Dual Redundancy CAN-bus Controller based on FPGA

Deepika.T.P *, Bhagya.P **

Department of Electronics and Communication
Don Bosco Institute of Technology
Bangalore, India

Abstract- In the present trend, CAN buses are implemented using software with the host computer monitoring the CAN as slave. Cause of which leads to bad reliability and real time performance. According to CAN specification version 2.0 of BOSCH gmbh, by downloading the IPcore to XILINX's Spartan 6 FPGA, hardware implementation of customized Dual redundancy CAN bus Controller is put forward in this Paper. It's verified that proposed design can meet the required real-time performance and reliability.

Index Terms- CRC, Dual redundancy CAN-bus Controller, FPGA, IP core, Verilog, XILINX

I. INTRODUCTION

As the automation design has grown up fast, Digital modules designed using FPGA (Field Programmable Gate Array) has find lots of applications in automobile industry, communication, medical equipments and industrial automation [7]. There are many serial communication protocols existing in the market like SPI, I2C, but these are not prioritizing the properties such as error management, anti- interference etc. So that Controller Area Network (CAN) which is a serial, asynchronous, multi-master communication protocol[1] for connecting electronic control modules, sensors and actuators in automotive and industrial applications has become a popular data bus for communication[8] in electronic modules. SJA1000, [1] Philips Semiconductors provides a stand-alone CAN controller that supporting system optimization, diagnosis and maintenance.

Device Net systems that are widely used in factory floor can be subjected to pretty severe environmental conditions that may cause Device Net cable wear out, broken, twist; and may also cause Device Net connector loose connection or other unexpected situations [2]. Providing redundancy mechanism can provide great improvement for the device net system reliability. Solution for this can be the use dual CAN buses [2].No matter how perfect the single- channel bus is designed, but if something happens due to short circuit or open circuit the whole network breaks down.

CAN is the de-facto standard for in-vehicle data transmission. It's the best automotive communication network providing flexible and robust communication [3]. CAN as a field bus technology, found increased applications in robotics, automotive, shop floor control. [4]. among field busses, the CAN bus provides advanced built-in features, which make it suitable for complex real-time applications [5].

There are many types of redundancy like redundancy of bus driver, bus controller, software system [2, 4, 9, 10, 11], but these implanted in software have the disadvantages of bad real time and performance.

So the best redundancy means is the redundancy management done by hardware logic circuit. But a CAN controller chip is a whole component whose function cannot be modified. Thus, a Dual Redundancy CAN-bus Controller (DRCC) based on FPGA chip, which is a programmable logic unit, is put forward in this paper.

II. DUAL REDUNDANCY CAN BUS CONTROLLER DESIGN

The block diagram of DRCC is shown in fig1. DRCC mainly comprises of the blocks BTL(Bit Time Logic) and BSP (BIT Stream Processor).BTL monitors the serial CAN –bus line and provides the bit time logic for CAN-bus, it does hard synchronization and re-synchronization , compensates for the propagation delay times and controls the sample point and the number of samples to be taken within a bit time.

BSPB takes responsible of Data Link Layer protocol and manages CAN Message through recognizing and handling frames like whether it is standard frame or extended frame, manages FIFO(CAN_FIFO),filters Message(CAN_ACF) and do cyclic redundancy check (CAN_CRC).

The Shift Register serializes the messages to be sent and parallelizes received messages. It's loading and shifting is controlled by the BSP.

Error management logic finds and manages the errors on the bus, it can distinguish between temporary and permanent errors. Both BTL and BSP comprise the control signal of Error Management Logic Unit.

CAN_FIFO does buffering the data by not using the top most level. CAN_ACF is an acceptance filter that accepts data when and only the IDENTIFIER of both REMOTE FRAME and DATA FRAME are same.

CAN_CRC depends on 15 bit shift register CRC_RG (14:0).If NXTBIT denotes the next bit of the bit stream; the CRC SEQUENCE is calculated as follows:

```
CRC_RG = 0;
        REPEAT
```

CRCNXT = NXTBIT EXOR CRC_RG(14);
CRC_RG (14:1) = CRC_RG(13:0);
CRC_RG (0) = 0;
IF CRCNXT THEN
CRC_RG (14:0) =CRC_RG (14:0) EXOR (4599hex);
ENDIF
UNTIL (CRC SEQUENCE starts or there is an ERROR condition).

III. SYSTEM DESIGN OF CAN CONTROLLER

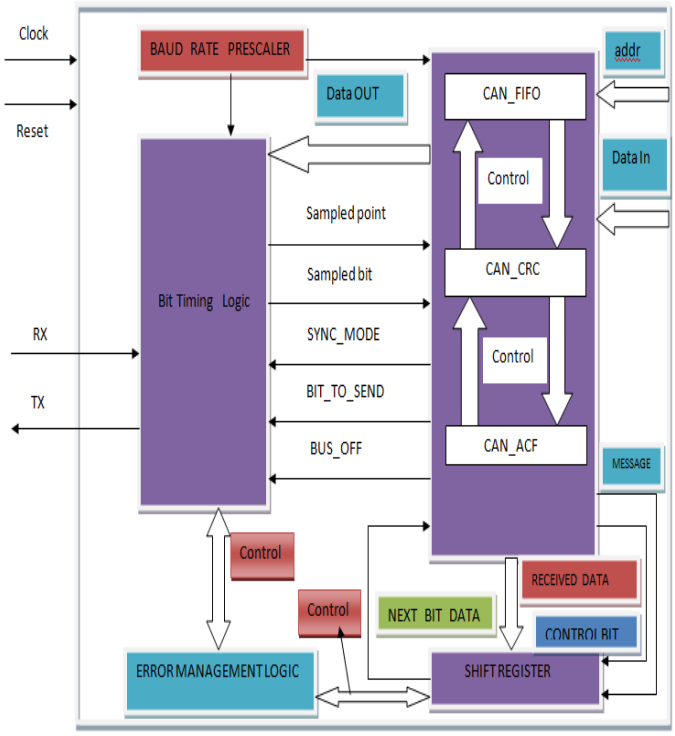

Fig.1. Dual Redundancy Can Bus Controller

The above Fig.1 shows the complete can controller module which contains the BSP (Bit processor logic), FIFO, Bit Timing logic, Acceptance filter modules; Register modules each module are build independently using CAN2.0 protocol and below explained are the individual module operation.

A. Design of Bit Stream Processor

The BSP translates the messages into frames and vice versa. It generates and discards the enclosing fixed format bits, inserts and extracts the stuff bits, calculates and checks the CRC code, performs the error management and decides which type of synchronization is to be done. It is evaluated is time after the sample point that is needed to calculate the next bit to be sent. At the sample point and processes the sampled bus input bit. It defines IPT (Information Processing Time).

B. Design of Bit Timing Logic

Bit timing configuration is done in two register bytes. The sum of Prop_Seg and Phase_Seg1 (as TSEG1) is combined with Phase_Seg2 (as TSEG2) in one register, SJW and BRP are combined in the other register. The data in the bit timing registers are the configuration input of the CAN protocol controller.

The processing of the bit time, the calculation of the position of the Sample Point, and occasional synchronizations are controlled by the BTL state machine.

C. Design of FIFO and Register module

The FIFO and register modules are used for storing the data in the memory. For which FIFO checks the status of the memory, if the memory is full then it will give the message fifo_full and fifo_empty if the data is empty in the memory. Register modules are used for storing the data with respect to clock and reset so that the data will not go to the Meta stability and there is less chance of the data loss.

IV. DESIGN IMPLEMENTATION RESULTS

Design implementation is explained below which shows the individual module snap shots and its working procedure.

A. Bit Stream Processor

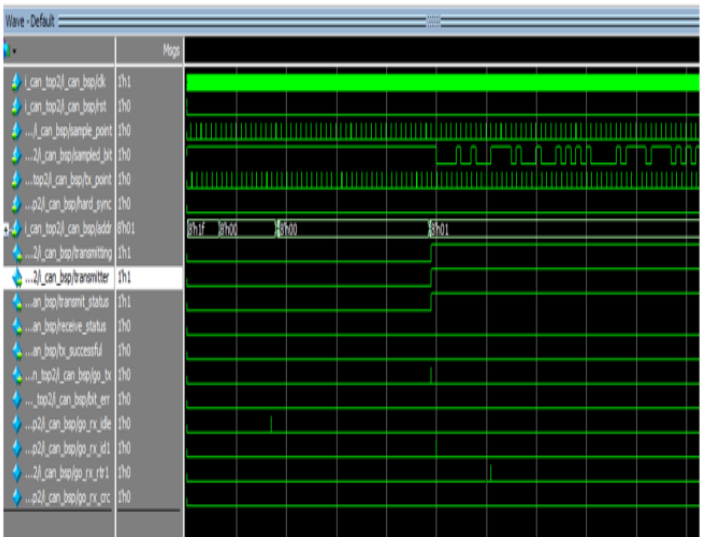

Fig.2. Bit Stream Processor Output

The above Fig.2 snapshot gives the information about BSP that which controls the complete CAN module where input signals sample_bit, tx_point, tx_state, sampled_point are sampled at every positive edge clk. With respect to input signals it drives the data_in and address. By the completion of CAN controller, the input signal hard_sync goes high which means that the complete module is reset and BSP will start the next address transaction.

B. Bit Timing Logic Module

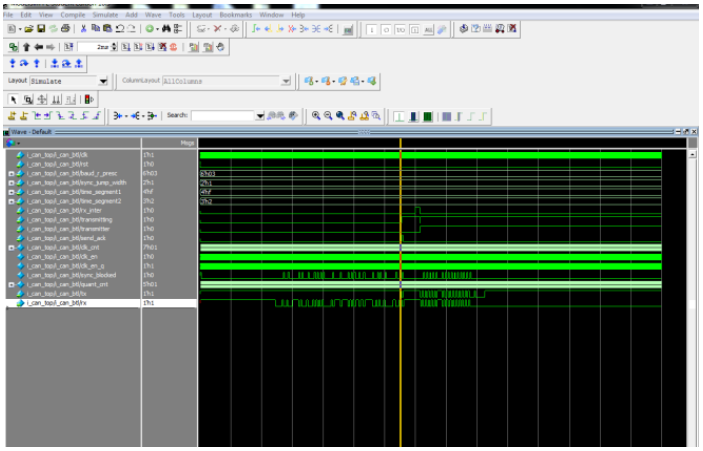

Fig.3. Bit Time Logic output

The Fig.3 snapshot gives the information about the bit timing logic of CAN controller. This will control all the registers and set the values of baud_r_presc (BRP), sync_jump_width (SJW), time_segment1 and also about the time_segment2 registers.

C. Register Module

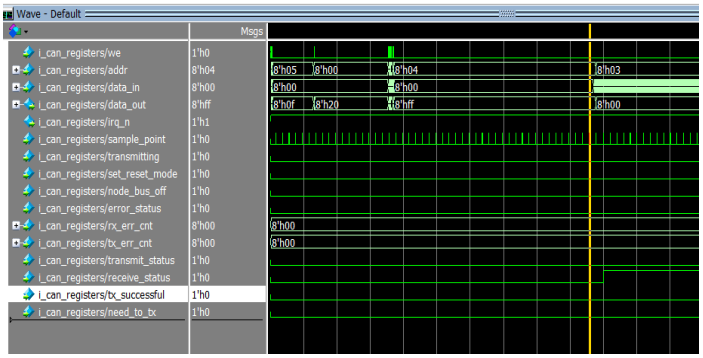

Fig.4.Register Module Output

The Fig.4 snap shot gives the register module where all the configurable registers are configured in this module.

D. CRC Module

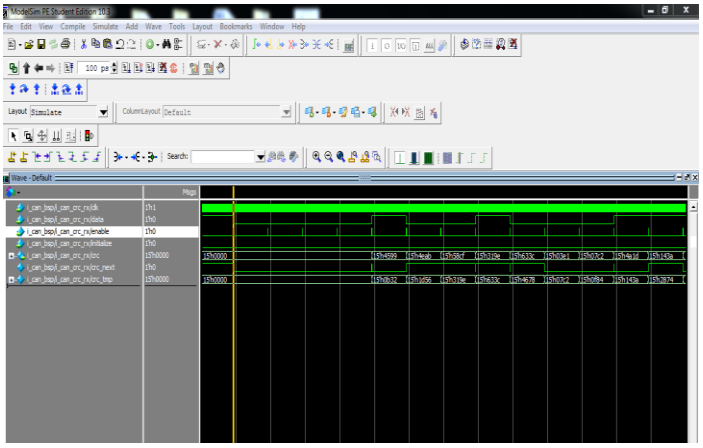

Fig.5. Cyclic Redundancy check Output

The Fig.5 snap shot gives the information of CRC calculation. The CRC should be calculated using above mentioned equation in title II.

E. CAN Acceptance Mode

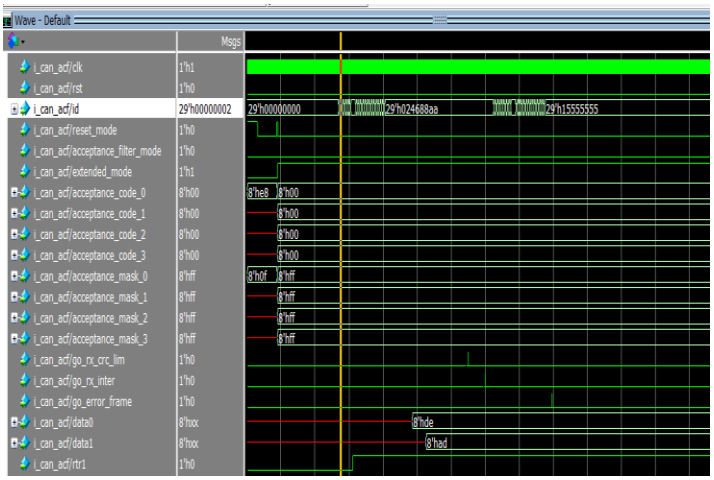

Fig.6. Acceptance Filter Output

The Fig.6 gives the information about the acceptance mode where it will gives the information about the id, node error, CRC_limit, error_frame and RTR information.

V. CONCLUSION

The DRCC IP Core, which is written by synthesizable, behavioral Verilog language, can be used as a module in a project and it has a bright scope for the future. By downloading the IP Core into a XILINX's SPARTAN-6, The Design and Implementation of Dual Redundancy CAN-bus Controller is successful on FPGA. It compensates for the disadvantage of software redundancy and guarantees the reliability and real-time performance.

ACKNOWLEDGMENT

I would like to thank Professor Bhagya.P for her help in testing and analyzing the CAN IP Core.

REFERENCES

[1] Philips Semiconductors. SJA1000 Standalone CAN controller. January 2000.

[2] Qing Jia, DeviceNet media redundancy iCC 2005.

[3] Robert Bosch GmbH, CAN Specification Version 2.0, September 1991.

[4] Jos´e Rufino, Dual-Media Redundancy Mechanisms for CAN, Technical Report, January 1997.

[5] M.A. Livani, J. Kaiser, and W.J. Jia, Scheduling hard and soft real-time communication in CAN, In Proc. of about the CRC the 23rd.Workshop on Real-Time Programming, Shantou, China, June 1998. IFAC/IFIP.

[6] CiA - CAN in Automation. CAN Physical Layer for Industrial Applications - CiA/DS102-1, April 1994.

[7] Ma Xiaojun, Tong Jiarong, "Design and Implementation of A New FPGA Architecture," ASIC, 2003. Proceedings. 5th International Conference, Vol.2, pp.816-819, October 2003.

[8] Yu Zhu, Can and FPGA Communication Engineering: Implementation of a Can Bus Based Measurement System on an Fpga Development Kit, Diplomica Verlag, 2010.

[9] C. Mateus, Design and implementation of a non-stop Ethernet with a redundant media interface. Graduation Project Final Report, Instituto Superior T´ecnico, Lisboa, Portugal, September 1993. (In Portuguese).

[10] Han Ju, Ke Jing, Jin Jiang, A kind of CAN bus redundancy method, Electronics Process Technology (China), Vol.19, No.4, 1999.

[11] Yu ChunLai, Xu HuaLong, Liu GengWang, Hou XiaoLing, Research of the Redundant Methods of CAN, Measurement & Control Technology(China), Vol.22, No.10, 2003.

[12] Xilinx Inc, Spartan-6 FPGA Family: Complete data sheet

[13] I Mohor, CAN Core, www.opencores.org.

[14] Jane Smith, Verilog Coding Guidelines, Cisco Systems, Inc.

AUTHORS

First Author – Deepika T.P, pursuing final year M.Tech in VLSI Design & Embedded systems from Don Bosco Institute of technology, Bangalore.
Email : deepikatp@gmail.com.

Second Author – Bhagya.P has done her M.Tech and is currently serving as Associate Professor in Electronics & Communication department of Don Bosco Institute of Technology, Bangalore.