TH-Notiz-012-2008

FAIR

Collection of BPM specifications

13. August 2008

Author(s): T. Hoffmann, P. Forck, P. Kowina, K. Lang

Version: 0.2

Glossary

Basic Mode: The modes which have to be prepared on the FPGA part. They differ in memory handling, BPM-package contents, pre-evaluation etc. (see Basic Modes)

Sub-Mode: Different evaluation modes of Basic Mode Data on the CCCP

BPM-package: Our self-defined data package for the BPM data plus time information

Signal-package: Our self-defined data package for the additional signal data (transformer, rf, dipole ramps etc.) from Libera 13

Sub-Systems: External parts or interfaces which have to be controlled via network (external GSI timing, database, control system)

CCCP: Concentrator and Control Computer

PTIF: Programmable GSI Timing Interface (TCP/IP based) for TTL pulse and gate generation

Basic Modes

1. **Bunch to Bunch Data (to be realized) PRIO 1**
2. **Raw Data (to be realized)**
3. **Full Cycle Data (not yet)**
4. **Oscilloscope (not yet)**

Known risks, open questions and information

- The workpackage for Kevin is very big. We think about outsourcing of the RAW Mode (FPGA). Perhaps this raw mode needs to be delayed.
- The specification and layout of a good concept is more time consuming then expected. This time is unfortunately not fully available. Some time optimization on the knowledge transfer and the discussions are required.
- This paper may change if further discussions lead to new results. This is not a binding and final specification sheet.
- The CTRI driver for the timing is not working with the 64bit PC. It is unclear how this problem can be solved. Also open is the use of "custom event source"
- The display of online data can be updated cycle by cycle, with the planned fast cycle mode (4Hz) another way is may be suitable (discuss)
- The described BPM-package is not yet approved. The final package size is dependent on the performance of the network and the CCCP parts and requires further discussions as this part is very important for the precision of the measurement.

Mode 1: Bunch to Bunch Data:

This mode is the most essential mode for GSI. In combination with the bunch recognition algorithms the position for each plane (22 Bit, see formula) is sent with additional overload/underload values and time information (using gate start, gate stop and rf) for data/bunch consistency checks to the CCCP. The data consistency checks and further data handling (averaging etc.) have to be performed on the CCCP. See formulas, see timing diagram.

Sub Modes

- **Closed orbit**

  The data of all 12 stations is averaged over a free selectable
  amount of turns (default 1000)
  Display see Posi program
  Set of start and stop event

- **Bunch tracing**

  the b2b data is used to calculate specific values called **Betatron phase advance** between subsequent BPM stations. For this, the b2b data must be complete and error free. The data may be saved on the CCCP for the calculation. This operation should deliver the result in reasonable time (<60s,wish), no other display (graph) is foreseen. A selection of a measurement time window (max.= complete cycle length) is required. The algorithm for this calculation is not yet defined.

- **Trending without averaging (see Posi) for one station**

  set trigger
  set timelength
  one graph per cycle (ms time scale)

- **Trending with averaging for one station**

  set bin size
  free run
  one dot per bin (default 1000)
  graph in s time scale

- **Overload graph like trending**

  Display of the 4 status bits for overload per cycle or per measurement gate (to see possible overload at the beginning, middle or end of a cycle)

- **Calibration**

  access to amplifiers via MIL bus (UFC)
  special sequence of commands (?)
  display like normal measurement, but no consistency check required. A mark within the data and a visible label have to be foreseen.
  On calibration mode the real analog signals are disconnected from the daq.
  The functionality of both amplifiers (old and new type) for calibration should be prepared.

- **Zero line identification (may be together with calibration)**
  measure the ADCs without bunches (pause or injection?)

- **Bunch merging (low priority)**

  merges bunches e.g. 4 to 1 or 5 to 1
  the harmonic value has to be known (SISMODI)
  workaround: may be measurement of the first part of the cycle with old harmonic and then later measurement of the second part with new harmonic.

- **FFT Analysis of 1 BPM (tune)**
- **Turn by Turn (1 BPM)**

Timing

- without CTRI: start and stop trigger via PTIF (Kevin), into I/O of Libera. Fesa only Server-Action.
- with CTRI: mixed RT-Action via FESA and PTIF plus I/O input on Libera.
- Eventually we design a GSI-CERN-Timing telegram converter. This would be perfect also for other FESA projects at GSI.

GUI

open

BPM-package / Signal-package (Data Format and Readout)

BPM-package:
e.g. :
Hor. Pos(22 Bits) + Ver. Pos(22 Bits) + 12 Bit t_1-t_{rf} + 12 Bit t_2-t_{rf} + 16 Bit T_{rf} + Overload and Underload bits

Signal-package
- rate of package (not of every bunch required)
- format of package

Mode 2: Raw Data:

This mode delivers all ADC values of all or some of the 12/13 Liberas plus the rf signal to the CCCP hard drive. A trigger setting is required, the maximum measurement length should be displayed. This is an expert mode. This operation should be optimized to a fast storage but may take some time (max. few minutes)

Sub Modes

n.a.

Timing

PTIF (so far)

GUI

- Select the Liberas (all or some or one), starting point (trigger), VrtAcc, Filename
- Show progress and finish
- Select one or multiple measurements (cycle by cycle, if technical possible, or next possible cycle after readout)
- add trigger info (BPMs, Event, Delaytime and VrtAcc) into the data file header
- switch between binary and ascii storage (for binary the format should be documented)

BPM-package / Signal-package (Data Format and Readout)

A, B, C, D , t_{rf}
Transport of data from Libera memory via SFP port to the CCCP memory/file.
Readout of 13th Libera

For both modes:

Clock Splitter Input Assignment

4 Channels are available: possible assignment

1. rf–Signal (already converted to countable pulses, TTL)
2. external sampling clock, 125 MHz, TTL
3. start trigger
4. stop trigger

Libera Input Assignment

Start and Stop trigger:	PMC I/O module on Libera
rf-Signal:	MC (operational??)
???:	Arm
???:	Trigger

Required hardware (buy or develop)

- 125 MHz high precision clock 19" rackmount
- Discriminator/Level adapter for rf signal
- GSI-Cern timing converter

Sub Systems

PTIF: via TCP/IP (Expert: Kevin)

UFC: Userface-access to GSI hardware (BPM amplifiers, set range, set test-generator etc.)
http://bel.gsi.de/docs/ufc.html

Oracle: DB access to significant informations (harmonic SIS-Modi)

FPGA Requirements

- Raw data stream onto the SFP port (outsourcing ?)
- Bunch to Bunch data stream onto the SFP port
- Working algorithm

- Readout of RF pulses via machine clock input (MC) creating time stamps (clock), bucket counter.
- Creating data stream 2x Position, 2x2 status bits overload plus FPGA algorithm for the status bits (Kowina), t_1-t_0, t_2-t_0, t_{rf}
- Add on data telegram with the 13 Libera data
- Creating readout for 13 Libera (binned/averaged data for 4 ADC channels, low performance)
- Synchronisation of Liberas (ADC ticks)
- Mode for ZERO level measurement (Calibration mode?)
- Preparation of programmable timing cards

Time-Synchronisation

To synchronize the time between all Liberas the command settime from the SBC has to be executed, an additional hardware trigger (clock splitter, which input at Libera?) to all Liberas should do the synchronization. The synchronization is required to obtain time information on which cycle was measured. A very high precision is not required. This time stamp has to be added once per cycle to the data e.g. at the reset trigger for the rf-counter.
Idea: The time info can be added to the signal-package on the 13 Libera.

Sampling-Synchronisation

Insert an external clock (precise 125 MHz) via clock splitter to the Liberas.

Additional Signals (13. Libera)

A 13th Libera shall be installed and integrated like the others to insert machine dependent data for correlation. Signals such as synchrotron transformer, dipole or quadrupole magnet, cavity rf or rf-Master shall be attached to the ADC inputs. These inputs can be binned/averaged already on the FPGA, this option should not limit or reduce the performance of the BPM part.

ToDo for GSI

- get names and properties for the amplifiers
- organize source code example for UserFaceCall (get and set with controls)
- migrate CTRI driver to 64 bit system
- prepare FPGA mode "Raw Data" and other programming
- Create a list with all BPM constants for position calculation (constante and offset)
- Observe the zero line, if it is stable or not to check the frequency when this measurement has to be performed.

- Definition of BPM- and Signal package at the Libera A, B, C, D = AB=x plane, CD=y plane
- Description of the calibration sequence.
- Buy a precise external clock 19" Rackmount
- Organize discriminator for rf pulses

Remarks

For the time being, a trigger module (PTIF) is foreseen, which is accessed via TCP/IP. The event number and a dedicated machine number (0-15) can be chosen (or all machines). That means the FESA and the PC do not know, which machine is running, only the selection of the trigger itself provides this info.

The system will change soon on many different sub parts. It should be foreseen, that these changes can be easily applied:
e.g. the type of amplifier changes and therefore the interface and the functionality (test/calibration procedure). The timing hardware and therefore the interface may change.

For GUIs and their graphical displays there are many possible ways, so this is flexible. Good experience was made with a mode, which stores all data, but displays only reduced data fitting to the display size (pixels). With zoom in functions, the next deeper extract of the stored data is used. So the display is dynamically created dependent on the zoom depth and the picture size (pixel).and is much faster than drawing all the points. This can be demonstrated with Ablass.

Whenever information is of interest it should be displayed in the GUI (e.g. Date, Time, VrtAcc, Events, Delays, gate length, Status, Gain etc). Standard functions for GUIs such as screenshot, scaling, markers, zoom, ROI etc should be available.

In principle the readout can be done at the end of each cycle. For performance reason (on the network) it might be suitable to make the readout after a defined filling grade of the memory and to transfer already blocks of data while the measurement is still running.

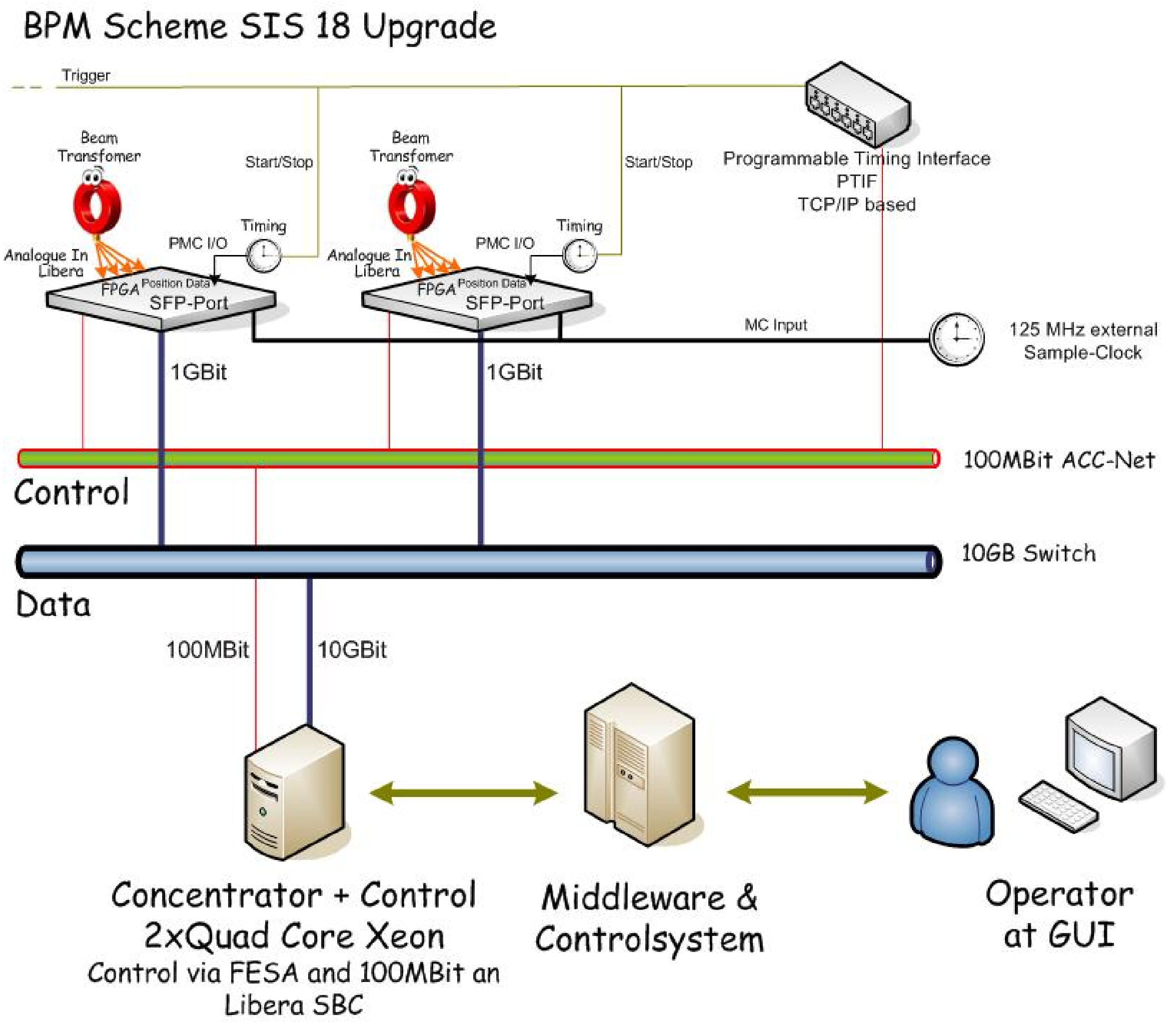
Formulas:

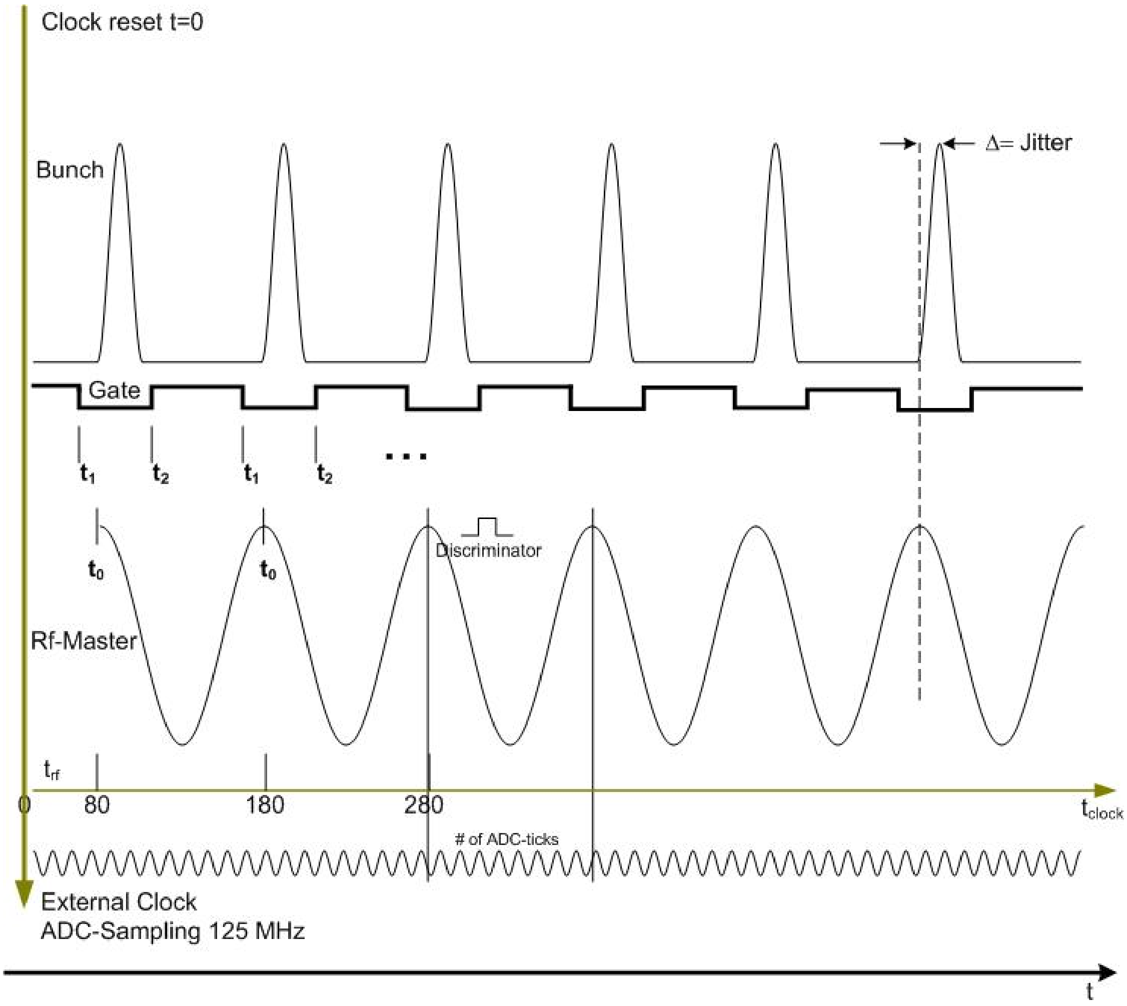
Position of X-plane: $x = K_{BPM}\dfrac{A-B}{A+B} + \delta_x$ K=BPM-constant, δ = offset

Position of Y-plane: $y = K_{BPM}\dfrac{C-D}{C+D} + \delta_y$

Not required yet Sum of plane X: $\sum_x = A + B$

Not required yet Sum of plane Y: $\sum_y = C + D$




Clock reset t=0
Bunch
Δ= Jitter
Gate
t_1
t_2
t_1
t_2
. . .
Discriminator
t_0
t_0
Rf-Master
t_{rf}
0
80
180
280
t_{clock}
# of ADC-ticks
External Clock
ADC-Sampling 125 MHz
t