Evolving the System for Remote Control, Configuration and Management

T.C.Sarma, A.Venugopal & *T.Srujana
. Data Archival and Real-time Systems Division,
National Remote Sensing Agency
(Govt. of India, Dept. of Space)
Balanagar, Hyderabad- 500037
040 - 23884288, 23884287
sarma_tc@nrsa.gov.in

***Abstract*- With the advent of high bit Giga bit network, it is easy to access any system/network globally The development of reconfigurable and field up gradable logic devices enabling the digital system designs to opt for the development of built-in testability, diagnoisabilty of the devices with an application of remote management as the basic feature. The timing systems that were developed by the authors were supplied internationally for the Indian Remote Sensing Satellite data users. As the number is increasing day by day, the authors have developed the remote management function of these systems while providing the built-in test provisions with remote access. The concept of remote management is embedded into the system design with the embedded micro controller and the Xilinx field programmable/updatable feature. After realizing this concept of remote management the systems can be managed remotely for maintenance, trouble shooting and upgradation. The Paper deals with the design concepts that were used in introducing the remote management for all the timing systems as per IRIG (Inter Range Instrumentation Group) format that were developed by the authors at Data Archival & Real Time Systems Division of NRSA, Dept. of Space, Govt. of India. *Key words:* Remote Programmability of the devices. Embedded microcontrollers, Serial Interfaces for remote management**

I. Introduction

Timing Systems are one of the important constituents of Remote Sensing Satellite Data Reception Station for basic data geo referencing and synchronized operations globally. At nearly all facilities where data is collected, the need exists to time tag data for the purpose of its correlation and indexing. The Global Positioning System (GPS) has evolved as a popular and accurate source for precision time. A standard Timing System comprises of a GPS Receiver, Time Code generator, Distribution Amplifier, Time Code Translators and Remote Time Display Units. The Time Code Generator clocked by a precision TCXO and with a provision for synchronization with GPS is the master source of serial IRIG-A time code. At the receiving system, the serial time code needs to be processed to extract the time information and provide a Binary Coded Decimal (BCD) output as well as a Display of the time being translated, the Time Code Translators are designed to provide these functions. Several of these systems as per IRIG standards were developed and deployed in different parts of the world as a part of Indian Remote Sensing Satellite data Acquisition and processing system by the authors In order to manage these systems remotely the concept of remote configurability, controllability and management were developed by the authors as an innovative approach. The concept is as follows.

Complex Programmable Logic Devices (CPLDs) have grown sufficiently in capacity and functionality to support complete platforms on a single chip. It is now possible to implement processors, memories and high speed I/Os on a single Device. In addition to the high capacity and additional functionality, reprogrammable CPLDs also allow designs to be easily changed even after hardware has been designed. This capability allows versatile hardware to be designed as a platform on which applications can be developed, making it easier to hit moving standards or changing customer requirements. By updating the Xilinx PLD-based hardware design using the product's network connection is becoming more realistic instead of deploying field service technicians to the installation site.

This paper describes in detail the design approach followed to develop a timing system that is designed for Remote diagnostic, test & upgrade options. The design of the remote management feature is made generic to be adaptable to other hardware systems.

II. Design Description

The Time Code Generator and Translator unit is designed to perform the dual functions of Time Code Generator or Translator as per the user selection as per IRIG standards. Time Code Generator provides the GPS synchronized IRIG-A Serial time, which forms the reference time for the Station for tracking and data archival. It also has provision for user presetable time. In the Translator Mode the unit provides the BCD Time for tagging to the RAW data in real-time. It also provides a serial interface for system time and network time synchronization. The Figure 1.0 below depicts the system block diagram. The system is designed around the Xilinx CPLDs. To realize the concept of a remote management of the system, an Embedded Microcontroller and a Siemens Infrared Data Transceiver (IRM3105) are used.

.*Student Trainee involved with this Development

1-4244-0549-1/06/$20.00 ©2006 IEEE.

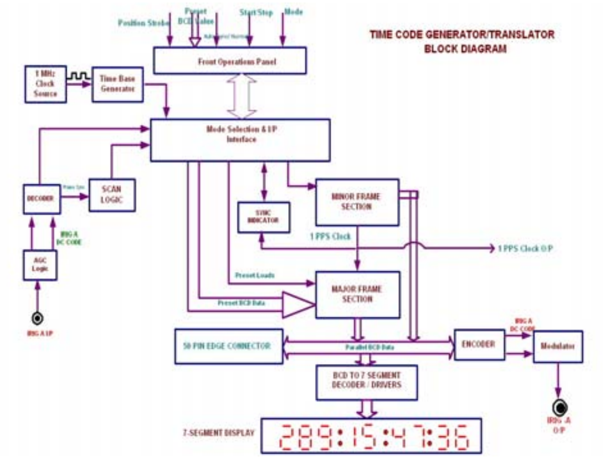


Figure 1.0 System Block Diagram

The Siemens infrared transceiver receives updated XC9500 programming information in the form of an XSVF (Xilinx Serial Vector Format File) file wrapped in RS232 format. The figure 2.0 shows a block diagram of the XC9500 Remote Field Upgrade design.

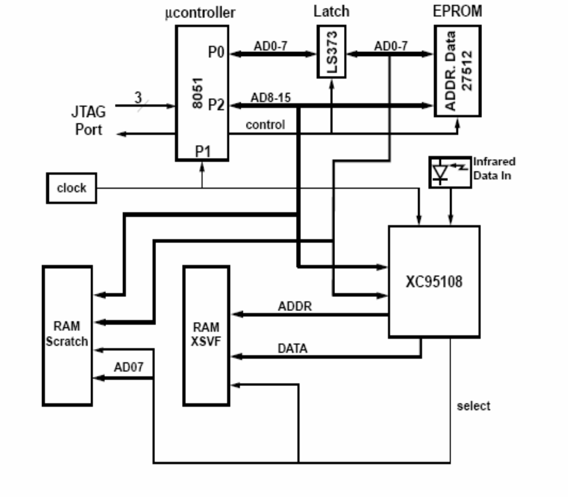


Figure 2.0

The XC95108 CPLD contains the functionality described in the VHDL code. The VHDL design contains a UART receiver, timing interface control engine bus controller and top level connectivity file At the beginning, a CPLD JEDEC design file is first transformed into a corresponding XSVF file. The file is driven by software through a serial port connected to the Siemens IR data transceiver. The receiver end of the transceiver is attached to the XC95108 where the bit-stream will be intercepted by the UART receiver and loaded into the XSVF RAM. This is done with a two step protocol process.

First, the UART captures each data byte, and then passes it to the timing interface control engine which manages the data arrangement in the XSVF RAM memory. Once the data transfer is complete, the interface control engine flags the embedded microcontroller, indicating that the updated XSVF data is ready in XSVF RAM. The microcontroller will drive the data down the JTAG port which is attached to the CPLD to be upgraded/diagnosed. The EPROM holds the 8051 code which performs the embedded download operation.

A. *UART Design details*

The UART design contains 4 states: MARK, DATA, STOP_BIT, and ERR, and is designed to parse the XSVF file, stripping out the start and stop bits, while registering the 8 data bits. The MARK state is essentially the rest state once the XSVF data transfer is complete, but also serves to verify the start bit in an RS232 data sequence. A start bit is verified by 8 counts while input IR_IN is "low". Once the start bit is verified, the state machine moves to the DATA state. A data bit is registered into an 8 bit register containing an "offset" that is incremented by one each time a bit is registered.

Data bits are registered after 16 counts. Once the 8 data bits are registered, the state machine moves to the STOP_BIT state. In the STOP_BIT state, the counter counts to 16 and verifies that IR_IN is "low". If IR_IN is "low", the state machine will advance to the MARK state when IR_IN transitions to a "high". If IR_IN is "high" after 16 counts, the state machine will enter the ERR state to signal that a transmit error has occurred. While in the STOP_BIT state, the UART flags the timing interface control engine to indicate that data is ready to be written into XSVF RAM. Figure shows the UART Receiver State Diagram.

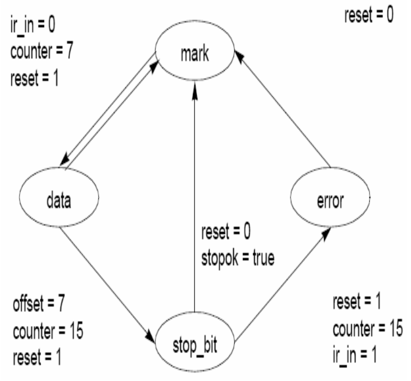


Figure 2.1 UART Receiver State Diagram.

B. *Timing Interface Control Engine*

The timing interface control engine is the heart of the remote field upgrade design. It contains 4 states: HIGH, LOW, DOWNLOAD, and ISP. The first two bytes of the XSVF file represent the number of XSVF file bytes to be transferred. States HIGH, and LOW, register the high, and low bytes, respectively, into signal "length_count". While in the DOWNLOAD state, the "length_count" is decremented each time a byte is written into XSVF RAM. The timing interface control engine also increments the internal address counter, "add_cntr", and controls the "write" signal allowing data to be written into XSVF RAM.

Once the entire XSVF file has been written into XSVF RAM, the timing interface control engine enters the ISP state. The ISP state interrupts the embedded processor, which eventually vectors off to it's interrupt service routine to reconfigure an in- system XC9500 CPLD with the updated information in XSVF RAM. The timing interface control engine works in concert with the bus controller. The bus controller monitors the timing interface control engine's state bits, and passes control of the XSVF data bus accordingly.

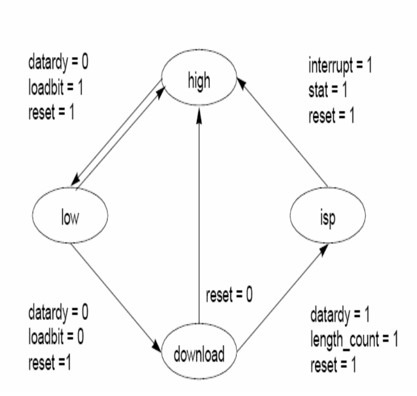


Figure 2.2: Time interface control engine State Diagram.

The DOWNLOAD state allows the XC95108 to retain control of the XSVF data bus while keeping the processor data bus in high impedance. The ISP state passes control of the XSVF data bus to the processor when the appropriate conditions are met (i.e., PSE = 1, UC_ADD(15) = 1, and READN = 0). The remaining states, HIGH and LOW, keep the XSVF data bus, and processor data bus in high impedance. The figure depicts the State diagram of the ime interface control engine.

C. Programming Core System CPLD via a Remote Source

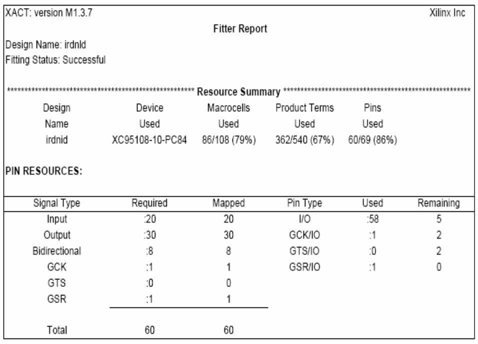
Serial Vector Format (SVF) is a syntax specification for describing high level IEEE 1149.1 (JTAG) bus operations. Xilinx Serial Vector Format (XSVF) is a compressed, binary version of the SVF file designed specifically for embedded applications. XC9500 CPLDs use the IEEE 1149.1 Boundary Scan Standard for in-system programming (ISP). Once the binary XSVF file is complete with number of bytes, the reader should use the binary to Intel Hex translator to create an Intel Hex file suitable for embedded applications. The Intel Hex converted XSVF file is then transferred remotely to the embedded application described herein, and thereby upgrades an in-system XC9500 CPLD.

D. Field Upgrade

While designing a system for remote testability and upgradeability, special precautions are needed in the early design stage itself. This section describes the measures take by the authors in designing the system. Enough room for growth has to be provided and the CPLD should not be packed to 100%, Pin-locking refers to allowing the CPLD software to fit a design based on an algorithm that tends to spread equations throughout the CPLD. Once the design is fit, future design changes can be made without compromising the original pin-out. The robust pinlocking feature XC9500 CPLD has been exploited to design the system for ISP. However, it's still important to recognize that if a design is likely to grow, it's best not to force the software to override the pin-locking algorithm, and pack the design. Some techniques used for leaving room for growth are as follows

III. Implementation

The Design density was optimized not to exceed 85%. Figure 3.0 shows a partial summary of the fitting results



```
XACT: version M1.3.7                                                              Xilinx Inc
                                   Fitter Report
Design Name: irdnld
Fitting Status: Successful

****************************** Resource Summary ******************************
        Design          Device          Macrocells     Product Terms      Pins
        Name            Used            Used           Used               Used
        irdnid      XC95108-10-PC84   86/108 (79%)   362/540 (67%)    60/69 (86%)

PIN RESOURCES:

     Signal Type      Required      Mapped      Pin Type      Used      Remaining
        Input           :20           20          I/O          :58          5
        Output          :30           30          GCK/IO       :1           2
     Bidirectional      :8            8           GTS/IO       :0           2
        GCK             :1            1           GSR/IO       :1           0
        GTS             :0            0
        GSR             :1            1
                    ______________________
        Total           60            60
```

Figure 3.0

It can be observed that the Macrocells Used is 79%, or 86 out of 108. Also notice that out of 69 total user I/O, 9 are left over. This leaves a cushion for future changes. Traces were provided on the circuit board that physically connects I/O to future applications. The "Create Programmable Grounds" option was exercised in the implementation phase of the design. For specific I/O pins that needed to be reserved, “Dummy” functions such as OUT = IN were used.

IV. Results

The XC9500 Remote Field Upgrade design was verified using the VHDL test bench with the Synopsys VHDL debugger Software. The design was translated using the Synopsys Design Compiler, v3.4b. It was fit to an XC95108-10PC84 CPLD with the M1.3.7 XACT CPLD Software. The photograph below depicts the system deployed in the Data Archival & Real-time System used to Archive and process the data from the Indian remote Sensing satellites at the Indian and International Ground stations

V. Conclusion

In the present technological era where the entire globe is treated as a single village, the concept of remote management of units, systems and functions become an essential feature. In this concept the authors made an attempt and succeeded to implement the remote control, configure and management functions for all the equipments that were developed and deployed by them in different parts of the globe with the associated developments in the from of Hardware, software and firmware with easily interactable GUIs to provide the automated report generation etc. It is certainly necessary to implement these concepts to every system and functionality to really visualize the globe as a single unit. Authors have also developed several modules for the remote management of the systems for the remote sensing satellite data handling and associated functions.

References

[1] A. Stoica, D. Keymeulen, R. Tawel, C. Lazaro and Wei-te Li. “Evolutionary experiments with a fine-grained reconfigurable architecture for analog and digital CMOS circuits.” In [6].

[2] R. Zebulum, A. Stoica and D. Keymeulen. A Flexible Model of a CMOS Field Programmable Transistor Array Targeted for Hardware Evolution. Third Int. Conference on Evolvable Systems:From Biology to Hardware (ICES2000)

[3] A. Fukunaga and A. Stechert. Evolving nonlinear predictive models for lossless image compression with genetic programming GP-98

[4] J. Koza, F.H. Bennett, D. Andre, and M.A Keane, “Automated WYWIWYG design of both the topology and component values of analog electrical circuits using genetic programming”, *Proceedings of Genetic Programming Conference*, Stanford, CA , pp. 28-31, 1996

[5] A. Thompson, “An evolved circuit, intrinsic in silicon, entwined in physics”. In *International Conference on Evolvable Systems*. Springer-Verlag Lecture Notes in Computer Science, 1996, pp. 390-405.[6] M. Sipper, D. Mange, A. Perez-Uribe (Eds.) Evolvable Systems: From Biology To Hardware*, Proc. of the Second International Conference, ICES 98*, Lausanne, Switzrland, Springer-Verlag Lecture Notes in Computer Science, 1998.

[7] A. Stoica, D. Keymeulen and J. Lohn (Eds.) *Proc. of the Firs NASA/DoD Workshop on Evolvable Hardware*, July 19-21, 1999, Pasadena, CA IEEE Computer Society Press.

[8] Stoica, A. Toward evolvable hardware chips: experiments with a programmable transistor array. Proceedings of 7th International Conference on Microelectronics for Neural, Fuzzy and Bio8 Inspired Systems, Granada, Spain, April 7-9, IEEE Comp Sci. Press, 1999.

[9] Butnariu, D. and Klement, E. P., Triangular norm-based measures and games with fuzzy coalitions, Kluwer Academics, 1993

[10]M.M. Gupta and J. Qi, Design of fuzzy logic controllers based on generalized t-operators. Fuzzy Sets and Systems, Vol. 40 pp 473-389

[11] [10]M.M. Gupta and J. Qi, Theory of t-norms and fuzzy inference methods. Fuzzy Sets and Systems, Vol. 40, 431-450

[12] Stoica, A Synaptic and somatic operators for fuzzy neurons: which T-norms to choose? In Proc. of 1996 Biennial Conference of the North American Fuzzy Information Processing Society - NAFIPS, Berkeley, CA, June 19-22, 55-58

[13] Stoica, A. On hardware evolvability and levels of granularity. Proc. of the International Conference Intelligent Systems and Semiotics 97: A Learning Perspective, NIST, Gaithersburg, MD, Sept. 22-25, 1997

[14] A. Stoica, D. Keymeulen, V. Duong and C. Lazaro. Automatic Synthesis and Fault-Tolerant Experiments on an Evolvable Hardware Platform. In R. Profet et al.(eds.), Proc. Of IEEE Aerospace Conf., March 18-25, 2000, Big Sky, MT, IEEE Press