

# 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 tele-command 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

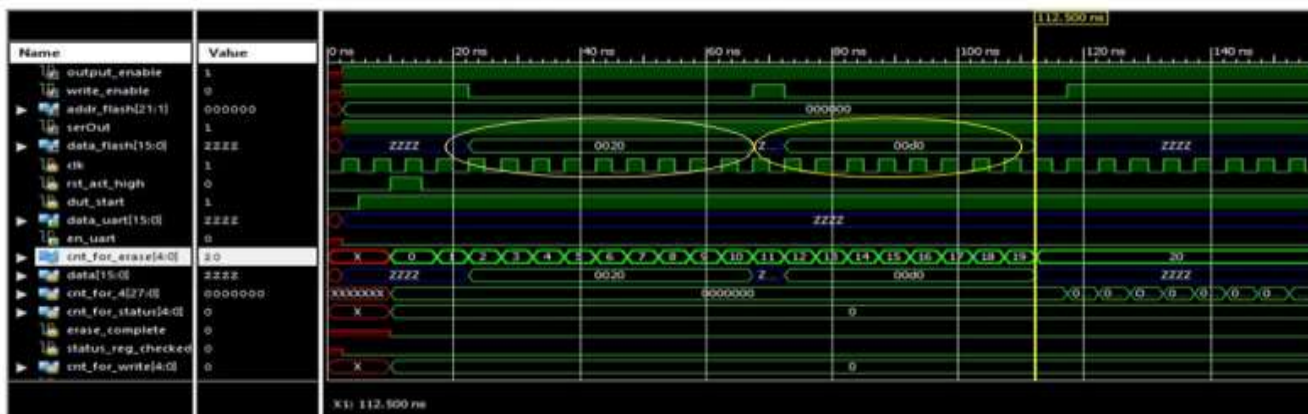


Figure 1. Erase Operation

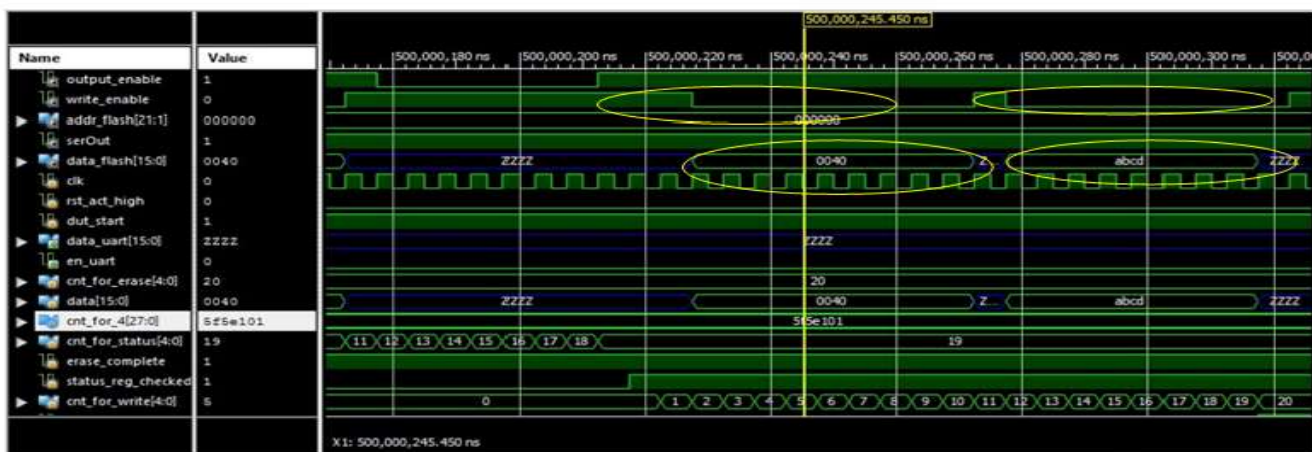


Figure 2. Write Operation

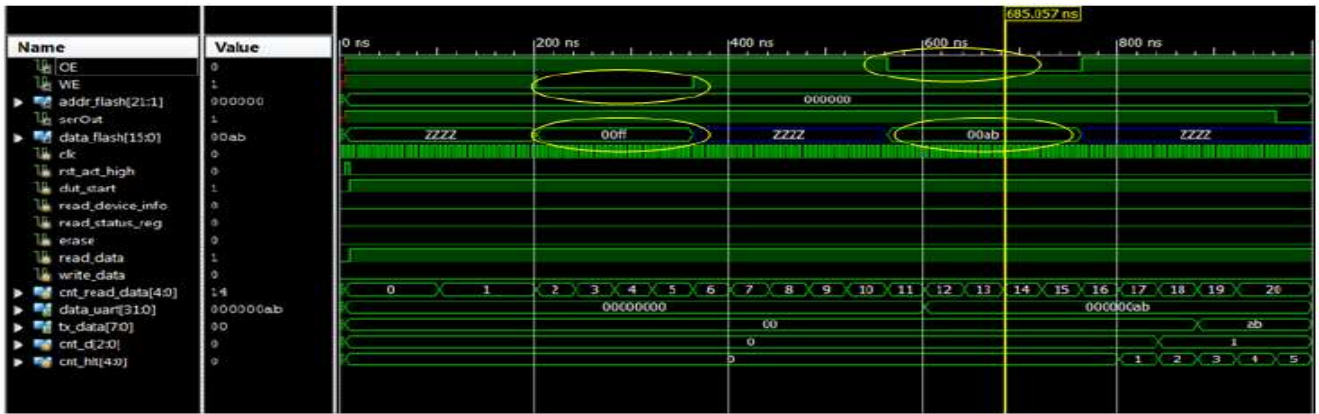


Figure 3. Read Operation

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

Figure 4. Protocol for the tele-decoder

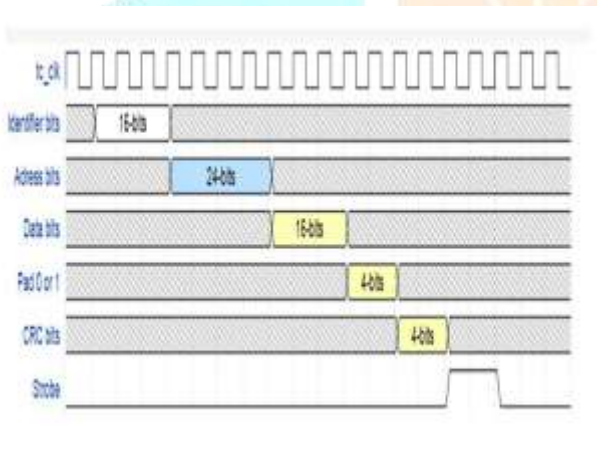


Figure 5. Waveforms of the tele-decoder

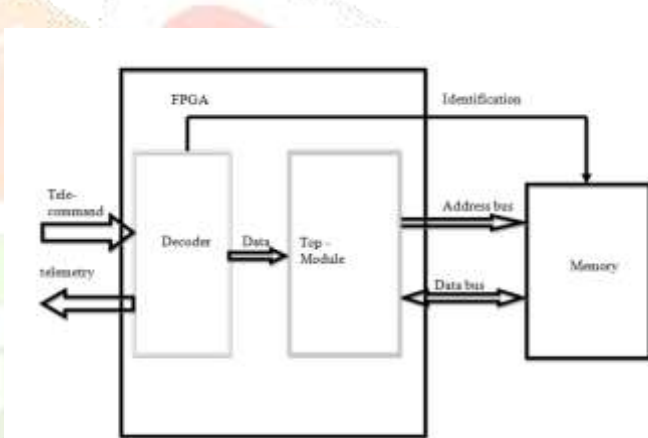


Figure 6. Detailed Design

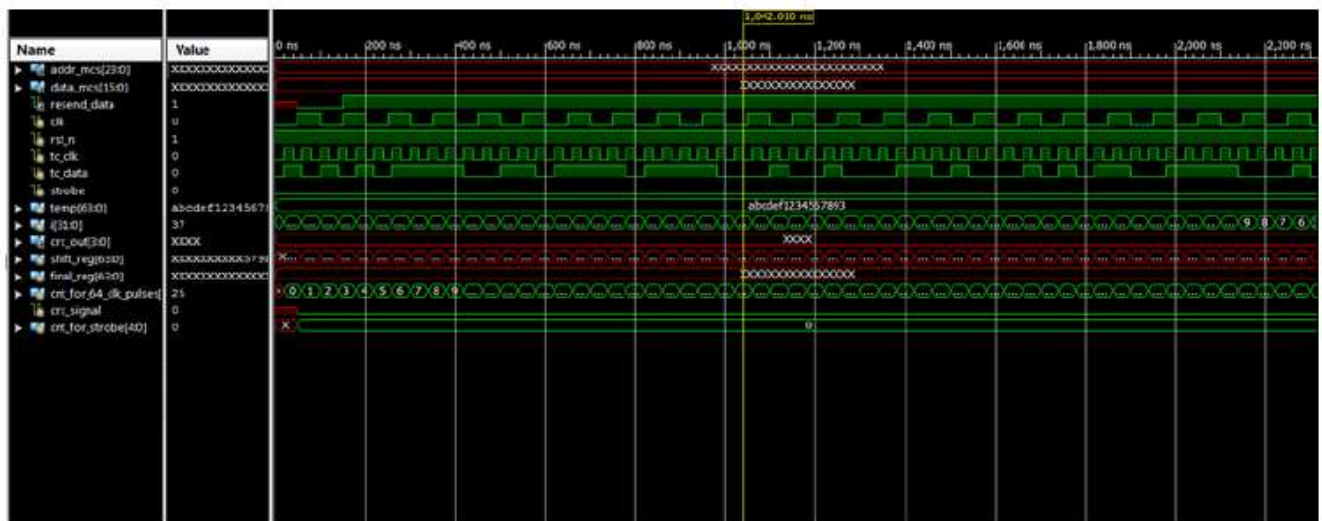


Figure 7. Protocol for the tele-decoder

