June 3, 2002

Manual for COMET (COMpressor for Electron Tracks)

Data compressor for the 2560 wires of the electron MWPCs used in the Singlet μp Capture in a Hydrogen TPC : R-97-05.20

René Prieels, D. Michotte, L. Bonnet (UCL)

Contents

1 General overview

The aim of this note is to present the system COMET developed in Louvain to compress the data produced by the 2560 wires of the electron MWPCs surrounding the muon TPC. The localization of the crossing points of the e^- (e^+) with the chambers may help in the vertex reconstruction of the decaying muon and in the discrimination of the signal over the background. The cathode and anode wires, 2560 in total, are arranged in 2 times 3 planes.

All the wires will be continuously scanned by 20 compressor modules each handling 128 wires. The action of these modules is to collect only the hit wires in order to reduce the data flow in the acquisition process. The 128 wires feeding one compressor are physically grouped in 8 times 16 wires, then logically in 4 times 32, half of them being treated by one FPGA chip. One compressor module consequently contains 2 such FPGAs. We call A and B the two groups of 32 bits analyzed by within one FPGA. When at least one wires is hit in A or B group, the 32 bits belonging to the group, and a 32 bits label containing the A or/and B code, the FPGA address and the time of detection are written in the internal FIFO of the FPGA.

A controller module continuously empty the content of all FIFOs (40 since there are 20*2 FPGAs) and send the corresponding data to the VME SIS3600 multi-event FIFO input register (see Fig. 1). At half full a signal from the SIS3600 is sent to the compressor controller which inhibits all compressor's action. This half full signal is dispatched externally to the DAQ which in turn takes over the inhibit signal during the readout of the SIS3600 FIFO. The inhibit is relaxed at suitable time by the DAQ. A time clear signal may also be send to the compressor controller to reset the time for all the compressors.

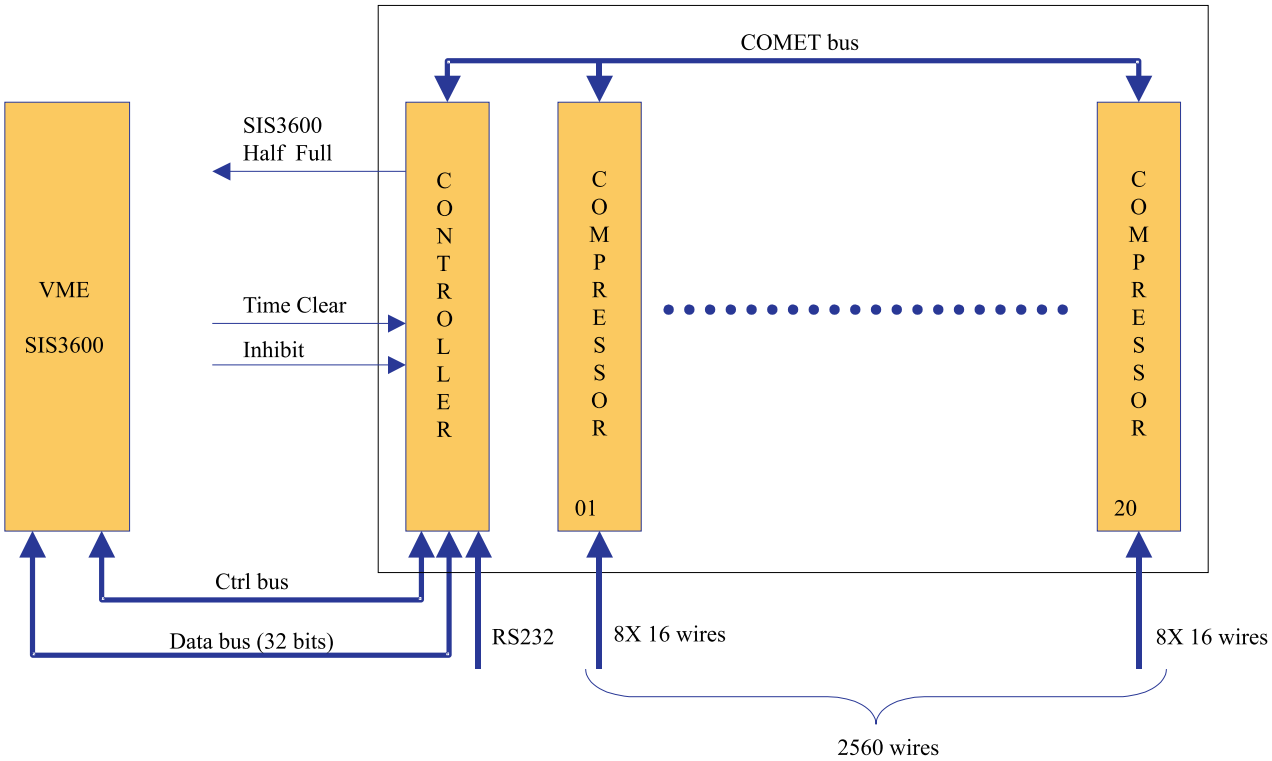

Figure 1: General concept of the COMET system.

2 Inputs and outputs of the controller module

Table 1 summarizes the inputs and outputs housed in the controller front panel or entering at the sides of it. Their locations can be seen on Fig.2.

Ready is a LED which tells that the processor inside the controller has correctly downloaded the codes inside each FPGA, has given a correct address to each of them and is scanning the content of each compressor's FIFO .

STR is a strobe output NIM signal synchronized with the data strobe signal given to the SIS3600 VME module. Used for checks.

CPF is a compressor full NIM output telling that at least one compressor FIFO has got a dangerous filling limit and that an internal inhibit of all data input is in action. It may be used by the DAQ to identified blocks that got some dead time.

LCF means : latch compressor full. It is the same as "compressor full" but the NIM signal is maintained until a time clear signal is given. It may be read by the DAQ at readout time to check if overflow occurred during the compression process.

HF is the VME FIFO half full signal. This NIM DC level stays as long as the half full state of the SIS3600 is reached. It has to be used by the DAQ system to start the readout sequence of the SIS3600.

Table 1: Inputs and outputs of the controller module

Symbol	Signals	Type	Action	LED color or position
	Ready	none	status	LED green
STR	Strobe	NIM	Output	LED green
CPF	Compressor Full	NIM	Output	LED red
LCF	Latch Compressor Full	NIM	Output	LED red
HF	VME FIFO half full	NIM	Output	LED red
TST	Logic test signal	NIM	Input	LED green
TCL	Time clear ($width = 3T_0/2$)	NIM	Input	LED yellow
FCL	Fast clear	NIM	Input	LED yellow
INH	Inhibit (level)	NIM	Input	LED red
CLK	Clk (50 MHz)	NIM	Input	LED yellow
EXT	Ext	none	status	LED green
INT	Int	none	status	LED red
RS 232	Load of CPU with program and FPGA code Input of commands	D9 RS232	Input	front
	2x32 bits data	2x34 pins flat connectors: ECL	Output	side
	8 control lines	20 pins flat connectors: ECL	In/Out	side

TST is a logic NIM signal input giving a rate for the generation of pseudo hit wire pattern or for triggering a pulse to the preamplifier cards. It requires the TST mode or the PLS mode status.

TCL is the time clear NIM input signal. It resets the time and the pseudo pattern sequence. Its length should be equal to 3 times the clock period used in the FPGAs.

FCL This NIM input signal could fast-clear the SIS3600 VME module. Up to now it was never tested.

INH is an inhibit DC level: NIM input, active low. Internally to the controller, it will be put in OR with other inhibits: like the STOP, the FIFO's compressor overflow, the half full SIS3600 status.

CLK is the NIM input for an external clock (one period = T_0.). Its rate is multiplied by two in each compressor unit before being fed to the two FPGAs housed in one compressor module.

EXT is an LED telling that the clock used is the external one (50 MHz).

INT is an LED telling that the clock used is the internal one.

RS 232 is a D9 connector for the RS232 bus used in the soft downloading of the controller program or, by default, for transmitting the R/W commands originating from the soft control program.

3 Inputs and outputs of the compressor modules

Table 2 summarizes the input, outputs and LEDs housed in the controller front panel or entering at the sides of it. Their locations can be seen on Fig.2. Each compressor consists of two channels (CH1 and CH2) processing 2x32 wires each named A and B. The front panel LEDs show the status in each channel. Their names are listed in table 2.

Table 2: Inputs and outputs of the compressor module belonging to one given channel. Below the double horizontal line are the items belonging to both channels

Symbol	Signals	Type	Action	LED color or position
NRM	Normal mode			LED green
TST	Test mode			LED yellow
PLS	Pulse mode			LED yellow
TOK	Token			LED green
EMPT	Compr. FIFO empty			LED green
ORA	Hit in A			LED yellow
ORB	Hit in B			LED yellow
FULL	Overflow in FIFO			LED red
4x16 Data	4x(16 data + ctrl)	4x40 pins: LVDS	I/O	side
GND	Ground	DC		-
6V	Power 6V	DC		-

NRM is an LED telling that al least 1x16 data input are scanned by the compressor.

TST is an LED telling that al least 1x16 data input are the pseudo pattern.

PLS is an LED telling that al least 1x16 data input are preamplifier data generated simultaneously from all 16 channels by one pulse sent by the compressor.

TOK is an LED which is lightening when the controller act on the corresponding channel.

EMPT is an LED indicating that the compressor FIFO is empty.

ORA is an LED which lights when a hit is found on at least one wire of group A.

ORB is an LED which lights when a hit is found on at least one wire of group B.

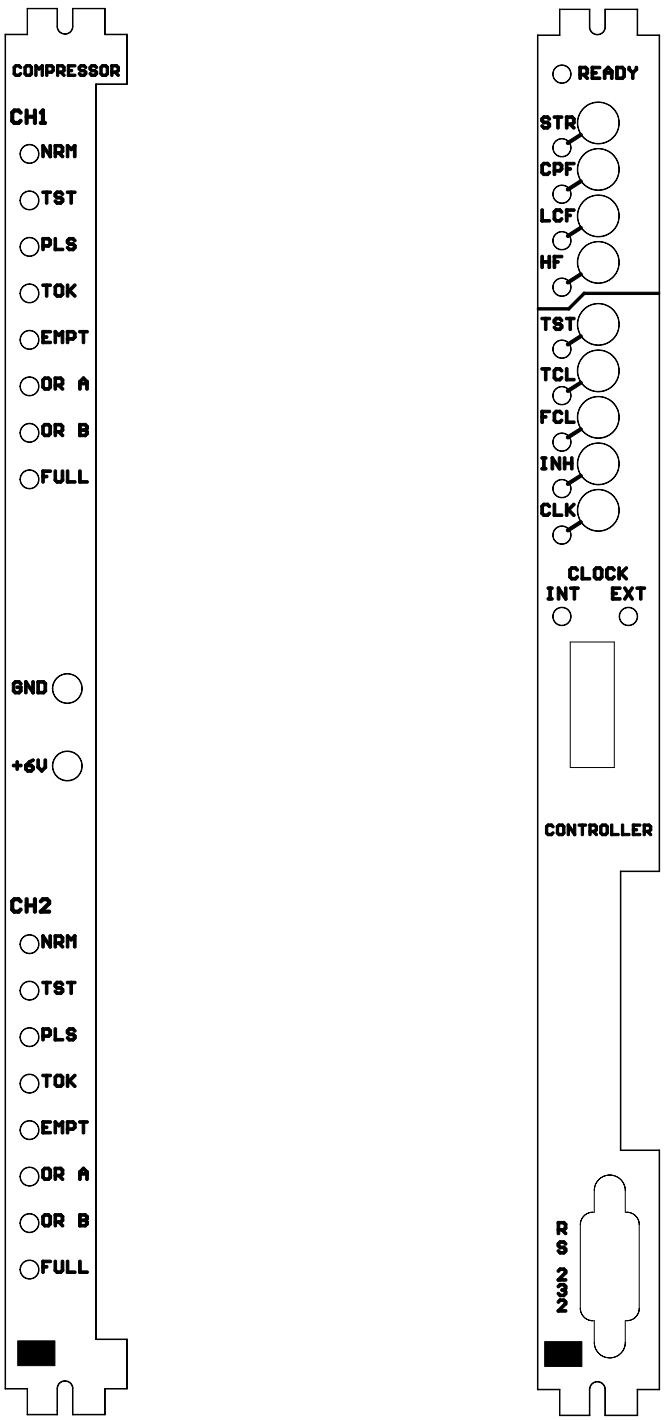

Figure 2: Front panel of the controller and compressor modules.

FULL is an LED indicating that the compressor FIFO is nearly full. When this nearly overflow occurs, the controller which receive the OR of such signals from all the FPGAs (i.e. CPF), forces all compressors to the inhibit state, and waits the CPF to disappear by the readout of the FPGAs, before relaxing the inhibit state. Let us precise that in fact the FULL LED signal is a latched version of this nearly overflow state and that it will be reset by a time clear signal.

GND is a ground input to be send to the preamplifiers.

6V is a 6V DC power input to be send to the preamplifiers.

4x16 Data are 4x40 pins twisted pair cables. It is the half of all data sent to one compressor module. Each cable is composed of 16 twisted pairs transporting the data from the preamplifiers in the LVDS standard. The data should have at least a width of $3T_0/2$, T_0 being the period of the clock. It contents also the 6V power supply and the ground needed by the preamplifier of one group of 16 wires and drives about 100 mA. This flat cable dispatch also, from the compressors to the preamplifier cards, an adjustable DC level playing as a unique discriminator threshold for one group of 16 wires. In pulse mode it sends an amplitude-tunable analogue pulse of fast rise time and long fall time to mimic real signals.

4 Rates, spaces, times

The compressor qualifications are constrained by the clock speed and by the various FIFO volumes

1. The maximum clock rate we tested is 50 Mhz ($T_0 = 20$ ms). The latter is multiplied by two before being fed into each FPGA. So the effective clock is 100 MHz. Due to the hit detection architecture and to the writing process, a scanning cycle takes 3 time periods. **The time precision of the detected hits is then equal to 30 ns.**

2. The VME SIS3600 FIFO has a 32 bits word full capacity of 32 k. The half full is then reached after a loading of 16 k words. Supposing that each hit (in a word of 32 bits) is accompanied by its time data of 32 bits, an half full VME FIFO corresponds of 8 k hits or 1.3 k electrons crossing 6 planes. **With a maximum muons stop rate of 50 kHz, the half full is reached in 26 ms with a transfer rate of 2.5 MB s^{-1}**

3. Each FPGA 10K30E-208 is provided with an internal double port FIFO of 512x32bits size. The in laboratory test have shown that **the compressor FIFO starts to experience overflow at a transfer rate of 16 MB s^{-1} which is a factor of 6 above the maximum expected rate.**

4. The label (address and time) accompanying the data are stored in a 32 bits word as shown here below:

MSB								LSB
	BAxx	xxxx	tttt	tttt	tttt	tttt	tttt	tttt

where the values of B and A are the MSB specifying the group, xxxxxx indicates the FPGA number and the 24 bits t give the time in units of 30 ns. Indeed we need 2 bits to select the A or/and B group to which the hit wire correspond. The FPGA address (there are 40 units) takes 6 bits. Consequently there remains $32 - (2 + 6) = 24$ bits for the time in units of 30 ns. **This corresponds to a round over time of 503 ms.**

In conclusion the first limitation comes from the VME FIFO size. This could be improved by a factor of 4 by increasing the FIFO size as foreseen by the manufacturer in ordering a supplementary buffer.

5 Setting the working modes of the compressors

All the commands needed to set the working modes of the compressors are based on an identical syntax and use the RS232 serial transfer line with the ascii standard. It has the following structure:

| function-name | FPGA-address OR 0x80 | function-code OR sub-address | Data |

where the FPGA address run from 0x0 to 0x27 and the twisted flat cables sub-address from 0x0 to 0x3. The sub-address runs from 0x0 to 0x7 in case of the MSK command (see table 3) due to the fact that the data-command-size is only one byte wide and that a flat wire cable contains 16 wires. The function-name, function-code and data values are summarized in table 3. The data given in the RAT command has the meaning given in table 4. Since the

Table 3: functions and data entering in the command syntax

Function-name	Function-code	Data	Comment
MOD	0x20	0x01	sets in normal mode
MOD	0x20	0x02	sets in test mode
MOD	0x20	0x04	sets in pulse mode
MSK	0x40	0x0-0xFF	sets a mask pattern on 8 wires
THR	0x18	0x0-0xFF	sets the level of the discr. threshold
LEV	0x10	0x0-0xFF	sets the amplitude of the pulse to preampl.
RAT	0x04	0-23	select a rate for the test or pulse mode
RD	any	0x??	reads the data associated to the fct-code

rate is common for all of sub-addresses, the sub-address is disregarded in the RAT command. Moreover the normal mode supersedes the rate values and acts on one defined sub-address.

Table 4: Values of data entering in the RAT command syntax

n	Rate
0	no pseudo pattern nor pulses
1	external rate given by the TST input of the controller
2 - 23	internal rate pulses. Rate = clock divided by 2^{n-1}

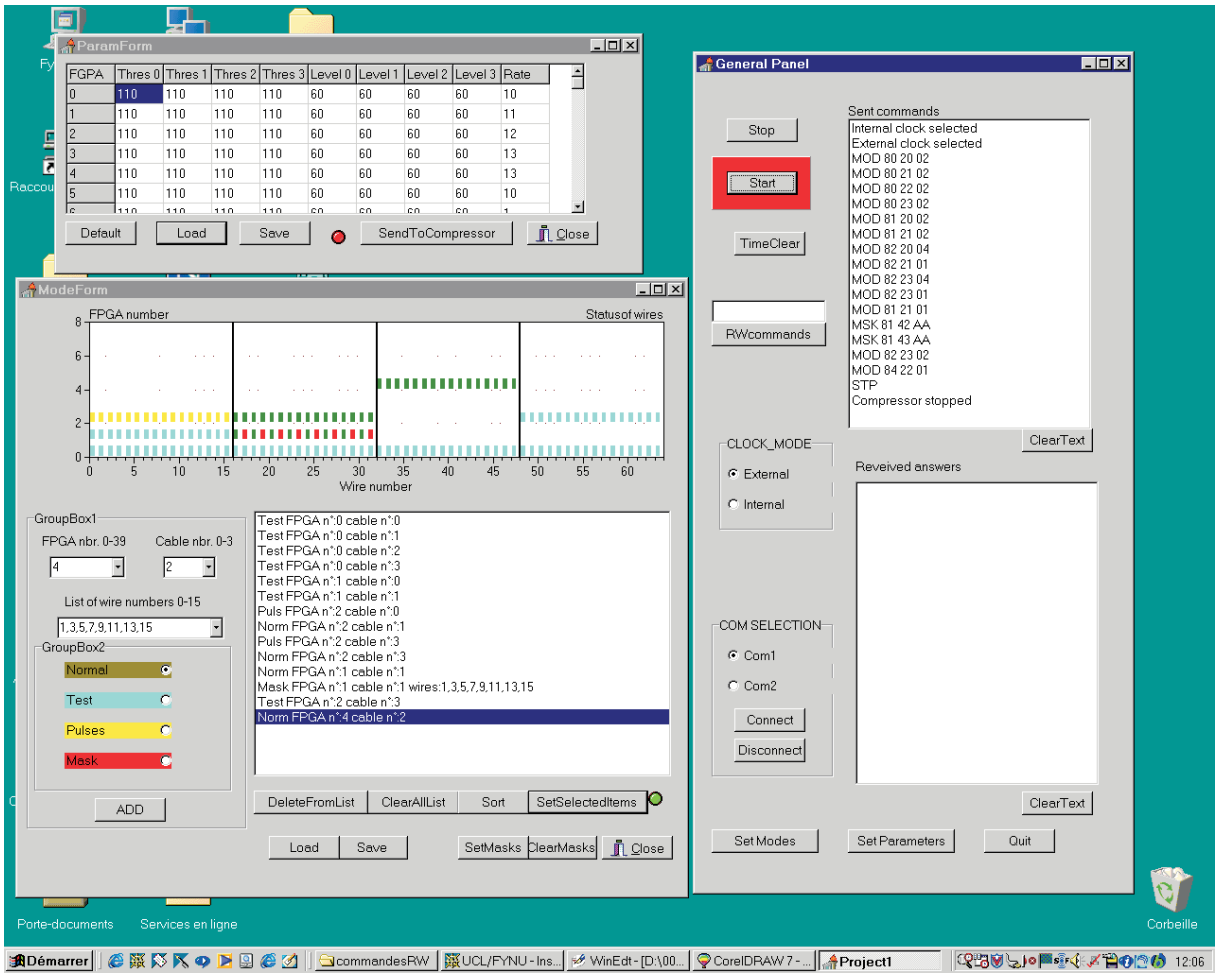

Figure 3: Slow control program presentation.

In supplement to these parameterized commands, there are simple commands (see table 5) for stopping, starting the compression, for choosing the clock mode or sending a soft time-clear signal, etc... All these commands, the parameterized ones and the simple ones are generated

Table 5: Simple functions

Function-name	Comment	Function-name	Comment
SMK	sets all mass	RST	reset
CMK	clear all mask	TCL	send a time-clear
STT	start	INT	internal clock
STP	stop	EXT	external clock

in an user friendly slow control program based on DELPHI programmation which makes use of windows and buttons (see Fig 3).

The parameterized commands loads data in various FPGA's registers which can be read back via a RD command and checked. Each newly sent command is checked by the slow control program and the result is communicated in a special window in case of wrong behaviour.