# IN ORBIT RE-PROGRAMMABILITY OF DIGITAL SUBSYSTEMS OF SATELLITE

<sup>1</sup>Miral Patel

<sup>1</sup>Student <sup>1</sup>Electronics and Communication, <sup>1</sup>VGEC, Chandkheda, Ahmedabad, India

*Abstract:* Every Spacecraft/ Satellite has many digital subsystems which are widely used for a various kind of controlling and data processing applications. Field Programmable gate arrays are one of the most common elements of these subsystems considering the last minute programmability. Presently, One Time Programmable (OTP) and SRAM based FPGAs are used in the space programs. Once the functionality of FPGA is fixed at ground, then it cannot be altered in space. Re-Programmability of these devices would be useful to fix the bugs, change the functionalities or addition of the new functionalities in due time course resulting in extension of mission life. Here in this project the concept of the full Re-Programmability and partial Re-Programmability of SRAM based FPGA as a part of the satellite or space craft would be demonstrated. The concept of Re Programmability of FPGA devices in orbit is achieved using rewritable program memory, re programmable rad-hard FPGA devices and a self developed unique protocol as part of the main logic/program.

# *Index Terms* - Re-Programmability; SRAM based FPGA; Space Applications; Rad-hard FPGA; One Time programmable FPGA

# I. INTRODUCTION

Digital subsystems are widely used for various kinds of controlling and data processing applications. FPGAs are very basic and common element of the digital subsystem of Satellite. There are two types of FPGAs used in the space. One is OTP FPGAs and second one is SRAM based FPGA. SRAM based FPGAs are Re-Programmable. Once the functionality of the FPGA is fixed then we cannot change it after launch. Thus, we cannot use the feature of Re-Programmability of FPGA. Thus after launch, the satellite works only for the assigned task and addition of the functionality cannot be done. After launch if some bugs detected then the correction cannot be done.

Our main objective behind this research is to achieve Re-Programmability in Orbit. Here we project the solution to this problem of Re-Programmability by defining a unique protocol. This protocol will be supplied to satellite through tele-commands from the earth-stations. The decoder in the design will identify the string of 64-bit, which was supplied from the earth-station. After identification of the string the device will decode the 64-bit data. In that string the data along with its address will be given. This new data will be stored into memory. On power up the FPGA will work according to new program. This is how Re-Programmability will be achieved.

# II. WORK FLOW OF THE SYSTEM

Step 1: First step is the data with finite clock pulses will be supplied to the satellite. Data packets will be received from the ground.

**Step 2:** When the data will be supplied, it should be identified by our device. So that we come to know that the data is for our device or not. As it is defined in the protocol that first 16-bit of the data are for identification. The tele-command decoder in the design will identify the code from the supplied data. If the identification fails then do nothing with the existing file and continue with the existing program file.

Step 3: After identification the process of CRC check will be done. So that we can make decisions further. Once the CRC check is done then the decoder will store the data at particular location. Last four bits are of CRC bits. If the CRC bits are not same then the user will request to resend the data with CRC failed information.

Step 4: When this process of identification and CRC checking is completed, the device will wait till the end of the file. According to the given data the contents of the memory will be updated.

Step 5: The data will be stored into the specified location of the memory. After that again the CRC check will be done. If the CRC bits matches then the device will send telemetry command that "Received Successful". And if it fails then the device will request to resend the data via telemetry.

Step 6: After that device will wait for the program command. Then this data will be copied to the program memory. On reboot command FPGA will be configured through new program. This is how functionality changes.

# **III. READ AND WRITE IN MEMORY**

There are some device commands which we have to follow to interact with the memory which we want to use. By using these we can read, write and erase the memory.

First we will see write operation. To write into memory first of all we have to erase the blocks. After erasing it, the 7<sup>th</sup> bit of the status register should be checked. If it is one then we can do writing operation. Erase the block Erase operation is mandatory for

programming the memory. For that two write cycles are needed. One write cycle is for setup and another is for confirmation. The first cycle will send 20h to data bus. After that D0h will be given to data bus for the confirmation of the erase operation. According to the device commands we have to write 40h in data bus to give command of writing. And in second write cycle the data which you want to write should be written.

Reading of the memory also requires two cycles. In first cycle FFh to the data bus should be given. And in second cycle the data is captured.

#### IV. PROTOCOL AND TELE-COMMAND DECODER

Bit classification and length of the protocol is given in this section. This protocol is defined. As well as the tele-command decoder is also designed to decode the protocol in Orbit.

The length of the protocol is 64-bit. The classification of bits is shown in the figure below. And also the waveforms of the telecommand decoder with its simulation are given below.

#### V. DETAILED DESIGN

The detailed design of the module is shown in the figures. This will contain tele-command decoder, top module and memory. Tele-command from the ground will be decoded in the tele-command decoder. After that the filtered data will go into top module of the design. According to the application the data will be stored into memory. Copy of the data will always present in the memory. So that if sometimes the data fails to store then that portion of data will be used as the information

for the FPGA. And FPGA will work as per the data.

The top module in the design will contain three basic blocks.

# VI. SIMULATIONS AND BLOCK DIAGRAM

								112.500 mil	
Name	Value	i0 me	120 mil	140.00	160 m	180 ma	1100 m	1120 me	1140 mi
12 output enable	Vanie	The state of the s	Citra da tara	a strate to the state of the st			Last an and the last	A PROPERTY AND A	the standard
We write enable			111			1			
addr_flash[21:1]	000000	X				000000			
The serout	1. Contraction (1997)						~		
► 🖬 data_fiasti135:0	2222	0 2222		0020	X		0000	Z	222
Line ethic	1	FLITT			C PT T				
Last_act_high	ō.								
14 dut_start	1					117 12			
<ul> <li>data_uart[15:0]</li> </ul>	2222	0				2222			
Les en_uart	0								
E ant_for_erase[4:0]	10		TXTXTXT	d XeXXX	X 4 X 10 X 11	XIIIXIIIXIIIX	(12X10X15X10X10		20
► w data[15:0]	2222	2112		0020	)7-	<u> </u>	0000		
Col. for statistical	0000000	and a second			900000			X0 X0 X0 X	0 X9 X0 X0 X
16 state complete									
18 status rea checked									
> with for writela:01	6	XX				0			
						500,000,24	5.450 ns		
Name	Value	500,000	0,180 ns 500,00	0,200 ns 500	,000,220 ns	500,000,240 ns	500,000,260 ns	500,000,280 ns 50	10,000,300 ns 500,0
output_enable	1								
write_enable	0								
addr_flash[21:1]	000000					00000			
le serOut	1								
data_flash[15:0]	0040		22222		C	0040	30	abcd	
1 cik	0								
1 rst_act_high	0								
dut_start	1								
<ul> <li>data_uart[15:0]</li> </ul>	ZZZZ					222			
1 en_uart	•								
cnt_for_erase[4:0]	20					20			
data[15:0]	0040		2222			0040	)z_(	abcd	2222
cnt_for_4[27:0]	5f5e101					515e101			
cnt_for_status[4:0]	19		14 15 16 17	(18)			19		
erase_complete	1	5 C							
status_reg_checked	1								
cnt_for_write[4:0]	5		0	X	1/2/3/4	DEDED Z	10 11 (1)	1 13 14 15 16	17 18 19 20
		X1: 500,000,245.	450 ns						

Figure 2. Write Operation



Figure 3. Read Operation

Identifier	Address	Data bits(16)	Pad0 or 1(4)	CRC bits(4)
bits(16)	bits(24)			

Figure 4. Protocol for the tele-decoder





Figure 5. Waveforms of the tele-decoder



									1,042.010 m						
Name	Value	0 fts		200 hs	+400 ns	1670 mi	800 ns	μ,	DO ns	1,290 m	1,400 ns	1,600 ns	1.800 ns	12,000 hs	2,100 rs
addr_mcs[23:0]	X000000000000						×	фос	100000000	20000000		And a state of the			
Idata mes(15:0)	XDXXXXXXXXXXXXXX								0000000000	000000					
Te resend data	1														
14 08	-ti-														
le ru n	1				1		100			California de la					12.82
The to clk	0	L T T	1 A A L	AAAA	<b>I</b> II I I I I I I I I I I I I I I I I I	LABAR	INGRO		REAL	Innan	Inna	<b>NAA</b> N	LINNI	Innan	INDI
là tc data	0	177												5	
Th stobe	0														
Image: Marging (63:0)	abode£12345678								abcdef1234	67813					
<ul> <li>W #35:01</li> </ul>	37	0000	DDDC	DEEE	00000		toogga	Тx					<u>*</u>	D	8 2 6
etc:_out3:0]	XXXX								30000						
shift_regiosoj	*********	×iii		Vertility and an		and the section land	and and and and	4.,	and the second second	And Ant. Sur. Sur.	Careford Tall And	and the star Name	San Yan Yan Yan Y	and the Second	a month
▶ M final_reg(620)	x0000000000000000000000000000000000000					the right reduced and the			000000000000000000000000000000000000000	DOCCOCK		and a second	ty free point of the		
> M cnt for 64 clk pulses	25	• •	123	)SC7.1	هموص	¢>>>>	10000	±X.			leee			<u>dece</u>	boex
the crt_signal	0														
ent for strobe(40)	e .	X							0	i					
Contraction of the other	189 - C														





Figure 8. Output of the tele-command decoder

#### REFERENCES

[1] Robert Giterman, Lior Atias, and Adam Teman, "Area and Energy-Efficient Complementary Dual-Modular Redundancy Dynamic Memory for Space Applications" IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 25, NO. 2, FEBRUARY 2017

[2] Xiaoxuan She and N. Li, "Reducing Critical Configuration Bits via Partial TMR for SEU Mitigation in FPGAs" IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 64, NO. 10, OCTOBER 2017

[3] Zhonghua Zhou, Kang Wang, Jiapeng Wu, Qunyang Wang, Dongbo Pei, "A High Reliability Design for Space-Borne Large-scale SRAM-Based FPGAs with Low Requirement of Memory Space" 2015 5th International Conference on Information Science and Technology (ICIST) April 24--26, 2015, Changsha, Hunan, China

[4] Sadia Ahmad, Minahil Zahra, Salma Zainab Farooq, Adnan Zafar, "Comparison of EDAC Schemes for DDR Memory in Space Applications" 2013 International Conference on Aerospace Science & Engineering (ICASE)

[5] Richard Dorrance, Student Member, IEEE, Fengbo Ren, Student Member, IEEE, Yuta Toriyama, Amr Amin Hafez, Student Member, IEEE, Chih-Kong Ken Yang, Fellow, IEEE, and Dejan Markovi'c, Member, IEEE, "Scalability and Design-Space Analysis of a 1T-1MTJ Memory Cell for STT-RAMs" IEEE TRANSACTIONS ON ELECTRON DEVICES, VOL. 59, NO. 4, APRIL 2012

[6] Jiaqi Yang, Hui Guo, Lei Chen, Yanlong Zhang, and Xuewu Li,, "SEU Sensitivity Evaluation of JTAG Circuit Used for SRAM-based FPGA" School of Microelectronics, Xidian University, Xian, 710071, China.Department of FPGA, Beijing Microelectronics Technology Institute, Beijing, China.

[7] Sarath Mohanachandran Nair, Rajendra Bishnoi, Mohammad Saber Golanbari, Fabian Oboril, Fazal Hameed and

Mehdi B. Tahoori, Senior Member, IEEE, "VAET-STT: Variation Aware STT-MRAM Analysis and Design Space Exploration Tool" IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

#### Book:

[1] Verilog HDL – Samir palnitkar

#### Website:

[1] www.xilinx.com

[2] www.wikipedia.org

[3] www.digikey.com