

# PEP II 16-CHANNEL CORRECTOR CONTROLLER USING BITBUS

R. Olsen, Stanford Linear Accelerator Center, P.O. Box 4349, Stanford, CA 94309

## Abstract

A 16-channel controller has been implemented to control the PEP II corrector switching power converters. The design and performance are discussed.

## 1 INTRODUCTION

The PEP II injection and storage rings require more than 1000 corrector magnets. In order to economically meet this requirement, a new power converter termed MCOR12 was designed[1]. It is equipped with 16 switching power converter modules rated at +/- 50 volts, 12 amperes. A bulk power converter supplies the raw DC power to a number of MCOR's. The 16 switching converters are housed in a 17-slot, 6U Eurocard crate with non-standard module spacing. The 17th slot is standard single-width, and is occupied by the controller module. Control of each power converter is by means of a control voltage input with a range of +10 to -10 volts for full scale current of each polarity. Readback is provided by primary and secondary shunt circuits with outputs of the same voltage range. A status bit is provided from each converter to indicate that the converter is on-line, and the controller provides a reset and inhibit function common to all converters.

## 2 DESIGN REQUIREMENTS

A fundamental requirement of the controller was that it be able to ramp the 16 channels simultaneously on demand using a cosine function after the appropriate control information had been downloaded. This requirement, to a substantial degree, determined the architecture of the system and the use of a DSP to perform the main control functions. Communication was to be at a high level, and setpoints would be specified using current and timespan. Accuracy of the control voltage outputs and shunt readbacks was to be 0.1% minimum.

## 3 COMMUNICATION PROTOCOL

Communication with the controller is by means of BITBUS, a subset of SDLC [2]. The individual packet length is limited to 43 bytes due to limitations in the 8044

on-chip SRAM. Superimposed on the basic protocol is a high-level protocol devised by the SLAC controls group [3].

This protocol provides the mechanism to send commands to multiple channels in each message, and to send multiple packets in a single message block. The protocol is designed to make maximum use of available bandwidth in the signalling channel. In order to effectively synchronize the ramp function between multiple controllers, hardware ramp start, ramp stop, and reset lines are included in the BITBUS interconnect cable.

## 4 ANALOG CONTROL VOLTAGE OUTPUTS

Severe constraints on available volume to package the requisite functions dictated that one DAC per channel, although the simplest configuration, was not a viable alternative. Thus, the configuration chosen was a single 16-bit DAC feeding 16 individual sample and holds through a 16-channel isolation switch. The 16-channels are refreshed once per millisecond using a serial port on the TMS320C31. The serial data stream contains a 16-bit data field, a 4-bit address field, and a 4-bit control field. The control field selects the main DAC or one of two 12-bit DAC's used to adjust the offset and gain of the main DAC. The data stream is demultiplexed by a pASIC which contains all the logic necessary to control the refresh and calibration functions.

## 5 ANALOG READBACKS

### 5.1 Data Acquisition

There are a total of 56 readback channels in four blocks. These are:

- 1) 16 control voltages
- 2) 16 main shunts
- 3) 16 secondary shunts
- 4) 8 calibration functions

These signals are carried through 16 to 1 FET switch multiplexers, to an instrumentation amplifier. This amplifier feeds the selected signal to a CS5016 analog to digital converter. This ADC is a self-calibrating stand-alone system which is able to maintain high accuracy over a wide temperature. Control of the analog data acquisition is under control of a subroutine executed in ISR1 at the 1 ms. interval. One sample is acquired each millisecond. Multiple samples are acquired for each channel. To capture all channels requires approximately 2 seconds.

---

Work supported by the Department of Energy under contract No. DE-AC03-76SF00515

## 5.2 Calibration and Monitoring

In addition to sampling the 48 signal channels, the following parameters are converted

- 1) + reference
- 2) reference inverted
- 3) ground
- 4) control DAC output
- 5) reference sample and hold
- 6) ground fault setpoint
- 7) ground current
- 8) bulk power converter voltage

By using a combination of the first five parameters, the offset and gain of each channel is dynamically controlled. Since the control voltage outputs must remain fixed, a 17th sample and hold is provided such that a first order correction may be performed based on the gain of the sample and hold which is approximately 0.9997

## 6 PROCESSORS

The controller uses two processors. An INTEL 8044 is used as the BITBUS slave, and a TI TMS320C31 is used to provide control and computational functions and to refresh the sample and holds.

### 6.1 TMS320C31 Main Processor

The TMS320C31 has a number of features which made it an attractive choice. It is a 32-bit floating point processor with a 32 Mhz clock with an instruction cycle time of 120 ns. (in this application). It has 4 external hardware interrupts, an on-chip boot loader, a serial port, two timers, a DMA channel, and 2K of on-chip SRAM. All of these features were used in the implementation of the multi-channel BITBUS controller. The front panel holds 17 LEDs which display status information and two 9-pin connectors for two serial ports for local control and status display.

#### 6.1.1 Basic System Organization

The system refreshes the sample and holds once per ms. by a service routine ISR1 on interrupt level 1 under control of TIMER1. All other periodic functions are performed by the same routine. ISR1 executes from internal SRAM and refreshes the sample and holds over the serial port using the DMA channel. In this way, bandwidth on the main memory bus is conserved. All memory access and control functions are performed by a pASIC.

### 6.1.2 Boot Loading the Controller

The controller is equipped with two 64K\*8 EPROM's in the region 400000h-41ffffh of 'C31 address space. In the current implementation, one holds the code for the 'C31 the other code for the 8044 although overlap is possible. Boot loading is initiated by a reset or power-up. After reset, the shared SRAM is mapped to f00000h, the 8044 is held reset, the 'C31 is in the boot loader mode, and an interrupt is applied to INT1 which initiates 8-bit mode to 400000h. Upon completion of the 'C31 load cycle, control is passed to a routine in the on-chip SRAM which remaps the SRAM to start at location 00h and branches execution to the SYSINIT routine in main memory. With the 8044 held reset, the 'C31 has access to its program store which is first cleared, then loaded with the executable code copied from the EPROM. The 8044 is then released from the reset state which gives it control of its program store and it then proceeds through its initialization.

### 6.1.3 Inter-Processor Communication

The two processors exchange messages by means of shared memory. The controller is equipped with 32K of SRAM in the 'C31 address space. The lower 32K bytes of 8044 data store is mapped into the top 8K words in the 'C31 address space. The 8044, of necessity, operates asynchronously. When the 8044 accesses the shared memory space, a hold request is issued to the 'C31 which replies with a hold acknowledge. Since it cannot be guaranteed that memory access will be granted in the time required by the 8044 bus cycle, a dummy read always prefixes any access. Since the 8044 only requires shared memory access to transfer blocks of data, a hold request will not be relinquished at the end of the current cycle. For each access to shared memory, a release timer is reset. If no further accesses are made, the timer will time-out in 340 us. Memory is normally released by writing to a specific memory location. The timer is provided to allow the 'C31 to regain control in the event of a failure in the 8044. Several hardware flags are provided which are used to synchronize message passing without the necessity of accessing shared memory.

### 6.2 I8044 BITBUS Slave processor

An INTEL 8044 functions as the BITBUS slave. This device consists of an 8051 core processor along with the serial port hardware and control firmware in an on-chip PROM necessary to implement the BITBUS protocol. It has a Harvard architecture, with a program store and a data store. The program store is read-only, and no instructions are provided which can write to it. The 8044 has no capability for either WAIT or HOLD states. In addition, severe constraints in on-chip ram restrict the message buffer size to two 43-byte buffers.

In this application, the program store is an SRAM which is loaded from the TMS320C31 during the boot loading process. In order to facilitate debugging using an emulator, the chip is configured to use external memory only. The executable code was copied from the on-chip prom and is loaded to program store during the boot loading procedure.

### 6.2.1 Basic System Organization

The 8044 runs under control of the INTEL DCX51 operating system. Four tasks are defined. TASK0 performs the serial I/O and the BITBUS protocol and is hidden from the user. TASK1 is invoked by TASK0 when a message has been received. It copies the message from the buffer in the memory pool RAM, saves it in 1 of 16 temporary buffers in data store, and deallocates the buffer to return it to the memory pool. TASK3 is invoked from the 'C31 by ISR1 to test for the presence of any waiting messages. If a message is waiting, it will be copied to the shared memory. TASK2 is invoked by the 'C31 when processing of the current message is completed. The reply is copied from shared memory to an assigned buffer in the memory pool, and a request is sent to TASK0 to send the reply message to the master.

## 7 INTERLOCKS

### 7.1 Power Converter Module Interlocks

The MCOR12 system is provided with a number of interlocks to assure safe and reliable operation. Each power converter module is equipped with a latching over-current and under voltage detector. An over-current failure in a converter is latched and the status bit returned to the controller is set

#### 7.2 Controller Interlocks

A daughter board is used to provide controller interlock functions. This board contains a ground fault detector and the controller fault latches.

##### 7.2.1 Under Voltage

The bulk power converter voltage is monitored and returned to the control system. In addition, the power converter must be reset with a minimum voltage present on the supply bus before the converter will become active

##### 7.2.2 Ground Current

Each MCOR12 has a ground current resistor. This current is monitored and returned to the control system.

##### 7.2.2 Ground Fault

The ground fault detector will latch a fault if this current exceeds a preset value.

### 7.3 System Interlock Handling

When an interlock occurs, the appropriate bit is set in the controller status word. The next message requesting device status will inform the system of the failure. A message RESET INTERLOCK may be sent to reset. If the reset is not reset, the affected channel(s) will remain off-line. A successful reset will return them to service.

## ACKNOWLEDGEMENT

The author wishes to express gratitude to Dave Brown, Mike Brown, Mark Crane, Linda Hendrickson, Tom Himel, Dave Macnair, Jeff Olsen, and Lee Ann Yasukawa for their able assistance in finding a number seemingly intractable problems. We also wish to express our appreciation to Len Genova for his continued support of this project.

## REFERENCES

- [1] G.E. Leyh, A.R. Donaldson, SLAC, Stanford CA 94309
- [2] "8-Bit Embedded Controllers", Intel order number 270645
- [3] Lee Ann Yasukawa, "PEP II BITBUS Power Supply Interface Control Document", SLAC internal document