

### Phystream Ltd

22 Elm Way, Melbourn,  
 Royston, Herts  
 United Kingdom, SG8 6UH  
 Phone: +44 (0)20 8469 4019  
 E-mail: [query@phystream.com](mailto:query@phystream.com)  
 URL: [www.phystream.com](http://www.phystream.com)

### Features

- Hardware/Software co-development platform
- Channelized data stream processor
- Supports efficient system design
  - Early system evaluation through the use of the phyCore simulator
  - Flexible partitioning through the use of indirect hardware/software communications channels
  - Predictable performance, through the removal of hardware/software latency
- Rapid development
  - System definition using the phyCore simulator provides a means of publishing a complex design
  - Co-design using the phyCore simulator removes time-consuming integration
  - A soft solution is easily adapted to changing requirements
- Standard tools and methods
  - GNU gcc, gprof, make
- Hardware bus functional model

CoreFacts	
Provided with Core	
Documentation	User Guide, Design Guide
Design File Formats	Xilinx NGC netlist
Constraints Files	Embedded within NGC
Verification	Test Bench, Test Vectors
Instantiation templates	VHDL, Verilog
Reference designs & application notes	Sample Application
Additional Items	PowerPC C Interface Library
Simulation Tool Used	
Modeltech Modelsim v5.8	
Support	
Support provided by Phystream Ltd	

**Table 1: Example Implementation Statistics**

Family	Example Device	Fmax (MHz)	Slices <sup>1</sup>	IOB <sup>2</sup>	GCLK	BRAM	MULT	DCM/DLL	MGT	PPC	Design Tools
Virtex-II Pro™	XC2VP7-7	200	3152	95	7	19	0	2	0	1	ISE 6.1.03i

Notes:

1) Actual slice count dependent on percentage of unrelated logic – see Mapping Report File for details

2) Assuming all core I/Os and clocks are routed off-chip

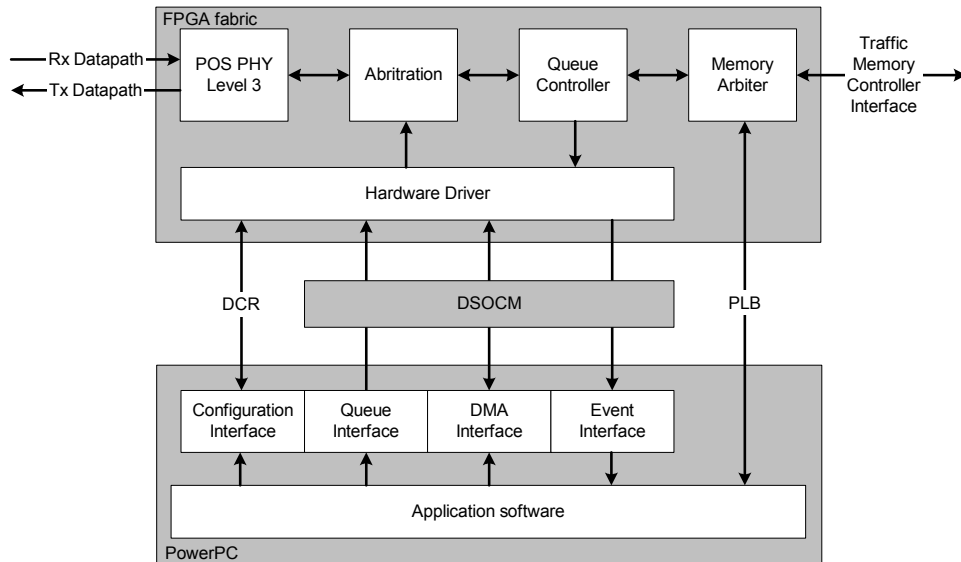


Figure 1: phyCore Dataflow Processor Block Diagram

### Features (continued)

- Hardware Macro
  - 200 MHz operation provides greater than 1Gbps throughput
  - Standard POS-PHY L3 traffic data interface
  - Generic memory controller interface
  - Flexible traffic queuing function (packet, cell)
  - Support for 512 queues
  - DMA Controller
  - Hardware implementation of driver functions
- Software Interface Library
  - High-level interface to queuing function
  - Automatic event handling in response to queue and DMA events
- Development can be accelerated using phySim simulator
  - High-level simulation tool for rapid system modelling, software development and hardware test vector generation
  - Desktop simulation for highest performance (over 100000 times faster than hardware simulation)
  - Instruction set simulator for functional accuracy
  - Operate on target hardware for early prototyping

### Applications

- Multi-protocol layer 2 termination (CES/IMA/HDLC)
- Packet classification and forwarding
- Media coding/decoding
- Multi-protocol encapsulation
- Digital Signal Processing – encryption, compression etc

### General Description

phyCore comprises hardware and software that combine with an immersed processor to produce a powerful dataflow processor. phyCore constitutes a hardware real-time operating system, managing a set of tasks that are triggered by the movement of data through its queuing system. Hardware

implementation of operating system functions offloads the immersed processor, greatly increasing achievable throughput.

The control of a data flow is passed between datapath hardware and software, allowing optimal partitioning between the two. The designer can exploit the parallelism of hardware, in order to accelerate low-level functions, whilst gaining the efficiency and flexibility of software implementation for more complex high-level functions. phyCore combines the low-level integration required to couple hardware and software operation with easily understood user interfaces. phyCore is applicable to most data processing tasks, and is particularly well suited to highly channelized applications.

phyCore supports flexible partitioning of functionality between hardware and software. It is important to make correct partitioning decisions early in the design cycle in order to achieve maximum performance and minimum development time. The phyCore application-level simulator, phySim, is a tool that allows the designer to explore partitioning trade-offs at the very beginning of the development, and at near real time execution speed. Its speed, and ease of use, makes the simulator ideal for system modelling and definition. Furthermore, the system model can be used to provide an accurate simulation for software development, and can also produce hardware test vectors. This co-development approach ensures system compliance, and virtually eliminates the hardware/software integration activity.

phyCore is an event-driven data processing technology that has many benefits over fixed silicon (ASIC and NPU) solutions:

- Provides access to the full processing power of immersed processors
- Free and flexible hardware/software partitioning
- Performance bottleneck of fixed silicon eliminated
- Easy coupling to programmable hardware interfaces
  - Unlimited application with architectural flexibility
- Powerful modelling platform built on industry standard tools
  - Enables transparent progression from model to product
  - Early and accurate performance estimation and rapid prototyping
- Design template provides solid design framework

## **Functional Description**

### **POS-PHY L3**

The traffic data interface is a POS-PHY Level 3 compliant interface. Data is presented in fixed size packets to the POS-PHY L3 interface by an external hardware convergence function. This performs the necessary translation between the external traffic format and the phyCore encapsulation format. Additional fields provide indirect communication with software, and autonomous forwarding of data.

The receive and transmit datapaths have direct access to a set of input and output queues respectively. This allows hardware datapaths to operate independently of the software operation. The receive hardware datapath can make default forwarding decisions for input data, directing the data to a particular output or software queue.

The receive datapath can add a hardware tag to the encapsulated data. phyCore manages the synchronization of the associated queue function and presentation of the tag data to the software. This indirect communication allows independent, and parallel, operation of the hardware and embedded software, enabling maximum throughput.

### **Arbitration**

The arbitration block determines the access priorities to the queue control block, for the processor and two datapaths. A weighted priority algorithm ensures that the real-time critical hardware transmit interface

receives the highest priority, but without starvation of the lower priority receive hardware and software interfaces.

### **Queue Controller**

The queue controller provides primitive queuing functions for use by the hardware and software datapaths. These can be used to gain access to the underlying traffic queues. The software can combine these primitives into more complex operations to support, for example, packet operation. The queue controller informs the hardware driver of completed queue operations via an event queue.

### **Memory Interface**

The memory interface provides a generic interface for connection to a memory controller. The memory interface presents address, data and burst flow control signals.

### **Hardware Driver**

The hardware driver is an extension of the phyCore software interface library. It performs functions, such as translation between hardware and software data structures, which are usually associated with driver software. This boosts overall performance in two ways: firstly, the software load is greatly reduced by the removal of frequent driver functions, and secondly, hardware/software latency is removed by decoupling the operation of the hardware and software.

The hardware driver buffers queue commands issued by the software, and then translates the commands for presentation to the queue control block. Completed command indication is received back from the queue controller through an event queue. These events are translated into software queue event handler calls. Associated data, which may carry partial processing results from the hardware datapath, are formatted and stored in shared memory for access by the event handlers.

The hardware driver includes a DMA controller. DMA commands are translated for presentation to the DMA controller. The DMA controller provides a block memory copy capability between traffic memory and DSOCM (data-side on-chip memory). The hardware driver performs the command translation, execution, and generation of a software callback, independent to the software operation. This removes the bus latency that results from direct access to the traffic memory.

### **Queue Interface**

The queue interface provides the application software access to the traffic queues. The application software can add locally generated data, transfer data between queues, concatenate queues, and fetch data. In addition, callbacks can be registered with each queue command. These are executed on completion of the associated command. This allows the application software to be structured as a set of event handling functions, linked by events that are triggered by queue state changes. The processing for a particular data flow suspends as it waits for access to the traffic queues, and resumes when access has completed, removing any hardware/software latency.

### **DMA Interface**

The DMA interface provides the application software with the means to access data in traffic memory. In order to maintain decoupling of hardware and software, a DMA callback can be registered with each command. Processing of a data flow is suspended while the DMA command is pending, and can be resumed by a callback on completion of the DMA command.

### **Event Interface**

The event interface provides the application software access to the queue of software callbacks generated by the hardware driver. In addition to managing the flow of tasks, the event interface provides access to context data associated with a particular task call. Where appropriate, the associated data includes the hardware tag produced by the receive hardware convergence function. This allows hardware

pre-processing and filtering functions to pass results to the associated software function. It is often the case that the addition of a small amount of application-specific hardware can produce a disproportionately large increase in performance.

### **Configuration interface**

The phyCore interface library provides functions to configure the phyCore sub-system, including the phyCore hard and soft IP, and traffic memory. Configuration of the phyCore hard IP is performed over the DCR (device configuration register) interface.

### **Core Modifications**

Phystream provide a Xilinx NGC netlist to which modifications are not possible. However, modifications can be undertaken as part of an additional service.

### **Verification Methods**

Extensive functional simulation has been performed at both the RTL and gate level using Modelsim 5.8, as well as with the Phystream application-level simulator, phySim. The core has also been successfully operated on a Virtex-II Pro reference platform.

### **Recommended Design Experience**

The design team should have experience in HDL design for FPGAs, and experience of the C processing language. Familiarity with the gnu tools gcc, gprof and make would be advantageous.

### **Available Support Products**

A number of optional items are available for separate purchase:

- phySim, Phystream's application-level simulator
- Training
- Customer development support
- A range of compatible standalone hardware blocks, including:
  - SDRAM memory controller
  - UTOPIA L1/L2 master and slave
  - Programmable telecom bus interface
  - ATM transmission convergence
  - 3-DES encryption/decryption
- Application software:
  - Circuit Emulation Service (CES)
  - Inverse Multiplexing for ATM (IMA)
  - Combined CES/IMA

## Ordering Information

This product is available directly from Xilinx AllianceCORE member Phystream Ltd under the terms of the SignOnce IP License. Please contact Phystream Ltd for pricing and additional information about this product. Contact information for them is on the front page of this datasheet. To learn more about the SignOnce IP License program, contact Phystream Ltd or visit the web:

Email: [commonlicense@xilinx.com](mailto:commonlicense@xilinx.com)  
URL: [www.xilinx.com/ipcenter/signonce](http://www.xilinx.com/ipcenter/signonce)

## Related Information

### Xilinx Programmable Logic

For information on Xilinx programmable logic or development system software, contact your local Xilinx sales office, or:

Xilinx, Inc.  
2100 Logic Drive  
San Jose, CA 95124  
Phone: +1 408-559-7778  
Fax: +1 408-559-7114  
URL: [www.xilinx.com](http://www.xilinx.com)