. The image sensor does not supply the image data in a row-by-column format. The GPIXEL sensor supplies image data in eight-channel format. This task is implemented in the swapping module [39]. Instead of entering the data image incoming from the sensor, it is possible to enter a test image to debug the design using multiplexers without using the image sensor.

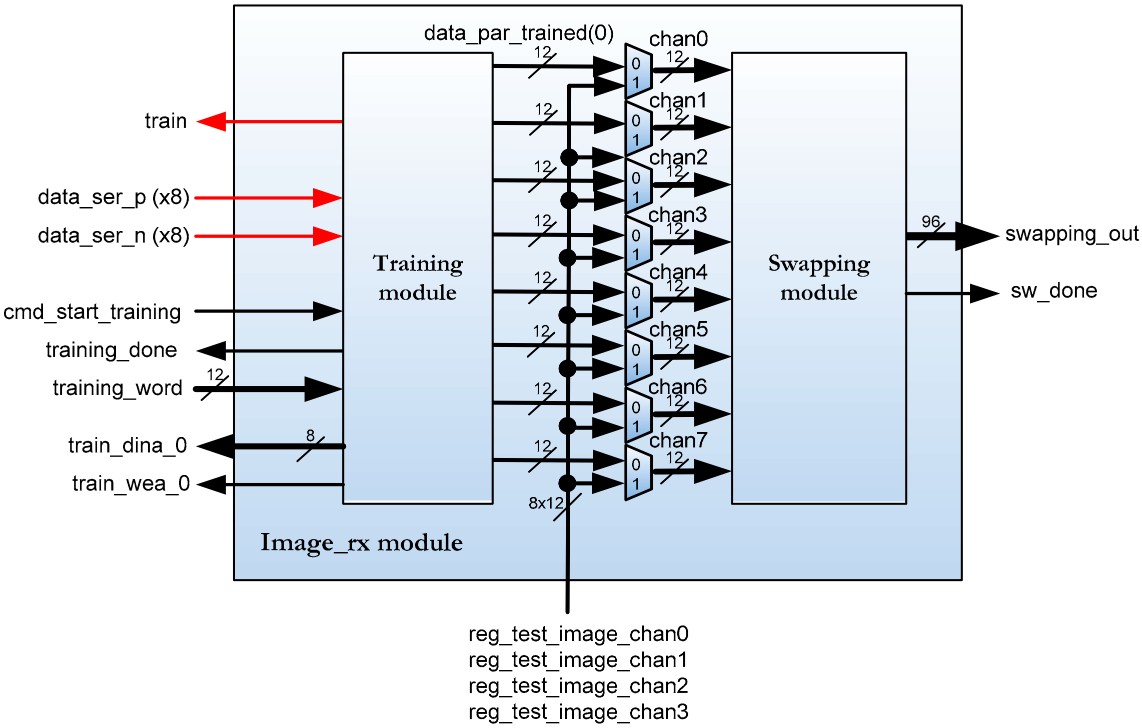


Figure 4. *Image_rx* block diagram.

The training module is described in verilog code and it has the interface shown in Figure 5.

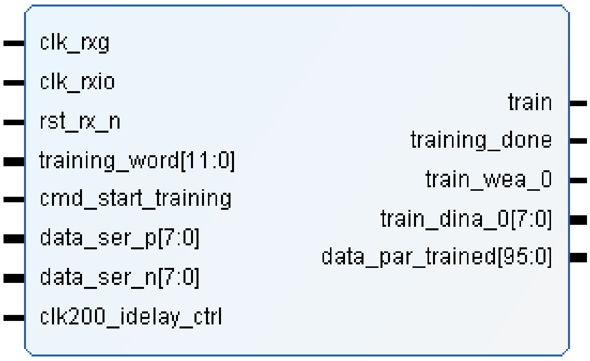
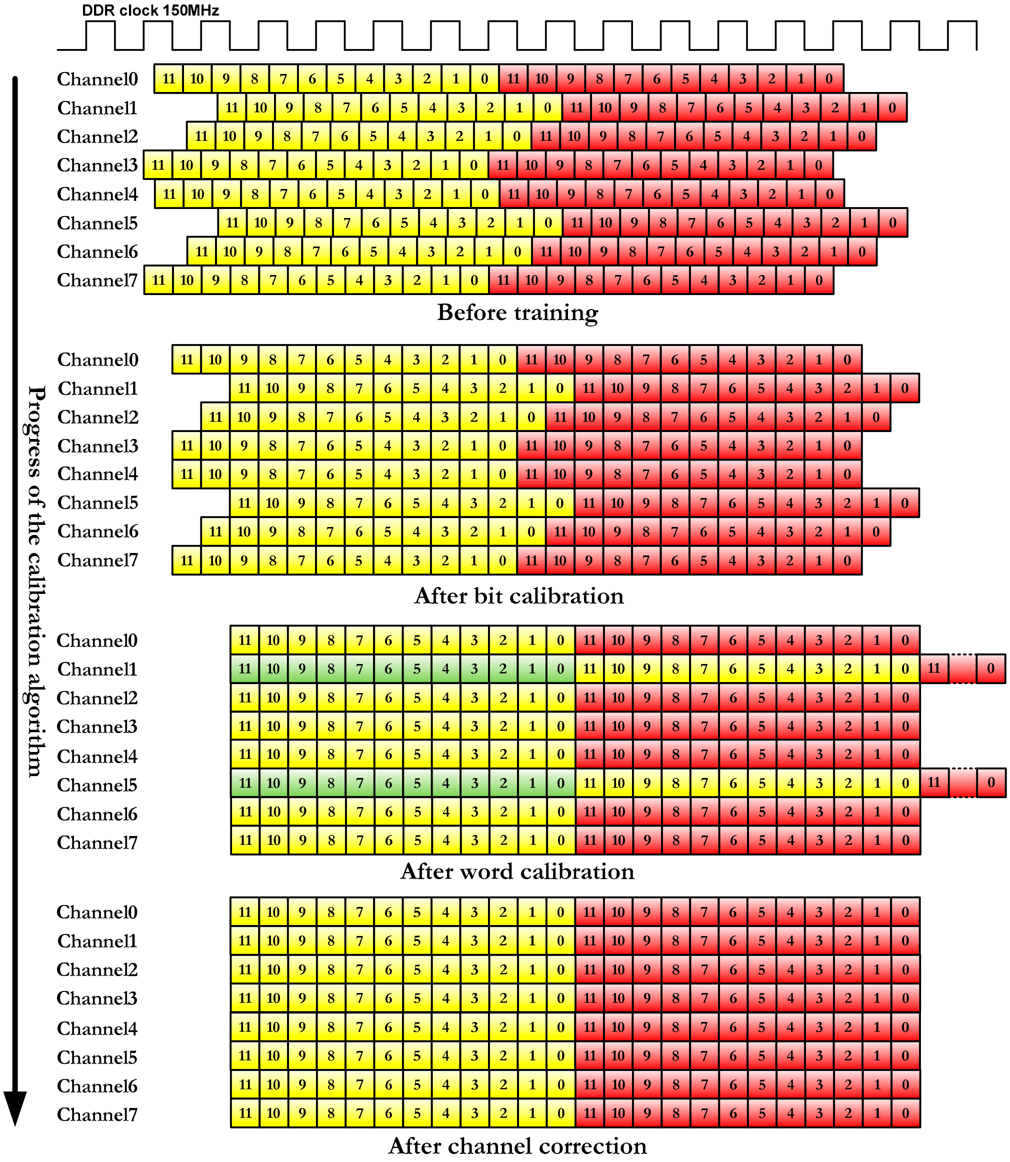


Figure 5. Module interface of training.v.

This sub-module calibrates the data channels in the FPGA so that the data are received correctly aligned (*data_ser_p/n* input ports). Each of the 8 channels includes a 12-bit deserializer with an adjustable input delay, called IDELAY2. Since data are transmitted at 300 Mbps, a 150 MHz clock is used to sample the incoming signal on rising and falling edges (*clk_rxio clock* signal). For this task, the IDDR (input dual data rate) components of the FPGA input blocks are used. These components are controlled by a component of the Artix-7 FPGA, called ICONTROL, which operates at 200 MHz (*clk200_idelay_ctrl* clock signal in Figure 5). These three components are new to the Xilinx FPGA 7 series [25,26]. Basically, a pulse on the *cmd_start_training* port indicates that channel calibration should be performed using the *training_word* value and the *train* port is set to one. When the calibration is finished, the calibrated and parallelized data are available on the *data_par_trained* port and the *training_done port* is set to one. Finally, *train_dina_0* and *train_wea_0* ports are used to send the data information for the calibration results.

4. Channel Calibration

The calibration of each channel is done in three steps as shown in Figure 6. The first step is the calibration of the bit, in which controlled delays are induced to ensure that the data are sampled in the center of the eye and not on one edge [41]. The correct reading of the bits is guaranteed with the previous step, but when inferring delays, it is possible that the 12 bits of each pixel are not aligned. Thus, in the second stage, the received data are rotated until they coincide with the control data previously agreed with the image sensor (*training_word* port in Figure 1). The third and last step is used to synchronize the channels so that all data are received in the same number of clock cycles. The IP RAM-based Shift Register is used to make this setting. All the parameters obtained in these three stages are written in a FIFO to later send them to the DPU to check the result of the calibration (*train_wea_0* and *train_dina_0* signals in Figures 4 and 5). If any channel has not been calibrated correctly, the process is repeated.



Before training

After bit calibration

After word calibration

After channel correction

Figure 6. Calibration process.

Prior to performing this process, a training word is configured in the image sensor (for example, 0 × 98e). When the training module sends a train command to the image sensor (*train* signal set to one), it sends the training word on all channels all the time as pixel data. This constant pixel value is used to calibrate the channels according to Figure 6.

As previously mentioned, the image sensor is configured to send a training word continuously on all channels when the *train* signal in Figure 1 is set to one. Figure 7 depicts data sent by the GSENSE400 using 12-bit X"98e" as training word.

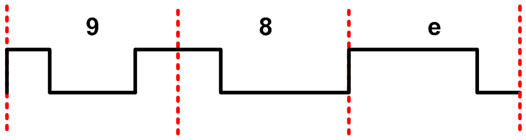


Figure 7. Data outcoming from image sensor to FPGA when train signal is set to high and training word is X"98e".

For each channel, the first thing that is performed is the bit calibration in order to sample each received bit in the middle (Figure 6). This exact sampling point is achieved by delaying the input signal by a few picoseconds. When the received data change with respect to previous data, it means that an edge has been detected in the incoming signal.

The component that performs these delays in the Spartan-6 FPGA is IODELAY2 [46]. Each IOB in the Spartan-6 FPGA contains this delay line that can be configured either for use as an input delay or output delay. This component can deserialize signals up to 1050 Mbps [47]. An 8-bit delay value allows delays from 0 to 255 taps to be achieved. The delay taps have an average value of 40 ps [23]. Two edges of the incoming signal are detected by delaying it using the IODELAYE2 component. The number of taps for each edge are *loc_eye_start* and *loc_eye_end* in Figure 8. With these tap values, the *loc_eye_mid* value is obtained.

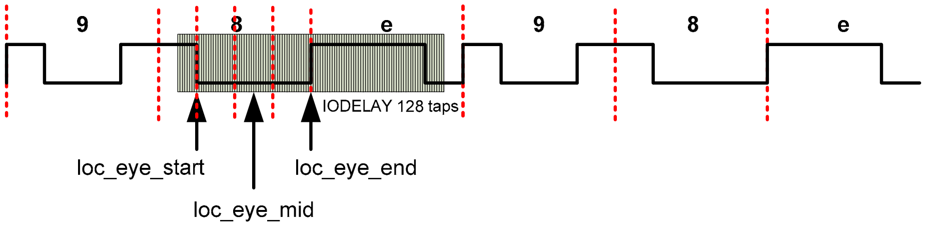


Figure 8. Bit calibration using IODELAYE2 component.

Setting the delay line to *loc_eye_mid* delays the input data by exactly one half an input clock cycle, allowing data sampling in the middle of the input data eye. When the bit calibration is finished, the received word may not be aligned. The training word 98E has been chosen because the rotations of the bits produce different values: 98e, 31d, 63a, c74, 8e9, 1d3, 3a6, 74c, e98, d31, a63, 4c7. In word calibration, the received word is rotated until it matches the expected word. This number of rotations is the *loc_word* parameter. In the worst case, 12 rotations should be performed. Finally, a variable shift register adjusts the number of cycles for each channel so that all data are synchronized (*loc_chan* parameter).

At the end of each channel calibration, the calibration parameters of this iteration (*loc_eye_start*, *loc_eye_mid*, *loc_eye_end*, *loc_word*, *loc_chan* and *loc_nok*/*loc_ok*) are written into a memory. The software must read these data to ensure the success of the calibration or the need of another calibration. When the 8 channels are calibrated, the main FSM is set to the high *training_done* signal and goes to an idle state waiting for a new command for start training.

4.1. Migration of the Calibration Algorithm to Artix-7 FPGA

The Xilinx FPGA family 7 represented a substantial change in the architecture and internal components of these devices compared to the previous ones. Because of this, upgrades of firmware implementations to newer FPGAs are not easy [48]. Several modifications were carried out with respect to the IMaX firmware for Spartan-6:

- Migration from Spartan-6 components to Artix-7 ad hoc components: substitution of IODELAY by IDELAY2, substitution of IDDR2 by IDDR. The 7-series devices have dedicated registers in the ILOGIC blocks to implement input double-data-rate (DDR) registers. This component is similar to the IDDR2 component of the Spartan-6 FPGA and direct replacement is possible [46,48]. However, the IDELAYE2 component of the Artix-7 FPGA differs greatly from its equivalent component of the Spartan-6 and direct replacement is not possible. The 6-series IODELAY component has a 256 tap-delay but the 7-series IDELAY2 has only 32 tap-delay [49]. Figure 9 depicts a comparison between both components;
- This time, the data capture mechanism is dependent on the IODELAY2 component instead IDELAY. The 6-series IODELAY component has a 256 tap-delay but the 7-series IDELAY2 has only a 32 tap-delay of nominally 78 ps and the minimum capture frequency = 78 × 31 = 2418 ps = 415 Mb/s. The frequency of the design is lower (300 Mb/s) and it could result in no edges at all being found. So, the previous algorithm using Spartan-6 FPGA has been modified to consider this situation;
- The bit calibration for Spartan-6 searches 2 edges but the modification for Artix-7 FPGA searches only one edge. If an edge is found in the delay line, then the final delay is statically set to be this value, ±16 taps. If no edge is found, the delay line is set to be 16 taps. In either case, the delay is set to be at least 16 taps away from the edge of the eye, which is acceptable at these lower bit rates [41]. Figure 10 depicts the 3 possible situations in the edge detection mechanism. At left, the edge is detected with taps less than 16 (for example 3), so 16 is added (set to sample a number of taps equal to 19). In the center, no edge is detected, so sampling is set at 16 taps. On the right, the edge is detected with a tap greater than 16 (for example, 20), so 16 is subtracted (sampling with 4 taps).

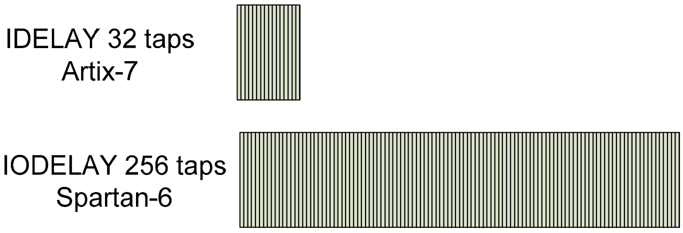


Figure 9. Comparison between IODELAY Spartan-6 FPGA and IDELAY 7-series FPGA.

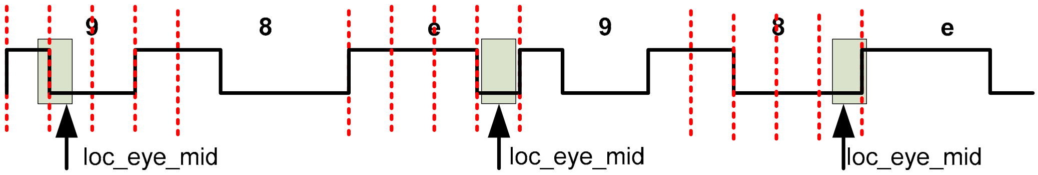


Figure 10. Edge search with the IDELAYE2 component.

Regarding the calibration of the channels, this is still sequential, with state machines similar to the Spartan-6 FPGA implementation controlling the calibration. The most relevant change is made in the bit calibration process, where an edge is searched instead of two, and in the calculation of the number of sampling taps (*loc_eye_mid*) because the *loc_eye_end* parameter is not used (taps value for the second edge is also eliminated). It is important to note that the calibration time is reduced since the calibration being equally sequential, now the number of taps to be covered is reduced from 256 to 32. The value of the reduction depends on the detection of the edge, but it will always be less than the Spartan-6 based solution looking for 2 edges.

4.2. FPGA Concurrent Channel Calibration

The migration of the channel calibration supposes an acceleration of the algorithm caused by the change in the calculation method of the sampling instant in the bit calibration (see Figure 6).

The acceleration of the channel calibration algorithm is much more relevant if the channels are calibrated simultaneously rather than sequentially. This is achieved by using more logic resources of the FPGA. The motivation for the change of the algorithm at the cost of logical resources is due to the fact that the Artix-7 FPGA is larger than the Spartan-6 and an increase in the logic resources used for this task does not compromise the rest of the tasks that the firmware must perform (communication, control, configuration, and so on).

The training module has been separated into 2 hierarchical levels in order to carry out the parallelization. At the lowest hierarchical level, the deserializer of each channel is implemented with its own local training control. This sub-module is named *training_one_channel*. At the top hierarchical level, each channel is monitored and all calibration data are written after all 8 channels have completed the process.

The top level of the training module for concurrent calibration has the same block diagram and port descriptions that the sequential one (see Figure 5). The *training_one_channel* sub-module is described in verilog code and it has the interface shown in Figure 11. Table 1 also shows a brief description of each port of the module.

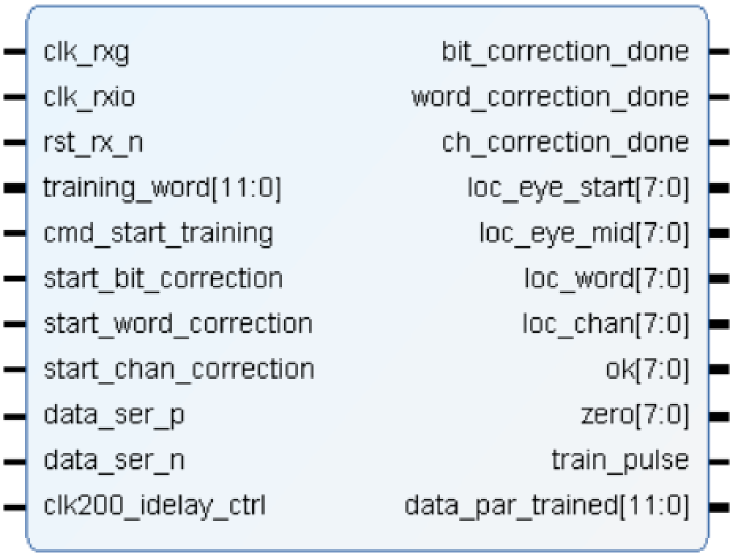


Figure 11. Module interface of training_one_channel.v.

Figure 12 depicts a simplified diagram of the local FSM of each channel for concurrent calibration. The yellow region corresponds to the bit calibration. The red region (S_WORD_ALIGN state) corresponds to the word calibration. The blue region (S_CHAN_ALIGN state) corresponds to the channel calibration. Now, the FSM must wait for the *start_bit_correction*, *start_word_correction* and *start_chan_correction* commands to start each of the three calibration levels. At the end of each calibration stage, a signal is sent to the general calibration control (in the upper hierarchical module). These signals are *bit_correction_done*, *word_correction_done*, and *chan_correction_done* in Figure 11. Note that the writing of the calibration results is not done in this sub-module. This task is carried out in the upper module as opposed to sequential calibration.

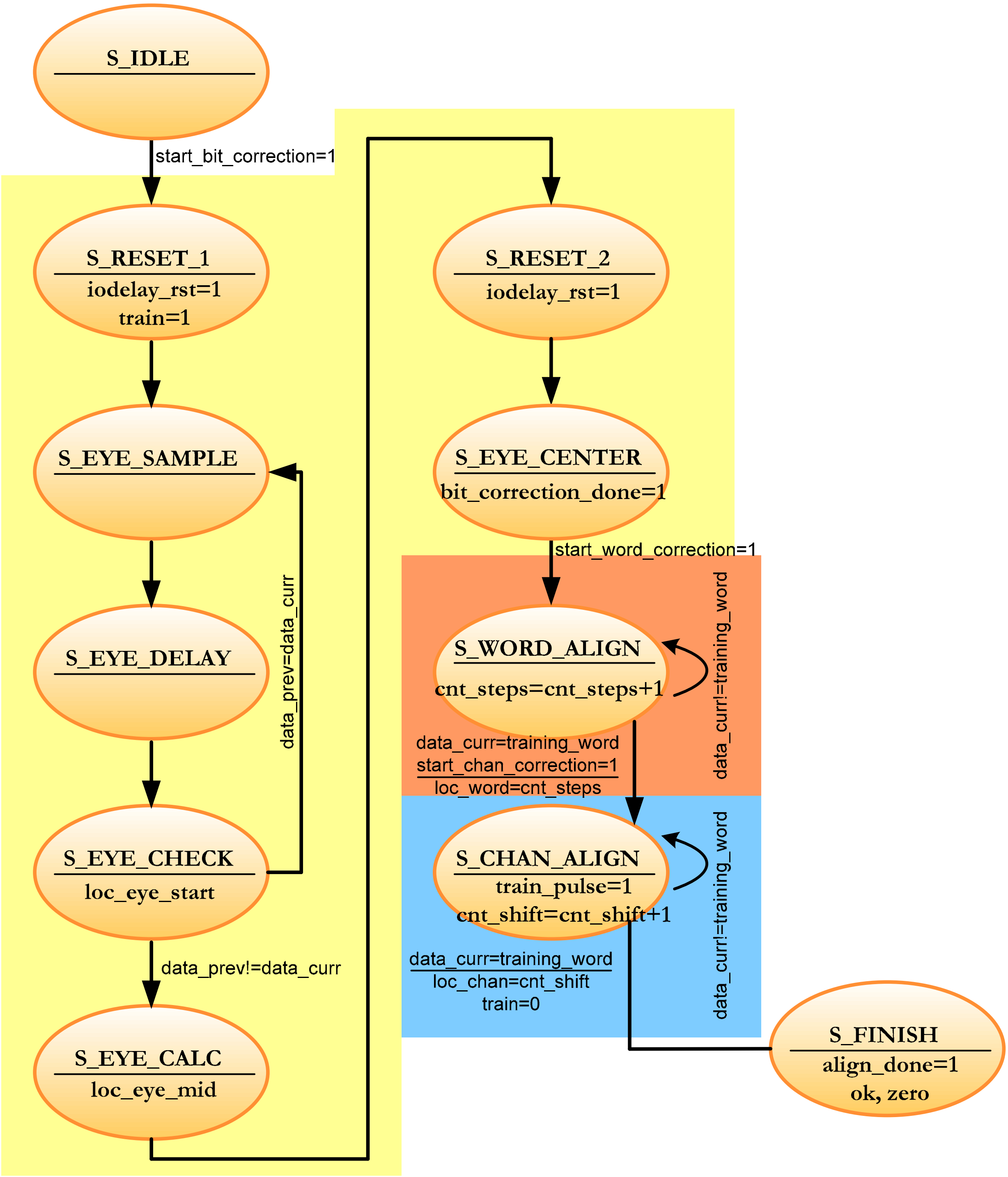


Figure 12. Simplified FSM for one channel calibration in the concurrent architecture.

Table 1. Port description of training_one_channel.v module.

Port	Description
clk_rxg	25 MHz clock
clk_rxio	150 MHz clock for sampling channels
rst_rx_n	low level reset
training_word	word used for calibration task by comparison
cmd_start_training	command for start training task
start_bit_correction	command to start bit correction
start_word_correction	command to start word correction
start_chan_correction	command to start channel correction
data_ser_p/n	8-LVDS channels from the sensor
clk200_idelay_ctrl	200 MHz reference clock
data_par_trained	parallelized and calibrated data (8 channels)
bit_correction_done	flag for the end of bit correction
word_correction_done	flag for the end of word correction
ch_correction_done	flag for the end of channel correction
loc_eye_start	Tap value for the edge detection
loc_eye_mid	Tap value for sampling
loc_word	Number of rotations
loc_chan	Number of shift register
ok	Calibration channel ok
zero	Zero value (for debug)
train_pulse	Train pulse order for channel calibration
data_par_trained	parallelized and calibrated data (1 channel)

Figure 13 depicts the simplified state machine that controls the concurrent calibration of the 8 channels. The machine starts in the S_CTRL_IDLE state. When the *cmd_start_training* signal goes to high, the next state is S_START_BIT_CAL, which sets the *train* signal to 1, so it instructs the sensor to send the calibration word on all channels all the time. At the same time, the *start_bit_correction* signal is set to high, instructing the 8 submodules to begin the bit calibration in Figure 6. It remains in this state until all submodules indicate that they have finished the bit calibration (*bit_correction_done* = FF). When this happens, the *start_word_correction* signal is set to high in the S_START_WORD_CAL state, so the word calibration begins. As in the previous state, the machine remains in this state until all the sub-modules complete this calibration (*word_correction_done* = FF). Channel calibration begins in the S_START_CHAN_CAL state by setting the *start_chan_correction* signal to high and sending a pulse in the *train* signal. Each channel counts the number of clock cycles it takes to receive the training word to set the variable shift register with the correct depth. All channels report that they have finished the channel calibration (*chan_correction_done* = FF) and all the calibration parameters obtained are written into a FIFO. The green shaded area is equivalent to writing the calibration parameters for each channel in the sequential calibration but for all channels.

The data reception module in the FPGA consists of 2 sub-modules as mentioned in Section 3: the training module and the swapping module (see Figure 4). In the concurrent implementation of channel calibration, the training module has the architecture shown in Figure 14. Each channel now has its own local control logic module that runs the algorithm in Figure 12 and controls the calibration process for each channel. A global control module that manages the entire training module according to the algorithm shown in Figure 13 supervises these local logic sub-modules. The ports described in Figures 5 and 6 are grouped in *the control signal* bus in the diagram of Figure 14.

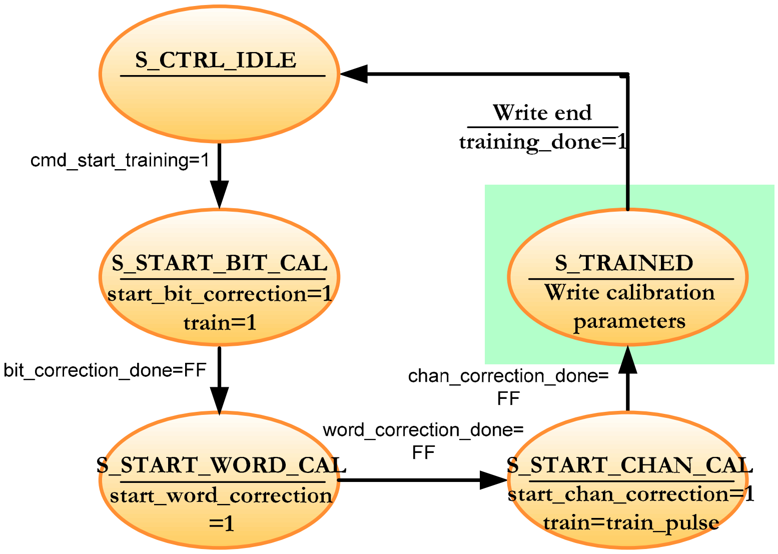


Figure 13. Simplified FSM for the overall calibration system in the concurrent architecture.

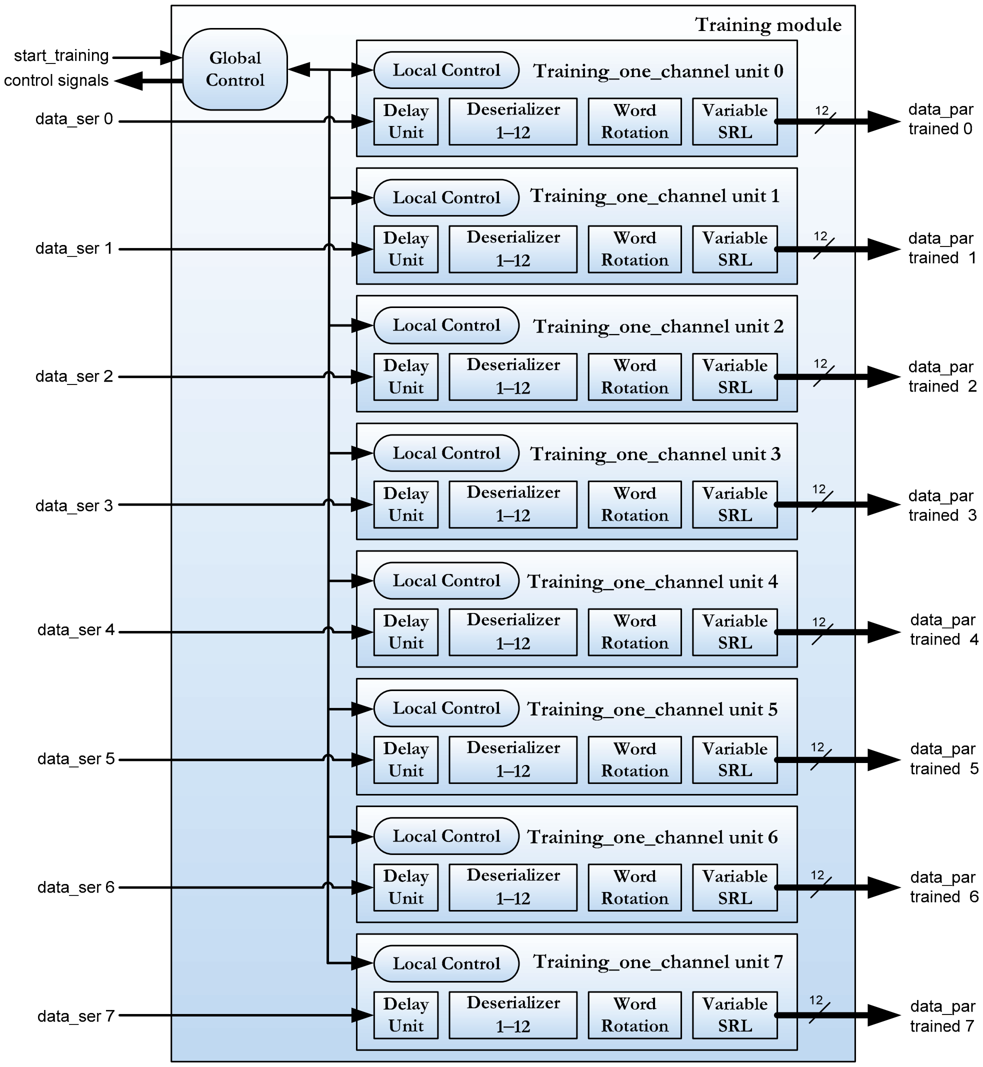


Figure 14. Block diagram of the concurrent channel calibration for the training module.

5. Results

An enhanced channel calibration for the GSENSE400 image sensor of the TuMag instrument has been implemented. The implementation consisted of two steps: first, there was the migration of the calibration algorithm from the Spartan-6 FPGA to the Artix-7 FPGA, preserving the architecture of the module and adapting the design to the components of the 7-series of Xilinx FPGAs; and finally, the calibration algorithm was parallelized to minimize training time. The codes have been developed using Vivado 2017.4 for simulation, debugging and implementation.

As mentioned above, the migration of the calibration algorithm to the new FPGA resulted in a reduction in training time. This is due to the fact that the component that infers delays on the incoming signals, the IDELAY component, has a lower number of taps than the equivalent component of the previous FPGA. In addition, the algorithm had to be adapted to look for one edge in the incoming signal instead of two, as mentioned in Section 4.1.

Once the migration to the new FPGA had been carried out, the possibility of modifying the architecture of the training module was studied to calibrate the 8 channels of the image sensor simultaneously.

Figure 15 shows a comparison between the sequential algorithm and the parallel algorithm. The sequential algorithm calibrates the channels one after another using an 8-to-1 multiplexer. The *chan_sel* signal (in orange in the figure) selects the currently active channel. The calibration parameters are written at the end of the calibration of each channel (*fifo_train_wen* and *fifo_train_din* signals in magenta in the figure). The parallel calibration algorithm can be seen below the sequential algorithm in Figure 15. The operation of the parallel algorithm has been described in Section 4.2. All channels are in the bit calibration phase when the *start_bit_correction* signal is set to high. When all channels have finished this phase (*bit_correction_done* signal set to high), word calibration proceeds (*start_word_correction* signal is set to high) and ends when all channels set the *word_correction_done* signal to high. In the same way, the channel calibration of all channels starts simultaneously when the *start_chan_correction* signal is set to high and when all channels finish this stage, the *ch_correction_done* signal is set to high. At this moment, the calibration parameters of all the channels are written and the algorithm ends with the *training_done* signal set to high.

Table 2 shows the computed times measured for channel calibration using the three available architectures. There is a relevant reduction in the calibration time of the migration from the Spartan-6 FPGA to the Artix-7 FPGA. It is due to the component that infers the delay has 8 times fewer taps. In addition, this fact implies that the algorithm should look for an edge in the incoming signal instead of two edges, as mentioned in Section 4.1. In addition, for the same FPGA, the speed-up of the parallel implementation compared to the sequential one is 8.68, as expected.

Table 2. Calibration times for the GSENSE400 image sensor.

Training Module	Time
Sequential calibration (Spartan-6)	4122.58 µs
Adapted sequential calibration (Artix-7)	524.88 µs
Concurrent calibration (Artix-7)	60.44 µs

The FPGA resources used by each of the three sub-modules are detailed in Table 3. It can be seen that the parallel implementation uses more resources than the sequential one. Parallel channel calibration uses 4.4 times more LUTs and 2.2 times more flip-flops. Even so, the amount of resources used is relatively low (less than 5% and 2%, respectively), so it is acceptable to speed up the calibration at the cost of this increase in logical resources. The remaining space on the FPGA is more than enough for the implementation of all the other control tasks that the FPGA must perform.

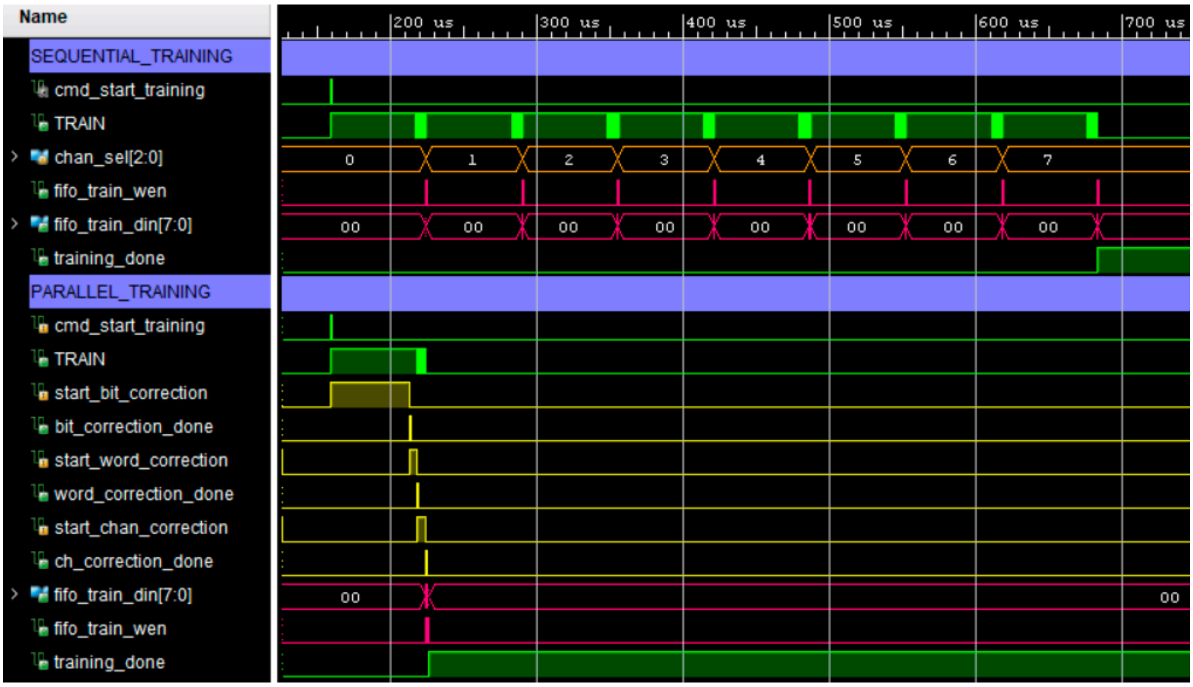


Figure 15. Comparison between the sequential and the parallel algorithm.

Table 3. XC7A50T Artix-7 FPGA logic resources (available and used).

	Available Resources	**Sequential Calibration**	**Concurrent Calibration**
LUT	32,600	303 (0.93%)	1332 (4.09%)
Flip-flops	65,200	463 (0.71%)	973 (1.49%)

GSENSE400 is a relatively new image sensor with high dynamic range, high sensitivity and low noise. The few developments that they have currently been implemented for this sensor use FPGAs older than the Artix-7, such as the Virtex-4, Virtex-5 and the Spartan-6 [27–30]. These designs employ sequential calibration and have used an old development environment that Xilinx no longer supports (Xilinx ISE). Our implementation using an Artix-7 FPGA has been developed using Vivado, the current Xilinx tool. This makes it easy to upgrade the FPGA firmware with new IP libraries. Currently, there is an IP in Vivado for Xilinx 7-family FPGAs that automatically calibrates channels in a way that is easy for firmware developers. This IP is called ISERDES and is a deserializer of incoming data that includes calibration [50,51]. This IP only supports deserializing data of 1:2, 1:4, 1:6, 1:8, 1:10 and 1:14 using two ISERDES modules in a master–slave system, so it is incompatible with deserializer 1:12 that the driver needs to receive the incoming data from the GSENSE400 image sensor. Deserializers based on Artix-7 FPGAs can be found for other image sensors [52,53]. They use the ISERDES IP because they require 1:8 deserializers in their developments. This work results from the development of a very specific instrument. It is because of this that it is compared to previous versions of itself. The works that are referenced in the literature do not specify the calibration time or the FPGA resources that the deserializer uses together with the calibration of the channels. The comparison is difficult for developments using the ISERDES module, because the channel deserializers are not 1:12. A qualitative comparison of the design can be made according to [52]. Table 4 shows the comparison with a design that uses an Artix-7 to perform 1:8 deserialization using the ISERDES IP. In our design, the bandwidth per lane is limited by the GSENSE400 image sensor.

Table 4. Comparison of systems using the Artix-7 FPGA deserializer.

	Liu et al. [52]	**Our System**
System resolution	3840 × 2160	2048 × 2048
Maximum bandwidth per line	891 Mbps	300 Mbps
Deserialization ratio	1:8	1:12
Calibration implementation	ISERDES IP module	Structural HDL code
Hardware cost	Small	Small

Figure 16 shows the results of acquiring an image with the prototype in which the USAF (United States Air Force) resolution test chart has been used. Figure 16a–c shows failed calibration workouts on channels seven, one; and one and five, respectively. A failed calibration is notified through the control signals detailed in Section 4. The calibration is repeated as many times as necessary until it is successful, as in the case of Figure 16d.

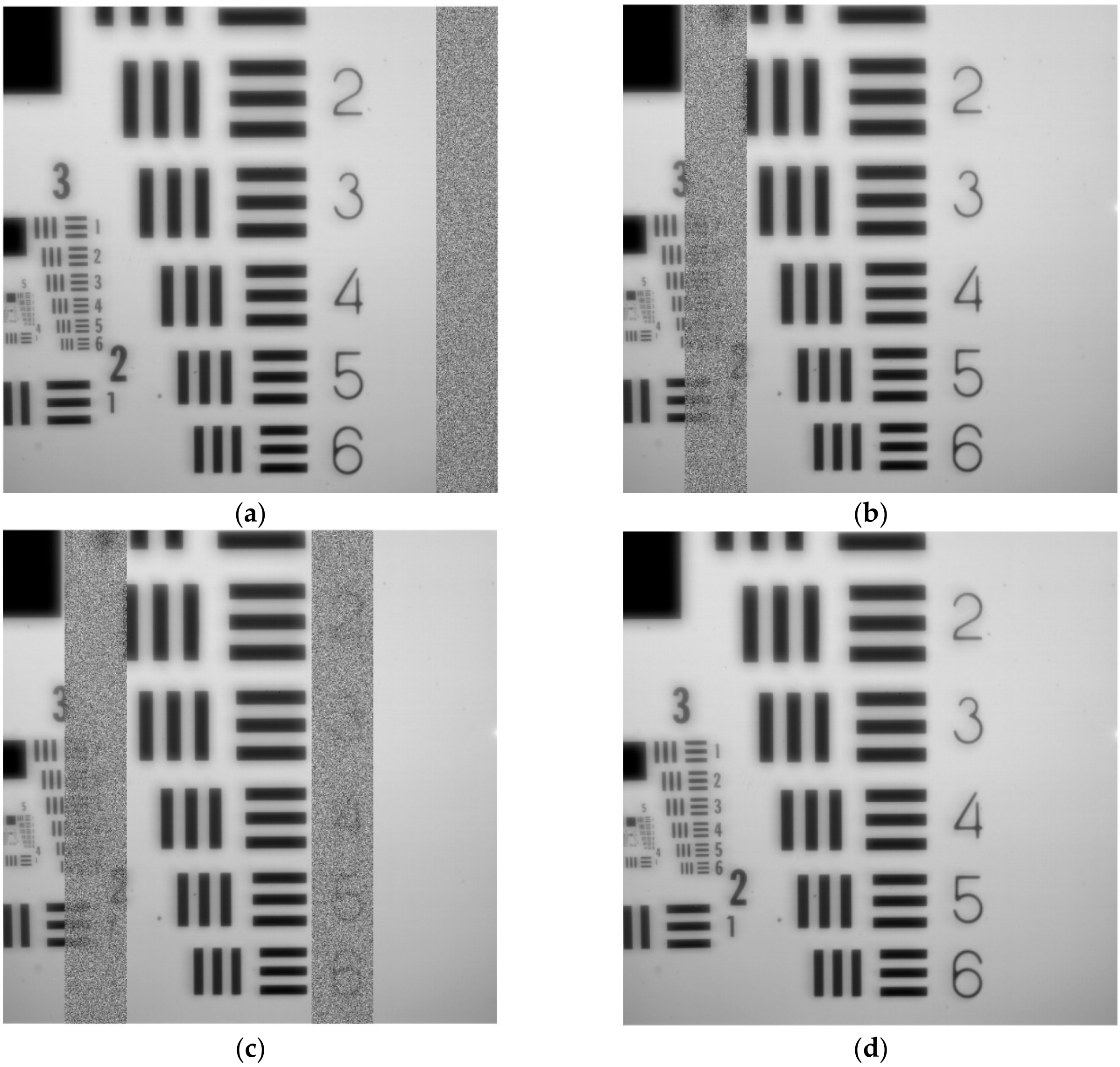


Figure 16. **A** 12-bit 2048 × 2048 image from GSENSE400 using the USAF pattern: (**a**) calibration error in channel seven; (**b**) calibration error in channel one; (**c**) calibration error in channels one and five; (**d**) successful calibration.

6. Conclusions

The concurrent calibration for the channels of the GSENSE400 image sensor has been successfully designed. The calibration process includes bit, word and channel calibration for each channel of the image sensor. Parallel calibration is 8 times faster than expected, as the sensor consists of 8 channels that are now calibrated simultaneously. Although the amounts of logical resources of the FPGA are greater than in the sequential implementation, these are not relevant with respect to the amounts of resources available in the Artix-7; so, there are no problems in parallelizing the training algorithm.

In general, it is preferable to use functional HDLs (hardware description languages) using FPGA-based design technologies. This makes the design independent of the technology (type of FPGA). However, the calibration algorithm has been carried out using specific components of the 7-series family Xilinx FPGAs, since the ISERDES IP cannot be used, so the algorithm has been designed using structural HDL. That is why the implemented algorithm is limited to Spartan-7, Artix-7, Kintex-7 and Virtex-7 FPGAs, being incompatible with FPGAs from other manufacturers and other FPGA families from Xilinx itself.

Technological advancements lead to the expectation that the next generation of image sensors will be greater than the current ones. An increase in the number of channels affects the design very little, as it is a modular and versatile design. The design is easily extrapolated to a greater number of channels to be calibrated.

Author Contributions: Conceptualization, E.M., M.R.V., D.H. and D.Á.G.; methodology, E.M., M.R.V. and D.H.; software, E.M., M.R.V., D.H. and A.M.G.; validation, M.B.; investigation, B.R.C. and D.O.S.; writing—original draft preparation, E.M.; writing—review and editing, E.M. and M.R.V.; supervision, M.B. and B.R.C.; project administration, B.R.C.; funding acquisition, B.R.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the RTI2018-096886-B-C53 Project of the Ministry of Science and Innovation.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Quintero Noda, C.; Katsukawa, Y.; Shimizu, T.; Kubo, M.; Oba, T.; Ichimoto, K. Sunrise-3: Science target and mission capabilities for understanding the solar atmosphere. In Proceedings of the 19th Space Science Symposium, Sagamihara, Japan, 9 January 2019.
2. Gandorfer, A.; Solanki, S.K.; Woch, J.; Pillet, V.M.; Herrero, A.A.; Appourchaux, T. The Solar Orbiter Mission and its Polarimetric and Helioseismic Imager (SO/PHI). *J. Phys. Conf. Ser.* **2011**, *271*, 8. [CrossRef]
3. Pillet, V.M.; Del Toro Iniesta, J.C.; Álvarez-Herrero, A.; Domingo, V.; Bonet, J.A.; Fernández, L.G.; Jiménez, A.L.; Pastor, C.; Blesa, J.G.; Mellado, P.; et al. The Imaging Magnetograph eXperiment (IMaX) for the Sunrise Balloon-Borne Observatory. *Sol. Phys.* **2011**, *268*, 57–102.
4. Barthol, P. *The Sunrise Balloon-Borne Stratospheric Solar Observatory*, 1st ed.; Springer: New York, NY, USA, 2011; pp. 1–123.
5. Barthol, P.; Gandorfer, A.; Solanki, S.K.; Schüssler, M.; Chares, B.; Curdt, W.; Deutsch, W.; Feller, A.; Germerott, D.; Grauf, B.; et al. The Sunrise Mission. *Sol. Phys.* **2011**, *268*, 1–34. [CrossRef]
6. Solanki, S.K.; Barthol, P.; Danilovic, S.; Feller, A.; Gandorfer, A.; Hirzberger, J.; Riethmüller, T.L.; Schüssler, M.; Bonet, J.A.; Pillet, V.M.; et al. Sunrise: Instrument, Mission, Data, and First Results. *Astrophys. J. Lett.* **2010**, *723*, L127–L133. [CrossRef]
7. Solanki, S.K.; Riethmüller, T.L.; Barthol, P.; Danilovic, S.; Deutsch, W.; Doerr, H.P.; Feller, A.; Gandorfer, A.; Germerott, D.; Gizon, L.; et al. The Second Flight of SUNRISE Balloon-borne Solar Observatory: Overview of Instrument Updates, the Flight, the Data, and First Results. *Astrophys. J. Suppl. Ser.* **2017**, *229*, 16. [CrossRef]
8. Kaithakkal, A.J.; Riethmüller, T.L.; Solanki, S.K.; Lagg, A.; Barthol, P.; Gandorfer, A.; Gizon, L.; Hirzberger, J.; Rodríguez, J.B.; Iniesta, J.D.T.; et al. Moving Magnetic Features Around a Pore. *Astrophys. J. Suppl. Ser.* **2017**, *229*, 13. [CrossRef]
9. Centeno, R.; Rodríguez, J.B.; Iniesta, J.D.T.; Solanki, S.K.; Barthol, P.; Gandorfer, A.; Gizon, L.; Hirzberger, J.; Riethmüller, T.L.; van Noort, M.; et al. A Tale of Two Emergences: SUNRISE II Observations of Emergence Sites in a Solar Active Region. *Astrophys. J. Suppl. Ser.* **2017**, *229*, 3. [CrossRef]

10. Requerey, I.S.; Iniesta, J.C.D.T.; Rubio, L.R.B.; Bonet, J.A.; Pillet, V.M.; Solanki, S.K.; Schmidt, W. The History of a Quiet-Sun Magnetic Element Revealed by IMaX/SUNRISE. *Astrophys. J. Suppl. Ser.* **2014**, *789*, 12. [CrossRef]
11. Requerey, I.S.; Cobo, B.R.; Iniesta, J.D.T.; Suárez, D.O.; Rodríguez, J.B.; Solanki, S.K.; Barthol, P.; Gandorfer, A.; Gizon, L.; Hirzberger, J.; et al. Spectropolarimetric Evidence for a Siphon Flow along an Emerging Magnetic Flux Tube. *Astrophys. J. Suppl. Ser.* **2017**, *229*, 5. [CrossRef]
12. Berkefeld, T.; Schmidt, W.; Soltau, D.; Bell, A.; Doerr, H.P.; Feger, B.; Friedlein, R.; Gerber, K.; Heidecke, F.; Kentischer, T.; et al. The Wave-Front Correction System for the Sunrise Balloon-Borne Solar Observatory. *Sol. Phys.* **2011**, *268*, 103–123. [CrossRef]
13. Riethmüller, T.L.; Solanki, S.K.; Barthol, P.; Gandorfer, A.; Gizon, L.; Hirzberger, J.; Van Noort, M.; Rodríguez, J.B.; Iniesta, J.D.T.; Suárez, D.O.; et al. A New MHD-assisted Stokes Inversion Technique. *Astrophys. J. Suppl. Ser.* **2017**, *229*, 14. [CrossRef]
14. Orozco Suárez, D.; Bellot Rubio, L.R.; del Toro Iniesta, J.C.; Tsuneta, S.; Lites, B.W.; Ichimoto, K.; Katsukawa, Y.; Nagata, S.; Shimizu, T.; Shine, R.A.; et al. Quiet-Sun Internetwork Magnetic Fields from the Inversion of Hinode Measurements. *Astrophys. J.* **2007**, *670*, L61–L64. [CrossRef]
15. Castelló, E.M.; Valido, M.R.; Expósito, D.H.; Cobo, B.R.; Balaguer, M.; Suárez, D.O.; Jiménez, A.L. IMAX+ camera prototype as a teaching resource for calibration and image processing using FPGA devices. In Proceedings of the 2020 XIV Technologies Applied to Electronics Teaching Conference (TAEE), Porto, Portugal, 8–10 July 2020.
16. Katsukawa, Y.; del Toro Iniesta, J.C.; Solanki, S.K.; Kubo, M.; Hara, H.; Shimizu, T.; Oba, T.; Kawabata, Y.; Tsuzuki, T.; Uraguchi, F.; et al. Sunrise Chromospheric Infrared SpectroPolarimeter (SCIP) for sunrise III: System design and capability, in Ground-based and Airborne Instrumentation for Astronomy VIII. In Proceedings of the SPIE Astronomical Telescopes + Instrumentation, Online, 14–18 December 2020. [CrossRef]
17. Feller, A.; Gandorfer, A.; Iglesias, F.A.; Lagg, A.; Riethmüller, T.L.; Solanki, S.K.; Katsukawa, Y.; Kubo, M. The SUNRISE UV Spectropolarimeter and Imager for SUNRISE III. In Proceedings of the SPIE 11447, Ground-Based and Airborne Instrumentation for Astronomy VIII, Online, 14–18 December 2020. [CrossRef]
18. Toshihiro, T.; Katsukawa, Y.; Uraguchi, F.; Hara, H.; Kubo, M.; Nodomi, Y.; Suematsu, Y.; Kawabata, Y.; Shimizu, T.; Gandorfer, A.; et al. Sunrise Chromospheric Infrared spectroPolarimeter (SCIP) for SUNRISE III: Optical design and performance. In Proceedings of the SPIE 11447, Ground-Based and Airborne Instrumentation for Astronomy VIII, Online, 14–18 December 2020. [CrossRef]
19. Kubo, M.; Shimizu, T.; Katsukawa, Y.; Kawabata, Y.; Anan, T.; Ichimoto, K.; Shinoda, K.; Tamura, T.; Nodomi, Y.; Nakayama, S.; et al. Sunrise Chromospheric Infrared spectroPolarimeter (SCIP) for SUNRISE III: Polarization Modulation unit. In Proceedings of the SPIE 11447, Ground-Based and Airborne Instrumentation for Astronomy VIII, Online, 14–18 December 2020. [CrossRef]
20. Uraguchi, F.; Tsuzuki, T.; Katsukawa, Y.; Hara, H.; Iwamura, S.; Kubo, M.; Nodomi, Y.; Suematsu, Y.; Kawabata, Y.; Shimizu, T. Sunrise Chromospheric Infrared spectroPolarimeter (SCIP) for SUNRISE III: Opto-mechanical analysis and design. In Proceedings of the SPIE 11447, Ground-Based and Airborne Instrumentation for Astronomy VIII, Online, 14–18 December 2020. [CrossRef]
21. Gpixel. *GSENSE400 4 Megapixels Scientific CMOS Image Sensor. Datasheet V1.5*; Gpixel Inc.: Changchum, China, 2017.
22. Xilinx Inc. 7 Series FPGAs Data Sheet: Overview. Product Specification. DS180 (v2.6.1). 2020. Available online: https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf (accessed on 22 November 2021).
23. Tang, X.; Liu, G.; Qian, Y.; Cheng, H.; Qiao, K. A Handheld High-Resolution Low-Light Camera. In Proceedings of the SPIE 11023, Fifth Symposium on Novel Optoelectronic Detection Technology and Application, Xi'an, China, 12 March 2019; Volume 110230X. [CrossRef]
24. Xilinx Inc. Spartan-6 Family Overview. Product Specification. DS160 (v2.0). 2011. Available online: https://www.xilinx.com/support/documentation/data_sheets/ds160.pdf (accessed on 22 November 2021).
25. Xilinx Inc. Artix-7 FPGAs Data Sheet: DC and AC Switching Characteristics. Product Specification. DS181 (v1.26). 2021. Available online: https://www.xilinx.com/support/documentation/data_sheets/ds181_Artix_7_Data_Sheet.pdf (accessed on 22 November 2021).
26. Xilinx Inc. Spartan-6 Libraries Guide for HDL Designs. User Guide. UG615 (v14.7). 2013. Available online: https://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/spartan6_hdl.pdf (accessed on 22 November 2021).
27. Xilinx Inc. Xilinx 7 Series FPGA and Zynq-7000 All Programmable SoC Libraries Guide for HDL Designs. User Guide. UG768 (v14.7). 2013. Available online: https://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/7series_hdl.pdf (accessed on 22 November 2021).
28. Heng, Z.; Qing-jun, M.; Shu-rong, W. High speed CMOS imaging electronics system for ultraviolet remote sensing instrument. *Opt. Precis. Eng.* **2018**, *26*, 471–479. [CrossRef]
29. Ma, C.; Liu, Y.; Li, J.; Zhou, Q.; Chang, Y.; Wang, X. A 4MP high-dynamic-range, low-noise CMOS image sensor. In Proceedings of the SPIE 9403, Image Sensors and Imaging Systems 2015, San Francisco, CA, USA, 13 March 2015; Volume 940305. [CrossRef]
30. Wang, F.; Dai, M.; Sun, Q.; Ai, L. Design and implementation of CMOS-based low-light level night-vision imaging system. In Proceedings of the SPIE 11763, Seventh Symposium on Novel Photoelectronic Detection Technology and Applications, San Francisco, CA, USA, 12 March 2021; Volume 117635O. [CrossRef]
31. Magdaleno, E.; Rodríguez, M.; Rodríguez-Ramos, J.M. An Efficient Pipeline Wavefront Phase Recovery for the CAFADIS Camera for Extremely Large Telescopes. *Sensors* **2010**, *10*, 1–15. [CrossRef] [PubMed]

32. Magdaleno, E.; Lüke, J.P.; Rodríguez, M.; Rodríguez-Ramos, J.M. Design of Belief Propagation Based on FPGA for the Multistereo CAFADIS Camera. *Sensors* **2010**, *10*, 9194–9210. [CrossRef] [PubMed]
33. Pérez, J.; Magdaleno, E.; Pérez, F.; Rodríguez, M.; Hernández, D.; Corrales, J. Super-Resolution in Plenoptic Cameras Using FPGAs. *Sensors* **2014**, *14*, 8669–8685. [CrossRef] [PubMed]
34. Rodríguez, M.; Magdaleno, E.; Pérez, F.; García, C. Automated Software Acceleration in Programmable Logic for an Efficient NFFT Algorithm Implementation: A Case Study. *Sensors* **2017**, *17*, 694. [CrossRef]
35. Alvear, A.; Finger, R.; Fuentes, R.; Sapunar, R.; Geelen, T.; Curotto, F.; Rodríguez, R.; Monasterio, D.; Reyes, N.; Mena, P.; et al. FPGA-based digital signal processing for the next generation radio astronomy instruments: Ultra-pure sideband separation and polarization detection, in Millimeter, Submillimeter, and Far-Infrared Detectors and Instrumentation for Astronomy VIII. In Proceedings of the SPIE, Edinburgh, UK, 26 June–1 July 2016; Volume 9914. [CrossRef]
36. Schwanke, U.; Shayduk, M.; Sulanke, K.-H.; Vorobiov, S.; Wischnewski, R. A Versatile Digital Camera Trigger for Telescopes in the Cherenkov Telescope Array. In *Nuclear Instruments and Methods in Physics Research Section A: Accelerators Spectrometers Detectors and Associated Equipment*; Elsevier: Amsterdam, The Netherland, 2015; Volume 782, pp. 92–103. [CrossRef]
37. Lecerf, A.; Ouellet, D.; Arias-Estrada, M. Computer vision camera with embedded FPGA processing. In Proceedings of the SPIE 3966, Machine Vision Applications in Industrial Inspection VIII, San Jose, CA, USA, 21 March 2000. [CrossRef]
38. Heller, M.; Schioppa, E.; Jr Porcelli, A.; Pujadas, I.T.; Ziętara, K.; Della Volpe, D.; Montaruli, T.; Cadoux, F.; Favre, Y.; Aguilar, J.A.; et al. An innovative silicon photomultiplier digitizing camera for gamma-ray astronomy. *Eur. Phys. J. C* **2017**, *77*, 47. [CrossRef]
39. Magdaleno, E.; Rodríguez, M.; Hernández, D.; Balaguer, M.; Ruiz-Cobo, B. FPGA implementation of Image Ordering and Packing for TuMag Camera. *Electronics* **2021**, *10*, 1706. [CrossRef]
40. EASii IC. *CoaxPress Device IP Specification. Hard Soft Interface Document. IC/130206*; EASii IC SAS: Grenoble, France, 2017.
41. Sawyer, N. LVDS Source Synchronous 7:1 Serialization and Deserialization Using Clock Multiplication. Xilinx XAPP585 (v.1.1.2). 2018. Available online: https://www.xilinx.com/support/documentation/application_notes/xapp585-lvds-source-synch-serdes-clock-multiplication.pdf (accessed on 2 December 2020).
42. CoaxPress Host IP Specification. *Hard Soft Interface Document. IC/130249*; EASii IC SAS: Grenoble, France, 2017.
43. Magdaleno, E.; Rodríguez, M. TuMag Camera Firmware design and implementation report. SR3-IMAXP-RP-SW730-001. Revision A. Release date 2020-08-01, La Laguna, Spain, September 2021.
44. GenICam Standard. Generic Interface for Cameras. Version 2.1.1. Emva. Available online: https://www.emva.org/wp-content/uploads/GenICam_Standard_v2_1_1.pdf (accessed on 2 December 2020).
45. MicroBlaze Processor Reference Guide. Xilinx UG984 (v2018.2). 2018. Available online: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_2/ug984-vivado-microblaze-ref.pdf (accessed on 2 December 2020).
46. Xilinx Inc. Spartan-6 FPGA SelectIO Resources. User Guide. UG381 (v1.7). 2015. Available online: https://www.xilinx.com/support/documentation/user_guides/ug381.pdf (accessed on 29 November 2021).
47. Sawyer, N. Source-Synchronous Serialization and Deserialization (up to 1050 Mb/s). Xilinx XAPP1064 (v.1.2). 2013. Available online: https://www.xilinx.com/support/documentation/application_notes/xapp1064.pdf (accessed on 29 November 2021).
48. Xilinx Inc. 7-Series FPGAs Migration. User Guide. UG429 (v1.2). 2018. Available online: https://www.xilinx.com/support/documentation/sw_manuals/ug429_7Series_Migration.pdf (accessed on 11 February 2022).
49. Xilinx Inc. 7-Series FPGA SelectIO Resources. User Guide. UG471 (v1.10). 2018. Available online: https://www.xilinx.com/support/documentation/user_guides/ug471_7Series_SelectIO.pdf (accessed on 30 November 2021).
50. Defossez, M.; Sawyer, N. LVDS Source Synchronous DDR Deserialization (up to 1600 Mb/s), XAPP1017 (v1.0). 2016. Available online: https://www.xilinx.com/support/documentation/application_notes/xapp1017-lvds-ddr-deserial.pdf (accessed on 14 February 2022).
51. Xilinx Inc. SelectIO Interface Wizard v5.1. LogiCORE IP Product Guide PG070. 2016. Available online: https://www.xilinx.com/support/documentation/ip_documentation/selectio_wiz/v5_1/pg070-selectio-wiz.pdf (accessed on 14 February 2022).
52. Liu, F.; Wang, L.; Yang, Y. A UHD MIPI CSI-2 image acquisition system based on FPGA. In Proceedings of the 40th Chinese Control Conference (CCC), IEEE Conference, Shanghai, China, 26–28 July 2021; pp. 5668–5673.
53. Lee, P.H.; Lee, H.Y.; Kim, Y.W.; Hong, H.Y.; Jang, Y.C. A 10-Gbps receiver bridge chip with deserializer for FPGA-based frame grabber supporting MIPI CSI-2. *IEEE Trans. Consum. Electron.* **2017**, *63*, 209–2015. [CrossRef]