Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.



THIS SPEC IS OBSOLETE

Spec No: 001-54468

Spec Title: INTERFACING LUPA1300-2 WITH FPGA - AN54468

Sunset Owner: Evelyn Beard (EYB)

Replaced By: NONE

Interfacing LUPA1300-2 with FPGA

AN54468

Author: Anoop Kumar Upadhyay
Associated Project: No
Associated Part Family: CYIL2SM1300AA
Software Version: None
Associated Application Notes: None

OBSOLETE

Application Note Abstract

This application note describes the interface between the image sensor and FPGA, implemented in the LUPA1300-2 demonstration kit (CYIL2SM1300-Eval). It also gives an overview of the demonstration system architecture and the method used to synchronize channels.

Introduction

LUPA1300-2 is a high speed CMOS image sensor with 1296 x 1024 (1.3 Megapixel, SXGA) resolution. The maximum operating frequency of the sensor is 315 MHz and it reads out 500 frames per second at full resolution. The sensor is based on 6T-pixel architecture and works in snapshot pipelined shutter mode (The integration for the next frame and readout of the present frame happens simultaneously). The image sensor can operate in Master, Slave, and Triggered modes.

The sensor has 12 LVDS data channels and one LVDS sync channel. Data is sent out at Double Data Rate (DDR). At a nominal speed of 310 MHz, data comes out at 620 Mbps. Therefore, data can be transferred over a longer distance, simplifying the surrounding system with low noise and distortion.

The sync channel provides the synchronization codes for the receiver, indicating Frame Start (FS), Line Start (LS), Line End (LE), ROT, FOT, Frame End (FE), image data, CRC, and training patterns. Pixels are always read in multiples of 24 (1 kernel) and each kernel corresponds to one timeslot. The last timeslot contains not only eight valid pixels, but also two dummy, six grey, and eight black pixels.

Figure 1 shows the timing diagram for full frame readout.

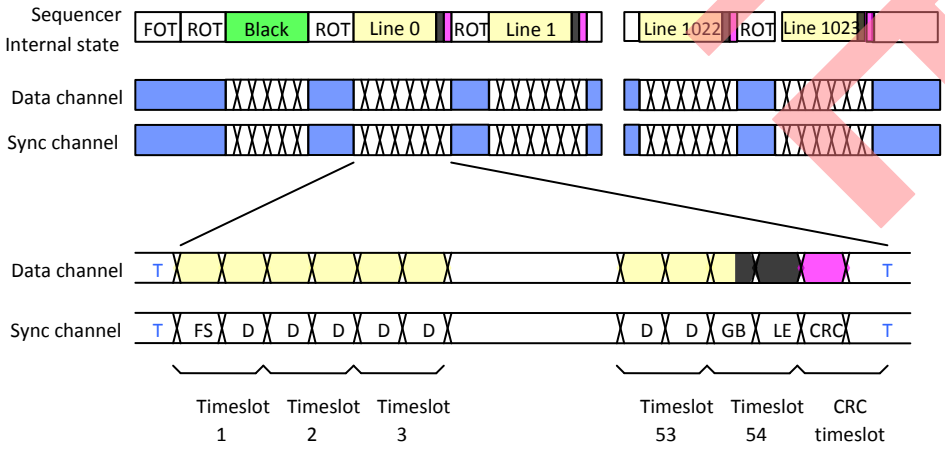
1 Timeslot = 1 Kernel = 24 Pixels Equation 1

Total Timeslots = 54 Equation 2

Before starting to read out the actual pixels, one dummy line is read to allow column FPN calibration (green colored block in Figure 1).

To evaluate link integrity when operating the sensor-camera system, a CRC code is sent during ROT. The polynomial used to generate the CRC checksum is (x^10+x^9+x^6+x^3+x^2+x+1) and 10 bits are calculated in parallel.

Figure 1. Full Frame Mode Readout Operation



Serial Link Interface Operation

The image sensor's serial link interface is based on a Mesochronous clocking system. This means that all parallel links operate at the same frequency, but their phase can be different. The skew between all the channels including the sync channel is less than 150 ps. The receiver (FPGA) performs per-pin skew compensation.

The host (FPGA) must send the LVDS clock as input to the image sensor. To compensate for possible large on-chip delays, the sensor transmits back the same clock with the same delay as the data suffers.

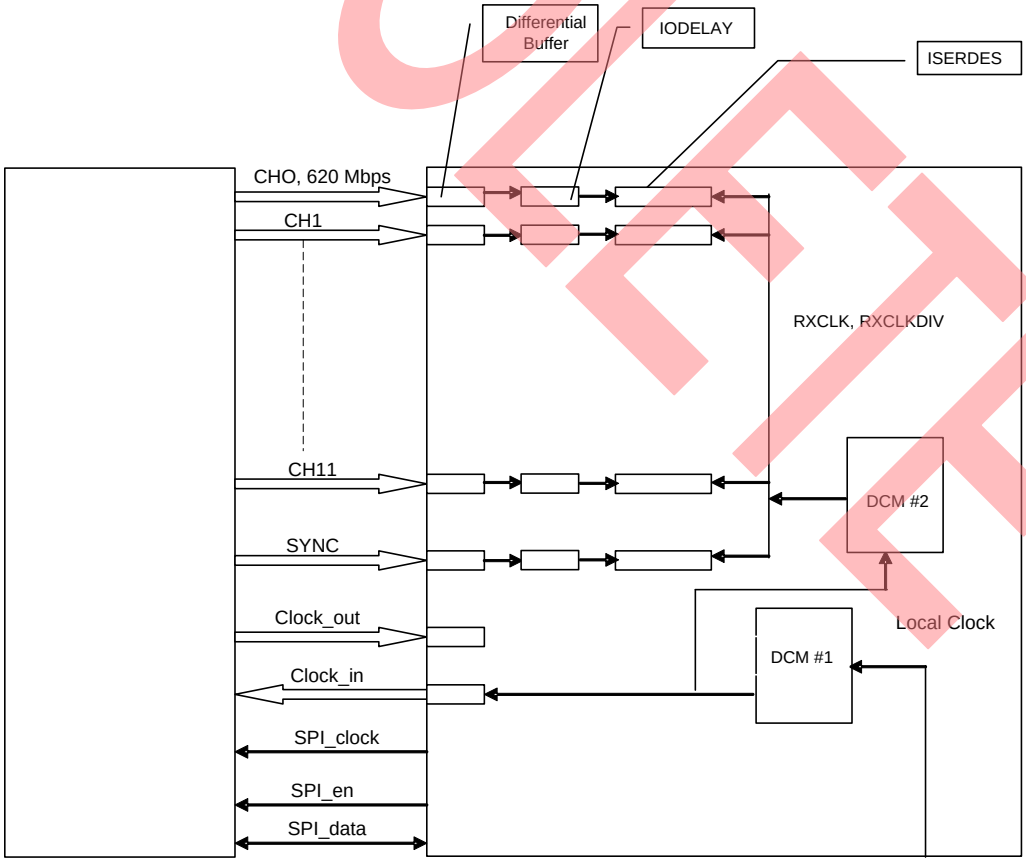
Training

To support per-pin skew compensation, the sensor provides a training mode. In this mode, a fixed code is transmitted that allows the Virtex-4/5 FPGA to recover data on all parallel channels and synchronize them. At power on, the sensor keeps sending the training pattern (which is 0x32A by default, but can be customized) on all channels including the sync channel. The receiver should synchronize and lock all the channels for proper data latching by performing bit and word alignment techniques.

When the host achieves the lock, it enables the sequencer through a SPI upload and initiates image capture.

Interfacing Block Diagram

Figure 2. Block Diagram Showing Interface between Sensor and FPGA



Bit and Word Alignment Techniques

The data on the high speed serial link can drift due to reasons such as Mesochronous clocking system (skew), jitter, PCB trace delays, and process, voltage, and temperature (PVT) variations. The receiver performs per-pin skew compensation using bit and word alignment techniques for every channel.

Bit and word alignment techniques are implemented using ISERDES components (ChipSync). This is available in every I/O of Xilinx' Virtex-4/5 devices.

Bit Alignment: Positioning the sampling edge of the clock at the center of the data eye window by adding delay (using Delay Taps) to the data path.

Word Alignment: Ensuring that the parallel data bits are in correct order at the output of the ISERDES by using the BitSlip function available in the ISERDES.

Note Refer to Xilinx application note XAPP855 for implementation details on bit and word alignment.

OBSOLETE

Differential Buffer: Converts the differential data to single ended data in FPGA.

IODELAY: Controls the delay to be introduced in data path using Delay_Taps value. 1 Tap=75 ps at 200 MHz reference frequency.

ISERDES: Available in every I/O of the Virtex-4/5 device; converts serial data into 10-bit parallel data output after performing bit and word alignment.

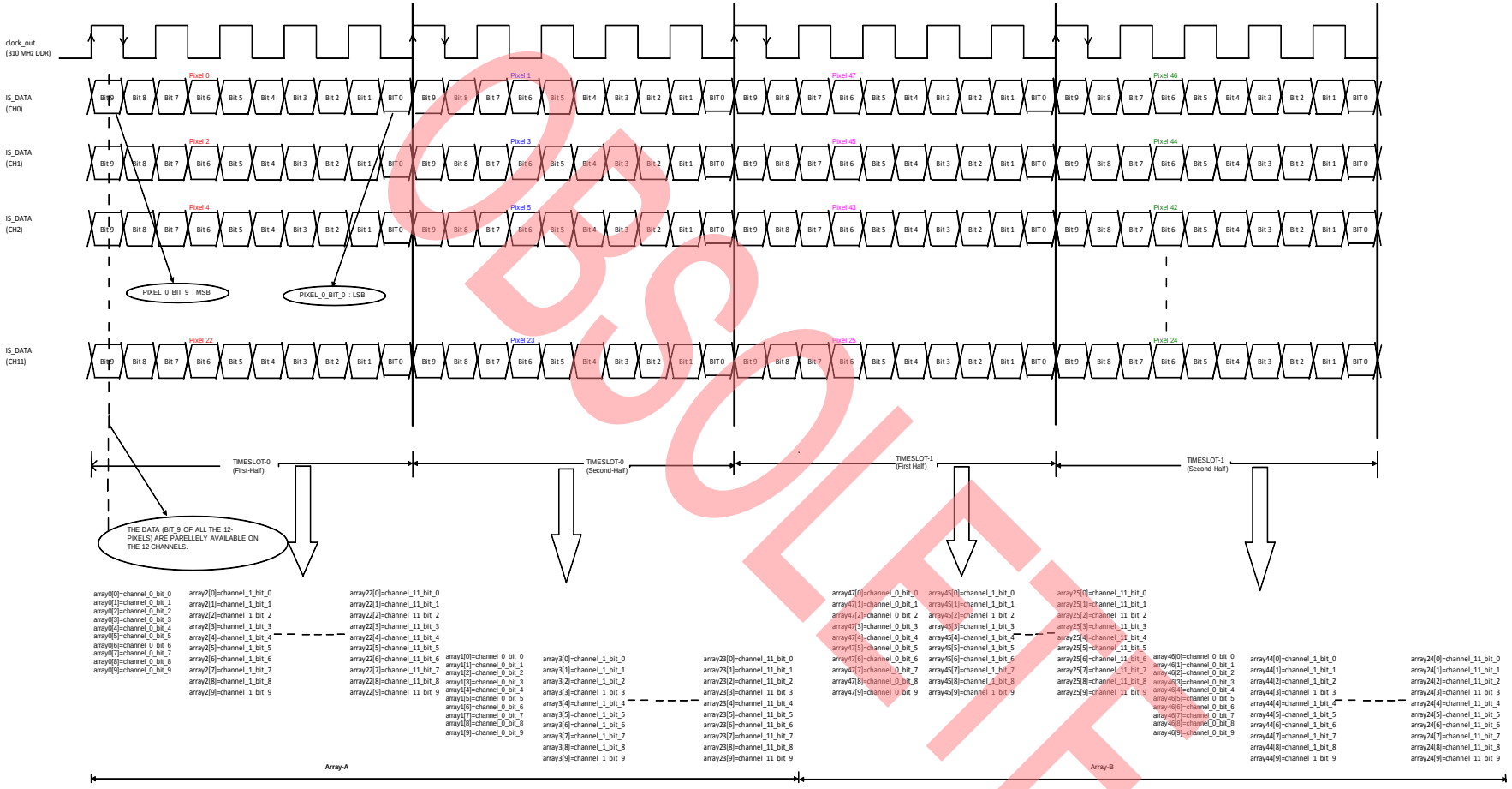
DCM #1: Generates the image sensor operating clock of 300 MHz from an external input clock (33 MHz) from a clock chip.

DCM #2: Generates both latching clocks (RXCLK (300 MHz) and RXCLKDIV (60 MHz)) for ISERDES either by using a local clock or image sensor clock_out.

The sensor sends back 300 MHz clock, which is used to latch the data inside the receiver (FPGA) because it has a fixed phase relationship with the data channels. However, the demonstration system does not use it, because bit and word alignment circuitry can work on local clock of 300 MHz.

OBSOLETE

Figure 3. Timing Relationship between Sensor Clock and Data Bits



Refer the Excel file accompanying this application note for the complete timing diagram.

OBSOLETE

The data bit on a channel is almost edge aligned with the clock given out by the image sensor.

The data on the LVDS channels is sent out MSB first. Each pixel data is represented by 10-bits. Therefore, after every five sensor clocks, the receiver gets 12-pixels in parallel through the12 data channels.

The sequence of pixels from the sensor varies depending on the timeslot, as explained in Table 1.

Table 1. Pixel Sequence

Timeslot	ch0	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	ch9	ch10	ch11
Timeslot-0 (First half)	0	2	4	6	8	10	12	14	16	18	20	22
Timeslot-0 (Second half)	1	3	5	7	9	11	13	15	17	19	21	23
Timeslot-1 (First half)	47	45	43	41	39	37	35	33	31	29	27	25
Timeslot-1 (Second half)	46	44	42	40	38	36	34	32	30	28	26	24

In the demonstration system, two sets (Array-A and Array-B) of local registers, each 240 bits wide, are used to store 24-pixels data.

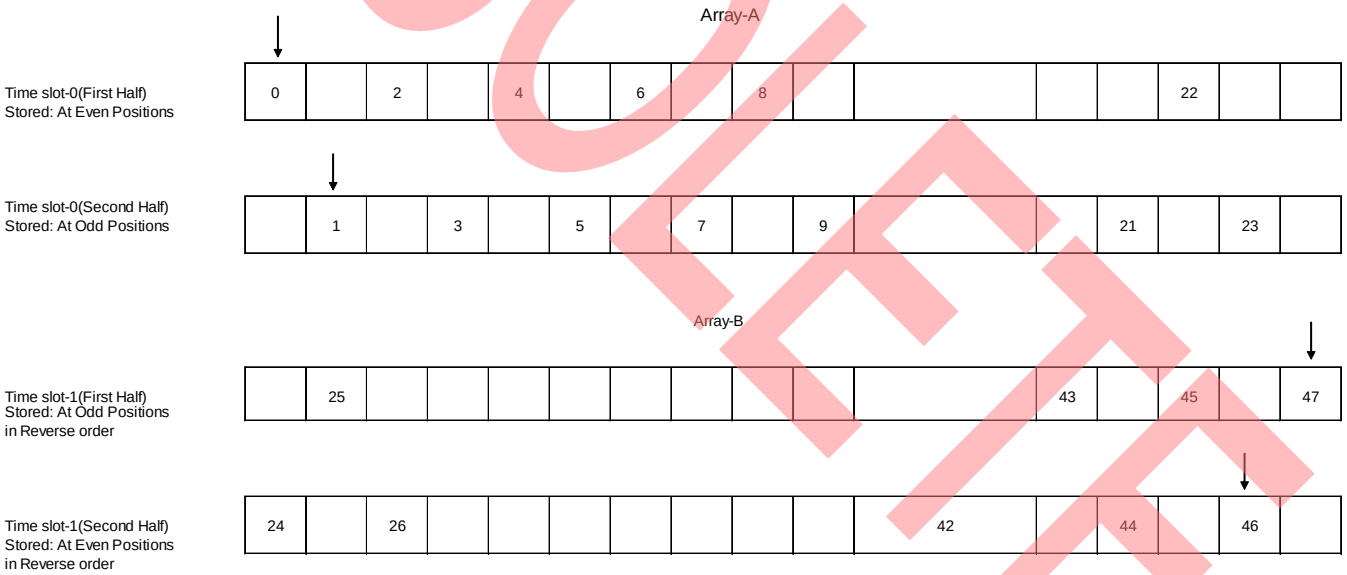
Timeslot-0 (First half) Array-A Even positions (0, 2, 4, 6… 22)

Timeslot-0 (Second half) Array-A odd positions (1, 3, 5, 7…23)

Timeslot-1 (First half) Array-B odd positions (25, 27… 45, 47); odd pixels in reverse order

Timeslot-1 (Second half) Array-B even positions (24, 26… 44, 46); even pixels in reverse order

Figure 4. Data Storing in Local Registers of FPGA



After every timeslot, 24 pixels are received, which can be rearranged in the correct order.

During reverse X/Y and subsampling mode, the timing relationship between the Clock_Out and data on the channels remain the same. However, the order of the pixels coming from the sensor changes, as shown in the following tables.

OBSOLETE

Table 2. Sequence of Pixels During Reverse X/Y Readout Operation

Timeslot	ch0	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	ch9	ch10	ch11
54a[1]	1295	1293	1291	1289	1287	1285	1283	1281	1279	1277	1275	1273
54b[2]	1294	1292	1290	1288	1286	1284	1282	1280	1278	1276	1274	1272
53a	1248	1250	1252	1254	1256	1258	1260	1262	1264	1266	1268	1270
53b	1249	1251	1253	1255	1257	1259	1261	1263	1265	1267	1269	1271
...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...
2a	47	45	43	41	39	37	35	33	31	29	27	25
2b	46	44	42	40	38	36	34	32	30	28	26	24
1a	0	2	4	6	8	10	12	14	16	18	20	22
1b	1	3[3]	5	7	9	11	13	15	17	19	21	23
CRC												

Notes
1. First half of 54th timeslot
2. Second half of 54th timeslot
3. Pixel-3

Table 3. Sequence of Pixels During Subsampling (Monochrome) Mode of Operation

Timeslot	ch0	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	ch9	ch10	ch11
1a[4]	0	2	4	6	8	10	12	14	16	18	20	22
1b[5]	46	44	42	40	38	36	34	32	30	28	26	24
2a	48	50	52	54	56	58	60	62	64	66	68	70
2b	94	92	90	88	86	84	82	80	78	76	74	72
3a	96	98	100	102	104	106	108	110	112	114	116	118
3b	142	140	138	136	134	132	130	128	126	124	122	120
4a	144	146	148	150	152	154	156	158	160	162	164	166
...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...
27a	1248	1250	1252	1254	1256	1258	1260	1262	1264	1266	1268	1270
27b	1294	1292	1290	1288	1286	1284	1282	1280	1278	1276	1274	1272
CRC												

Notes
4. First half of 1st timeslot
5. Second half of 1st timeslot

Table 4. Sequence of Pixels During Subsampling (Color) Mode of Operation

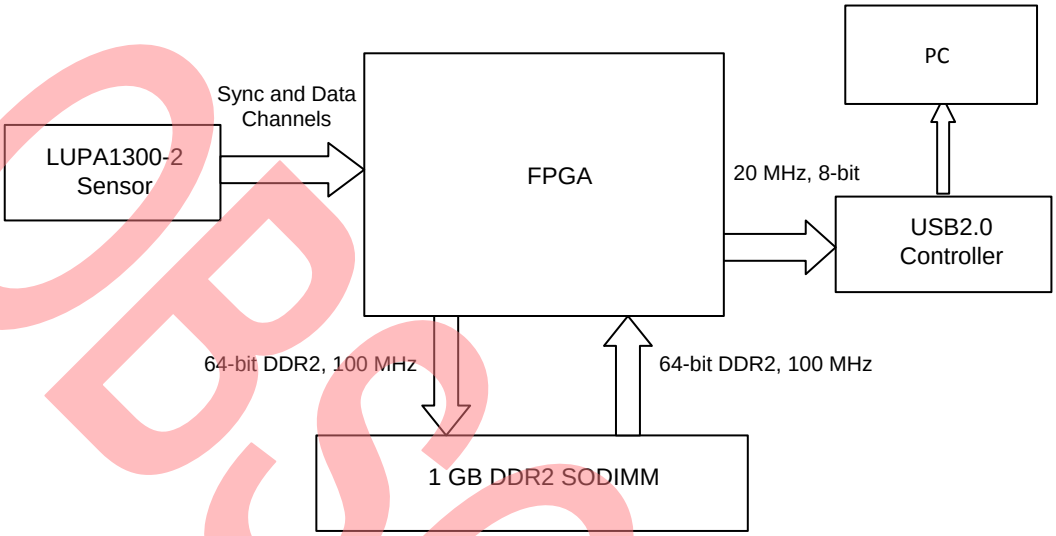
Timeslot	ch0	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	ch9	ch10	ch11
1a	0	45	4	41	8	37	12	33	16	29	20	25
1b	1	44	5	40	9	36	13	32	17	28	21	24
2a	48	93	52	89	56	85	60	81	64	77	68	73
2b	49	92	53	88	57	84	61	80	65	76	69	72
...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...
27a	1248	1293	1252	1289	1256	1285	1260	1281	1264	1277	1268	1273
27b	1249	1292	1253	1288	1257	1284	1261	1280	1265	1276	1269	1272
CRC												

OBSOLETE

LUPA1300-2 Demonstration Kit Implementation

LUPA1300-2 demonstration kit is developed to showcase the features and capabilities of the image sensor such as full frame readout, ROI, subsampling, reverse-X/Y, and multiple window. In this system, the sensor operates at 300 MHz. All channels are synchronized and locked after performing the bit and word alignment for every channel separately and independently using the training pattern (0x32A) sent by the sensor.

Figure 5. Demonstration System Block Diagram

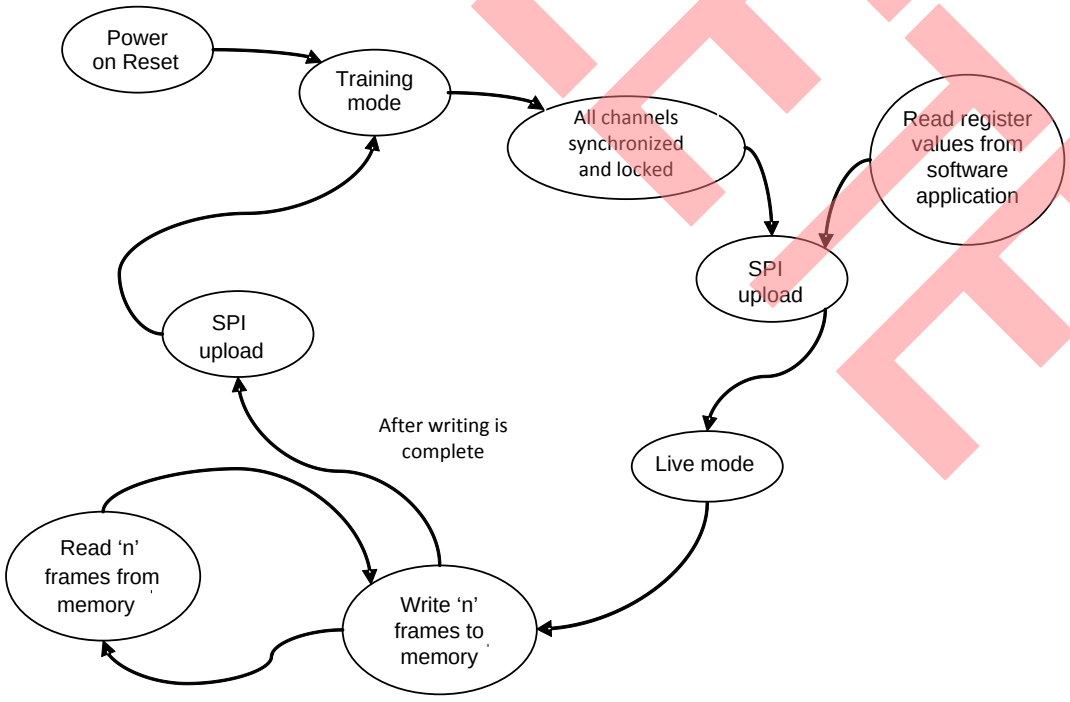


FPGA: XC5vLX50T – 1FF1136

The live pixel data is stored in external memory (1 GB SODIMM) with the help of a DDR2 SDRAM controller. 400 continuous frames are stored (though the image sensor sends 500 FPS at 315 MHz) at full resolution in 1 GB module. But at lower resolution (VGA, ROI), up to 1600 continuous frames can be stored in the 1 GB module.

The data is stored into the external memory in real time, but data read out from the memory is done with a much slower clock of 20 MHz because of USB2.0 protocol limitations.

Figure 6. State Diagram Showing Sequence of Steps Followed in Demonstration System



OBSOLETE

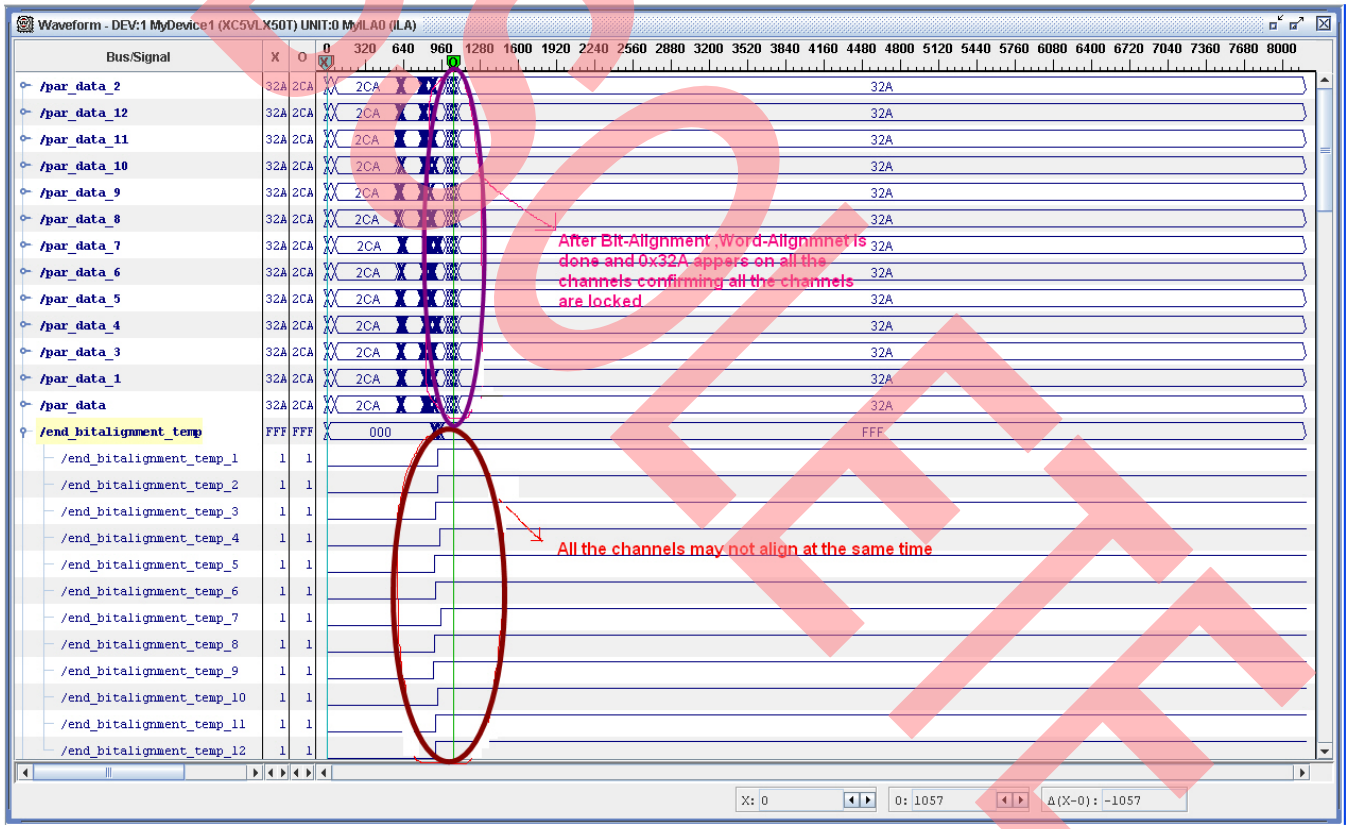
Explanation

1. At power on, image sensor continues to send the training pattern (0x32A) for the receiver to synchronize and lock the channels.
2. After locking the receiver channels, the new settings in the demonstration kit application software are loaded. The sequencer is enabled for live image capture through SPI.
3. 'n' continuous frames are stored in the external memory of 1 GB SODIMM. Maximum n is 400 for 1 GB at full resolution.
4. When the writing is complete, then frame readout is performed at the rate of 20 MHz (USB2.0).
5. While stored frames are being read out from the memory, the sequencer is disabled through SPI so that the training pattern comes on all the channels and the receiver starts synchronizing all the channels again. This way, the system counters against any drift on the channels due to skew, jitter, trace delays, and process, voltage and temperature (PVT) variations.

After locking all the channels, it goes back to Step 2 and the sequence repeats.

In the demonstration system, the channels are synchronized and locked before capturing a burst of continuous frames. Synchronize the channels dynamically (preferably before the capture of every burst of frames); otherwise, the system loses the synchronization over a period of time due to drift in the channels.

Figure 7. Chipscope Result for 10-Bit Parallel Data Out at the Output of ISERDES During Training Mode



PROFIT

Recommendations

- It is advisable to keep the image sensor in reset until the stable clock is applied to it.
- When generating the image sensor clock by using DCM/PLL, wait for the 'LOCK' signal of the clock synthesizer to go high ensuring a stable and clean clock given to the sensor. Until then, the image sensor can be in reset mode.
- According to the LUPA1300-2 internal architecture, the SPI_clock should be less than (1/30) of the image sensor operating frequency.
  SPI_clock < (1/30) * Image sensor_clock.
- When uploading the internal registers of the sensor through SPI, wait for the present frame readout to complete. Disable the sequencer and load all the required registers. At the end, enable the sequencer to send the data with new settings.
- Implement bit and word alignment using ISREDES (ChipSync Technology) available in every I/O of Vitex-4/5 devices.
  Note To implement a design in other vendor FPGAs such as Altera, consider the ISREDES counterpart, which performs a similar operation.
- The minimum frequency at which the bit and word alignment algorithm works is 206 MHz with IDELAY reference frequency of 200 MHz. (for more information refer Xilinx's application note XAPP855).
- All FPGA channels may not lock at the same time. The locking period of individual channel depends upon the amount of drift and variation present in that channel.
- LVDS works on 2.5 volts supply voltage with common mode voltage of 1.2 volts and pk-pk swing of 700 mV across a100 ohm termination resistor at the input of the receiver (FPGA).
- Always instantiate a 100 ohm termination resistor across the p and n differential pair using correct I/O standards inside FPGA. This is better than connecting it externally across the pins, which saves lot of board space and makes the system small.
- All data channels and clocks are differentially terminated logic. Probe p and n channel at the receiver end, preferably after the termination resistor. Note that when probing between the Tx (IS) and Rx (FPGA), signals may have lower swings, but at the receiver end the value may be correct.
- For all the differential signals the p and n trace length should match correctly. The Tx (IS) and Rx (FPGA) should also be placed close to each other as 1 inch of trace length introduces 150 to 180 ps delay for FR4 laminate Microstrip with 50 ohm characteristic impedance.
- Maintain all voltages at the pins of the image sensor as listed in the data sheet. Use linear regulators for sensor power supplies.
- Firmly fix the image sensor to the board; otherwise, the channel data is corrupted. Even a small shake corrupts the data to such an extent that the FPGA channels may not lock.
- Before enabling the image capture, all 13 channels should be locked. Otherwise, artifacts can be observed in the image.
- Dynamically synchronize and lock the channels to counter any drift and variations occurring on the channels. Otherwise, the system loses the synchronization over a period of time.
- Many macros such as ISERDES, ODDRs, and IDDRs are used when implementing bit and word alignment algorithm and DDR2 SDRAM memory controller interface, which generate a lot of heat during its operation. A proper heat sink or fan must be used in the system to dissipate the extra heat generated.

Summary

This application note emphasizes the interface between the image sensor (transmitter) and the FPGA (receiver). It describes the timing relationship between the sensor clock and data channels. The synchronization and locking of all channels are achieved using the training pattern (0x32A) sent by the sensor and implementing the bit and word alignment technique. The order of the pixels is based upon the timeslot and the mode of operation of the sensor. It also gives an overview of demonstration system architecture and the method implemented for sensor data latching.

PROFILE

About the Author

Name: Anoop Kumar Upadhyay.

Title: Sr. Systems Engineer (Logic Design)

Background: Bachelors in Electronics and Communication and Masters in Embedded System Design and VLSI.

Contact: uua@cypress.com

Document History

Document Title: Interfacing LUPA1300-2 with FPGA - AN54468

Document Number: 001-54468

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2734041	UUA	07/09/09	New application note
*A	4303526	EYB	03/11/2014	Obsolete document. Completing Sunset Review.

OBSOLETE

All trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com/

© Cypress Semiconductor Corporation, 2009-2014. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.