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THIS SPEC IS OBSOLETE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: 001-54598

Spec Title: PIXEL REMAPPING IMPLEMENTATION IN LUPA1300-2 DEMONSTRATION KIT - AN54598

Sunset Owner: Evelyn Beard (EYB)

Replaced by: None

Pixel Remapping Implementation in LUPA1300-2 Demonstration Kit

AN54598

Author: Anoop Kumar Upadhyay
Associated Project: No
Associated Part Family: CYIL2SM1300AA
Software Version: None
Associated Application Notes: AN54468

OBSOLETE

Application Note Abstract

This application note explains the remapping technique implemented in LUPA1300-2 demonstration system to align non consecutive pixels readout to consecutive pixels for creating proper images. It also describes the remapping method during Reverse-X/Y readout and Subsampling modes of operation.

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, reading out 500 frames per second at full resolution. The readout can be boosted by enabling subsampling and Region of Interest (ROI).

The LUPA1300-2 is based on 6T-pixel architecture and works in snapshot pipelined shutter mode. (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). Therefore, for a nominal speed of 310 MHz, bit rate is 620 Mbps. Data can thus be transferred over a longer distance simplifying the surrounding system while maintaining low noise and distortion.

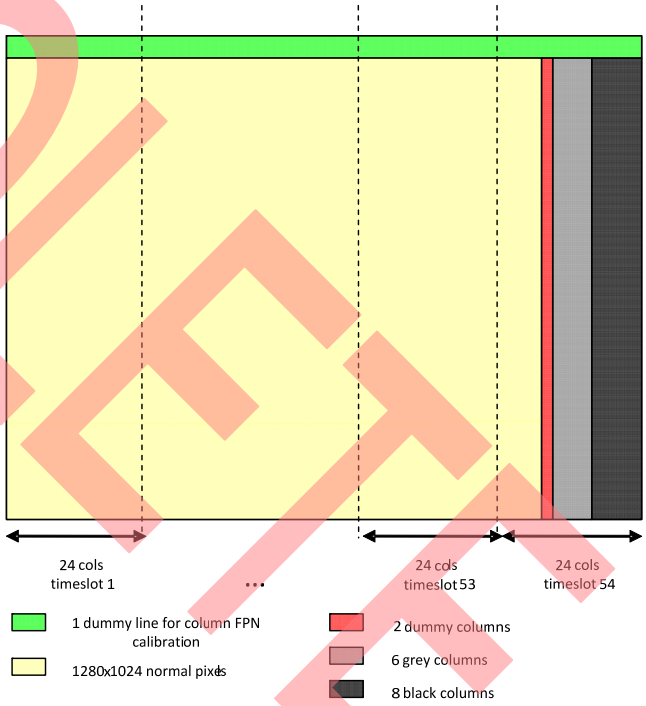
The sync channel gives all the synchronization codes for the receiver, indicating Frame Start (FS), Line Start (LS), Line End (LE), ROT, FOT, Frame End (FE), image data, CRC, and training pattern.

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 as shown in Figure 1.

1 Timeslot = 1 Kernel = 24 Pixels Equation 1

Total Timeslots = 54 Equation 2

Figure 1. Sensor Readout Format



Sensor Interface

The host (FPGA) sends the LVDS input clock to the sensor. Sensor logic works with the master clock and generates all the daughter clocks internally. To compensate for possible large on-chip delays, the sensor transmits back the LVDS clock with the same delay that the data suffers. The sensor's serial link interface is based on Mesochronous clocking scheme (all parallel links operate at the same frequency, but their phase can be different (less than 150 ps)).

The data coming on high speed serial links 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 on per channel basis.

In FPGA, all the LVDS pairs are terminated and passed through differential buffers followed by ISERDES blocks for serial to parallel conversion

The data bits on a channel are almost edge aligned with the clock sent out by the sensor. The data on the LVDS channels are sent out MSB first. Bit alignment algorithm is used to center align the latching clock edge at the middle of the data eye window. Word alignment algorithm is responsible for generating a properly aligned 10-bit word from incoming serial data.

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

Note If the design requires other vendors' FPGA such as-ALTERA and ACTEL, look for the corresponding ISERDES counterpart.

Each pixel data is represented by 10 bits and is available at the output of ISERDES after performing bit and word alignment. Therefore, after every five sensor clocks, the receiver gets 12 pixels in parallel through 12 LVDS data channels.

Refer Cypress application note AN54468 for more information on bit and word alignment techniques.

Pixel Readout Sequence

To increase the performance of the sensor, the order of pixels varies depending on the timeslots and working mode of the sensor as shown in the table below.

Table 1. Overview of Pixel Sequence for One Full Frame Readout

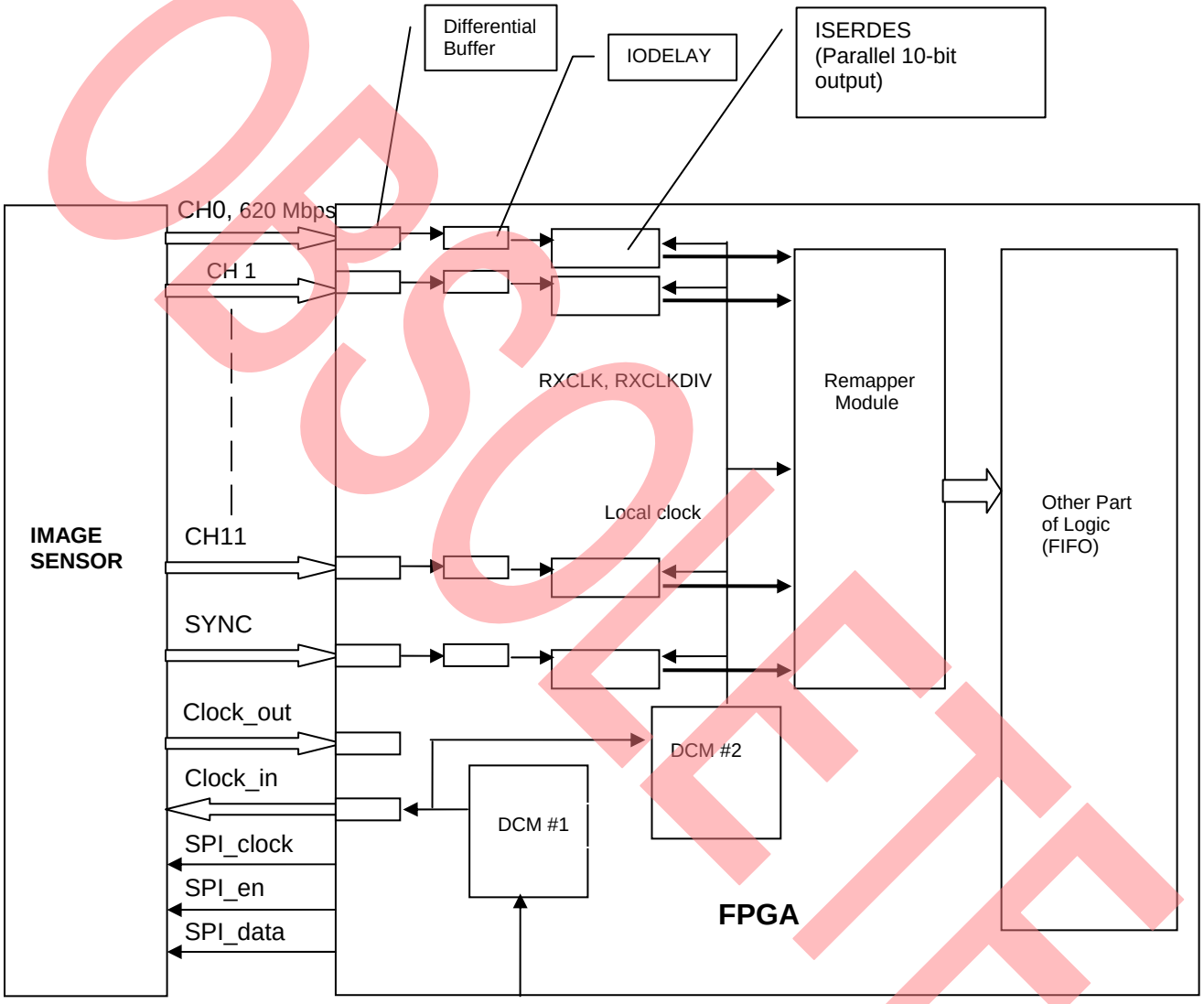
timeslot	ch0	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	ch9	ch10	ch11
1a	0	2	4	6	8	10	12	14	16	18	20	22
1b	1	3	5	7	9	11	13	15	17	19	21	23
2a	47	45	43	41	39	37	35	33	31	29	27	25
2b	46	44	42	40	38	36	34	32	30	28	26	24
3a	48	50	52	54	56	58	60	62	64	66	68	70
3b	49	51	53	55	57	59	61	63	65	67	69	71
4a	95	93	91	89	87	85	83	81	79	77	75	73
4b	94	92	90	88	86	84	82	80	78	76	74	72
5a	96	98	100	102	104	106	108	110	112	114	116	118
5b	97	99	101	103	105	107	109	111	113	115	117	119
6a	143	141	139	137	135	133	131	129	127	125	123	121
6b	142	140	138	136	134	132	130	128	126	124	122	120
7a	144	146	148	150	152	154	156	158	160	162	164	166
7b	145	147	149	151	153	155	157	159	161	163	165	167
8a	191	189	187	185	183	181	179	177	175	173	171	169
8b	190	188	186	184	182	180	178	176	174	172	170	168
9a	192	194	196	198	200	202	204	206	208	210	212	214
9b	193	195	197	199	201	203	205	207	209	211	213	215
10a	239	237	235	233	231	229	227	225	223	221	219	217
10b	238	236	234	232	230	228	226	224	222	220	218	216
11a	240	242	244	246	248	250	252	254	256	258	260	262
11b	241	243	245	247	249	251	253	255	257	259	261	263
12a	287	285	283	281	279	277	275	273	271	269	267	265
12b	286	284	282	280	278	276	274	272	270	268	266	264
...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...
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
54a	1295	1293	1291	1289	1287	1285	1283	1281	1279	1277	1275	1273
54b	1294	1292	1290	1288	1286	1284	1282	1280	1278	1276	1274	1272
CRC												

OBSOLETE

Remapping Method Implemented in FPGA

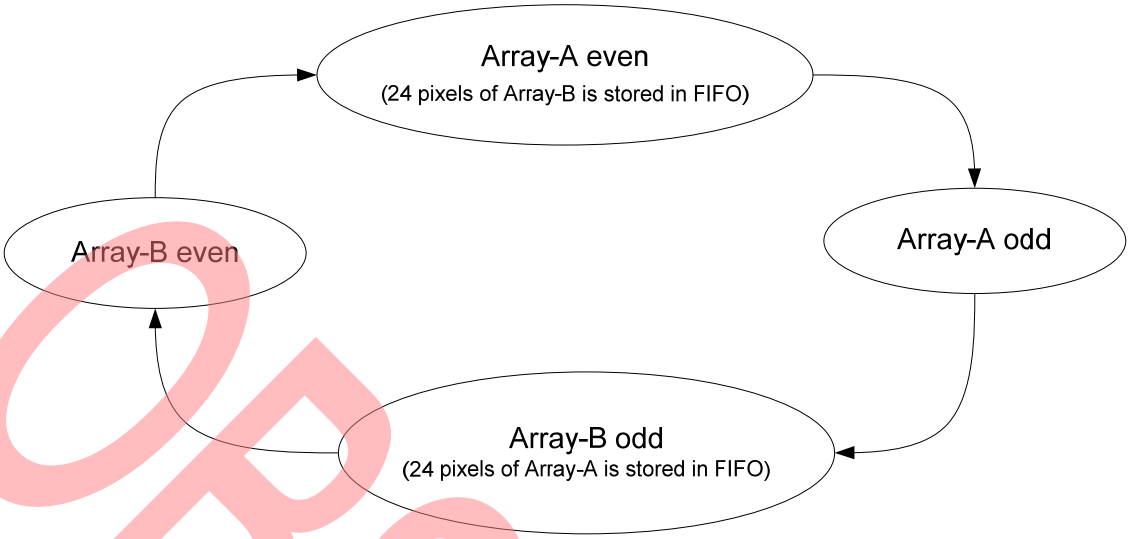
After bit and word alignment, the 10 bit parallel output data from ISEREDS is passed through the remapper logic block to align the pixels consecutively. After every timeslot, 24 pixels are received; these are stored in two sets (Array-A and Array-B) of local FPGA registers, each 240-bit wide. This is illustrated with the help of block diagram in Figure 2 and state diagram in Figure 3.

Figure 2. Block Diagram Describing Remapping after Bit and Word Alignment



OBSOLETE

Figure 3. State Diagram showing Intermittent Storage and Handling to Achieve Consecutive Pixels in a Frame



OBSOLETE

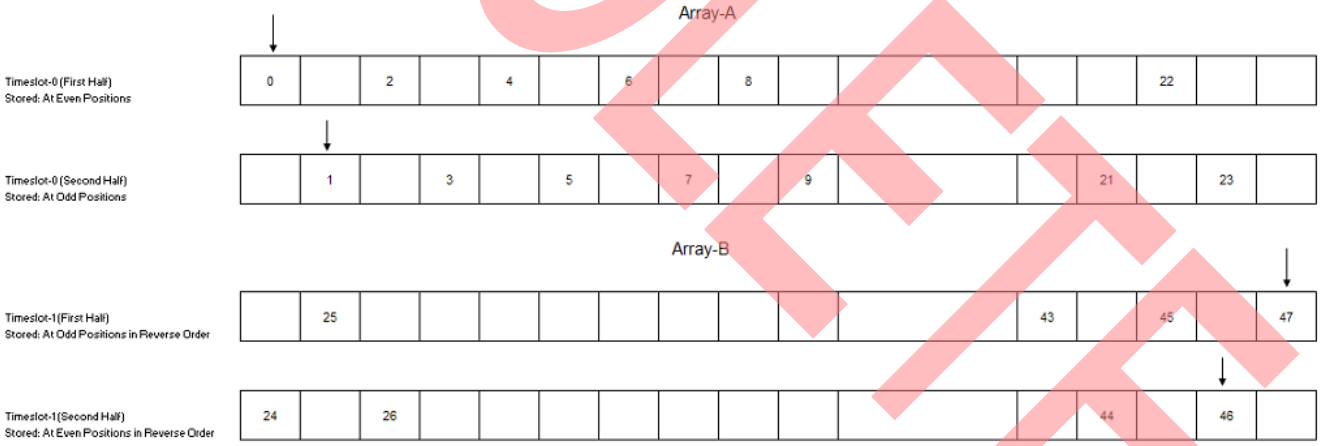
Timeslot-0 (First half) : Array-A Even positions (0, 2, 4, 6,…, 22)

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) in reverse order

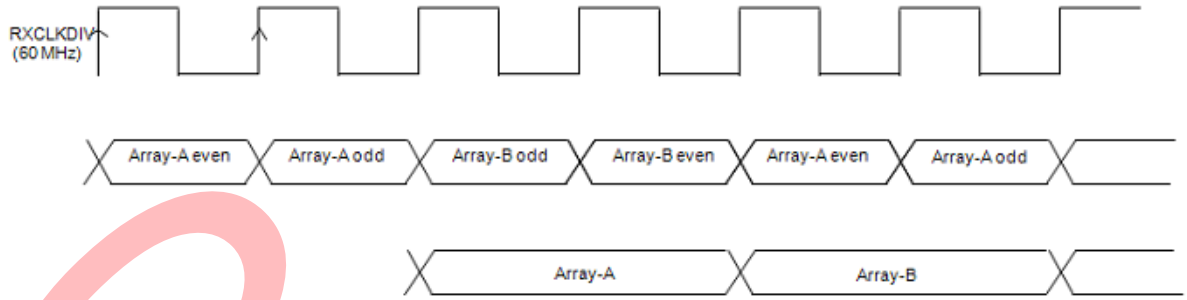
Timeslot-1 (Second half) : Array-B even positions (24 , 26 , …, 44 , 46) in reverse order

Figure 4. Sequence of Pixels Stored in Registers



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

Figure 5. Generation of 24 Pixels after Every Timeslot



During reverse-X/Y and subsampling modes of operation, the timing relationship between the clock out and data on the channels remains the same. However, the order of pixels coming out from the sensor changes as shown in the following tables.

Reverse-X/Y Readout

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	1295	1293	1291	1289	1287	1285	1283	1281	1279	1277	1275	1273
54b	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	5	7	9	11	13	15	17	19	21	23
CRC												

Both the pixel and line order is reversed during the sensor's reverse-X/Y mode. However, the pixels are remapped in the same manner as described earlier.

Subsampling Mode of Operation

Table 3. Sequence of Pixels during Subsampling (Monochrome) Mode

Timeslot	ch0	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	ch9	ch10	ch11
1a	0	2	4	6	8	10	12	14	16	18	20	22
1b	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												

OBSOLETE

During subsampling mode of operation, the pixels are stored in only two sets of registers, each 120 bits. This is explained in the state diagram in Figure 6.

Figure 6. State Diagram Describing Pixel Storage during Subsampling (Monochrome) Mode

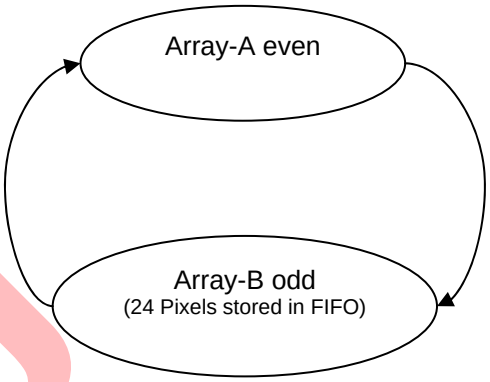


Table 4. Sequence of Pixels during Subsampling (Color) Mode

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												

Subsampling (Color) mode is not implemented in the demonstration system, but pixel data can be stored in a similar manner.

Summary

This application note describes the remapping method implemented in LUPA1300-2 demonstration system by using two sets of registers (Array-A and Array-B). The sequence of pixel readout during Normal, Reverse-X/Y, and Subsampling modes of operation is also explained.

OBSOLETE

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: Pixel Remapping Implementation in LUP1300-2 Development Kit

Document Number: 001-54598

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2741951	UUA	07/21/09	New application note
*A	4238109	CFT	01/07/14	Obsolete spec.

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. 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.

OBSOLETE