SP02 to DDU Data Format

Petersburg Nuclear Physics Institute / University of Florida
May 15, 2004
Version 2.1

Abstract

This document describes the SP02 to DDU event data format, as it is implemented in the SP02 design. During the test beam, only the VME interface will be used for read-out, but the Data Acquisition path implementation is easily expandable for pushing data to the DDU as well.

Event Structure

Each event is a set of three types of blocks:

- A Header Block,
- A FRONT Data Block and
- A SP Data Block.

There is only one Header Block in each event. The number of Data Blocks and its content and size may vary as defined by the Header Block. A block size is a number of 16-bit words, comprising the block. The Data Block may contain either complete data or zero suppressed data. The block size of a complete data block is always a multiple of four.

Normally, there should be an event for every L1 Accept signal, so the Event Counter increments by one from event to event. But if the VME readout fails to keep up with the trigger rate, the SP02 logic skips the event(s) that do not fit in the DAQ FIFO, while still incrementing the Event Counter on every L1 Accept. If this happens, there will be missing events in the readout. As the trigger rate further increases above the VME throughput, more missing events will be encountered.

Header Block

The Header Block carries description of the event structure that follows, as well as the corresponding event and bunch counter values. The Event Configuration Word has several fields:

- The FRONT_FPGAs Active mask field is 5-bit wide and each bit corresponds to one of the FRONT_FPGA chips F5…F1. If any bit is set to logical 1, the corresponding chip will be read out on L1 Accept;
- The MB Active mask bit, if set to logical 1 the Barrel muon data will be read out;
- The PT_LUT field allows one the option either to skip the PT LUT output data, or to spy on one of the three PT LUT outputs;
- The #_of_BX field regulates the number of bunch crossings, i.e. the number of Front and SP Data Blocks to be read out on every L1 Accept. The valid setting are from 0 (no Data Blocks follow the Header) to 7;

- The Zero Suppression bit determines if zero track stub / track data is skipped in the data blocks.

Except for the PT_LUT field, which value is set by the CSF_SFC register in the SP chip, all other fields are determined by the CSR_DFC register. Each relevant FPGA has a copy of the CSR_DFC register, so all FPGAs follow the same data readout rules.

The Header Block has a unique signature: hex 0xF is the most significant digit for all Header Block words. This signature could not be found anywhere else in the Data Blocks.

The DDU FPGA reads the first two words of the Header Block from the SP FPGA. The Bunch Counter stores its value at L1 Accept arrival, so the value corresponds to the first FRONT and SP Data blocks. There is no need to read out the Event Counter, since it is located in the DDU FPGA.

FRONT Data Block

The content and size of the FRONT Data Block are defined by the Event Configuration Word and the proper data.

The first two data words with Valid Pattern (VP) and Synchronization Error (SE) bits for each EMU muon are always present. These bits are received from the MPC, so no processing is yet performed on them due to transmission or/and synchronization errors that may be detected by the SP logic.

The DDU FPGA collects data only from those FRONT FPGAs that have been configured as active, so VP and SE bits for inactive FRONT FPGA muons will be zeros.

If Zero Suppression mode is not configured, then the data is collected for all three muons of all active FRONT FPGAs. Otherwise, the data is collected only for muons with VP bit set to 1. In any case, the muon order always goes from ME1A to ME4C.

The FRONT Data Block word count, if the block is present at all, is 2 words minimum and 32 words maximum.

The EMU muon data format is rearranged compared to the MPC data format to facilitate its reading in hex notation, see table below. Since the VP and SE bits have been extracted earlier, they are not present in the muon data.

Table 1: Data Format for EMU Muons

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Word
0	Wire Group ID [6:0]							CSC ID [3:0]				Quality [3:0]				ME/Fr1
0	BC0	BX0	L/R	CLCT Pattern # [3:0]				CLCT Pattern ID [7:0]								ME/Fr2

SP Data Block

The content and size of the SP Data Block are defined by the Event Configuration Word and the proper data.

The first two data words are always present. The first data word carries Modified Synchronization Error (SE) bits for each EMU muon. The SE can be set due to several reasons:

- If the FRONT FPGA detects a link error for just one BX, it sets the SE bit just for this particular BX;

- If the FRONT FPGA logic detects the loss of data synchronization for a particular link, it permanently sets the SE bit for this link, until it is cleared by L1 Reset;
- If the SP FPGA logic detects the loss of data synchronization for a particular FRONT FPGA, it permanently sets the SE bits for all three muons serviced by this FRONT FPGA;

The second data word carries the Track Mode bits and non-zero Quality marks for DT muons. The DDU FPGA collects DT data only if the MB is configured to be active.

The data is collected either unconditionally for both DT muons and all output tracks if Zero Suppression bit is set to 0, or data is collected only for DT muons with Quality > 0 and output tracks with Mode > 0. The PT LUT data may also present in the output block if in addition to the above conditions the data is requested by the PT LUT field setting.

The DT data format is shown in the table below:

Table 2: Data Format for DT Muons

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Word
0	0	Calib	Flag	0	0	0	Phi Bend [4:0]					0	Quality [2:0]			MB/Fr1
0	BC0	BX[1:0]		Phi [11:0]												MB/Fr2

The output track data format is rearranged compared to the Spy FIFO data format to make it more readable in hex notation and to free up the MSB. The MODE field is skipped, since it has been extracted earlier. MS Frame 4 comes only once with the corresponding muon data, as requested by the PT_LUT setting in the event header.

Table 3: Data Format for SP Tracks and PT LUT Outputs

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Word
0	RSV	SE	Eta [4:0]					HL	CHRG	FR	Phi [4:0]					MS/Fr1
0	BC0	BX0	SIGN	Delta23 Phi [3:0]				Delta12 Phi [7:0]								MS/Fr2
0	MS_ID [2:0]			MB_ID [2:0]			ME4_ID [1:0]		ME3_ID [1:0]		ME2_ID [1:0]		ME1_ID [2:0]			MS/Fr3
Rear Muon PT LUT Data								Front Muon PT LUT Data								MS/Fr4

The SP Data Block word count, if the block is present at all, is 2 words minimum and 16 words maximum.

Table 4: SP02 DAQ Data Format

SP02 DAQ Data Format																		
			Data bits															
Data Word	**Description**	**Comment for Zero Supression bit = 1**	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Event Header																		
HD0	Event Configuration Word	Always present	0xF				PT_LUT		MB Active	FRONT_FPGA Active					Zero Supr.	#_of_BX		
HD1	BC	Always present	0xF				Bunch Counter											
HD2	EC_LSB	Always present	0xF				Event Counter LSB											
HD3	EC_MSB	Always present	0xF				Event Counter MSB											
FRONT_Data_Block[1] for (BX = Bunch_Counter_Value)																		
FAB0	Muon Valid Pattern bits for Zero Suppression	if (#_of_BX) > 0 and at least one Active FRONT_FPGA	0	ME4C	ME4B	ME4A	ME3C	ME3B	ME3A	ME2C	ME2B	ME2A	ME1F	ME1E	ME1D	ME1C	ME1B	ME1A
FAB1	Synch Error bits (as they come from MPC)		0	ME4C	ME4B	ME4A	ME3C	ME3B	ME3A	ME2C	ME2B	ME2A	ME1F	ME1E	ME1D	ME1C	ME1B	ME1A
FAB2	track stub	if (#_of_BX) > 0, (F1_Active) = 1 and VP[ME1A] = 1	ME1A Frame 1 data															
FAB3	track stub		ME1A Frame 2 data															
FAB4	track stub	if (#_of_BX) > 0, (F1_Active) = 1 and VP[ME1B] = 1	ME1B Frame 1 data															
FAB5	track stub		ME1B Frame 2 data															
FAB6	track stub	if (#_of_BX) > 0, (F1_Active) = 1 and VP[ME1C] = 1	ME1C Frame 1 data															
FAB7	track stub		ME1C Frame 2 data															
FAB8	track stub	if (#_of_BX) > 0, (F2_Active) = 1 and VP[ME1D] = 1	ME1D Frame 1 data															
FAB9	track stub		ME1D Frame 2 data															
FAB10	track stub	if (#_of_BX) > 0, (F2_Active) = 1 and VP[ME1E] = 1	ME1E Frame 1 data															
FAB11	track stub		ME1E Frame 2 data															
FAB12	track stub	if (#_of_BX) > 0, (F2_Active) = 1 and VP[ME1F] = 1	ME1F Frame 1 data															
FAB13	track stub		ME1F Frame 2 data															
FAB14	track stub	if (#_of_BX) > 0, (F3_Active) = 1 and VP[ME2A] = 1	ME2A Frame 1 data															
FAB15	track stub		ME2A Frame 2 data															
FAB16	track stub	if (#_of_BX) > 0, (F3_Active) = 1 and VP[ME2B] = 1	ME2B Frame 1 data															
FAB17	track stub		ME2B Frame 2 data															
FAB18	track stub	if (#_of_BX) > 0, (F3_Active) = 1 and VP[ME2C] = 1	ME2C Frame 1 data															
FAB19	track stub		ME2C Frame 2 data															
FAB20	track stub	if (#_of_BX) > 0, (F4_Active) = 1 and VP[ME3A] = 1	ME3A Frame 1 data															
FAB21	track stub		ME3A Frame 2 data															
FAB22	track stub	if (#_of_BX) > 0, (F4_Active) = 1 and VP[ME3B] = 1	ME3B Frame 1 data															
FAB23	track stub		ME3B Frame 2 data															
FAB24	track stub	if (#_of_BX) > 0, (F4_Active) = 1 and VP[ME3C] = 1	ME3C Frame 1 data															
FAB25	track stub		ME3C Frame 2 data															
FAB26	track stub	if (#_of_BX) > 0, (F5_Active) = 1 and VP[ME4A] = 1	ME4A Frame 1 data															
FAB27	track stub		ME4A Frame 2 data															
FAB28	track stub	if (#_of_BX) > 0, (F5_Active) = 1 and VP[ME4B] = 1	ME4B Frame 1 data															
FAB29	track stub		ME4B Frame 2 data															
FAB30	track stub	if (#_of_BX) > 0, (F5_Active) = 1 and VP[ME4C] = 1	ME4C Frame 1 data															
FAB31	track stub		ME4C Frame 2 data															

SP_Data_Block[1] for (BX = Bunch_Counter_Value)																		
SPB0	Modified Synch Error bits (as SP gets it)	if (#_of_BX) > 0	0	ME4C	ME4B	ME4A	ME3C	ME3B	ME3A	ME2C	ME2B	ME2A	ME1F	ME1E	ME1D	ME1C	ME1B	ME1A
SPB1	MB non-zero Quality and Track Mode bits		0	0	MB1D	MB1A	MODE[3]				MODE[2]				MODE[1]			
SPB2	track stub	if (#_of_BX) > 0, (MB_Active) = 1 and Quality[MB1A] > 0	MB1A Frame 1 data															
SPB3	track stub		MB1A Frame 2 data															
SPB4	track stub	if (#_of_BX) > 0, (MB_Active) = 1 and Quality[MB1D] > 0	MB1D Frame 1 data															
SPB5	track stub		MB1D Frame 2 data															
SPB6	1st track data	if MODE[1] > 0	MS[1] Frame 1 data															
SPB7	1st track data		MS[1] Frame 2 data															
SPB8	1st track ids's		MS[1] Frame 3 data															
SPB8a	PT data	if MODE[1] > 0 and PT_LUT = 1	MS[1] Frame 4 Data															
SPB10	2nd track data	if MODE[2] > 0	MS[2] Frame 1 data															
SPB10	2nd track data		MS[2] Frame 2 data															
SPB11	2nd track ids's		MS[2] Frame 3 data															
SPB11a	PT data	if MODE[1] > 0 and PT_LUT = 2	MS[2] Frame 4 Data															
SPB12	3rd track data	if MODE[3] > 0	MS[3] Frame 1 data															
SPB13	3rd track data		MS[3] Frame 2 data															
SPB14	3rd track ids's		MS[3] Frame 3 data															
SPB14a	PT data	if MODE[1] > 0 and PT_LUT = 3	MS[3] Frame 4 data															
...																		
...																		
FRONT_Data_Block[#_of_BX] for (BX = Bunch_Counter_Value + #_of_BX - 1)																		
SP_Data_Block[#_of_BX] for (BX = Bunch_Counter_Value + #_of_BX - 1)																		