CERN Trajectory Measurement System
System Overview

Project	CERN-TMS
Date	2007-11-02
Reference	Cern-tms/systemOverview
Version	1.1.0
Author	Dr Terry Barnaby

Table of Contents

1. Introduction

This document is intended to give an overview of the CERN Trajectory Measurement System (TMS). It covers the hardware components and software components of the system.

2. Design Features

The TMS system is designed to measure the trajectory of particle beam's within the CERN Proton Synchrotron. It is able to measure the amplitude and x/y displacement of the individual particle bunches as they pass each of the 40 analogue sensors in the ring. The system integrates the data received for each particle bunch and stores the results in memory for later data access. In order to accurately measure the particle bunches the system uses phase locked loops to synchronise the data capture to the incoming data.

The system continuously samples 120 Analogue channels at 125MHz, 14 bits and processes this data in real-time to determine information on the position of particle bunches as they orbit at around 437kHz. The system captures and processes around 15 billion samples per second. Multiple Xilinx Vertex 4 FPGA's are employed in a modular system to capture and process the data. The system is controlled over a Gigabit Ethernet network from which portions of the resulting data can be accessed.

The main design features of the system include:

- High degree of modularity. The system is split into 3 + 1 independent processing modules. Within each processing module there can be 5 PU processing engines each processing 3 of CERN's PU's.

Each of these PU processing engines (PUPE) has 9 analogue inputs, 1 digital clock input, 16 general purpose digital I/O ports and a large Xilinx Virtex-4 FPGA.

- High degree of board level component re-use allowing quick and easy swapping of faulty components.
- Large level of common of the shelf (COTS) components.
- Based on the flexible, powerful and recent Xilinx Virtex-4 FX FPGA chips providing leading edge firmware configurable, high performance data processing.
- Spare FPGA capacity for additional algorithms.
- The system has 256Mbytes of memory available per PU. The Module Controllers have 1 GigaByte of RAM and the System Controller has 2 GigaBytes of RAM.
- Open-source software for flexibility and ease of customisation.
- Relatively low power and hence reduced heat dissipation and reliability.
- Housed in an industry standard compact PCI rack system for a robust, off the shelf, enclosure system.

3. Overall System Design

The system hardware design has been based, as much as possible, on common of the shelf (COTS) components available from Alpha Data and other sources. There is one, board level, component that has been specially designed and manufactured for the system: the Pick Up processing Engine (PUPE). This compact PCI board is based around Alpha Data's **ADM-XRC/FX100-10/1G FPGA** PMC module's design. It has a Vertex-4 FX100 FPGA together with 9 analogue to digital converter's and digital timing/test signal interface circuitry. The system uses the industry standard compact PCI (cPCI) rack mounted bus system to house the main processing boards, power-supplies and provide fan cooling to the system's hardware. The system has the following connections:

Name	*Number*	*Description*
ADC Input	120 + 30 spare	Analogue signal inputs. 2 Volts peak to peak into 50 ohms. Sampled at 125 MS/sec at 14 bits.
10 MHz system clock	4	Master system clock. The ADC's 125 Mhz sampling clock is optionally synchronised to this clock and all of the digital timing signals, except the Injection signal, will re-synchronised to this clock within each FPGA. Positive TTL into 50ohms.
FREF Input	4	Reference frequency. Positive TTL into 50ohms. (437KHz)
CYCLE_START Input	4	Start of a machine cycle. Positive TTL into 50ohms.
CYCLE_STOP Input	4	End of Last Flat Top, effectively end of cycle. Positive TTL into 50ohms.
CAL_START Input	4	Start of calibration period. Positive TTL into 50ohms.
CAL_STOP Input	4	End of calibration period. Positive TTL into 50ohms.

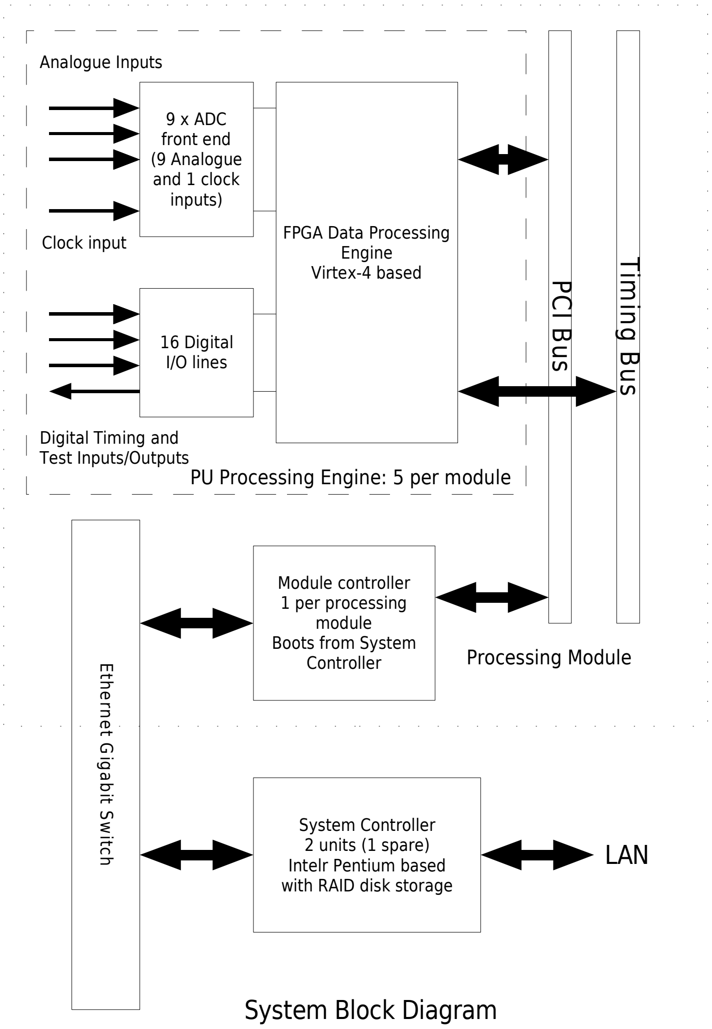
Name	*Number*	*Description*
INJECTION Input	4	Injection. Positive TTL into 50ohms.
HCHANGE Input	4	Harmonic changes. Positive TTL into 50ohms.
Spare Input	4	Spare digital inputs. Positive TTL into 50ohms.
Test outputs	51	Test signal outputs, one per PU. These will provide the synthesised FRef signal for test purposes but can be switched to provide certain other internal signals. 50Ohm Positive TTL outputs.
1000/100/10 base T Ethernet	2	Network connections to CERN's systems for control and data access. RJ45 connectors.

CERN's systems supply 4 complete sets of digital timing signals, all except for the injection signal, synchronised to the 10MHz master clock's positive edge. These signals are connected to the digital timing inputs of each processing module where they are distributed to each PU processing engine via a timing bus.

The system design is focused on providing 24/7 service with minimal down-time in the event of a component failure. In order to achieve this and to ease system maintenance, system development time and testing the system has been designed in a modular way. The system consists of 3 identical processing modules and one reduced processing module as a spare. Each of these processing modules has its own power supply and an 8 slot cPCI backplane. The cPCI backplane has a PCI bus for board communications. Housed within each processing module is a conventional CPU based module controller and up to 5 PU processing engines. Each of the PU processing engines has 9 ADC's and 16 digital I/O lines connected to a Virtex-4 FX100 FPGA based processing engine. Thus each PU processing engine can acquire and process the data from 3 of the Proton Synchrotron's pick ups (PU's). This architecture was chosen to reduce system cost while providing FPGA processing from one of the latest Xilinx FPGA designs available.

The PU processing engines are interconnected with an 8 signal, 16 wire timing bus. The first PUPE in a processing modules has an extra panel containing connectors for the external timing signals. This first PUPE engine is configured to transmit these timing signals to all PUPE's in the processing module over the timing bus. The timing signal bus consists of an IDC ribbon cable connected along the front panels of the PUPE's. There is also the future option for passing the digital timing signals using the J3 connector of the cPCI backplane.

The processing modules are controlled from a master system controller through a local Gigabit Ethernet switch. The system controller is used for booting the individual processing engines and overall system control, data access and management. There are in fact two system controllers for system redundancy. Remote systems communicate with the system through the system controller, the individual processing modules are on a separate virtual or perhaps physical Ethernet based network.


Analogue Inputs
9 x ADC front end (9 Analogue and 1 clock inputs)
Clock input
FPGA Data Processing Engine Virtex-4 based
16 Digital I/O lines
Digital Timing and Test Inputs/Outputs
PU Processing Engine: 5 per module
PCI Bus
Timing Bus
Module controller 1 per processing module Boots from System Controller
Processing Module
Ethernet Gigabit Switch
System Controller 2 units (1 spare) Intelr Pentium based with RAID disk storage
LAN

System Block Diagram

4. Physical Design

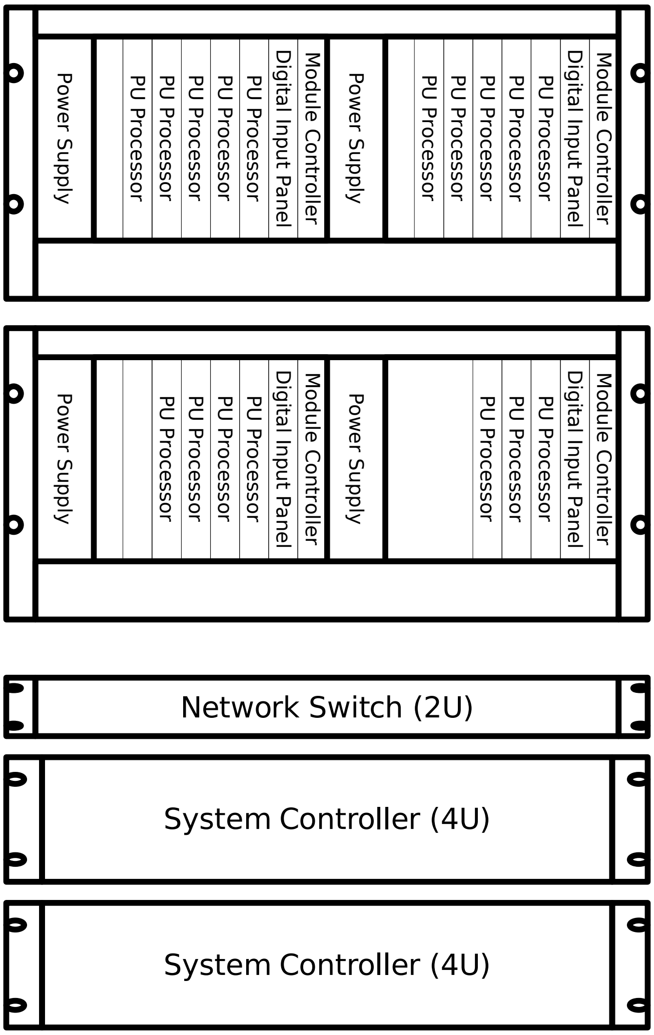
The system’s physical design is based on the industry standard 19inch rack format. The systems 3+1 processing units are housed within two 9U cPCI backplane enclosures. Each cPCI enclosure houses two separate processing modules each managing up to 15 PU’s. Each processing module has an independent power supply, set of fans for cooling and a module controller. The module controller is linked to the system controller using a Gigabit Ethernet interface.

Three of the processing modules implement the required 40 PU processors (actually 42). An additional processing module of 3 PU engines is spare, allowing either an individual PU processing engine to be replaced or a complete processing module to be replaced. The spare processing module can be left powered down for cold spares or powered up for hot spares. This spare processing engine could also possibly be used for testing new algorithms while the system is in use. There are spare cPCI backplane slots available in all of the processing modules for hot spares or additional processing or test modules.

Below the main processing modules is a Gigabit Ethernet switch that handles communications between the

System Controller and the individual processing module controllers.

There are two System Controllers for redundancy. Each of these controllers have twin, long life SATA disks in a RAID configuration and run with identical software.



Rack System

5. PU Processing Engine (PUPE)

The complete system has 17 PU processing engines each housed on a cPCI board. Fourteen PU processing engines are needed to support the 40 PU's and three boards are available as cold or hot spares. The PU processing engine consists of a custom cPCI board on which is mounted the Virtex-4 FX100 FPGA, ADCs and digital I/O interface components.

The design makes full use of the Virtex-4 FPGA to provide a flexible hardware design solution that can be tailored in the FPGA firmware design.

Full details on the PUPE’s design is contained within the pupeBoardDesign document.

The PUPE boards have been designed with built in Gigabit Ethernet interfaces. These provide the ability to use Gigabit Ethernet for all inter-board communications in future developments.

6. FPGA Firmware

The FPGA firmware is the heart of the system. It implements the major real-time data capture and processing within the FPGA fabric.

Full details of the FPGA firmware design is in the TmsPupeFirmware document.

7. Processing Module

Each processing module is independent of other processing modules. It consists of an 8 way cPCI backplane with power supply, a conventional CPU based module controller and 4 or 5 PU processing engines.

7.1. Module Controller

The module controller is a COTS component. It has a conventional low power Intel x86 Core Duo CPU, some boot FLASH memory, 1 Gigabyte of RAM, a cPCI bus interface and triple Gigabit Ethernet ports. The actual unit to be used is the Concurrent Technologies PP 410/03x.

The module controller boot's from the main system controller over the Ethernet interface and runs a small Linux based operating system. It is responsible for booting and managing the 5 PU processing engines (15 Proton Synchrotron PU’s). Communications between the system controller and the individual PU processing engines is also handled.

8. System Controller

The system controller is a standard Dual Intel Pentium Xeon based computer system. It is housed in a separate 4U 19” rack enclosure. The system controller has 2 Gigabyte’s of memory and dual SATA disk drives in a RAID configuration for disk redundancy. These disks contain all of the TMS’s software, FPGA firmware and configuration information. The system controller has dual Gigabit Ethernet interfaces, one connected to the Gigabit switch that communicates with the processing module’s controllers and one connected to the sites LAN for remote access to the system.

The system controller does not need a monitor, keyboard or mouse although we have provided these for use if required. All system configuration and maintenance can be carried out over the Ethernet network. The system supports the IPMI over LAN control interface for managing low level BIOS access if needed for complete software re-installation.

The system controller runs the Linux operating system.

Two identical system controllers are provided for system redundancy. The second controller can be left powered down and then booted when required or left continuously running. Remote applications will be able to connect to either controller based on the IP address. Switching between the two system controllers is handled by configuring there network interfaces appropriately.

As well as providing a control and data interface to the Trajectory Measurement System, the software on the system controller implements a system boot, system configuration, system test and fault diagnostics functions. This are made available to operators via a web based interface as well as through a command line API.

9. Test Signal generator

As part of the system design and development process we have produced a simple analogue and digital test signal generator. This is based on an 8 channel 150MSample/second arbitrary waveform generator PCI board. We have used the Chase DA8150-12-2M-PCI, http://www.chase2000.com/ board. A simple software application, tmsSigGen, has been developed to drive this board with appropriate PS test signal patterns. We can drive the Sigma, DeltaX, DeltaY, SYSTEM_CLOCK, FREF, CYCLE_START, INJECTION, H-CHANGE and CYCLE_STOP signals with the signal generator.

10. System Software and API's

All of the system software is based on the Linux operating system. This provides a reliable an flexible system that can be easily maintained locally and remotely. All of the software is Open Source and thus all source code is available. All of the system's special software will be available in source code form with an Open Source license.

All communications is through the system controller which supports a simple API to control and gather data from the system. The system controller will interrogate the individual PU processing engines via the local Gigabit Ethernet network and the module controllers. CERN can control and acquire data across the network interface from a remote system via the network based API or install their own programs on the system controller which will communicate with the system using the same API.

The system controller's API will accept cycle information from CERN's system describing each Proton Synchrotron's machine cycle. This information allows the FPGA configuration to be correctly set and the data captured to be tagged with the appropriate cycle information. This information is distributed to all of the PU processing engines along with configuration data such as the position of the PU engine within the Proton Synchrotron's ring.

The full design of the system software is given in the Tms Software document.