ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-WEAPL01

PRESENT AND FUTURE OF HARMONY BUS, A REAL-TIME HIGH SPEED BUS FOR DATA TRANSFER BETWEEN FPGA CORES

M. Broseta†, X. Serra-Gallifa, ALBA Syncrothron, Cerdanyola Vallès, Spain
J. Avila-Abellan, S. Blanch, G. Cuní, D. Fernández-Carreiras,
O. Matilla, M. Rodriguez, J. Salabert, ALBA Syncrothron, Cerdanyola Vallès, Spain

Abstract

When feedback loops latencies shall be lower than milliseconds range the performance of FPGA-based solutions are unrivalled. One of the main difficulties in these solutions is how to make compatible a full custom digital design with a generic interface and the high-level control software.

In ALBA, we have simplified the new equipment development process with the use of Harmony Bus (HB). Based on the Self-Describing Bus, developed at CERN/GSI, it creates a bus framework where different modules share timestamped data and generate events. This solution lets the high-level control software in a Single Board Computer or PC, to easily configure the expected functionally in the FPGA and manage the real-time data acquired.

This framework has been already used in the new Em# electrometer, produced within collaboration between ALBA and MAXIV, which is currently working in both synchrotrons. Future plans include extending the FPGA cores library and high-level functions.

NEW FRAMEWORK FOR REAL-TIME DATA PROCESSING

Feedback loops take the system output into consideration, which enables the system to adjust its performance, to meet a desired output response. They are usually used to correct errors. When designing systems with target response times close to µseconds, the possibility to use high level software is limited and the only possibility is via hardware based solutions. Among those, nowadays FPGA is the most selected technology due to their low processing time, relatively low cost, reprogrammable capability, massive parallel complex operation capability and the availability of high number of input/outputs. That makes them quite appropriate for those designs that require integration times under milliseconds range. On the other hand, they are complex to program and offer limited possibilities to store or share data with other high level applications. PC's or Single Board Computers (SBC) are a good solution for those designs where the processing time is not so critical as required, latencies are higher than tenths of milliseconds and they offer complex data processing with multiple tools for data sharing to other high level applications or data transmission via network.

Since 2014, in ALBA computing division, we started to consider making our new in-house developments as versatile as possible and easily adaptable to changing needs from our scientists. Our target was to cover feedback system designs in the ranges from milliseconds to µseconds. In that sense we began the development of our new products based on a solution resulting from the combination of an FPGA and PC. The new design had to be flexible enough to perform real-time calculus, data pre-processing or feedback implementation, easy to configure and with sufficient capability of integration in any control system or configuration. The resulting design is shown in Figure 1, where the FPGA and the PC are connected via PCI Express and the PC is a Single Board Computer (SBC) with I2C and SPI ports to connect other peripheral devices, like a touch-screen display, control the power supply board or any other

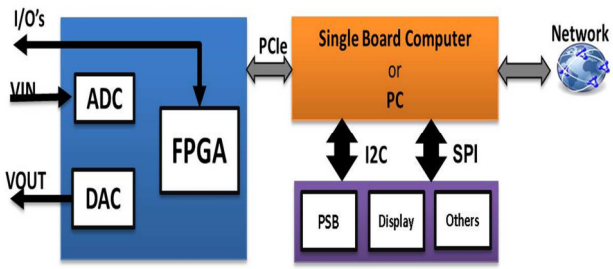

Figure 1: Electric block diagram of new ALBA development.

BASICS OF HARMONY BUS

Communication between FPGA and SBC is done via PCIe bus. This bus, even having a high transfer peak, is not easy to be deterministic and easy control the latency between different communications from high level software layers. To overcome this constrain, the new design clearly divides the functionality between acquisition and real-time data processing in the FPGA, while configuration and high-level data processing in the main processor (SBC). Following the versatility requirements, the FPGA was designed containing different functionality cores that are dynamically interconnected between them depending on the equipment configuration. Harmony Bus (HB) [1] is the easily configurable proposed bus to dynamically share data among the cores at high speed rate.

In the HB data is shared between FPGA cores in real-time with timestamp and with the identifier of the core that has generated it. Harmony follows the tree structure of Self Describing Bus (SDB) [2] developed by CERN, although it is not mandatory that all the cores defined in SDB had also and Harmony Bus connection. The SDB uses a standard bus protocol, the Wishbone bus [3], for the data exchange between the FPGA and the high level

†mbroseta@cells.es

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-WEAPL01

software. The SDB is used for cores diagnostics and set up, while HB is used for fast data sharing between cores.

FPGA cores are connected using Harmony Bus Bridges that act as bus arbiters, broadcasting the data to other bridges and sharing it with the rest of FPGA cores. Bridges register the upstream and downstream data to fit the timing constraints. Inevitably the bridges add latency and jitter to the messages, but the effect is negligible for the targeted latencies. HB can be understood as a pool where all components put its data and can read all other's data (see Figure 2).

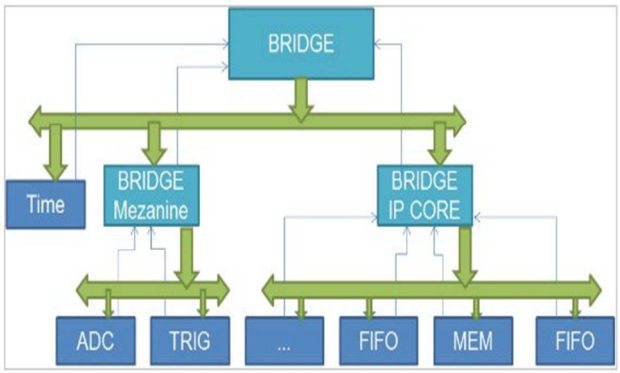

Figure 2: Data flow of Harmony Bus.

The HB is composed of 2 busses: one for upstream data and a second where bridges broadcast all data received. Both are 64-bit bus: 32 bits are reserved for data, 24 bits for timestamp (TS) and 8 bits are dedicated for data identification (ID). There are also 3 lines for bus control: one line indicating valid data in broadcast bus and 2 lines (data request and reading) to manage the upstream bus (see figure 3).

The data sent in HB have an ID of 8 bits which allows up to 256 different messages with id 0 and 255 reserved. The ID's can be configured dynamically to each component using the before mentioned SDB. At the end it is the control software hosted in the SBC who takes care of the ID assignation. In timestamping 24 bits with ns resolution are used, which involves that an overflow occurs every 16 milliseconds. To avoid it before it happens, a message with ID 0 is sent with a time reference. Finally the ID 255 is reserved to send error messages.

At the end is the control software hosted on the SBC that handles the ID assignment. With 24 bits for timestamp with a resolution of ns, it means an overflow occurs every 16 milliseconds. To avoid this before it is present, a message with ID 0 is sent as a time reference. Finally, ID 255 is reserved for sending error messages.

The use of this bus provides a high flexibility in the development of new functionalities or designs. Cores can be interconnected or not, at configuration time by the main software running in the SBC. Thanks to this, multiple different functionalities can be implemented in the FPGA dynamically without having to change its firmware. And the most important, without loosing the deterministic and low latency behaviour of the FPGA. The use of the SDB structure helps to simplify the documentation of the FPGA cores and helps the main software to self-detect and manage the FPGA contents.

EXTRA FUNCTIONALITIES FOR FEEDBACK SYSTEMS DESIGNS

As mentioned before, the FPGA integrates a set of cores that share data between them depending on the equipment configuration, using the Harmony Bus. The main software running on the SBC [4] features complex embedded control software based on Linux OS build with Yocto [5], with enough functionality to run the required applications. Thanks to the dynamic ID's and Harmony Bus, it can configure the FPGA cores required and the number of them depending on the equipment functionality. For the main software, FPGA cores are like a library of hardware modules that can be used individually or interconnected between them, to implement a specific equipment function. The list of the different cores available in the FPGA, divided by its type of functionality is as follows:

I/O cores:

- ADCCORE: Component that controls the ADC device using high-speed serial lines.
- DACCORE: This core controls the Digital to Analog converter.
- DIGITIAL I/O: Controls the digital input output ports. It can be used to implement a counter or an external trigger.
- SPI: Control external devices using the SPI protocol like port expanders, temperature sensors, EEPROM memories, etc…

Logic cores:

- ID GEN: generates data with and ID. It can be used to setup memories or for diagnostics.
- COUNTER: This a counter module to count to that counts harmony ID's previous selected.
- MEMORY: Ram memory to save and download data from fast BUS. It implements a circular buffer that can be started/stopped from fast or slow bus.
- TIMEBASE: Clock reference signal that can be externally referenced to a clock source.
- DMA *(future):* Core responsible to share data with the main software through DMA via PCIe.

Mathematic cores:

- AVERAGE: used to calculate a dynamic mean of data in 32-bit 2-Complement format. There is also one module per channel.
- FIFO: stores and delays any data specifying the ID of the data to delay. There is one module per channel.
- PID: This is one of the most important modules to use the equipment as a control loop feedback mechanism.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-WEAPL01

MATH *(future):* Mathematic coprocessor with the basic operations to perform over the input data +,-,* and /.

A typical configuration of the FPGA cores to implement a feedback system is shown in Figure 3. The analogue signal inputs acquired via the Analog to Digital Converter (ADC) are processed by the ADCCORE. The FIFO and the AVERAGE cores are used to get a mobile average of the input. The result is processed by the Proportional Integral Derivative core (PID) that calculates the error and then, sent to a DACCORE that commands the Digital to Analogue Converter (DAC). At any moment, the data sent through the harmony bus can be stored in the FPGA memory core for a further analysis and processing by the main software.

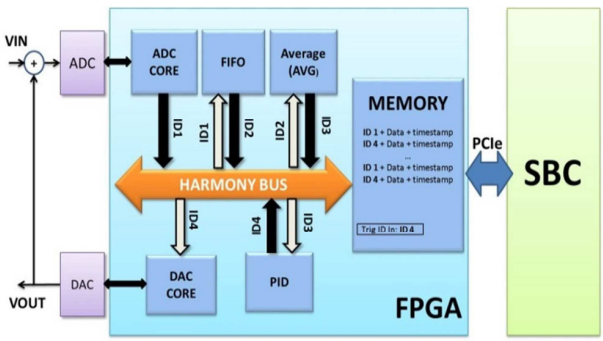

Figure 3: FPGA cores configuration with HB for a feedback system.

The huge quantity of input/outputs available in the design together with the use of dynamic ID and HB, allow that multiple setups can be configured in the equipment without the need of any gateware or software update. The main software also performs other slow or low priority tasks such as the user control and diagnostics: the information display in the touch-screen, software and firmware update, integration with external control systems through a standard communication protocol SCPI or remote control via telnet (using the SCPI protocol [6]) or via web through a tornado webserver [7]. It is written in Python to simplify the integration and development of new functionalities.

EXAMPLES OF IMPLEMENTATION

For a better understanding of the HB configuration used to implement a typical feedback system, a real example of this implementation will be detailed in this section using the ALBA electrometer Em# [8]. The main core for the feedback system is the PID core that provides control over the feedback loop. Figure 4 shows the data flow in the HB for a feedback system. Data acquired by the ADC, is read by the ADCCORE block at 200 KHz with and oversampling than can be configured between 0 to 64 samples. The output sent to the Harmony bus with the identifier ADC1_ID. Then it goes to configurable size FIFO that outputs the FIFO1_ID and this to the AVERAGE that outputs the AVG1_ID. The FIFO and AVERAGE modules are used to get the mobile average of the input. The size of the FIFO indicates the size of the mobile average in time. Each data in the FIFO means 320µseconds and the maximum FIFO size is limited to 1024 samples, what results in a maximum of 320 milli-seconds.

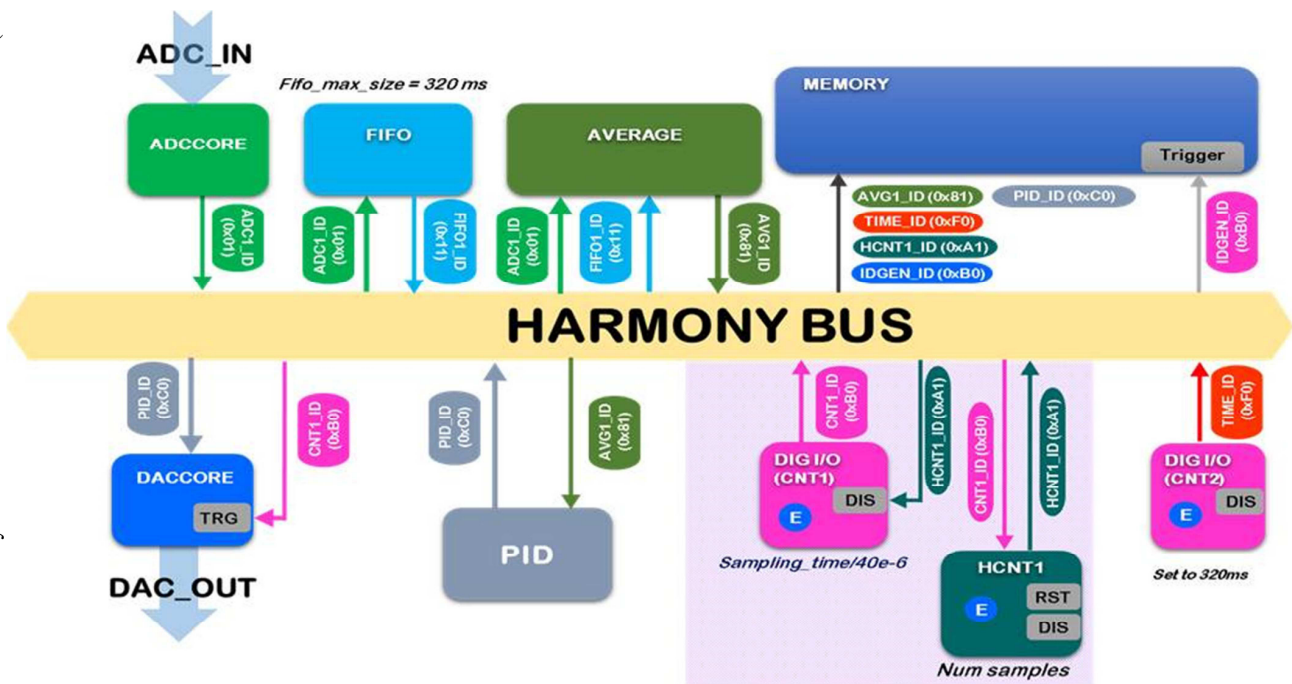

Figure 4: Harmony bus data flow for an acquisition on Em# project, using the Harmony Bus.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-WEAPL01

In the bottom side of the harmony bus, DIGI/O CNT2 provides the time base signal. Each 320ms a message is generated to be used as time reference. This message is always stored in the FPGA memory for future data processing by the main module. DIGI/O CNT1 and HCNT1 are two counters used to configure the sampling time and the number of samples to acquire.

The PID module is continuously processing the signal that comes from the AVERAGE module (AVG1_ID) and generating the corresponding corrected signal PID_ID. Finally, the PID_ID is processed by the DACCORE module to be converted to an analogue output value each sampling time trigger. All the data sent through the harmony bus is stored in the FPGA memory every sampling time trigger, to be processed by the main software. The main software uses this data with timestamp to provide diagnostics and monitoring tools that help to understand how the feedback system is working and reacting, at real-time.

Based on this new design concept, the first product has arisen, the new ALBA electrometer Em#. This equipment adds 4 current amplifiers (CA) to each analogue input and a current amplifier board where these CA's are mounted, to measure low noise currents lower than PicoAmp.

Other functionalities that can be activated with minor changes in the high level software are analogue or digital signal generators. They can be used to synchronize other systems or to easily characterize external systems in open loop, as shown in Figure 5. They can work in parallel to the acquisition or independently. To generate any voltage signal in the analogue voltage output previously defined and stored one of the FIFO's. Every data which is output from the FIFO triggers the IDGEN which stores again the data in the FIFO creating a circular buffer. The output of the FIFO is also used to generate an output in the DAC.

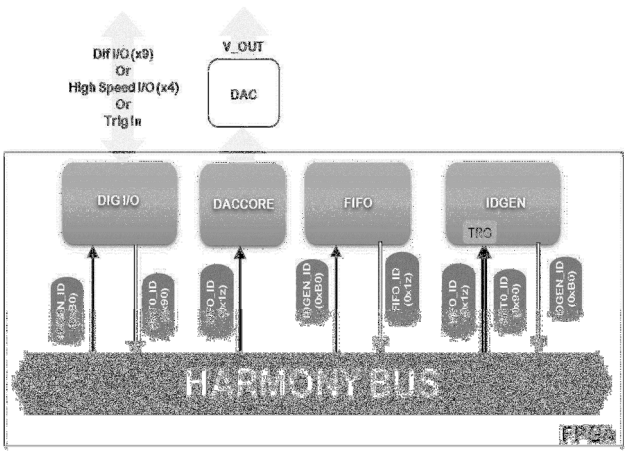

Figure 5: Data flow in Harmony Bus for a Function Generator.

WHAT'S NEXT

As mentioned before, Harmony permits the development of new low-level equipment's functionalities without the need of Hardware or Software modifications. Future plans include extending the library of available FPGA cores adding new functionalities and increasing the possibility of their combination. The roadmap of planned future includes:

- Direct Memory Access (DMA) from FPGA to the SBC. Actually data sharing between the FPGA and the SBC is using virtual memory spaces via PCIe. But depending on the configuration this can affect to the data read and processing speed. To improve the processing time a future plan includes the usage of DMA access.
- MATH core. A mathematical processor for real-time calculus in FPGA where it will be possible to do complex mathematical operations.
- External HB synchronization. Develop a core able to interconnection different Harmony Bus in different equipment's so they are able to share data in real-time using an SFP port.

For example one of the applications that could be done using this design would be to implement a closed loop using a piezo actuator in the second crystal of a Monochromator in a beamline. That could be used to correct the intensity or position of a beam in a synchrotron beamline as shown in Figure 6.

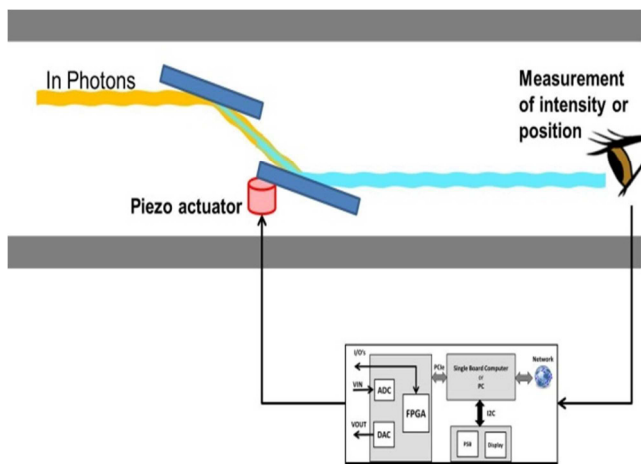

Figure 6: Monochromator with increased bandwidth.

A typical example like this will let the scientist to cover different configurations with just single equipment: the characterization of the piezo actuator using the function generator configuration; a feedback system using the feedback system configuration; read low noise currents using the electrometer; or any other that the scientist could have. All these configurations can be adjusted just selecting the appropriate FPGA cores configuration in the Harmony Bus, via high level commands and without having to modify the Hardware or Software. And in all these experiments, data with timestamp acquired at real-time is available for the scientist at any moment, to diagnose and monitor the status of the experiment.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-WEAPL01

CONCLUSIONS

Harmony Bus is a deterministic, low-latency and multi port bus oriented to share data among different FPGA cores with the objective of implementing feedback systems and providing timestamp services and diagnostic tools from high level software layers. It has been designed as a perfect complement to the SDB developed in CERN for the before mentioned functionalities.

Most of the feedback systems are hard to characterize and diagnose their dynamic behaviour. Harmony Bus offers tools to analyse, diagnose and process in real-time the timestamped data, while the system is working. It accepts commands to dynamically change the configuration of the system or change from a close loop system to an open loop depending on the status of the processed signal. Future possibilities could also allow the characterization of a feedback system using simulators.

The bus has been used successfully introduced in the development of the new ALBA electrometer Em# showing its capabilities as a real-time time acquisition bus.

ACKNOWLEDGEMENT

The authors of this paper want to thank all the people involved in the development of this new framework. They are: Javier Serrano and its OHR team at CERN; the complete Electronics division team at ALBA Synchrotron who have collaborated in the development and production of the Em# (the first product development Harmony Bus); Alfonso Burgos from ALBA Synchrotron MIS section who gave us technical support in the web development; and finally, to thank Peter Sjöblom and Antonio Milán from the MAX IV Laboratory for its collaboration in the development, testing and production of this project.

REFERENCES

[1] X. Serra, *et al.,* "A Generic Fpga Based Solution for Flexible Feedback Systems", ALBA-CELLS Synchrotron, presented at PCaPAC16, Campinas, Brazil, Oct 2016, paper FRFMPLCO06.

[2] A.Rubini, W. Terpstra, M. Vange, "Self Describing Bus (SDB) – Specification for Logic cores – Version 1.1", April 2013

[3] Wishbone Computer Bus, http://opencores.org/howto/wishbone

[4] M. Broseta et al., "Embedded Control Systems for Programmable Multi-Purpose Instruments", ALBA-CELLS Synchrotron, presented at PCaPAC'16, Campinas, Brazil, Oct 2016, paper THDAPLCO01

[5] Yocto Project, https://www.yoctoproject.org

[6] "Standard Commands for Programmable Instruments", European SCPI Consortium, May 1999.

[7] Tornado Documentation http://www.tornadoweb.org

[8] J. Avila-Abellan et al., "Em# Electrometer Comes to Light", ALBA-CELLS Synchrotron, presented at ICALEPS'17, Barcelona, Spain, Oct 2017, paper TUAPL04