



PACT



NoC Concepts with XPP-III

Overview

1. Overview

- Target Architecture
- Interface Types
- Communication Mechanisms

2. Building Blocks

- Crossbars
- DMAs
- Arbiters
- Building a Chip around the NoC

3. Motivation

- Hardware Design and Backend
- Software Support

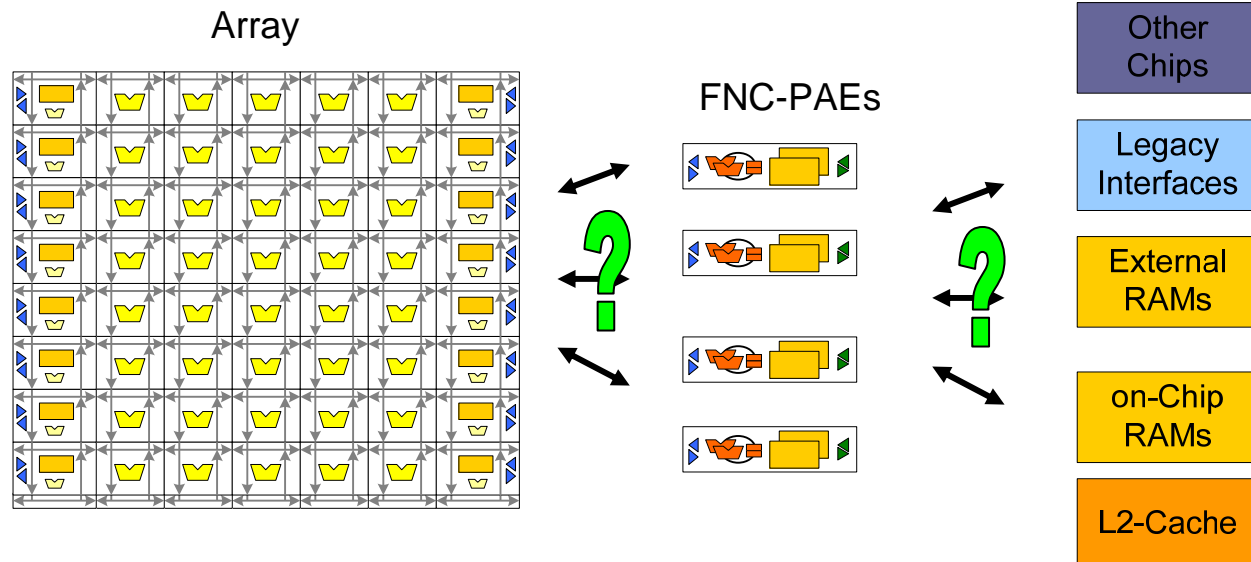
4. Examples and Potential

- HPDP Space processor
- Usage scenarios
- Outlook and Conclusion

XPP-III Signal Processor Architecture





- Signal Processor with 2 Types of Processing Engines:
- FNC-PAEs: 8-ALU VLIW-type control processors (16-bit)
- XPP-Array: Reconfigurable coarse grained Array architecture based on ALU-PAEs and RAM-PAE Elements

How to link processing cores and connect with memories and interfaces ?



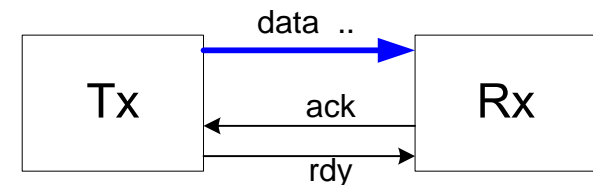
Basic Communication Mechanism: Streams

Three Point-to-point Stream Types:

- *Data stream:*  16-bit wide
- *Event stream:*  1-bit control information
- *Memory channel* with two timely independent streams:
 -  64-bit *data and address stream* (to memory)
 -  64-bit *read response stream* (from memory)

Simple Stream Protocol

- Any single transferred Data/Event/Memory word is transferred with handshake signals
- single *one-word* “packets”
- Sender: enters packets into a queue
- Receiver: reads packets from queue
- Queues may stall, but no data can be lost
- Programmer does not need to care about timing, only the order of packets must be maintained.



How to link Stream-Based Components ?

Solution Alternatives

- a. Convert streams to Bus Protocol, eg. AHB ?
 - Pipelining difficult – we loose bandwidth
 - High overhead per stream for Bus protocol converter
 - Bus topology (multi layer AHB) difficult to design for large chips and high frequency
 - From the shelf peripherals (DMA etc.) can be used
- b. Special Components that directly link streams
 - Full pipelining: maximum bandwidth
 - No overhead for stream converters
 - Pipelined communication structure adaptable to any chip topology
 - Special modules to be designed

NoC Alternatives

- Packet switched or circuit switched ?
 - (a) *Packet switched Network* requires hardware overhead in any node, packets need headers and overhead in the array software.
 - (b) *Circuit switched Network* is simpler and fully under software control. Only payload is transferred.

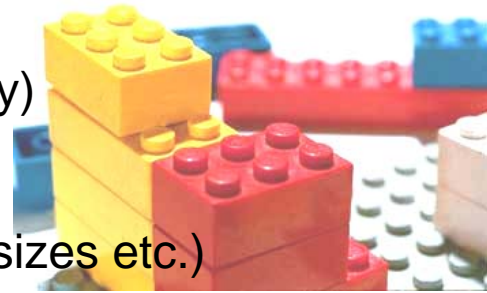
XPP-III Solution: (b) Circuit switched Network

- Crossbars (XBar) for *Data and Event streams*
- Arbiter for *Memory Channels*, routing with address-ranges
- DMA controllers convert *data streams* to *Memory channels*
- Software Setup and Routing Control via FNC-PAE processor
- API for easy setup of application communication topology

NoC Components

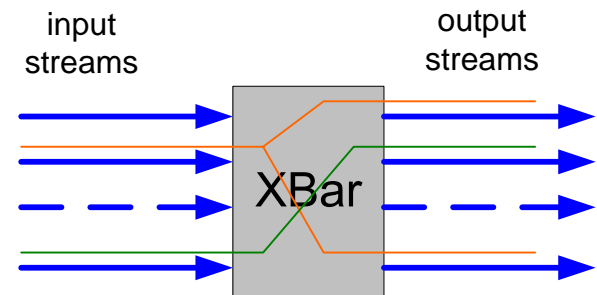
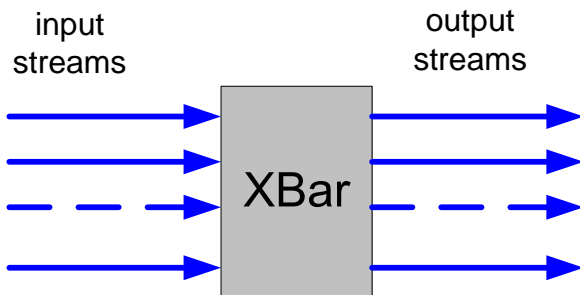
General

- All Components can be stitched together (Plug&Play)
- Any Chip topology possible
- IP models are configurable (number of ports, FIFO sizes etc.)
- Control via pipelined configuration bus through FNC-PAEs and interrupt



XBar

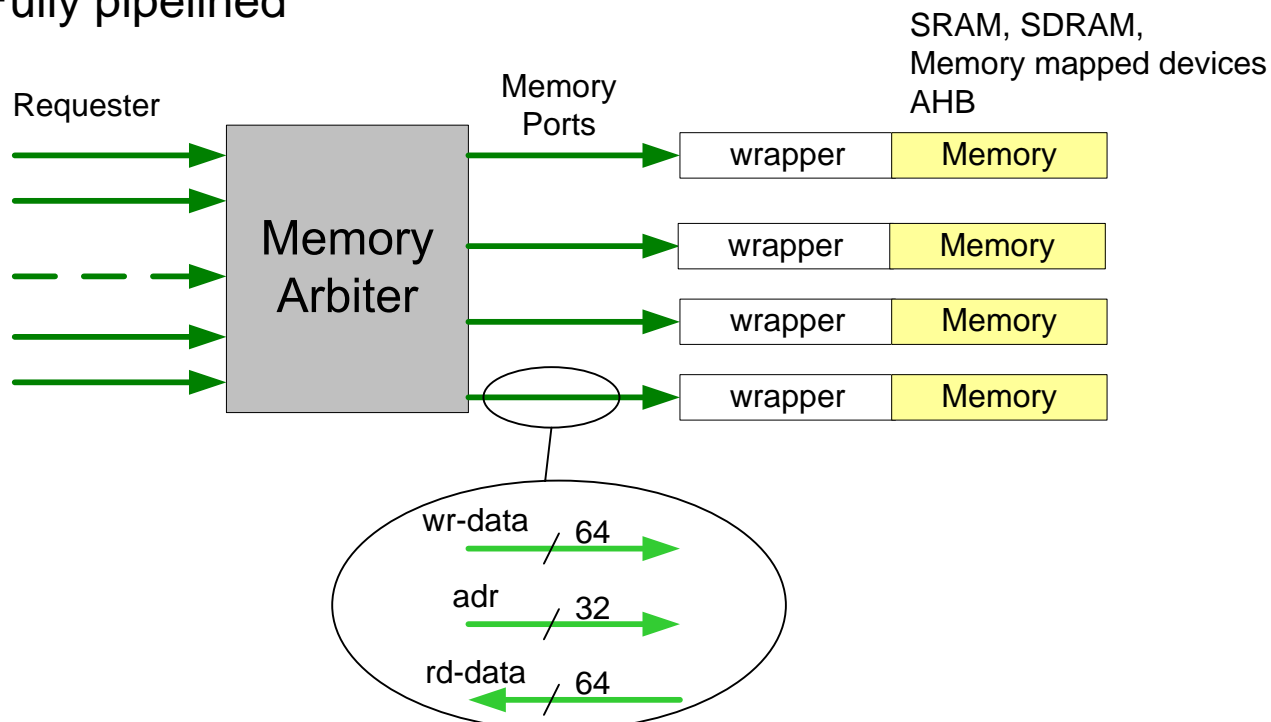
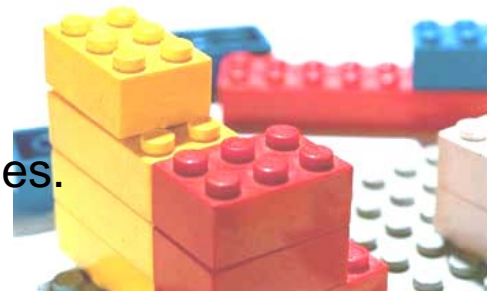
- *Data and Event streams*
- *1..31 inputs, 1 .. 31 outputs, non blocking*
- *1:n connections with full handshake*



Example

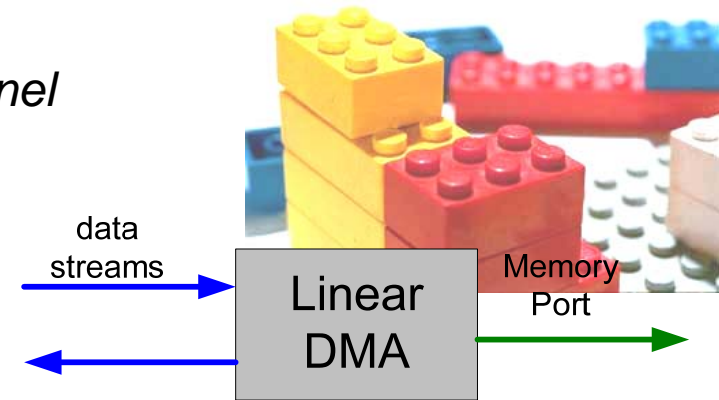
Memory Arbiter

- 1 .. 14 Requester Ports
- 1..4 Memory Ports, 4 simultaneous memory accesses.
- Programmable “fair” Requester priorities
- Support for Atomic Memory access
- Routing to Memory ports with programmable address ranges
- Fully pipelined



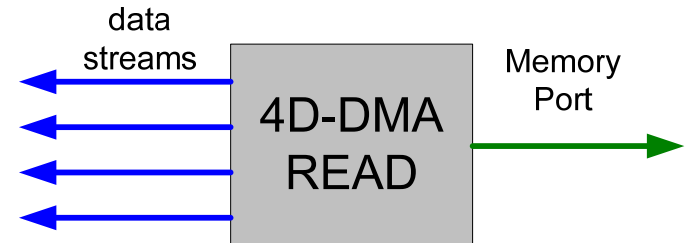
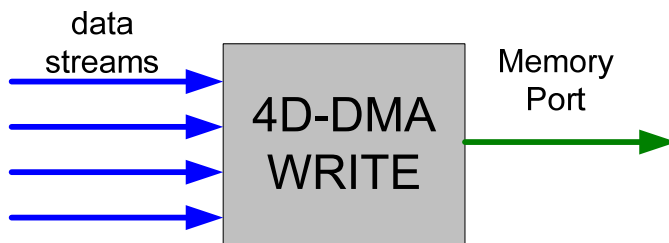
Linear DMA

- Converts *Data streams* to *Memory channel*
- Linear address pattern
- Optimized Memory access, 64-bit bursts
- Arbitrated Memory read and write



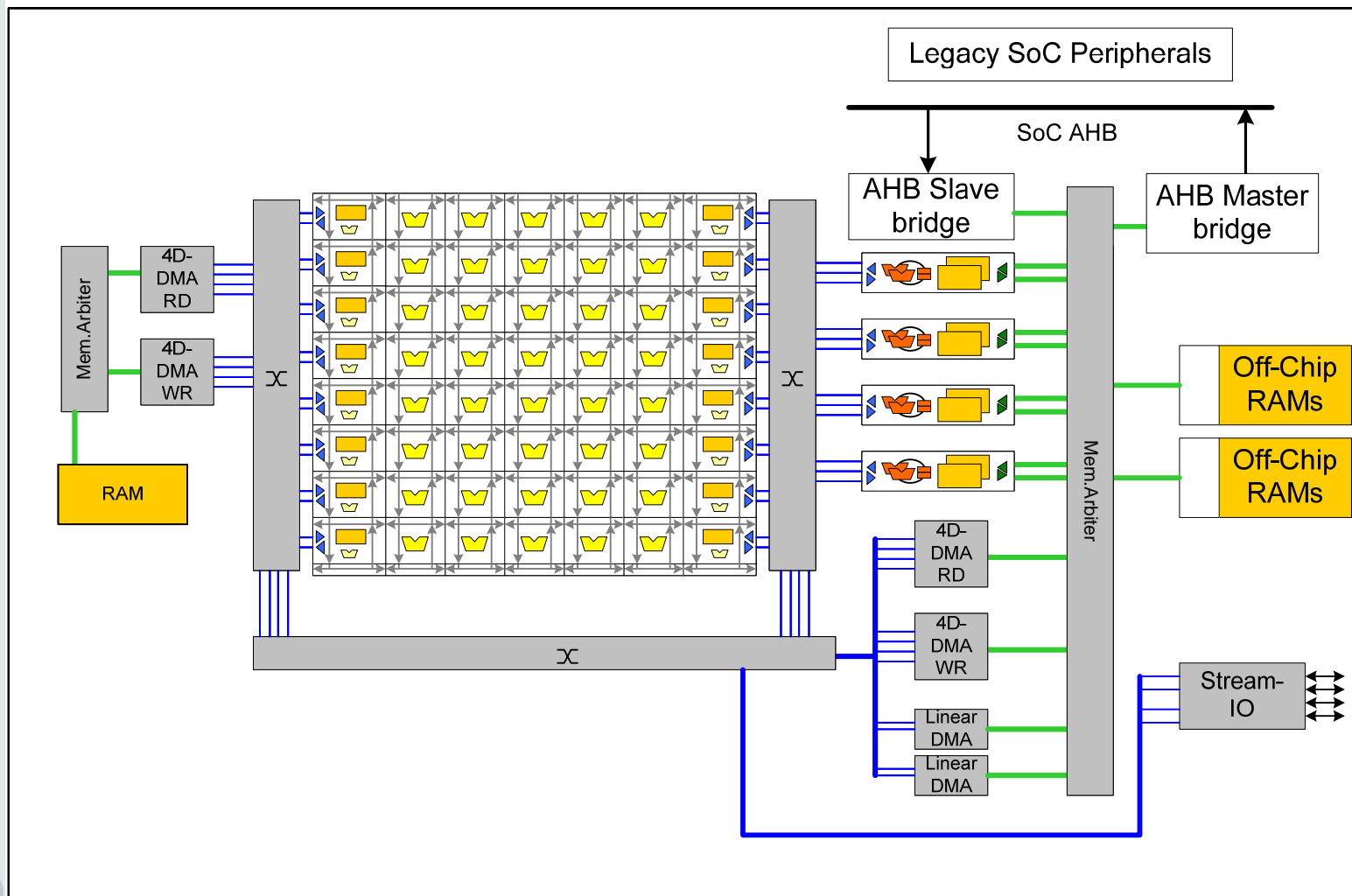
4D-DMA-Read, 4D-DMA-Write

- Converts up to 4 data streams to a *Memory channel*
- Complex address patterns (e.g. moving area in video frame etc.)
- Optimized Memory access, 64-bit bursts
- Interrupt on final physical memory write (for save software synchronization)



Building the NoC from Components

- 1.) Connect XBars
- 2.) Add DMA Controller and Memory Arbiter
- 3.) Add Memory wrapper, SoC Peripherals etc.



Motivation for the chosen NoC Architecture

Bandwidth

- Streaming applications (SDR, Video, Codecs) require guaranteed bandwidth on (most) channels
- Communication pattern is fixed per application
- Reconfigurable platform can serve multiple streams in parallel
- Automatic synchronization of streams and processing resources w/o software overhead (fully transparent)
- Allows full pipelining which is difficult with bus protocols.
- Save Clock domain crossing within streams

Chip Design

- Simple Plug &Play IP modules for any SoC Topology
- Easy verification due to simple protocol
- XBar/Arbiter structure can be specified for optimal floorplanning
- Additional stream-pipe insertion for higher frequencies and timing closure

Control through FNC-PAE Library

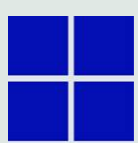
- FNC-PAEs are optimized for control tasks
- API for connectivity: “connect *device A* with *device B*”
- API for DMAs and other components
- Semaphores & Mutexes
- Cycle Accurate SystemC Simulator
- Multi-Chip Simulation

SoC interfacing

- Most SoCs are AMBA based
- Conversion of standard protocol to XPP NoC:
 - *AHB Master bridge* provides XPP-NoC access to AHB Peripherals, Memory etc.
 - *AHB Slave bridge* provides AHB-Processor access to XPP Memory (external)
 - *AHB FNC-IO bridge* that allows AHB-Processor access to XPP Components (e.g. DMA controller) via NoC

Stream interfacing to other NoC Architectures

- FIFOs for input and output
- Self synchronizing to external NoC speed and latencies
- Example in the following ..

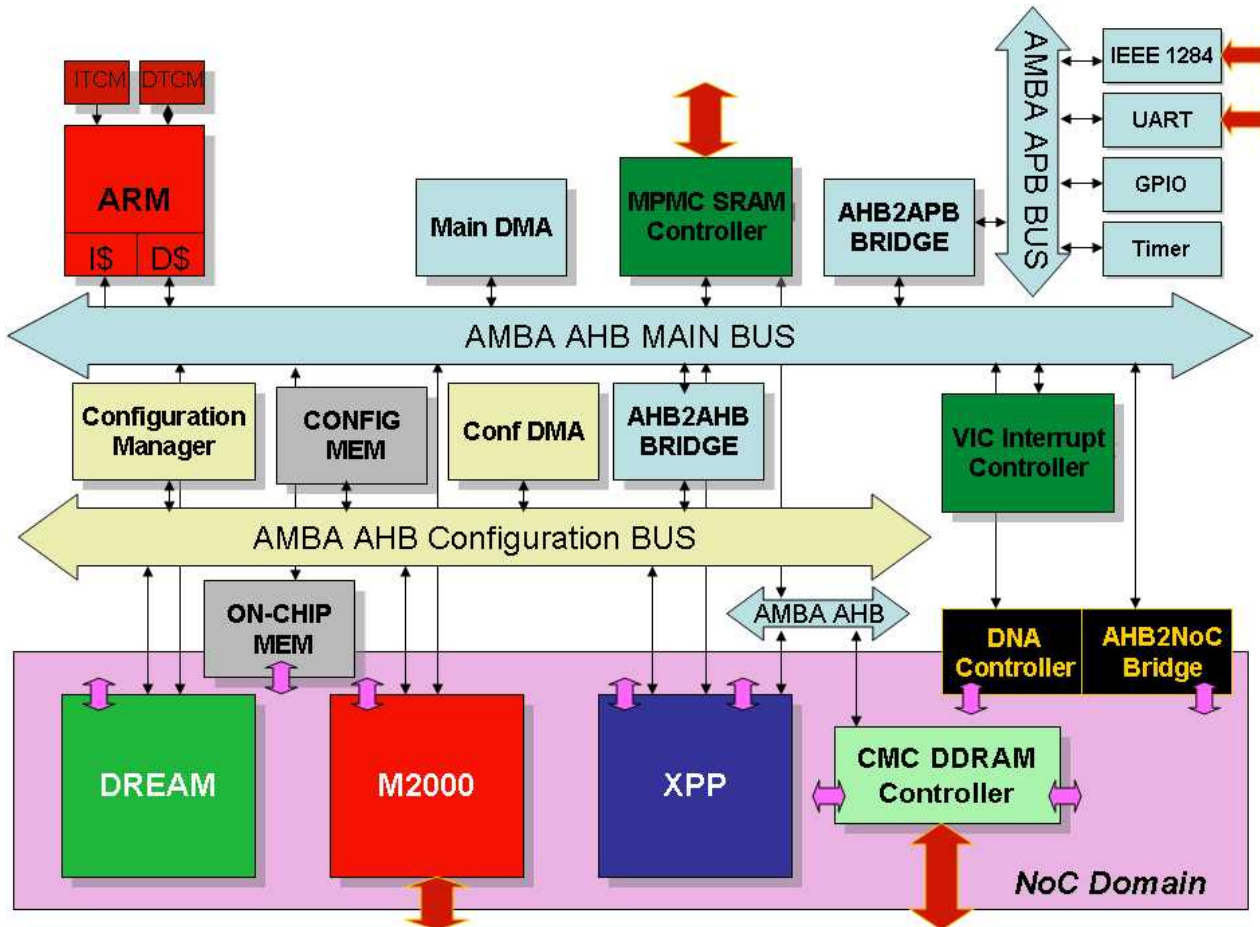


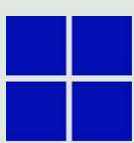
PACT

Example: MORPHEUS SoC

MORPHEUS Chip

- ARM Control Processor + 3 Reconfigurable Engines (HRE)
- HREs embedded into 8-node *ST Spidergon NoC* + DMA within nodes
- Fully functional Silicon (MPW) (ST CMOS 090)



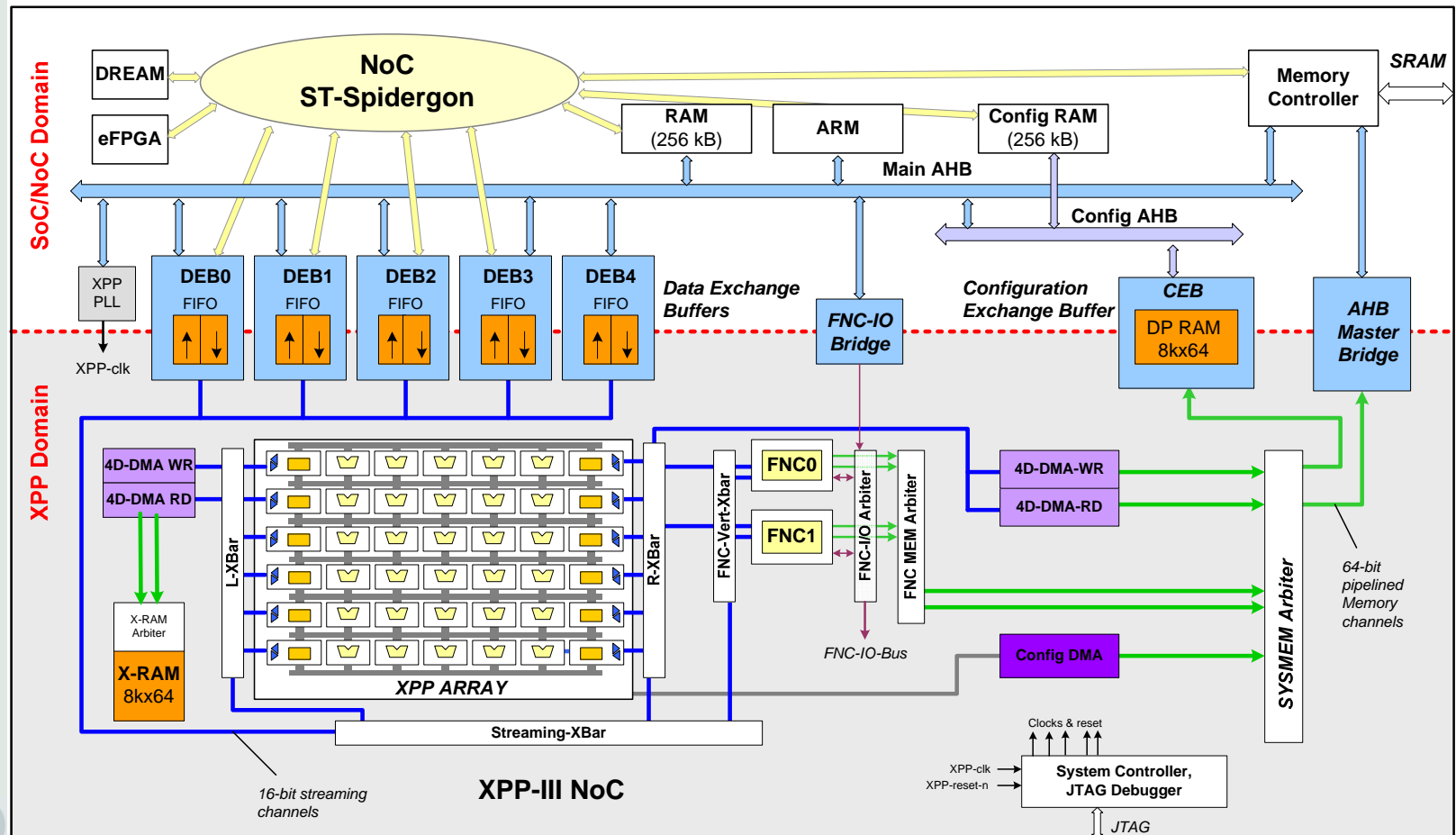


PACT



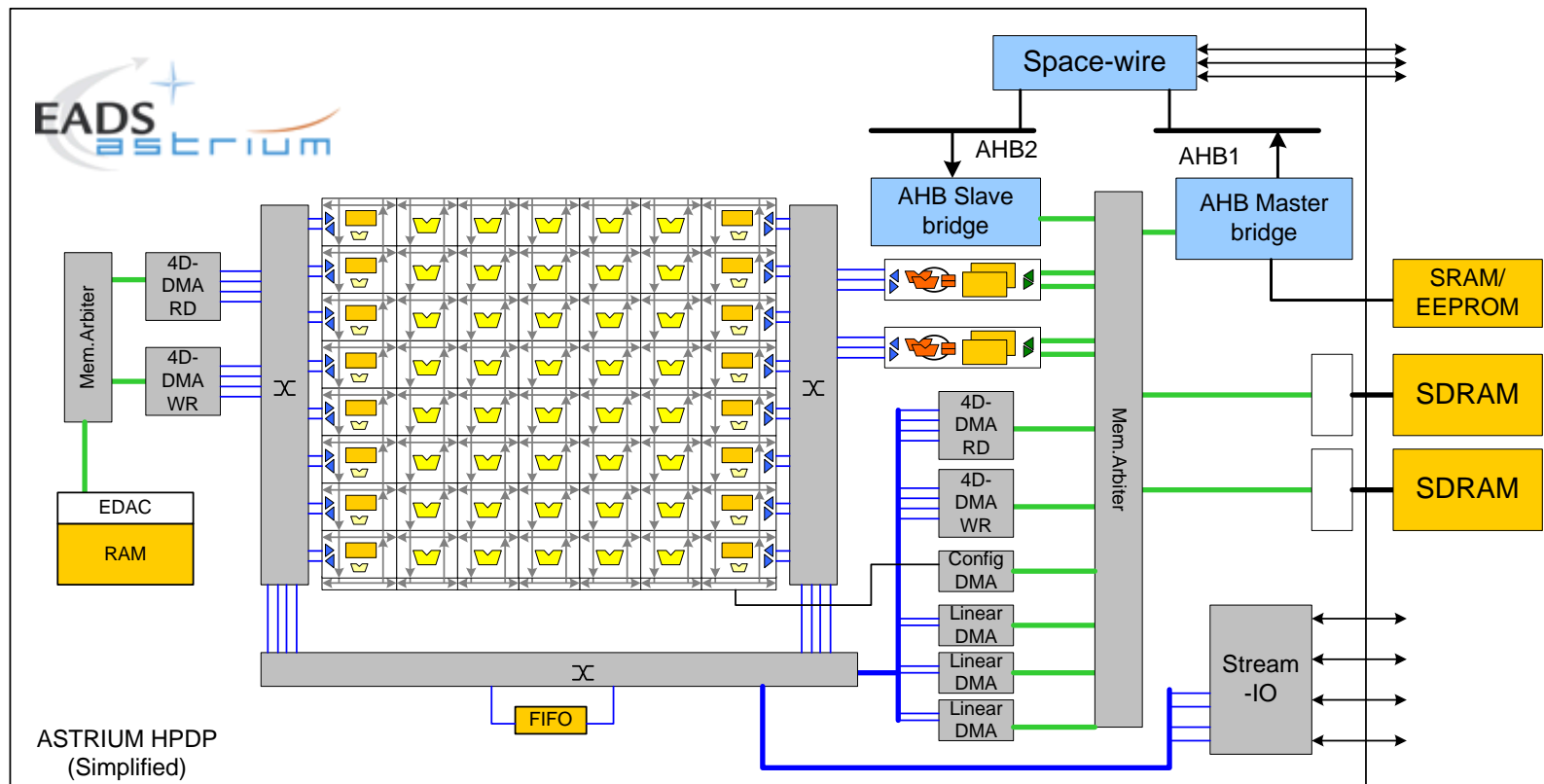
MORPHEUS XPP-III und SoC/NoC Interfaces

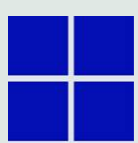
- Data Exchange Buffers (DEB): FIFOs to NoC and Main AHB
- Configuration Exchange Buffers (CEB): Dual Ported RAM, (XPP code)
- AHB Master bridge to Memory Controller
- FNC-IO-Bridge: ARM Access to Components



ASTRIUM's HPDP Processor

- XPP-III processor core
- Spacewire links
- *Stream-IO* for inter-chip communication (+ redundancy) extends on-chip NoC to other HPDPs or data sinks/sources.
- Multiple clock domains

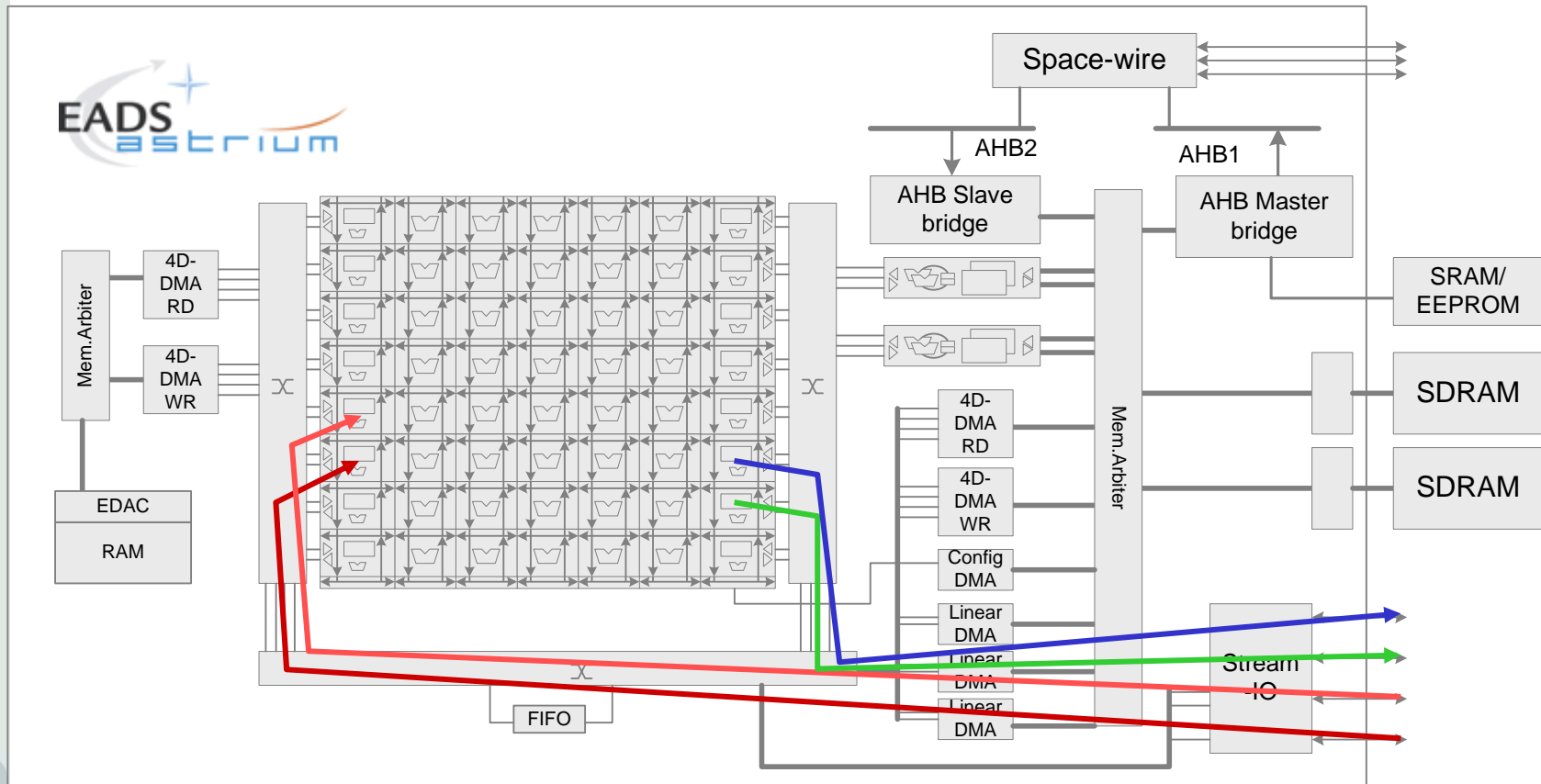


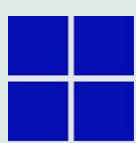


PACT

HPDP NoC Usage Scenario (1)

- Stream-IO complex in
- Stream-IO complex out

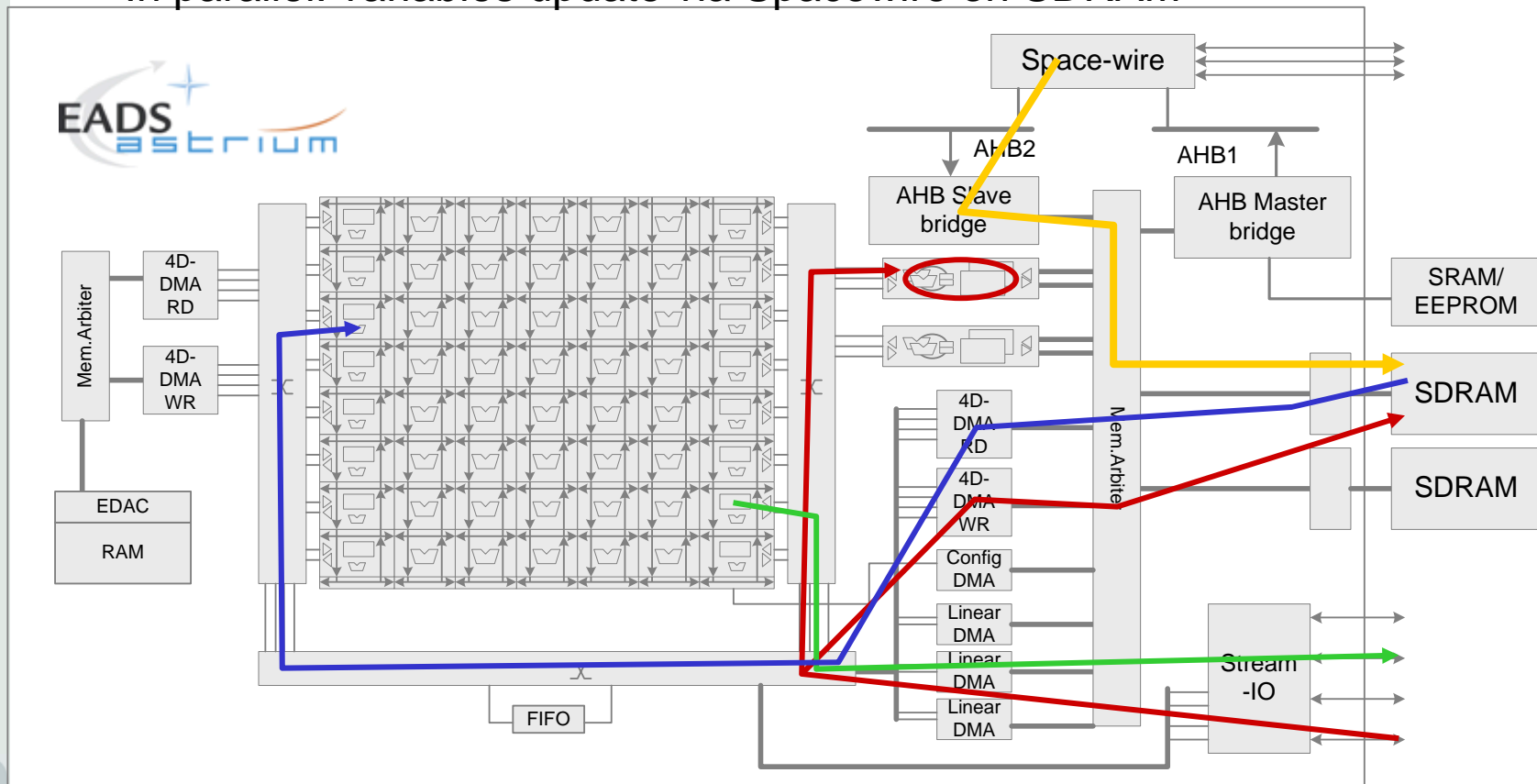




PACT

HPDP NoC Usage Scenario (2)

- Stream-IO in → FNC0 *and* Stream-IO in via 4D-DMA-WR → SDRAM
- FNC0 performs header detection and starts 4D-DMA-RD
- SDRAM via 4D-DMA-RD → Array
- Array → Stream-IO out
- In parallel: variables update via Spacewire on SDRAM

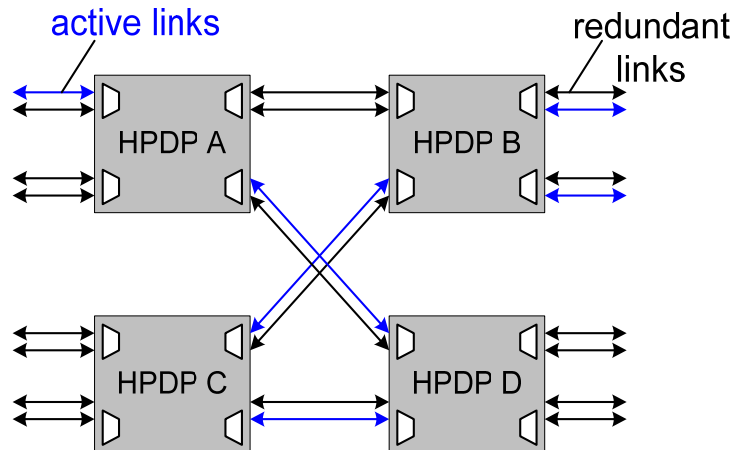


HPDP Chip

- Reconfigurable communication Network (for regenerative payload)
- Guaranteed bandwidth e.g. for SDR applications
- Implicit redundancy: defect paths can be bypassed
- Supervision of results: transparent stream snooping with FNC-PAEs
- parallel background operations without affecting bandwidth of running applications

System

- Inter-chip and inter-board redundancy with additional Stream-IO Pins (cold standby)



Available NoC Components (from the shelf)

NoC IP Modules

- Fully verified
- Configurable IP
- Cycle accurate SystemC model
- Typically PACT configures and connects the IP:
The verified core and Simulator is delivered to customers

IP

- XBars (Data, Event)
- 4D-DMA, Linear DMA, Array configuration DMA
- RAM-IO (direct Array memory I/O)
- Memory Arbiter
- FNC-IO Arbiter and Hierarchical decoder
- Stream-IO
- Stream FIFOs
- SRAM Memory wrapper
- Clock Domain crossing for all stream types
- Stream Pipes
- Protocol keeper (to stop clock for debugging)



NoC Roadmap

- XBars with dynamic routing controlled by event or data streams
- Wider data paths (eg. with parity, Floating Point-words)
- Extended latency schemes: guaranteed latency (Xbars and Arbiter)
- XBars with n:1 connects and guaranteed latency
- NoC Generator

XPP-III NoC

Superior flexibility and bandwidth

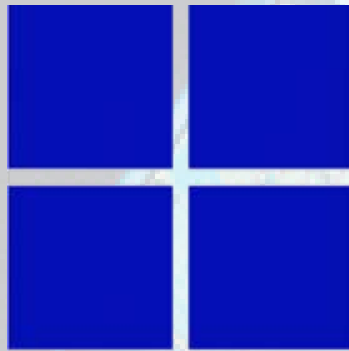
Plug & Play Modules

Communication topology under software control

Low area budget and high frequency, Silicon-proven

SDK and Simulator of a reference design or HPDP
available

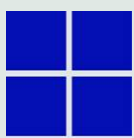
THANK YOU !



PACT

BACKUP

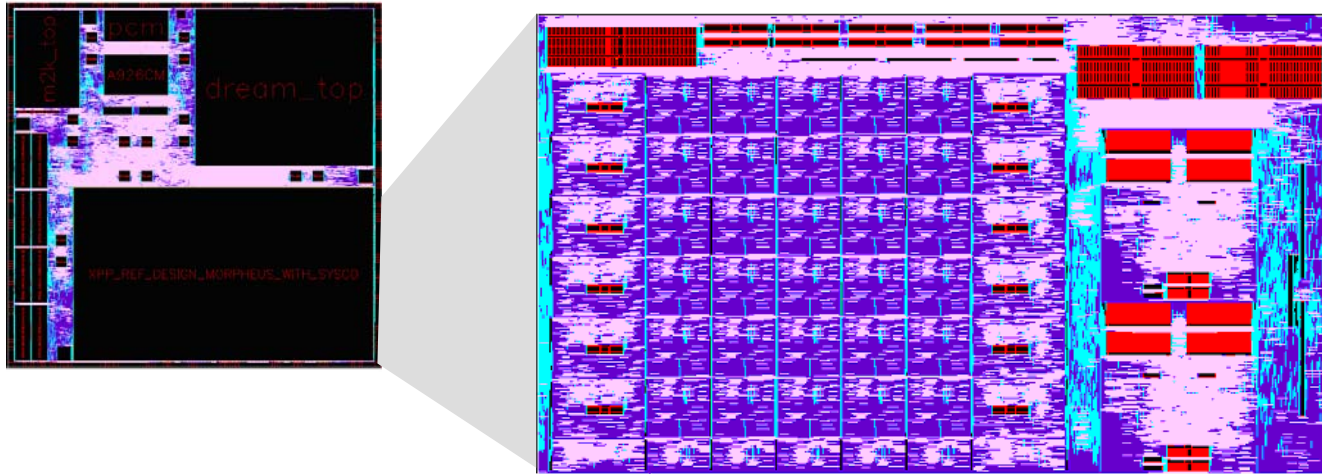




PACT



Morpheus chip



- SoC and XPP-III fully functional
- XPP-III section: 150 MHz @1.0V, 200 MHz @1,12V
- XPP-III dyn. power: 7.6 mW/MHz (max. Stress Test)
- SoC 110 mm², XPP-III ~40 mm² (no area optimisation)
- Application: Professional Video