The CCU25: a network oriented Communication and Control Unit integrated circuit in a 0.25 µm CMOS technology.

C. Paillard , C. Ljuslin, A. Marchioro

CERN, 1211 Geneva 23, Switzerland
Christian.Paillard@cern.ch

Abstract

The CCU25 is the core component of a newly developed field bus intended for the slow control, monitoring and timing distribution for the CMS silicon tracker.

As no commercial component could satisfy all requirements of radiation hardness, functionality and electrical properties, a new component has been developed.

Its architecture has been inspired by commercial token-ring type topologies. Each CCU25 contains various types of peripheral controllers and has dual network input and output ports allowing the cabling of a redundant network.

Inside the chip, critical circuitry is tripled and a majority-voting scheme is used to cope with errors caused by single event radiation effects.

The design was fabricated with a library in rad-tolerant 0.25 µm CMOS developed at CERN [1]. It contains approximately 50 000 cells and has 196 I/O pads for a die-size of 6x6 mm^2.

The detailed functionality is described and first prototype usage is reported.

I. INTRODUCTION

This paper reports on a circuit intended for the slow control, monitoring and timing distribution for the CMS silicon tracker.

The network architecture of the CCU25 has been inspired by commercial token-ring type topologies and is optimized for a ring network working at 40 Mbit/s. This is necessary as the network carries also the LHC synchronous clock information, the Level 1 trigger signal and the synchronous Reset to the front-end electronics. Each node in a ring is able to control various types of peripherals, such as 16 I2C master ports with several operating modes, a 4 byte-wide bi-directional static interface, a memory interface with 16-bit address and 8-bit data, a JTAG master and a trigger control block. The chip is therefore very flexible and usable as a generic embedded network controller in a number of different applications.

In a system, a 7-bit address is assigned to each CCU25 node thus allowing creating rings of up to 127 nodes.

II. RING ARCHITECTURE

The CMS tracker control system uses a ring topology configured as local area network. The architecture of a ring of CCU25s is functionally very similar to (and inspired by) the one used by commercial token ring networks (similar to the IBM Token Ring or FDDI) [2]. As shown in Figure 1 a module called FEC (Front-End-Controller) is the master of the network and uses two optical fibres for sending the timing and data signals to the slave modules (CCU25), which in turn use two fibres to transmit a return clock and the return data back. The ring consists basically of a number of node devices that are all capable of accepting and inserting packets in the ring. In the simplest implementation the ring consists of only two devices, the FEC and one embedded CCU25, thus resembling a point-to-point network. Due to the relatively long distance between the control room and the detector, the communication between the embedded electronics on the CMS tracker and the external electronics is expected to use a ribbon of four optical fibres. This data link is synchronized to the LHC clock frequency and has raw capacity of 40 Mbit/sec. The clock line provides the synchronization of the FEC and the embedded electronics. CCU25s are typically mounted on Control Modules (CCUM), housing the necessary ancillary electronics, such as line drivers, receivers and level translators. To minimize cost of the slow control system, it is expected to connect serially a certain number of Control Modules on local parts of the detector, resulting in a ring-like arrangement as shown in figure 1.

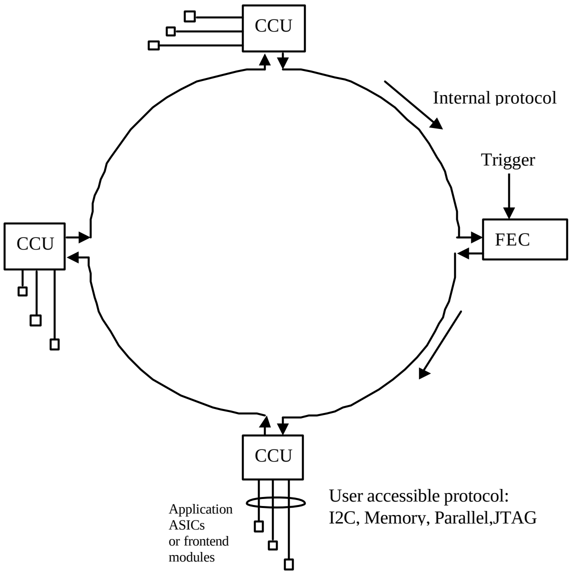

Figure 1: CCU ring architecture

III. COMMUNICATION ARCHITECTURE

The communication architecture used by the CCU25 is based on two layers:

The first layer (called the Ring) connects the FEC to CCU25s and the CCU25s between themselves; the protocol on this layer is message-based and is implemented in a similar way to standard computer LAN networks. The protocol used on this layer will be called the Ring Protocol.

The second layer connects the CCU25 itself to front-end chips or modules via so called Channels. The protocols used here are called the Channel Protocols.

The first layer is unified and common to all CCU25s, and is based on a LAN architecture transporting data packets to and from the FEC and "Channel" controllers. The second layer is specific to the channel, and different kinds of physical implementations of the channels are foreseen.

As shown in figure 2 the CCU25 contains the following blocks:

- One node controller (the CCU25 control itself is seen as a special channel capable for instance to report the status of the other CCU25 channels)
- Sixteen I2C master channel controllers
- One memory-like bus controller to access devices such as static memories, A/D converters etc.
- Four 8 bit parallel I/O interfaces
- One trigger distribution controller
- One JTAG master controller

The dual network layer architecture introduced above is necessary to support applications where long cables/fibres are used between the FEC and the CCU25s (therefore generating long delays) and to support the relatively slow buses chosen to interface to the front end chips, such as the I2C bus. This architecture assumes that the control is done by sending data packets (messages) to the respective channels, which interpret the messages as commands, execute them on their external interfaces (for example just a read or write operation to a memory bus) and conditionally return a status reply to the FEC via another message.

This protocol assumes that the remote devices controlled by the CCU25s are seen from the FEC as remote independent channels, each one with a particular set of control registers and/or allocated memory locations. The channels can operate independently from each other to allow concurrent transactions.

The high-level ring network layer, being a local area network-like protocol, is controlled through the FEC by software running on an appropriate microprocessor.

To decouple the operation of the channels from the one of the ring, the architecture assumes that all operations on the channels are asynchronous and do not demand an immediate acknowledgement. Basically this means that all commands carried by the ring in the form of network messages are posted to the channel interfaces. This is easy to implement for write operations, where practically it works by posting write operations to the channels. For read operations a read request is instead sent to the channel using a request packet; the channel performs the operation on its interface and returns a message to the requester using a separate packet. Broadcast operations are supported in a similar manner. Only write broadcasts are supported.

The handling of error conditions is largely a responsibility of the software driver running on the FEC. For simplicity, the CCU25 hardware has to be able to recognize error conditions, but all correction actions are always taken in software by the driver running on the FEC.

IV. INTERNAL STRUCTURES

Figure 2 show the internal structure of the CCU25.

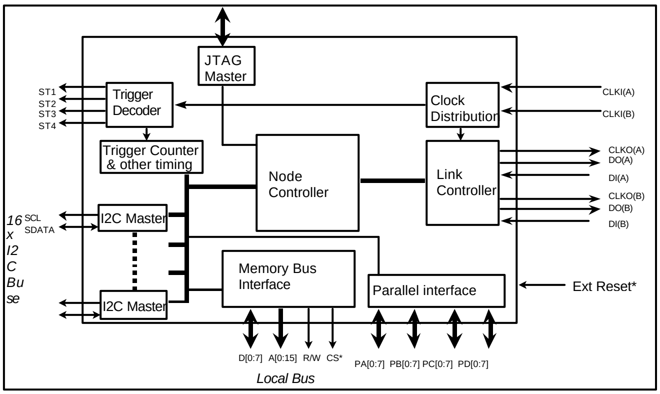

Figure 2: Block diagram of the CCU25

Channels receive commands from the node controller and control the actions on the bus to which they connect as masters. These commands are contained in the data part received by the link controller. The data content of a given packet is interpreted differently according to the channel to which it is addressed.

The command sub-field (typically the third byte in a message to a channel) contains the code for the requested operation. Each channel type has a set of valid commands; channels receiving an invalid command do not execute any action and they report the error condition to the node controller. Upon reception of a command, a channel performs the required operation on its interface. Depending on the particular command it can then return a reply to the node controller as a data block which will be transmitted to its destination in the ring under the form of a ring message packet.

The distinction between channels at the level of the FEC is performed essentially by software.

A. Node controller and link controller

The CCU node controller is a dedicated logic block inside each CCU25, which is needed mainly for network and internal channels supervision. The CCU controller is handled with the same protocol used to transfer data to the other port channels. These blocks contain several status and control registers for monitoring and configuring link operation: error counter, parity error counter, enable alarm, etc.

B. Master I2C interface

The I2C bus [3] is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL).

The operations performed by the I2C interface are:

- single byte read-write with normal 7-bit addressing
- read-modify-write single byte to address with mask
- special single byte read-write with "7+ 8-bit address" for APV6 [4] front end chip
- extended read-write with 10-bits addressing
- 16 independent I2C channels are available in the CCU25.

In the write cycle two modes are possible : with acknowledgement or without acknowledgement to the FEC. The transmission speed is programmable to 100 kHz , 200 kHz, 400 kHz and 1 Mhz.

The status register indicates the status of the previous transaction, the transaction number of the last successful operation, the number of the last incorrectly executed transaction, and the status of the data line.

C. Parallel interface

The parallel I/O bus channel is an interface, allowing parallel connections with programmable direction in groups of 8 bits. Input strobe and output strobe are also available with programmable width. Four independent parallel adapter channels are available in the CCU25.

D. JTAG Master

A simplified JTAG [5] master channel is implemented in the CCU25. The JTAG master generates three signals i. e. TCK, TMS and TDO, where TCK is the scan chain clock, TMS the mode control bit for the JTAG slave state machines and TDO the serial data sent to the scan chain. Data is returned on the TDI line. The JTAG controller isn't autonomous and part of the protocol is simply implemented in software. A JTAG packet consists of the normal header information of Destination, Source, Length, Channel Number and Transaction Number; then follows the Data and CRC16. Each byte of data contains four bits of TMS and four bits of TDO. This permits sending JTAG sequences of up to a length of ~8K bits. This is accomplished by packing 4 bits each of TMS and TDO per data byte.

E. Memory bus interface

The Memory Data bus implemented on the CCU can address a 64KB memory through a 16-bit address and 1 byte wide data interface. It can perform single and multiple byte read-write operations as on a normal byte-wide memory device. The operations foreseen are:

- single byte read-write to address
- multiple (up to 2K) bytes read-write to address with address auto-increment
- read-modify-write single byte to address with mask

Two programmable windows are provided, corresponding to 2 decoded output signal for external devices.

Several registers control the operation of the memory bus channel.

F. Trigger distribution

The trigger distribution logic block is a channel dedicated to the distribution of the four special trigger (ST1-ST4) signals. The special trigger signals are generated from the input clock and the level 1 trigger (T1) to the CCU25 and are distributed to the trigger destination front-end ASICs. The ST1-ST4 signals can be delayed a number of cycles by programming a delay value in a control register.

G. Alarm and reset

The CCU25 provides the basic hardware necessary to handle alarm conditions from front-end chips. Enabled alarms generate a special packet from the CCU25 to the FEC. The software running on the FEC must find the source of the alarm condition and reset it. The ALARM lines are level sensitive, the CCU node controller monitor the lines and sends out alarm packets to the FEC. When the alarm packet has been successfully transferred, or the maximum retry conditions reached, the corresponding enable is cleared.

V. PACKETS FORMAT

The packet contains two different types of information, one for signalling availability of the ring and the second to actually carry data.

The network data packet format is defined in Table 1:

SOF	**Destination**	**Source**	**Length**	**Data**	**Crc-16**	**EOF**
[1B]	DAddress [1 B]	Address [1 B]	[1 or 2 B]	[<128B or <32K B]	[2 B]	[2 B]

Table 1: Network data packets format

The Start of Frame (SOF), End of Frame (EOF), Source, Destination, Length and CRC fields are mandatory for all circulating data packets. The SOF field is defined as the unique “J-H/K” sequence, using the two special characters. The Data field is not interpreted by the ring protocol and is used exclusively by the channel adapter for internal addressing and data. These fields are explicitly defined by the functionality of each channel.

The data portion of the packet is shown in Table 2:

CH#	TR#	Channel Specific Command
[1B]	[1B]	[(Length-2) B]

Table 2: Format of the data field

Two data bytes are mandatory as payload at the beginning of each data packet:

- a channel number (single byte field), used to identify a device channel within a node

- a transaction number (single byte field with wrap-around) used to assure correct identification of an operation within a given channel. This field is always generated by the initiator of a transaction. For transactions initiated by the FEC, the transaction number should always be in the range of 1-255, because the special transaction number 0 is used for Alarms generated by the CCU25s.

The cyclic redundancy Check (CRC-16) field covers the packet content from the Destination address to the end of the Data field. The polynomial used for the CRC-16 calculations is: $X^{16} + X^{15} + X^{2} + 1$

VI. REDUNDANCY

As each ring could control a sizable number of front-end channels it is important to be able to guarantee a very high reliability for the ring as a whole. One malfunctioning element in a control ring will mean the loss of control of too many detector elements and it would clearly be unacceptable.

Figure 3 shows a redundant interconnection scheme based on doubling signal paths and bypassing of faulty CCU25s. A sames schema is used between the CCU25 and the FEC.

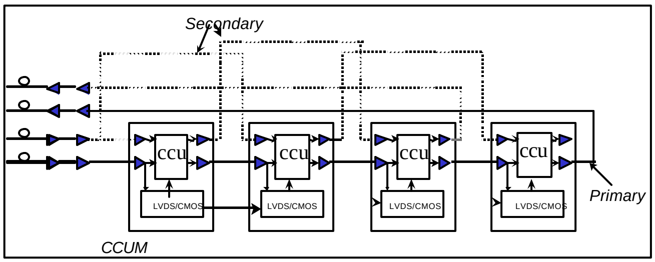

Figure 3: Primary and secondary interconnect scheme allow bypassing of a faulty CCU25.

VII. RADIATION TOLERANCE FEATURES

To be able to operate in the CMS tracker environment, the CCU25 is designed with rad-tolerance features. These features include:

- rad-tolerant library cells, for total dose tolerance
- redundant circuitry on critical logic blocks for SEU robustness

Redundant circuitry using three identical copies of the same logic, plus majority voting logic is used for the node controller. Because this block is critical for the correct functioning of the CCU25, it is included three times in the chip and all its output signals are selected after a voting circuit (purely combinatorial), which takes as valid output the signal for which at least 2 out of the three blocks agree. To make the rest of the logic blocks more robust against single event upsets, the following circuit features are also implemented:

- all data paths are protected by parity
- all finite state machines are encoded as one-hot circuits

These two features allow the logic in the CCU25 to identify nodes which have been corrupted by an SEU event. Whenever such an error occurs, the corresponding logic block aborts execution of the current operation and signals this event to the node controller. These features are sufficient to guarantee that the CCU25 will not perform a wrong operation on one of the slave chips attached to one of its channels.

VIII. DESIGN METHODOLOGY

Figure 4 shows the chip, which was completely synthetised from high-level Hardware Description Language (HDL) using the Synopsys [6] tools. The first prototype was implemented in a AMS [7] library (0.6 µm non rad-hard). A large part of source HDL was re-used for the synthesis on 0.25 µm rad-hard library [1]. In a design of this size (6 x 6 mm^2) the clock distribution is a extremly delicate part, and must be treated carefully. The clock tree has been generated with the CTGEN tool of Cadence [8] after placement of all 50 000 cells. The clock tree contains 5000 leaf nodes. The CAE tools estimated a skew of approximately 100 ps and an insertion delay 1 ns for.

IX. IRRADIATION TESTS

A complete chain of transmission has been installed at the Paul Scherrer Institute (PSI).

Sixteen CCU25s have been irradiated using a 300 MeV proton beam ($3x10^8$ proton/sec.). The total dose received was 1.4 Mrad in 33 hours. The number of errors detected was 4.58 SEU/chip/hour.

Estimation for LHC:

SEU/chip/hour @ LHC 4.21e-2

If 2000 CCU25 -> 84 SEU/hour worst case position in LHC

X. PROTOTYPE USAGE

Several CCUs (the rad-soft version) have been installed in a test beam [9] and a complete acquisition and control chain has been assembled and operated successfully.

Several CCUs are now in use for testing hybrid and silicon detectors in institus of the CMS collaboration.

XI. CONCLUSIONS

The CCU25 circuit combines two functions; the network interface and the frontend control interfaces. The network part carries also the timing and trigger information for the experiment.

The first test on IC tester (IMS) give a yield of 90 % on 200 circuits tested up to 80 Mhz clock frequency.

The CCU25 has been sucessfully integrated into system-level setups and beam tests.Radiation hardness has been demonstrated to be sufficient for the CMS application.

Detailed documentation is available [10] for the CCU and the associated software [11] for the FEC.

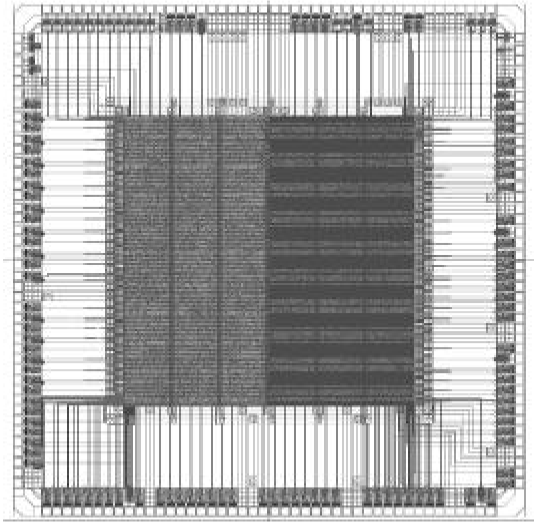

Figure 4: CCU25 Layout

Size 6x6 mm2

Package 196 pin bga

Power 250 mW at 2.5 V

XII. REFERENCES

[1] K. Kloukinas, F. Faccio, A. Marchioro and P. Moreira, "Development of a radiation tolerant 2.0V standard cell library using a commercial deep submicron CMOS technology for the LHC experiments" *Proc. of the fourth workshop on electronics for LHC experiments,* pp. 574-580, Rome, 1998

[2] IBM Token-Ring Network Architecture Reference, IBM Publication SC30-3374.

[3] "The I2C-BUS specification", Philips Semiconductors, Version 2.1, January 2000

[4] 'The APV6 Readout Chip for CMS Microstrip Detectors', M. Raymond *et.al.*, Contributed Paper to the LEB Conference 1997, London.

[5] IEEE Standard Test Access Port and Boundary-Scan Architecture, IEEE Std 1149.1-1990.

[6] Synopsys, Inc. Mountain View, CA 94043

[7] AMS austria micro system

AG A8141 Schloss Premstatten Austria

[8] Cadence Design Inc., 555 N. Mitilda Ave., Sunnyville, CA 94086.

[9] The CMS Tracker front-end and control electronics in an LHC like beam test

[10]CCU25 Manual

http://cmstrackercontrol.web.cern.ch/CMSTrackerControl/documents/Sandro/CCU25Specs%20v2-1.pdf

[11] FEC driver specifications

http://f.home.cern.ch/f/fdrouhin/www/#FEC%20Device%20Driver

.