ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA138

YCPSWASYN: EPICS DRIVER FOR FPGA REGISTER ACCESS AND ASYNCHRONOUS MESSAGING

J. Vasquez[†], E. Williams, J. D'Ewart, K. Kim, T. Straumann, SLAC National Accelerator Laboratory, Menlo Park, California, USA

Abstract

The Linac Coherent Light Source II (LCLS-II) is a major upgrade of the LCLS facility at SLAC, scheduled to start operations in 2020. The High Performance Systems (HPS) defines a set of LCLS-II controls sub-systems which are directly impacted by its 1 MHz operation. It is formed around a few key concepts: ATCA based packaging, digital and analog application boards, and 10G Ethernet based interconnections for controls. The Common Platform provides the common parts of the HPS in term of hardware, firmware, and software. The Common Platform Software (CPSW) provides a standardized interface to the common platform's FPGA for all high-level software. YAML is used to define the hardware topology and all necessary parameters. YCPSWASYN is an asynPortDriver based EPICS module for FPGA register access and asynchronous messaging using CPSW. YCPSWSYN has two operation modes: an automatic mode where PVs are automatically created for all registers and the record's fields are populated with information found in YAML; and a manual mode where the engineer can choose which register to expose via PVs and freely choose the record's filed information.

INTRODUCCION

The high repetition rate (1 Mhz) of the next generation Linac Coherent Light Source II (LCLS-II) has brought requirements that can be achievable only with FPGA-based solutions. In that sense, the High Performance System (HPS) was born [1]. As parts of the HPS, the common platform contains all the common hardware, firmware and software that are common to all applications.

YCPSWASYN is a general-purpose EPICS module for register access and asynchronous messaging with the FPGA. The module is based on asynPortDriver, and uses the Common Platform Software (CPSW) as communication layer with the hardware, which is described using YAML files.

THE HIGH PERFORMNCE SYSTEM

The 1MHz operation rate of LCLS-II has set new requirements for controls that are no longer achievable by software application, as was done for LCLS-I. The adopted solution was to implement the control systems with real-time requirements into FPGA-based system. This FPGA-based solution is called the High Performance System (HPS).

The HPS Comon Platorm

As all sub-system using the HPS will have a set of similar requirements, the Common Platform encompasses all the common areas: Intelligent Platform Management Interface (IPMI), Timing Network, Machine Protection System (MPS) network, and EPICS network.

The HPS common platform is based om Advanced Telecommunication Computing Architecture (ATCA). Each sub-system will use an ATCA Advanced Mezzanine Card (AMC) carrier board which contains all the common platform blocks. The AMC carrier supports two doublewide, full height AMC mezzanine cards. An AMC card is where the application specific hardware exists for a given HPS sub-system. Figure 1 shows a block diagram of the HPS ATCA carrier and AMC card.

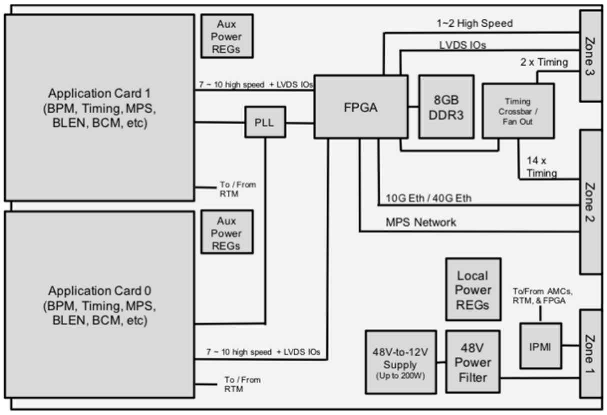

Figure 1: The HPS hardware block diagram.

The help minimize cabling, the ATCA back plane is highly utilized. We are using a dual-star back plane. In slot#1 (1st star connection), we use a Commercial Off The Self (COTS) Ethernet switch to connect Ethernet to all the AMC carriers via the back plane's zone2 interface. In slot#2 (2nd star connection), we use an AMC carrier to connect all the other AMC carriers to the timing and MPS networks via the back plane's zone2 interface as well. The slot#2 AMC carrier's external connection to the timing and MPS networks is via a Rear Transition Module (RTM). The ATCA Zone1 back place interface provides the AMC carriers with -48VDC power (up to 300W) and an IPMI network interface. Figure 2 shows a block diagram of the crate network connections.

[†] jvasquez@slac.stanford.edu

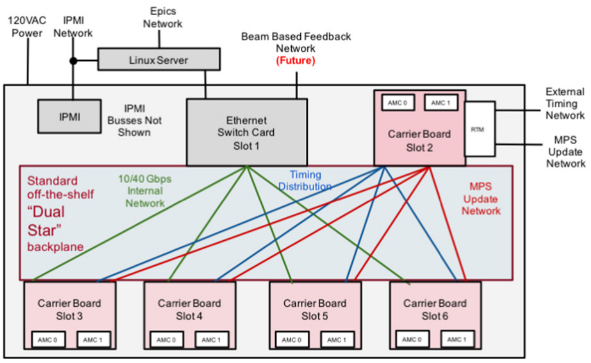

Figure 2: Crate network block diagram.

THE COMMON PLATFORM SOFTWARE

The Common Platform Software (CPSW) provides a common interface to the FPGA for all high-level software.

YAML to Describe the Hardware

CPSW provides a way of describing the hardware by using YAML files. The YAML files are provided by the firmware build system and include all relevant parameters (protocol stack details, register offset and sizes, etc.). A CPSW YAML interpreter automatically construct the system from these files.

Accessing the Entities

Once the hardware has been described using YAML, the user can navigate the hierarchy via Path object which identify an entity using names (string), similar to file system paths. The most basic operation is the lookup, which allows the user to find a specific entity.

Once that the desired entity has been found, an object which instantiates the desired user interface of the entity can be created. Subsequently, the property may be accessed via the instantiated interface.

THE YCPSWASYN EPICS MODULE

YCPSWASYN is an EPICS module based on asynPortDriver. It uses CPSW for accessing the hardware, which is describe using YAML.

YCPSWASYN is a general-purpose driver, doesn't have any application specific functions. It gives access to registers and asynchronous messages. It has two operations mode: auto and manual generation of PVs.

Auto-generation of PVs

In this mode, the module navigates the entire hierarchy defined in YAML and generates PVs for all elements found. This feature is very useful during testing and prototyping phases, enabling the engineers to create an IOC for an application board with minimal effort and time.

Based on the type of register and its properties, records of appropriated type are generated based on Table 1. The record fields are populated with the register properties defined in YAML. Currently, the supported fields are: SCAN, DESC, and Bx/MBBx state names.

The PV names are calculated based on the register path, following the mapping defined in an external text file. Each device name, from right to left, in the path is substitute by its map value. A device can be defined as the top of the PV name. If the device is not found on the maps, the device name is truncated to 3 chars. The register name at the tip of the path is left unmodified. A post-fix is added to the PV name based on the type of register:

- ":Rd" for RO registers,
- ":St" for RW registers,
- ":Ex" for Command registers

Table 1: Margin Specifications

Register information				EPICS PV
Register class	**N. elements**	**Encoding**	**Enum**	**Record type**
IntFiled	1	None	No	longin, longout
			Yes	bi, bo, mbbi, mbbo
		IEEE_754	No	ai, ao
	>1	None		waveform
Sequence Command	N/A			bo
IntField (stream)				waveform

A prefix macro *${P}* is place at the beginning of the name. This macro can be defined in the IOC startup script.

For example, the PV for the RO register located at */mmio/DigFpga/AmcCarrierCore/AxiVersion/BuildStamp* could be *${P}:C:AV:BuildStamp:Rd*.

A RW register will generate 2 record, one for writing and one for read back.

Manual Generation of PVs

The module gives the possibility to disable the auto-generation and define PV manually for specific registers. This gives the possibility to freely choose PV name and field properties, necessary when strict naming conventions must be followed.

The manual generation of PVs requires two steps:

- Map a register to an asyn parameter using a dictionary file, which is an external text file with two columns, one for the register's path, and the other one for the desired asyn parameter name,
- Create an appropriate record for the register, and use the parameter name in the INP/OUT filed in the standard way asynPortDriver does it.

CONCLUSION

YCPSWASYN is an EPICS module used to create EPICS IOCs for the SLAC HPS application boards. The module can automatically create PVs for all the FPGA's registers defined in YAML with minimum effort from the engineer. It also provides a manual mode, given more control to the engineer necessary, for example, to follow pre-defined naming conventions.

REFERENCES

[1] J. Frischet *et al.*, “A FPGA Based Common Platform for LCLS2 Beam Diagnostics and Controls,” in *Proc. IBIC’16*, Barcelona, Spain, 2016, paper WEPG15.