TESTING DIGITAL ELECTRONIC PROTECTION SYSTEMS

S. Gabourin, A. Garcia Muñoz, CERN, Geneva, Switzerland

Abstract

This paper outlines the core concepts and realisation of the Safe Machine Parameters Controller (SMPC) test-bench, based on a VME crate and LabVIEW program. Its main goal is to ensure the correct function of the SMPC for the protection of the CERN accelerator complex. To achieve this, the tester has been built to replicate the machine environment and operation, in order to ensure that the chassis under test is completely exercised. The complexity of the task increases with the number of input combinations. This paper also outlines the benefits and weaknesses of developing a test suite independently of the hardware being tested, using the "V" approach.

INTRODUCTION

The SMPC ensures the correct configuration of the LHC machine protection system, and that safe injection conditions are maintained throughout the filling of the LHC machine. The SMPC receives information in real-time from measurement electronics installed throughout the LHC and SPS accelerators, determines the state of the machine, and informs the SPS and LHC machine protection systems of these conditions.

The SMPC is built to ensure reliability and availability of the transmitted information. The reception of the information and the generation of the state of the machine are done in complete redundant VME boards. An arbitration module then determines the correct information to send from the redundant information provided. This redundancy ensures high system dependability.

At the same time, redundancy makes calculations and the overall system more complex and subject to numerous errors or unexpected behaviours that would not be an issue for a non-redundant system. For this reason, a dedicated tester, the Safe Machine Parameter Tester (SMPT), has been developed to identify possible weaknesses of the system, and to validate the correct function of the SMPC.

SAFE MACHINE PARAMETERS

Components

The SMPC (see Figure 1) is composed of three types of electrical boards, Receivers (CISR), Generators (CISG) and an Arbiter (CISA) all based on the same basic PCB design, but having different functionality thanks to different VHDL code in FPGAs.

For the LHC, there are two redundant CISR (CISRA and CISRB), each one decoding a pair of energies and a pair of intensities. These CISR pass information to 2 redundant CISG (CISGLA and CISGLB).

The CISG use energies and intensities from the CISR, in conjunction with flags directly received from LHC beam instrumentation, and other information provided by Ethernet, to derive energy, intensity, and a set of flags and values representing the LHC machine state. Some of these flags are transmitted directly to the extraction Beam Interlock System (BIS) ensuring the safe transfer of beam between the SPS and LHC machines [1].

Data from both CISG is sent to the arbiter board (CISA) which assembles the redundant information and determines the overall state of the system, before transmitting the information to client systems through the LHC General Machine Timing (GMT) .

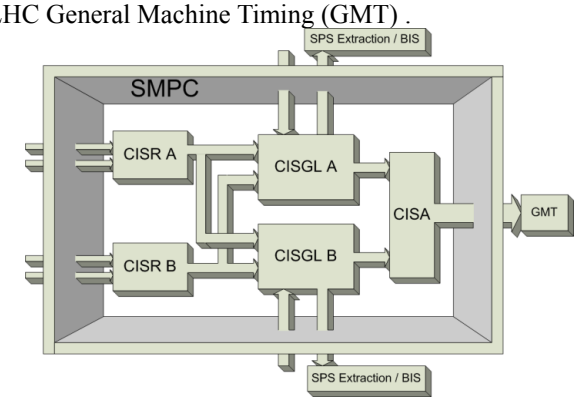

Figure 1: LHC SMPC and clients.

Each board has a pair of Field Programmable Gate Arrays (FPGAs). A "control FPGA" tasked with the execution of critical operations, in which the SMP mission-critical functions are implemented. And a "monitor FPGA" which does not interfere with the critical part, but receives information from it, formats and makes it available for observation and interpretation through the Java software supervision.

Functions

The main goal of the SMPC is to transmit energy and intensity values and to translate input information into states/flags. Principle flags are:

- The Beam Presence Flag (BPF), which evaluates to TRUE when a beam is circulating in the LHC. This is directly received by each CISGL and sent to SPS extraction and CISA.
- The Setup-Beam Flag (SBF), which evaluates to FALSE when the circulating beam is considered as dangerous for the tunnel equipment (magnets, etc) if any problem inducing the loss of the beam occurs.
- The Moveable Devices Allowed In / Stable Beams flags (MDI/STB) are used by the four main experiments (ATLAS, CMS, ALICE and LHC-b). The MDI flag is set to true when a specific energy is reached, when the beam is squeezed at each of the four interaction points, and when the appropriate beam mode is set. Similarly the STB becomes true when the beam mode is stable in addition to the MDI being true.

SAFE MACHINE PARAMETER TESTER

As the SMPC has a heavy dependency on external equipment, it is not possible to test it completely in the real machine environment, it receives information from and transmits information to a diverse set electronics in both the LHC and SPS machines. The SMPT reproduces the machine environment by sending all the different information to the SMPC, replicating the characteristics of each system connected to the SMPC. The SMPT is implemented in a separate VME crate, and a LabVIEW program supervises and controls the SMPT, as well as reading back the states of each element of the SMPC, checking if its behaviour is as expected.

The VME Crate SMPT

The task of the SMPT is to simulate all the inputs (energy, intensity, flags, etc), retrieve the output signals and cross-check the state of the SMPC versus the prescribed function in the specification. The SMPT uses the same hardware boards as the SMPC with different VHDL programs (see Figure 2):

- five CISTR are used to simulate energy and intensity frames (in yellow) sent to the CISRs, and the BPF sent to the back panel of the SMPC (in blue),
- a CISTA is used to read back and verify the output of the CISA which normally goes to the GMT (in blue),
- two CISTCL read the current loop flags directly sent by the back panel from the CISGL to the SPS extraction lines (BPF, SBF) and to the BIS.

On Figure 2, the SMPT VME crate is represented on top with the front panel outputs and inputs which connect to all SMPC connectors on the front panels (on bottom left) and back panel (on bottom right).

Fibres optical connections are represented in yellow, electrical ones in blue, and current loops in green.

LabVIEW Software

The SMPT is monitored and controlled through a LabVIEW program realising the actions required for each test. The front panel of the program, shown on Figure 3, represents the hardware state of each SMPC board, including their connections, and their internal states. A colour-coding is used to display the state of each element:

- dark gray for elements which are not tested (The majority of elements on picture 3),
- dark blue for elements which are currently or are going to be under test (The four SBF equations of CISGLB),
- green for elements successfully tested (the CISA energy and all its dependant tests) and
- red for elements whose test has failed (The intensity priority test in the CISGLA).

Three users' actions are possible:

- "Ctrl click" on an object (rectangle, label or frame) to launch this specific test. On picture 3, the SBF label of CISGLB was clicked to obtain the blue objects,
- "Ctrl shift click" to launch this specific test and all dependant ones. On picture 3, this has been done on the Energy object of the CISA which has induced the test of all green objects
- "Shift click" to show the diagnostic of a test which has been executed, this opens a window showing a breakdown of the test steps.

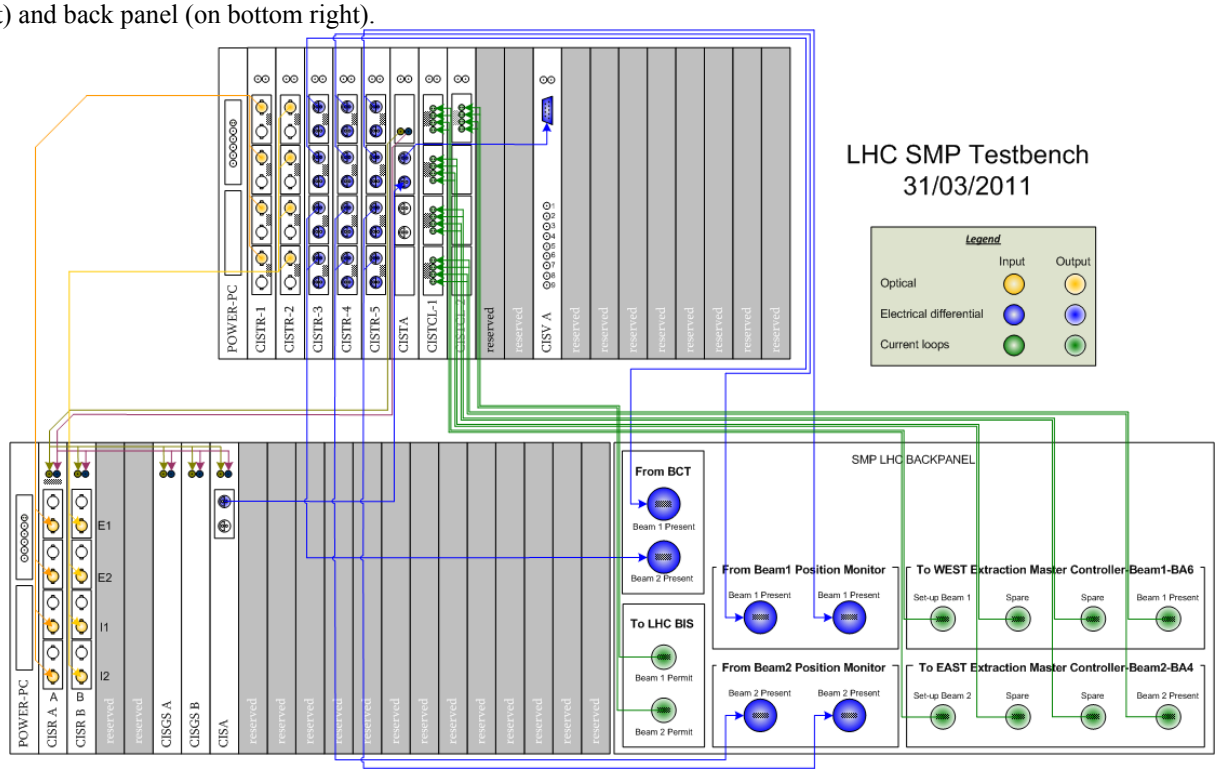

Figure 2: LHC Connexion between SMPT and SMPC.

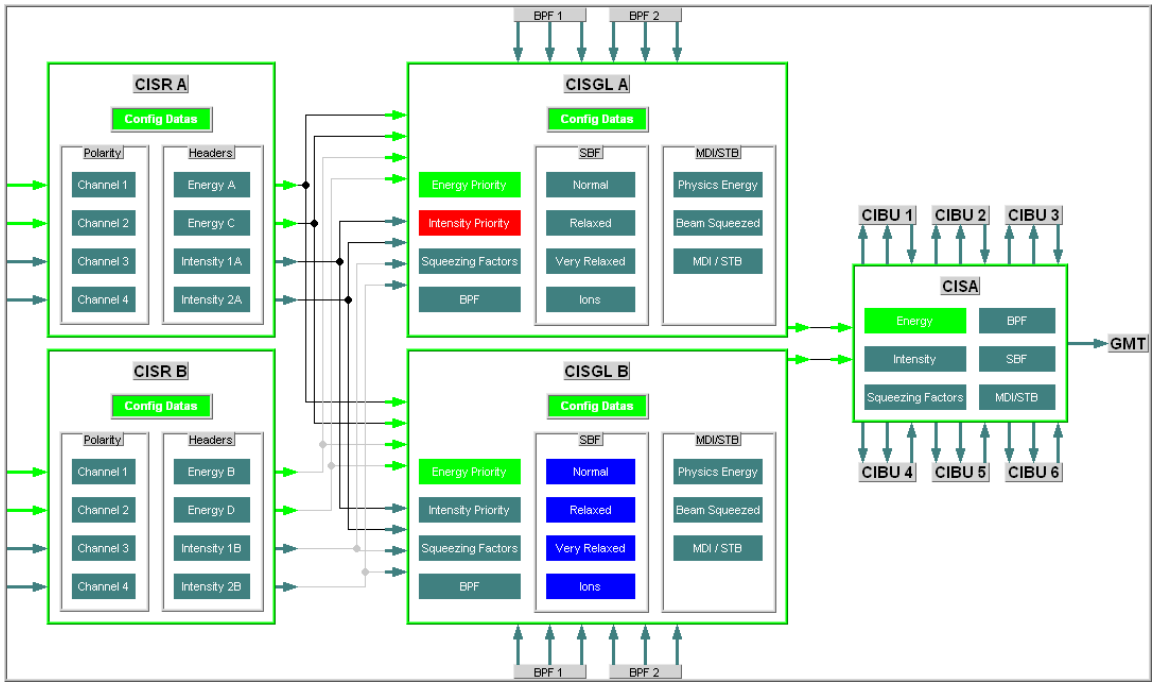

Figure 3: Front panel of the LHC LabVIEW tester.

FESA Layer

The communication between the program and the VME crates SMPC and SMPT is done through a CERN standard architecture called FESA. FESA is a C-based set of drivers developed at CERN to communicate with several devices, and it provides information which can be interpreted by several programming languages such as LabVIEW or Java.

In this application, FESA is used to access all registers and history buffers (HB) implemented in each monitor FPGA. Each HB is a 1024 record rolling buffer which contains specific events with their time of occurance and status, and which are written only when necessary by the mean of triggers inside the SMPC hardware. The registers and the HB allow information about the status, intermediate calculus and timing parameters of the different functionalities to be determined. This information is shown on the front panel of the LabVIEW tester in the picture 2.

The reading is done through simple requests or subscriptions with a refresh rate of one Hertz.

TEST METHODOLOGY

"V" Approach and Predicate Logic

The "V" approach is based on the desire to have the SMPC entirely reproduce to specification requirements. The SMPC was developed from the specification and, in parallel, a set of tests were implemented to validate the behaviour described in the specification, without taking into account how the SMPC was built. As a normal test system, the SMPT allows the validation of the behaviour of the SMPC with respect to the specification. This parallel development has the added advantage of highlighting any kind of ambiguity in the specification itself.

This "V" approach is a good way to consolidate the specification, at the same time, the initial specification was not written with this in mind. To avoid misunderstanding, misinterpretation or un-defined conditions, further effort was carried out to generate a documentation free of ambiguity, using Formal Methods (FM). FM are used to describe the behaviour of a system, independently of its realisation (hardware, software, abstraction...), even if there are several ways to realise a system, depending on the target where the logic is foreseen to be established. FMs in this case took the form of Predicate Logic, similar to a programming language, giving an exhaustive description of how the system has to work, with only a single interpretation.

For example, the SBF is calculated for each particle beam from several input conditions. There are four equations (Normal, relaxed, very relaxed, ions) which give an intensity threshold *THD* (i.e. a number of particles) for a specific LHC proton beam energy. If the LHC beam intensity is below this threshold, the SBF is true (the beam is a set-up beam), otherwise it's false.

For each beam, the SMPC receives two intensity values *IA* and *IB*, each one associated with a "valid intensity" flag, respectively *IA_valid* and *IB_valid*. The intensity value used for the calculation is the highest if both are valid. If not, the maximum intensity value (all bits to '1') is chosen as it automatically sets the SBF to false, which is the fail-safe value (as it is the more dangerous case). The SBF can be expressed as the expression:

$$SBF = (IA\_valid)\&(IB\_valid)\&(IA \leq THD)\&(IB \leq THD)$$

This expression is the basis used to build the SMPC and to define for the SMPT which tests can validate the SMPC behaviour.

Test Structure & Organisation

Full tests coverage is impossible to reach, as 428 bits are derived by hardware and software connections. Moreover, for the LHC SMPC, 82 32-bits registers can be written to change the system configuration and behaviour. This represents a total of 2^{3052} input combinations, without taking timing requirements into consideration.

The main approach to reduce the number of tests is to modify only the inputs which have effect on parameter under test. Moreover, complex functionalities can be tested in steps, by validating intermediate results first.

Tests involving thresholds are a particular case. The threshold and the threshold value plus one are fixed numbers which are always tested. But to be more complete, it's important to add tests of the intervals below and above the threshold. As all values cannot be considered, the solution kept is to choose a random value inside each different interval.

The test bench software has been built to facilitate the addition, modification or suppression of tests. The tests are not defined in the program itself but in an Excel file named "Tests to do" on Figure 4.

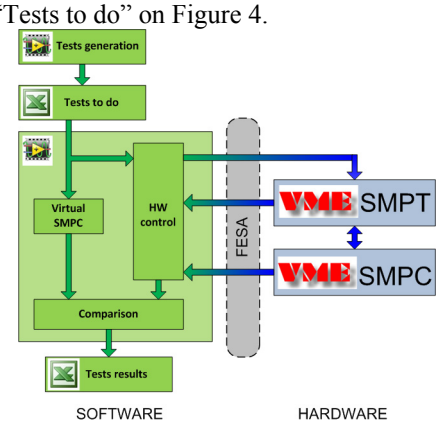

Figure 4: Tests organisation structure.

This file contains each possible elementary test divided in steps. A step is defined as a set of actions which are realised at the same time. Eight actions (all optional) are defined, each one in a separate worksheet:

- three write: for the hardware it can be frames or flags, for the software it is registers,
- four read: for the hardware it can be frames or flags, and for the software registers or HB records,
- one sets the delay before starting the next step.

The writing of this file requires a meticulous approach, so another LabVIEW program (Tests generation on igure 4) has been realised to facilitate the definition of tests.

Tests Sequence

As tests are defined by the program on-the-fly, it's impossible to know in advance what the expected results are. This is also due to some stimuli which have to be chosen randomly at the very start of tests. For this reason, it is mandatory to introduce a subprogram able to calculate at the end of each step the expected internal state (registers and HB) of every board. This "virtual" SMPC is built directly from the specification (as the real SMPC) and gives the possibility to validate the information gathered from the hardware.

The virtual SMPC reads the test steps and transforms them into an array which each cell is a combination of all states of all SMPC inputs for a specific time. The time granularity is 1ms, thus a 5 second test sequence generates an array of 5000 cells. The outputs signals and the FPGA memory are also simulated and calculated for each step, and then sent to the Comparison module.

In parallel, the test steps are read by another subprogram, the hardware control (HW control), which formats them into commands to be executed by the SMPT, waits for the real delays, and reads signals and memories of the SMPC and of the SMPT. All the gathered information is compiled to have a similar structure than the virtual SMPC results, and is also sent to the Comparison module.

The Comparison code compares the real and the virtual data and gives the results following the colour code of Figure 3 (green if all is the same and red otherwise). In case of failed test, it's possible to open the diagnostic window of the program; the results are also saved in an Excel file (Tests Results on Figure 4).

FUTURE PLANS AND CONCLUSION

The main improvement concerns the reduction of time consumption for tests, as well for the SMPC than for the virtual SMPC. In total the SMPC needs more than 5 hours to be tested; the virtual SMPC needs 20 minutes. Presently, the SMPC communicates with the hardware using FESA, and needs to wait for the data to be refreshed before readings. This refresh occurs every second, asynchronously with the test process. Then the time to wait before having new data is between a few milliseconds and a second. Instead of using FESA, LabVIEW could connect on the VME crate through a SSH connection and read the memories thanks to low-level programs which already exist for manual operation. This would allow an improvement in the time for testing the SMPC. Concerning the virtual SMPC, the actual method is to simulate iteratively the memories and outputs at every millisecond, which becomes huge for tests of more than 1 minute. This can be replaced by an equation independent of time, capable of evaluating the system state in a single calculation, given all inputs. In this case any test would complete within a few millisecond. Such an equation has already been defined for the CISR.

To conclude, FM and "V" approach ensure having specification out of ambiguity and a system which behaves as specified. These techniques are particularly efficient as the system is more complex, as for the SMPC version 3.1, currently in use in the CERN control room.

REFERENCES

[1] B. Todd et al. "The architecture, design and realisation of the LHC Beam Interlock System", ICALEPCS'05, Geneva, Switzerland, October 2005.