REMOTE BITSTREAM UPDATE MECHANISM FOR FPGA RECONFIGURATION

Jerlin Emiliya R[*]

Keerthana.N[**]

Delphy.D[**]

Guru.R[**]

Abstract:

Nowadays remote update process play important role in high volume application. Retrieve the device is too expensive in high sale application. Field Programmable Gate Array(FPGA) provide success for these through a network. Though FPGA suffer from security thread, A secure mechanism has been proposed. The encrypted bitstream are send to FPGA after its integrity are verified by MD5 algorithm. At the end of update process FPGA decrypt the encrypted bitstream and store in RAM. Our proposed scheme guarantee the integrity and confidentiality of bitstream during remote update process and it also protect from replay attack.

Keywords: FPGA, remote bitstream update, integrity.

---******---

[*] Associate Professor, Department of ECE, As-Salam College of Engineering & Technology,Aduthurai-612102, India

[**] Assistant Professor, Department of ECE, As-Salam College of Engineering & Technology,Aduthurai-612102, India

1. INTRODUCTION

Wireless sensor network development was motivated by military application such as battlefield surveillance. These network are used in many industrial and consumer application, such as industrial process monitoring and machine health monitoring and so on. In this paper remote bitstream update mechanism has been proposed for FPGA Reconfiguration.. Reconfiguration platform has feature that allow easy reuse of the node in several application avoiding redesigning the system from scratch. The node include an FPGA which is the core of the reconfigurable capabilities of the node. Reconfiguration area can be remotely or dynamically configured [8]. In order to reduce security flaw like replay attack and to transmit the bit stream confidently we go for remote configuration.

Here zigbee is used as wireless module. Zigbee is mainly suite for high volume application, though it is low powered we can able to transmit data over long distance by passing date through intermediate device. In this paper configuration controller are placed external to FPGA Control signal that are given at transmitter side are send to configuration controller and these controller check for data integrity. When integrity check is correct(ie 128 bit hash value match with original message) it will send reconfiguration boot command to FPGA otherwise process will stop. Depending upon control signal FPGA is reconfigured.

The paper outline is as follows. Section 2 presents Remote Configuration and their feature. Section 3 presents Reconfigurable architecture. Section 4 presents MD5 algorithm for data integrity check. Section 5. presents Experimental setup. Section 6 presents simulation and result. Section 7 presents conclusion and future work.

2. REMOTE CONFIGURATION

Remote update for hardware system is enabled by Field Programmable Gate Array (FPGA). In this work, previous ideas [7,10] are improved and implemented in order to achieve flexibility. Remote configuration has following feature: fix software bugs, adapt to changing user needs and environmental condition in which the network is deployed, shorten software development phase, make software robust, complete application replacement.

3. RECONFIGURABLE ARCHITECTURE

3.1 NEED FOR NETWORK RECONFIGURATION

- Node gets failed or low response time
- Signal strength seems to be quite weak
- Change the router configuration when number end device increase or decrease
- Add a new device with high secure.

Generally there are two type of configuration (i)STATIC (The device is not active during reconfiguration process. While data is send into FPGA the rest of device is stopped and brought up after configuration complete) (ii)DYNAMIC (Active reconfiguration permit to change the part of the device while reset of an FPGA is still running). In this paper dynamic configuration is used[4].

3.2 RECONFIGURABLE ARCHITECTURE

We introduce reconfigurable hardware block to Wireless Sensor Network. Which independently conduct simple sub-task instead of the CPU. Software can be modified by reprogramming the code memory, synthesized logic cores require a redesign of the chip.

3.3 PROPOSED BLOCK DIAGRAM

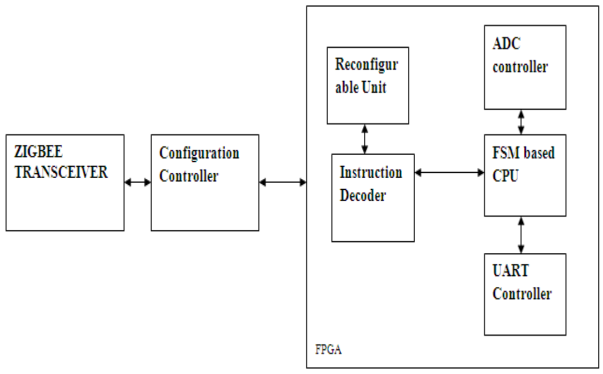


Figure 1: Hardware Implementation Block Diagram

3.3.1 BLOCK DIAGRAM DESCRIPTION

ADC controller

Generate the start of conversion, clock signal to external ADC. This controller monitor the output enable signal from ADC. It will read the sensor data which is in hex format and convert the hex value into decimal and ASCII format.

UART controller

UART Controller generate require baud rate for transmit the data(9600). Compress the sensor data and discard the redundant bits. UART arrange the sensor data into 10bit frame(8 bit data one start bit and stop bit) and shift the data in vitwise each 1/9600 clock period.

FSM based CPU

Generate control signal to UART controller and ADC controller. FSM read the opcode from instruction decoder and execute the task and monitor the reconfigurable unit for update the new bit stream.

Configuration controller

This controller will read the bits stream from Zigbee transceiver. Decrypt the bit stream data, controller will verify the data integrity through MD5 algorithm and generate reconfiguration boot command to FPGA.

4. MD5 ALGORITHM

Previously MAC (message authentication code) value is calculated when making configure with remote update server. This security analysis the integrity and the confidentiality of the bitstream for remote updating process [3,4.7&10]. In this paper MD5 algorithm is used for data integrity.

MD5 algorithm was developed by Professor Ronald L. Rivest in 1991. MD5 is basically MD4 with safety belts it is slower than MD4 but MD5 is high secure. According to RFC 1321, “MD5 message-digest algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input …The MD5 algorithm is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA”. MD5 is similar to MD4 with safety belts. MD5 is highly secure when compared with MD4, it has 4 distinct round but speed is low when compared with MD4.

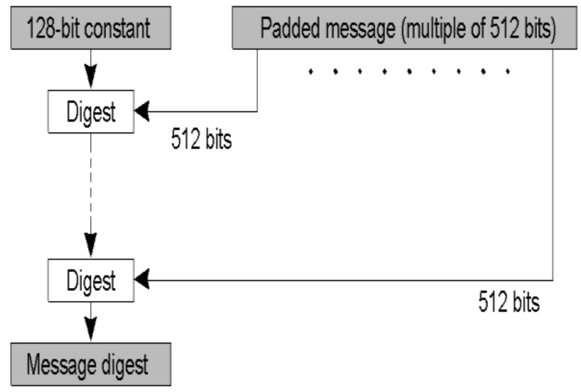


Figure 2 : MD5 Algorithm Structure

5. EXPERIMENTAL SETUP

Hardware implementation is realized with FPGA. FPGA is the programmable digital logic chip, that means programming to do any digital function. the second FPGA heavyweight is Altera . It is very easy to use with an HDL software suite. Since the silicon has a bit less feature and their architecture is not open. Altera quartus is another programmable logic device software from the altera[4].

5.1 ALTERA DE1 BOARD

The DE1(Development and Education) board allow the user to implement a wide range of designed circuits from simple circuits to various multimedia projects has many features. All connections are made through the Cyclone II FPGA device and it provides maximum flexibility for the user,. Thus the user can configure the FPGA to implement any system design.

5.2 CYCLONE II FPGA CONFIGURATION

There are two methods to configure the cyclone II FPGA (i) JTAG Programming (In this method of programming -*Joint Test Action Group*, the configuration bit stream is downloaded directly into the Cyclone II FPGA. The FPGA will retain this configuration as long as power is applied to the board; the configuration is lost when the power is turned off.) (ii) AS Programming (In this method, called *Active Serial* programming, the configuration bit stream is downloaded into the Altera EPCS4 serial EEPROM chip. when the power supply to the DE1 board is turned off also, it preserves non-volatile storage of the bit stream, so the information is retained. When the board's power is turned on, the configuration data is automatically loaded into the Cyclone II FPGA) in the EPCS4 device. In this paper AS Programming is used..

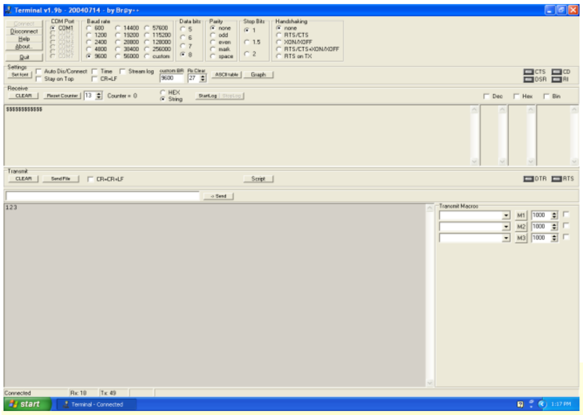
5.3 XILINX ISE VS ALTERA QUARTUS II

Altera quartus II has better GUI(Graphical User Interface) than Xilinx, it provide better HDL support than Xilinx.

6. SIMULATION AND RESULT

Here terminal V1.9b software is used. Terminal is a simple serial port (COM) terminal emulation program. It can be used for communication with different devices such as modems, routers, embedded uC systems, GSM phones, It is very useful debugging tool for serial communication applications. For generation of hash value Secure Hash Algorithm(SHA) is used which is similar to MD5. SHA generate 160 bit hash value for given input, it has 80 rounds.

Control signal are given at transmitter side using Terminal software



Terminal v1.9b - 20040714 - by Br@y++

COM Port: ● COM1 · Baud rate: ● 9600 · Data bits: ● 8 · Parity: ● none · Stop Bits: ● 1 · Handshaking: ● none · custom BR 9600 · Rx Clear 27 · Receive: 13 · Counter = 0 · ● String

```
$$$$$$$$$$$$
```

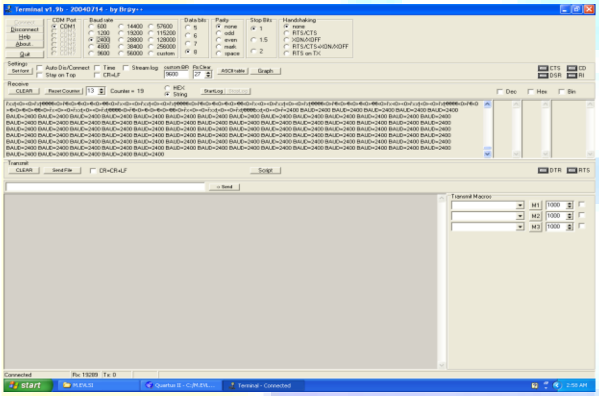
Transmit:

```
123
```

Connected   Rx: 18   Tx: 49

Figure 3: Terminal- Transmitter side

When control signal is 1 baud rate is configured to 2400



Terminal v1.9b - 20040714 - by Br@y++

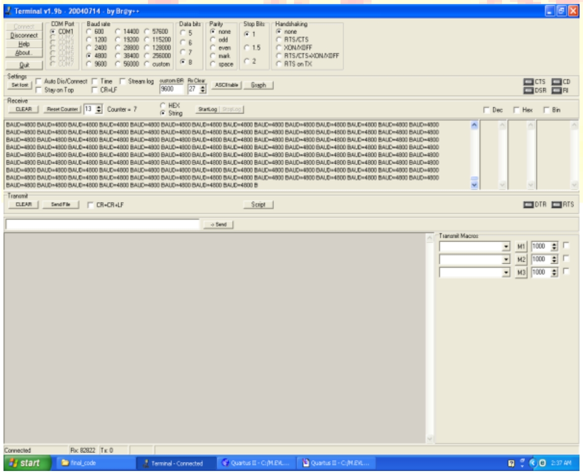
COM Port: ● COM1 · Baud rate: ● 2400 · Data bits: ● 8 · Parity: ● none · Stop Bits: ● 1 · Handshaking: ● none · custom BR 9600 · Rx Clear 27 · Receive: 13 · Counter = 19 · ● String

```
[illegible]
[illegible]=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400
BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400
BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400
BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400
BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400
BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400
BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400
BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400 BAUD=2400
```

Connected   Rx: 19289   Tx: 0

Figure 4: Receiver side showing configured baud rate 2400

When control signal is 2 baud rate is configured to 4800



Terminal v1.9b - 20040714 - by Br@y++

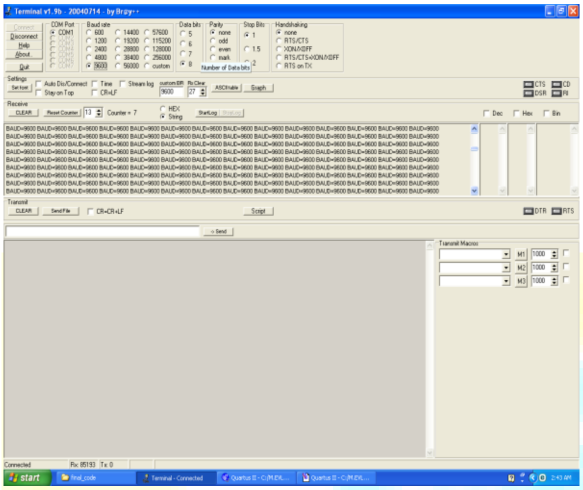
COM Port: ● COM1 · Baud rate: ● 4800 · Data bits: ● 8 · Parity: ● none · Stop Bits: ● 1 · Handshaking: ● none · custom BR 9600 · Rx Clear 27 · Receive: 13 · Counter = 7 · ● String

```
BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800
BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800
BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800
BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800
BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800
BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800
BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800
BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800
BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 BAUD=4800 B
```

Connected   Rx: 82822   Tx: 0

Figure 5: Receiver side showing baud rate configured to 4800

When control signal is 3 baud rate is configured to 9600



Terminal v1.9b - 20040714 - by Br@y++

COM Port: ● COM1

Baud rate: ● 9600

Data bits: ● 8

Parity: ● none

Stop Bits: ● 1

Handshaking: ● none

custom BR: 9600

Rx Clear: 27

Reset Counter: 13

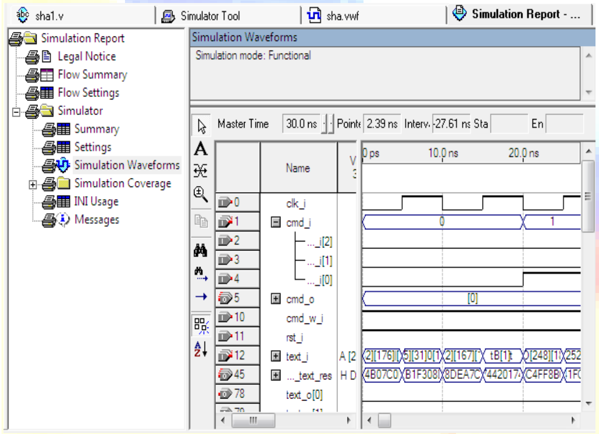
Counter = 7

● String

```
BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600
BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600
BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600
BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600
BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600
BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600
BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600
BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600
BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600 BAUD=9600
```

M1 1000 M2 1000 M3 1000

Figure 6: Receiver side showing baud rate configured to 9600



Simulation mode: Functional

Master Time 30.0 ns Pointer 2.39 ns Interval -27.61 ns

Figure 7: Generation of Hash value.

Above fig show hash value for given 32 bit input here 160 bit hash value is generated. These value is generated mainly for data integrity check.

7. CONCLUSION AND FUTURE ENHANCEMENT

Reconfigurable computing platform based on Sram FPGA is an important high-performance computing platform, we propose a secure scheme for the remote update of bitstream against the security threats. The scheme will reconfigure the FPGA component with the bitstream information. The operation of decryption to reduce the complexity of the remote bitstream update. The bitstream data is encrypted and data integrity is calculated on the remote update server-side. Thus the targeted FPGA is reconfigured in order to safely complete the update of the remote bitstream. The analysis shows that the proposed mechanism is able to ensure that the integrity and confidentiality of the update bitstream and prevent replay.

In this work we had achieved only partial remote configuration and configuration module is external in future internal configuration module is achieved.

REFERENCES

[1]. Ann Gordon-Ross, Alan D. George and Rafael Garcia (2009), “Exploiting Partially Reconfigurable FPGAs for Situation-Based Reconfiguration in Wireless Sensor Networks”, ISBN: 978-0-7695-3716-0.

[2]. Andreas Engel, Andreas Koch and Bjorn Liebig (2012), “Energy- Efficient Hetrogenous Reconfigurable Sensor Node For Distributed Structural Health Monitoring”, E-ISBN : 978-2-9539987-4-0.

[3]. An Braken, Nele Mentens, Jo Vliegen and Ingrid Verbauhede,”Secure Remote Reconfiguration of FPGA”. http://drops.dagstuhl.de/opus/volltexte/2010/2839.

[4]. Balasubramaniyan.G, Keerthana.N(IJRET 2014),”Secure Remote Protocol for FPGA Reconfiguration”.

[5]. Carlos Eduardo Pereira, David Cemin and Marcelo Gotz (2012),” Reconfigurable Agents for Heterogeneous Wireless Sensor Networks”, ISBN: 978-1-4673-5747-0.

[6]. T,Casstro,A and Riesgo (2007),"A Reconfigurable FPGA-Based Architecture For Modular Nodes In Wireless Sensor Networks," ISBN:1-4244-0606-4.

[7]. Florian Devi, Lionel Torres (2010),"Secure Protocol for Remote Bitstream Update Preventing Replay Attacks on FPGA", International Conference On Field Programmable Logic and Application.

[8]. Jia, Z, Liu, Xie, s (2012), "Hardware reconfigurable wireless sensor network node with power and area efficiency "

[9]. Johann Glaser, Jan Haase, Markus Damm, "A Novel Reconfigurable Architecture for Wireless Sensor Network Nodes''.

[10]. Run-feng Huan (ICAISE 2013),"Secure Mechanism for Remote Updates of Reconfigurable Computing Platform".