

Design of FlexRay Communication Network Using Active Star

Yi-Nan Xu (Corresponding author)

Department of Electronics and Communication, Yanbian University

977 Gongyuan Road, Yanji City, Jilin 133002, China

Tel: 86-433-273-2265 E-mail: ynxu@ybu.edu.cn

Chang-Ha Jeon & Jin-Gyun Chung

Division of Electronics and Information Engineering, Chonbuk National University

664-14 Ga Deokjin Dong, Jeonju City, Jeonbuk 561-756, South Korea

Tel: 82-63-270-2466 E-mail: jsure, jgchung@chonbuk.ac.kr

This work was partly supported by the IT R&D program of MKE/IITA. [No. B1120-0901-0002, A supporting program of IT cluster]

Abstract

FlexRay is a new standard of network communication system which has been developed for future automotive applications. It was devised by automotive manufacturers and semiconductor vendors. FlexRay provides a high speed serial communication, time triggered bus and fault tolerant communication between electronic devices for automotive applications. In this paper, FlexRay system is first designed using SDL (Specification and Description Language). For hardware implementation, FlexRay system is designed using Verilog HDL based on the SDL design result. The designed system is synthesized using Synopsys Design Compiler with the Hynix 0.18 μm cell library. For constructing a FlexRay communication network, the active is used since active star systems can provide high speed data transmission up to 10 Mbps. The performance of the active star network is tested using one transmitter node and two receiver nodes.

Keywords: FlexRay, Active Star, Communication Controller, SDL, FPGA

1. Introduction

FlexRay was developed for next generation automobiles by a consortium founded by BMW, Bosch, DaimlerChrysler and Philips in 2000. FlexRay is a new standard of network communication system which provides a high speed serial communication, time triggered bus and fault tolerant communication between electronic devices for future automotive applications. FlexRay supports a time-triggered scheme and an optional event-triggered scheme. The upper bound of the data rate is 10 Mbps and it provides two channels for redundancy (FlexRay Consortium, 2005).

FlexRay protocols are first designed using SDL (Specification and Description Language). Then, the system is re-designed using Verilog HDL based on the SDL source. In addition, the FlexRay system is combined with active stars. The combined system is implemented using ALTERA Excalibur ARM EPXA4F672C3. It is shown that the implemented system operates successfully.

In section 2, FlexRay structure is briefly reviewed. Section 3 presents the implementation of FlexRay system using SDL. Section 4 presents the implementation of FlexRay system using Verilog HDL. Section 5 presents the design of FlexRay communication network using the active star. Finally, brief conclusions are given in section 6.

2. FlexRay system architecture

Each FlexRay node consists of a host, communication controller (CC), bus guardian (BG) and bus driver (BD) as shown in Fig. 1. Host is the user software which controls the communication process. CC is an electronic component in a node that is responsible for implementing the protocol aspects of the FlexRay communications system. BG is an electronic component which protects slots against faulty media access and is optional. BD is consisted of a transmitter and a receiver that connects a communication controller to one communication channel. A node can be connected to both channels (e.g., Node A or Node D in Fig. 1) or only to a single channel (e.g.,

Node B or Node C in Fig. 1).

The communication takes place in the periodic equal length cycles depicted in Fig. 2. One communication cycle consists of four segments. Each communication cycle contains static segment (ST), dynamic segment (DYN), symbol window (SW) and network idle time (NIT).

A time division multiple access (TDMA) scheme is used for ST. ST is consisted of specific number (designated by global configuration parameter `gdStaticSlot`) of static equal-length. DYN is operated by a flexible time division multiple access (FTDMA) scheme and is optional element for non-periodic communication. DYN is consisted of specific number of minislots (designated by global configuration parameter `gNumberOfMinislots`). SW is used for run time tests, and NIT is a communication free period that concludes every communication cycle.

A FlexRay frame consists of three segments: header segment, payload segment and trailer segment as depicted in Fig. 3.

The first five bits define the basic features of the frame. Frame ID (11 bits) is defined as the slot position in the static segment. For the dynamic segment, Frame ID is used to indicate the priority of the frame: a lower identifier indicates higher priority. Payload Length (7 bits) is defined as the data length (payload length * 2 = number of data bytes). Header CRC (11 bits) is a Cyclic Redundancy Check, which is computed over the Sync Frame Indicator (1 bit), Startup Frame Indicator (1 bit), Frame ID (11 bits) and Payload Length (7 bits). Cycle Count (6 bits) is the serial number of the frame defined locally in the node. Payload Segment (0 ~ 254 bytes) contains main data. Trailer Segment (24 bits) is for Cyclic Redundancy Check, which is computed over the header segment and payload segment.

3. FlexRay system design using SDL

Specification and Description Language (SDL) is a high abstraction level system design language that is standardized as Z.100 by ITU Telecommunication Standardization Sector (ITU-T) (ITU-T Z.100, 1996). SDL is used for modeling, simulation and verification of communication protocols. Protocol design using this formal language ensures the compatibility between the requirements of the initial design and the final implementation. It reduces the design time by isolating the implementation details at the beginning (Ferenc Belina, Dieter Hogrefe and Amardeo Sarma, 1991). In this section, we present SDL design of FlexRay CC, BG and the functions in the protocol specification.

Fig. 4 illustrates the implementation of FlexRay CC and BG systems using SDL. The FlexRay CC system consists of fifteen blocks: Protocol Operation Controller (POC), MacroTICK Generation (MTG), Clock Synchronization Startup (CSS_A, CSS_B), Clock Synchronization Process (CSP), Media Access Control (MAC_A, MAC_B), Frame and Symbol Process (FSP_A, FSP_B), Coding and Decoding Process (CODEC_A, CODEC_B), Bit Strobing Process (BITSTRB_A, BITSTRB_B) and Wakeup Pattern Decoding Process (WUPDEC_A, WUPDEC_B).

FlexRay BG system consists of seventeen blocks: Bus Guardian Protocol Operation Controller (BG_POC), Communication Controller Supervision (CCS), Transmit Data Enable signal from the CC (TxEN), TxEN Supervision Process (TxENSUP_A, TxENSUP_B), Header check in Normal Supervision (NHC_A, NHC_B), TxEN active length check in Normal Supervision (NTALC_A, NTALC_B), Channel Asymmetry Detection Process (TxENASYM), Bus Guardian Enable (BGE), BGE Generation Process (BGEGEN_A, BGEGEN_B), Header Content Check Process (CHC_A, CHC_B), and Startup Supervision Process (STARTUPSUP_A, STARTUPSUP_B).

For the implementation the FlexRay system, we modified some processes as follows:

- i. In the FlexRay CC protocol specification, there is no indication on the signals from MAC_A (MAC_B) to CSP channels. In our implementation, an action point on A (action point on B) signals are loaded as depicted in Fig. 4.
- ii. In the proposed implementation, the BITSTRB_A (BITSTRB_B) and WUPDEC_A (WUPDEC_B) blocks are separated from the CODEC_A (CODEC_B) block, since, in SDL description, it is difficult to combine BITSTRB, WUPDEC and CODEC blocks as one block.
- iii. FlexRay protocol specification does not describe the details of the following functions: `prepbstream`, `prepCASstream`, `frameCRC`, `headerCRC`, `getpayloadlength`, `getRF`, `position`, `abs`, `length`, `sort`, `append`, and `member`. In this research, all of the functions are realized by the SDL, and its application to the implemented FlexRay system.
- iv. FlexRay protocol specification has three kinds of parameters: global configuration parameters, node

parameters, and protocol constants. In our implementation, the global configuration parameters and node parameters were decided based on DECOMSYS GmbH equipment, and the protocol constants are the same as those of FlexRay protocol specification.

Message Sequence Chart (MSC) is used for the design verification and simulations of designed FlexRay system. An example of MSC simulation result for transmit frame from MAC_A block to CODEC_A block is shown in Fig. 5.

4. FlexRay system design using Verilog HDL

FlexRay communication controller and bus guardian protocols can be realized in Finite State Machine (FSM) diagram form. FSM is a communication method to realize large and complicated system easily using software or hardware. In this work, FSM method was used for easy implementation of FlexRay system. For hardware implementation, FlexRay system was designed using Verilog HDL based on the SDL design result. The designed system was synthesized using Synopsys Design Compiler with Hynix 0.18 μm cell library. The designed system requires about 1,587,000 gates and 14 mW power consumption. It is shown that the designed system can successfully operate in the frequency range above up to 35 MHz. The design was verified using Cadence Verilog-XL and Simvision.

Fig. 6 shows the Simvision timing simulation result of transmit and receive frames in channel A and channel B after transmission frame (vTF) is transmitted from the transmission node. The combined FlexRay system is implemented using FPGA (Yi-Nan Xu, Y. E. Kim, K. J. Cho, J. G. Chung and M. S. Lim. 2008).

5. FlexRay active star design and application

The active star network uses point-to-point connections between active stars and nodes. There are two kinds of stars. One is an active star and the other is a passive star. The active star is mainly used in FlexRay communication system, since the active star designed by commercial transceiver chip could realize a data transmission up to 10 Mbps. On the contrary, the passive star is used to transmit data less than 1 Mbps with the method of cutting off bus between nodes manually.

Fig. 7 shows the example of a FlexRay network using star. The active star to which the nodes are connected has the function to transfer data streams on one branch to all other branches. Since the active star device has a transmitter and receiver circuit for each branch, the branches are actually electrically decoupled for each other.

Fig. 8 shows the FlexRay bus transceiver architecture used in the implemented active star system. The FlexRay bus transceiver is a high-speed automotive transceiver for fault tolerant and high speed applications, operating as the bi-directional interface between a generic communication controller and the twisted pair copper wires. The device enables two-way communications with the microcontroller with full mode handling, including the low-power modes.

The transmitter consists of a high side driver and a low side driver for the BP (Bus Plus) line as well as the BM (Bus Minus) line, as shown in Fig. 8(a). Fig. 8(b) depicts a schematic view of the receiver block, which is responsible for biasing the bus and receiving the signal streams. Fig. 9 shows the FlexRay active star circuit diagram using transceiver chip.

The FlexRay active star was designed using bus transceiver chip. The bus transceiver works in six different kinds of stations. They are standby, sleep, go to sleep, power on standby, and receive only and normal mode, respectively. In order to control the active star in such different situations, the additional controller designed by Verilog HDL language and FlexRay system are synthesized in this research. And the experiment is performed to confirm the application of designed active star. In Fig. 10, there are two frames for transmission from transmitter nodes, frame with ID of '01' in slot 1, and frame with ID of '10' in slot 2. If serials of frame data are transmitted in transmitter node, as shown in Fig. 11, through active star, the receiver node Rx_1 only receive 'frame 01' and the receiver node Rx_2 only receive 'frame 10'.

The entire realization of the test environments is shown in Fig. 12. From the host PC, it is verified that the data transfer is performed successfully.

6. Conclusions

In the present study, firstly, implemented FlexRay CC and BG protocol specifications and function parts using SDL and then, the system was re-designed using Verilog HDL based on the SDL source. The FlexRay system was synthesized using Hynix 0.18 μm technology. The designed system requires about 1,587,000 gates and 14 mW power consumption. And the FlexRay system was implemented using ALTERA Excalibur ARM EPXA4F672C3.

Also, the active star is designed using a commercial bus transceiver chip. The combination of this active star and the FlexRay system designed by FPGA are constructing FlexRay communication network.

References

- Ferenc Belina, Dieter Hogrefe and Amardeo Sarma. (1991). *SDL with Applications from Protocol Specification*. USA: Prentice Hall
- FlexRay Consortium. (2005). Electrical Physical Layer Specification, v2.1 Rev A: <http://www.flexray.com/index.php?sid=2ee5bfdd0586d7fd6641ebc28398e805&pid=47&lang=de> (Dec. 5, 2009)
- FlexRay Consortium. (2005). Preliminary Node-Local Bus Guardian Specification, v2.0.9: <http://www.flexray.com/index.php?sid=2ee5bfdd0586d7fd6641ebc28398e805&pid=47&lang=de> (Dec. 5, 2009)
- FlexRay Consortium. (2005). Protocol Specification, v2.1 Rev. A: <http://www.flexray.com/index.php?sid=2ee5bfdd0586d7fd6641ebc28398e805&pid=47&lang=de> (Dec. 5, 2009)
- Homepage of the austriamicrosystems, <http://www.austriamicrosystems.com>
- Homepage of the DECOMSYS GmbH, <http://www.decomsys.com>
- Homepage of the FlexRay Consortium, <http://www.flexray.com>
- ITU-T Z.100. (1996). *CCITT, Specification and Description Language (SDL)*
- Yi-Nan Xu, Y. E. Kim, K. J. Cho, J. G. Chung and M. S. Lim. (2008). Implementation of FlexRay Communication Controller Protocol with Application to a Robot System. *The 15th IEEE International Conference on Electronics, Circuits, and Systems*. Malta: IEEE, 2008. 994-997.

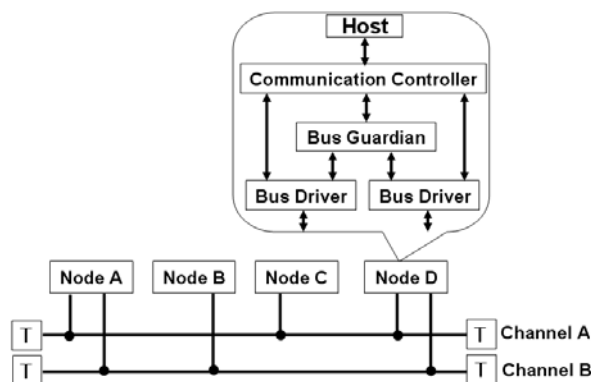


Figure 1. FlexRay node architecture and dual bus topology configuration

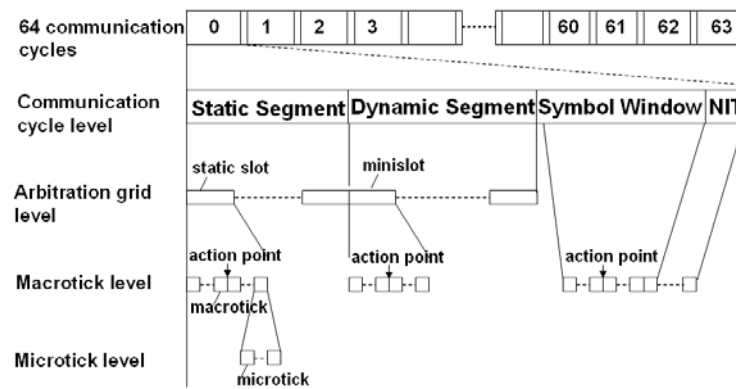


Figure 2. FlexRay timing hierarchy

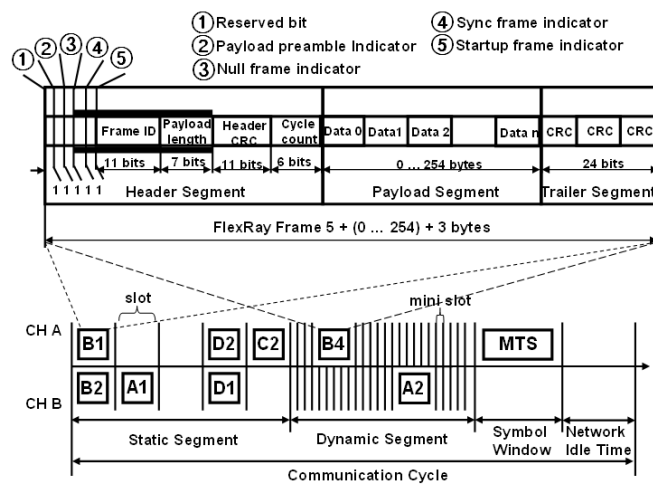


Figure 3. FlexRay frame format and communication cycle

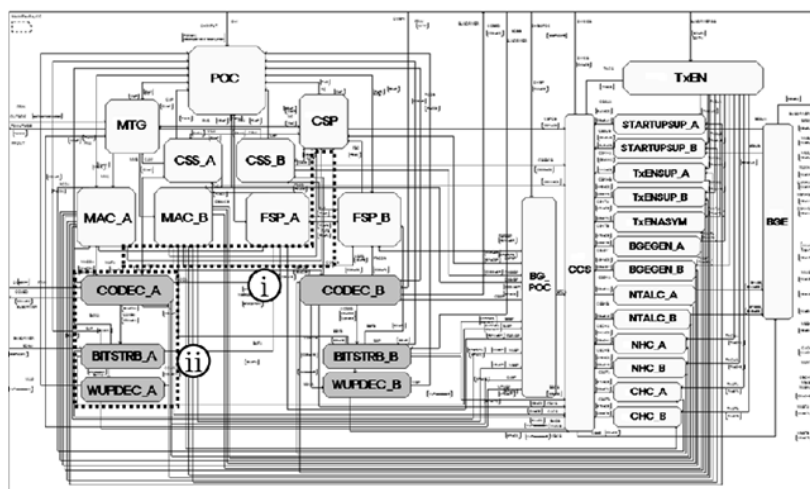


Figure 4. Hierarchical structure of FlexRay system

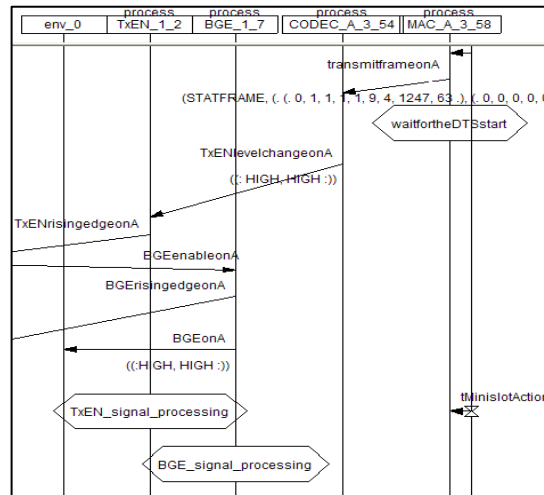


Figure 5. Example of MSC simulation result

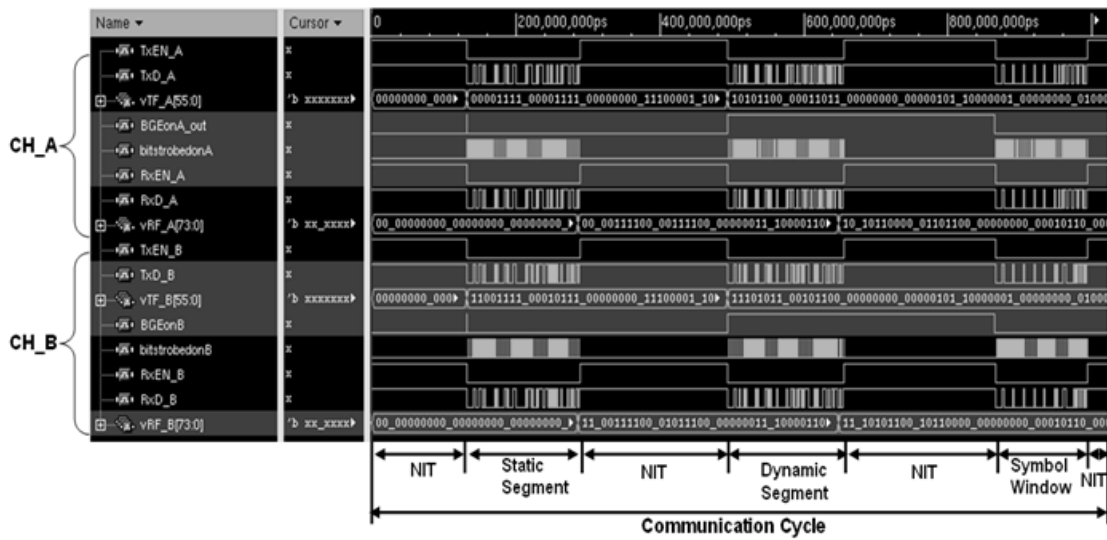


Figure 6. Timing simulation result of transmit and receive frames

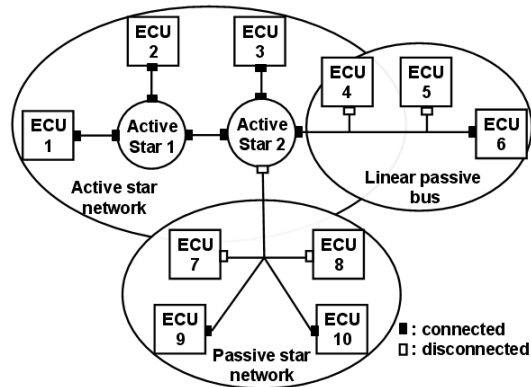


Figure 7. Example of a FlexRay network using star

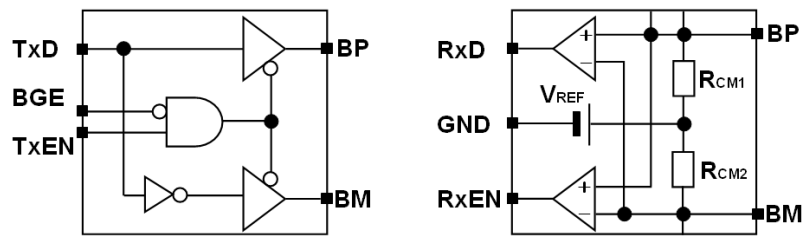


Figure 8. Bus transceiver schematic: (a) Transmitter principle schematic, and (b) receiver principle schematic

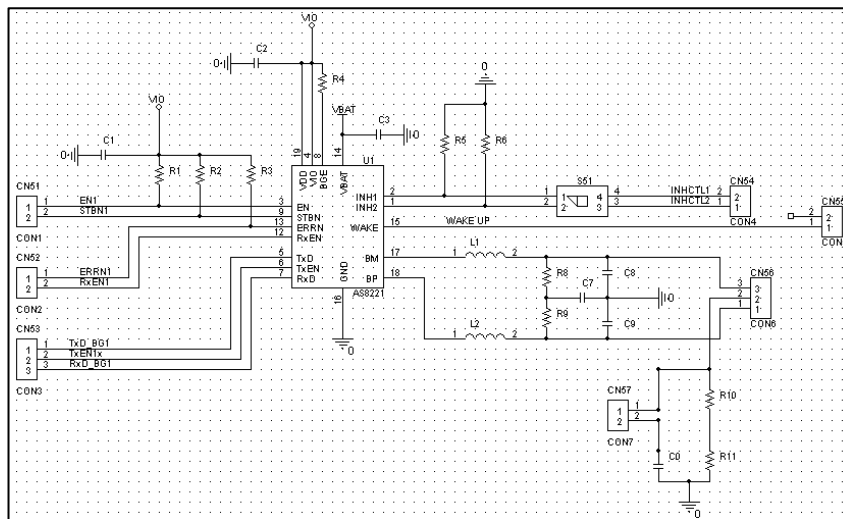


Figure 9. FlexRay active star circuit diagram

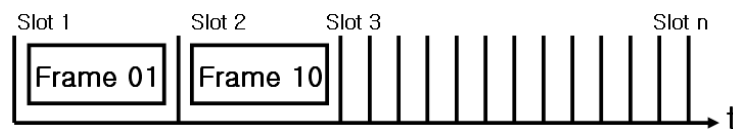


Figure 10. Allocated frame structure by slot

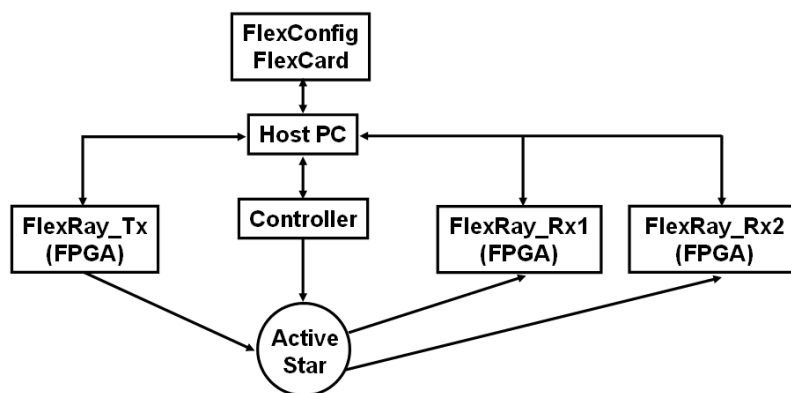


Figure 11. Block diagram of test environment

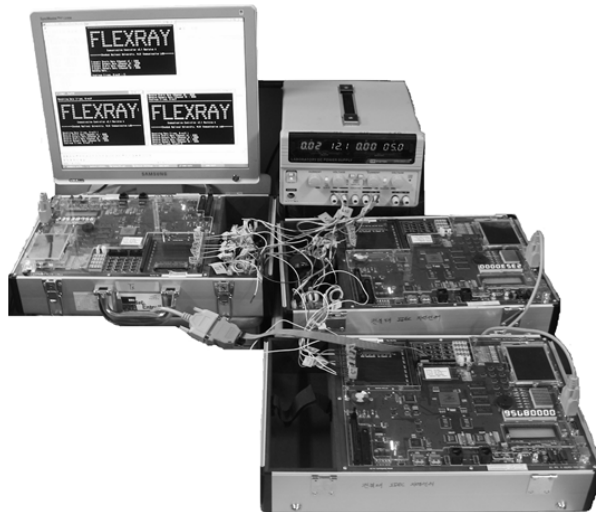


Figure 12. Actual test environment of FlexRay Network