TOPIC

EMBEDDED PRODUCTS

• Dyplo is an abbreviation for Dynamic
an operating system extension for
Dyplo delivers a full infrastructure for FP
hybrid (embedded) platform.
• Miami SOMs provide a scalable high
platform with a powerful CPU and the
on its SOM connectors.
This prevents design
complex PBC
mi Boards with
P will speed up y
The Kit combines a Miami and Florida with touch
development w
compatible Linux distribution
• IPware are I
d to integrate in your design to accelerate your development.
They come
enabled drivers.
software to kick-start your
ting
a FPGA provided
lenges of high density and
IPWARE
DYPLO INTERFACE

Embedded
in your future

Accelerate your development

Dyplo

software driven threaded FPGA development using partial reconfiguration techniques

Dirk van den Heuvel
Principal Consultant

March 15, 2016 / 15:35-16:00, Newton 1 and 2
SEFUW : SpacE Fpga Users Workshop, 3rd edition

TOPIC EMBEDDED PRODUCTS

Embedded in your future

Topic in a nutshell

- Real Embedded company; 170 employees
  - 130+ embedded software developers
  - 20+ FPGA designers
  - 10+ board designers
- Founded in 1996, privately owned
- 3 Business unit:
  - Since 1996: Consultancy: the Netherlands
  - Since 2006: Project execution: Europe and North America
  - Since 2014: Product development and sales: World Wide

XILINX® ALLIANCE PROGRAM PREMIER MEMBER

TOPIC EMBEDDED PRODUCTS

Embedded in your future

Our ecosystem

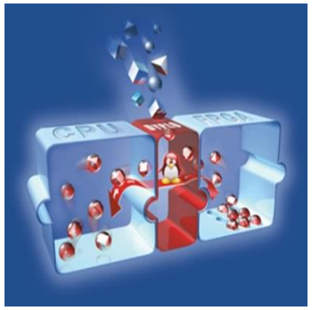
FLORIDA Powered by TOPIC

MIAMI Powered by TOPIC

DYPLO® Powered by TOPIC

IPWARE Powered by TOPIC

DSIGN Powered by TOPIC

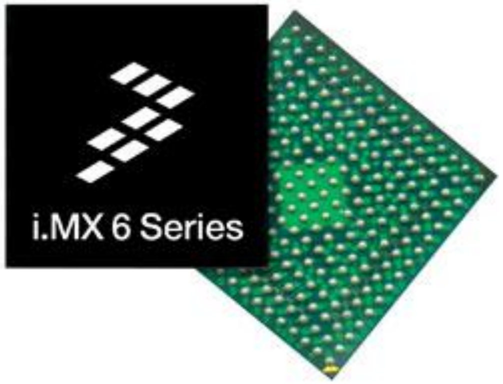


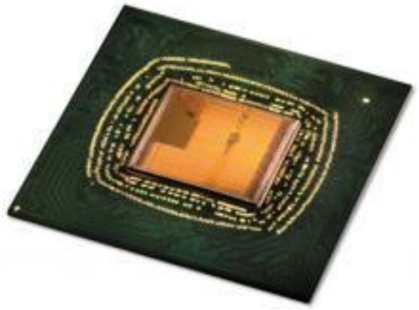
- **Accelerate** our customers developments is key in everything we do!
- Total **ecosystem** of embedded solutions
- Embedded **hard-and software** solutions
- All products are **combinable and compatible**
- High **quality** solutions for **today**, build with the **future** in mind

TOPIC EMBEDDED PRODUCTS

Embedded in your future

Advances in (MP)SoC technologies
Heterogeneous architectures



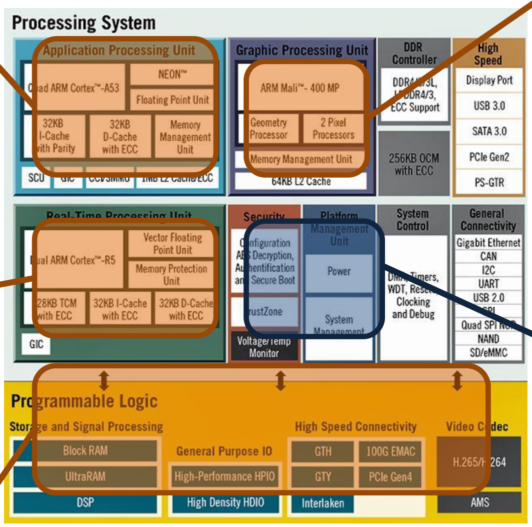


Embedded in your future

Processing capabilities

64 bit quad core application processor

graphics processor



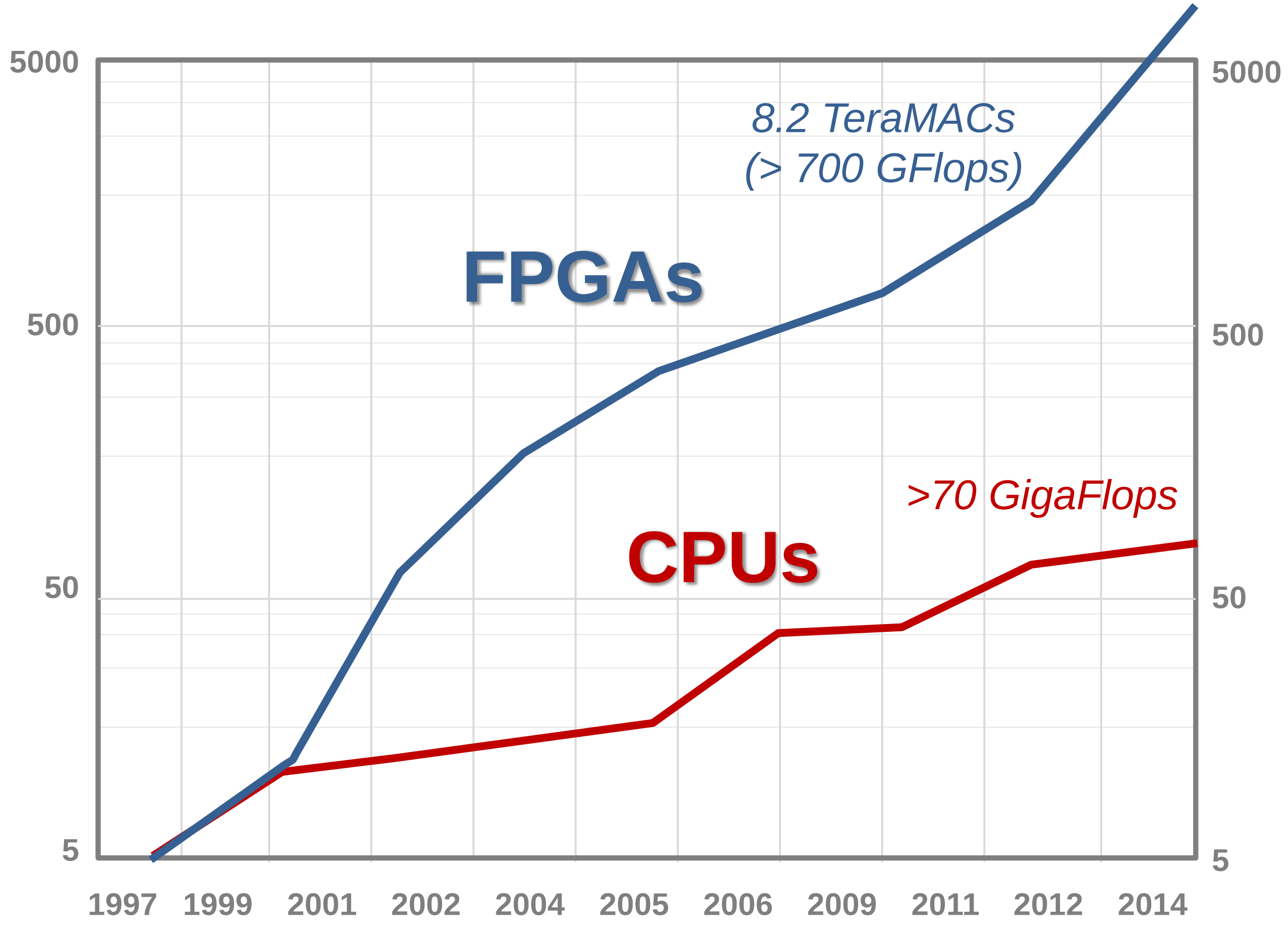
32 bit dual core real-time processor

in-system peripherals

FPGA fabric

Embedded in your future

Issues with heterogeneous systems



FPGA performance (Kintex Ultrascale)

CPU performance (quad core i7)

How to program?

Embedded in your future

Common programming model?

- Application processors
  - (Certified) Operating systems
  - Multi-core programming support
  - Programming in C, C++(11) and higher abstractions
- Microcontrollers
  - Minimal functionality operating systems
  - Libraries with (certified) functionality
  - Programming in C/C++
- GPU
  - Coexists with an application processor
  - Programmed using e.g. OpenCL API
  - SIMD like instructions
- DSP
  - Limited OS functionality
  - Programmed using C with dedicated compilers
- FPGA
  - No operating system (right?)
  - Programmed using VHDL/Verilog every time from the bottom-up
  - IP usage for design speedup
  - Increased usage of C, C++ and OpenCL for algorithmic part

TOPIC EMBEDDED PRODUCTS

Embedded in your future

Integration bottleneck

- Distributed connected heterogeneous processing units
- Proprietary connectivity, no common interconnect, no common API
- C/C++(11) most common programming language, but not portable code
- Programming abstraction is very different
- Software programming needs “threaded” hardware
- No common code base
  - Application is sum of many partial applications

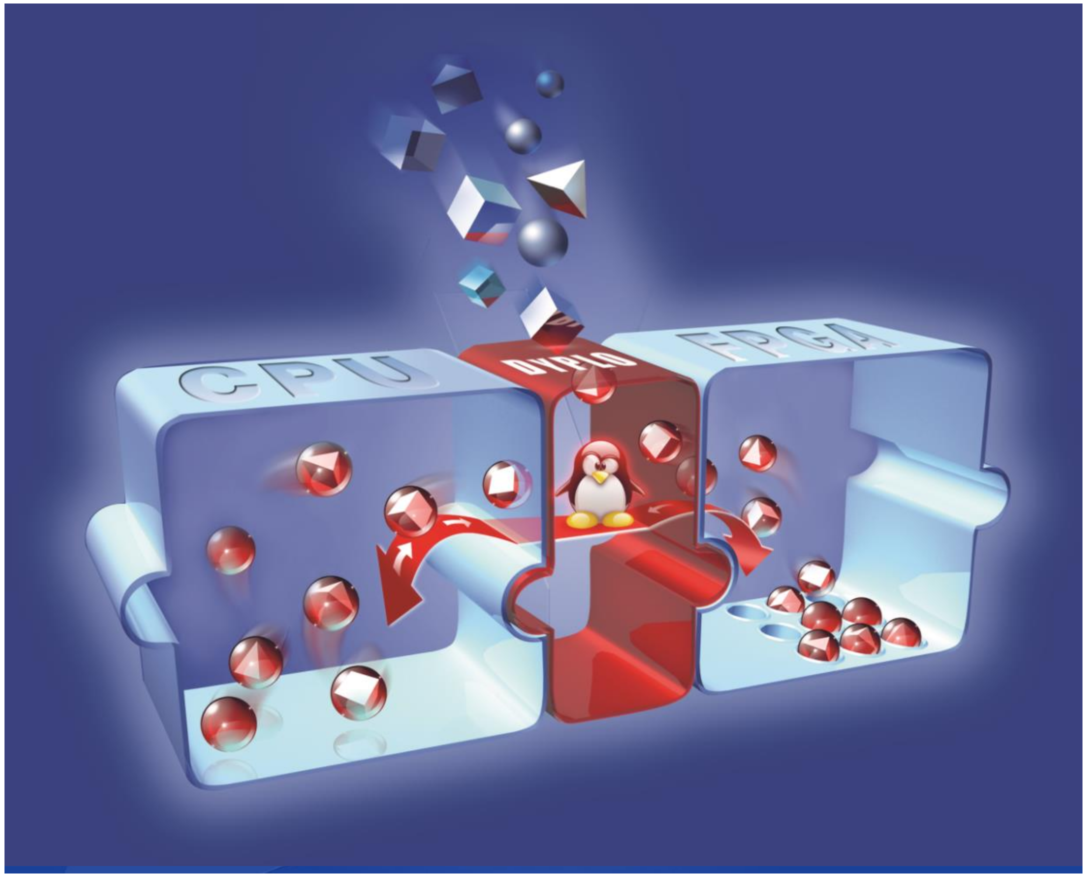
Embedded in your future

The Dyplo Concept

DYnamic Process LOader

Providing developers the ability to connect to various processing units of choice while dynamically loading, distributing and controlling tasks

Hardware/software integration

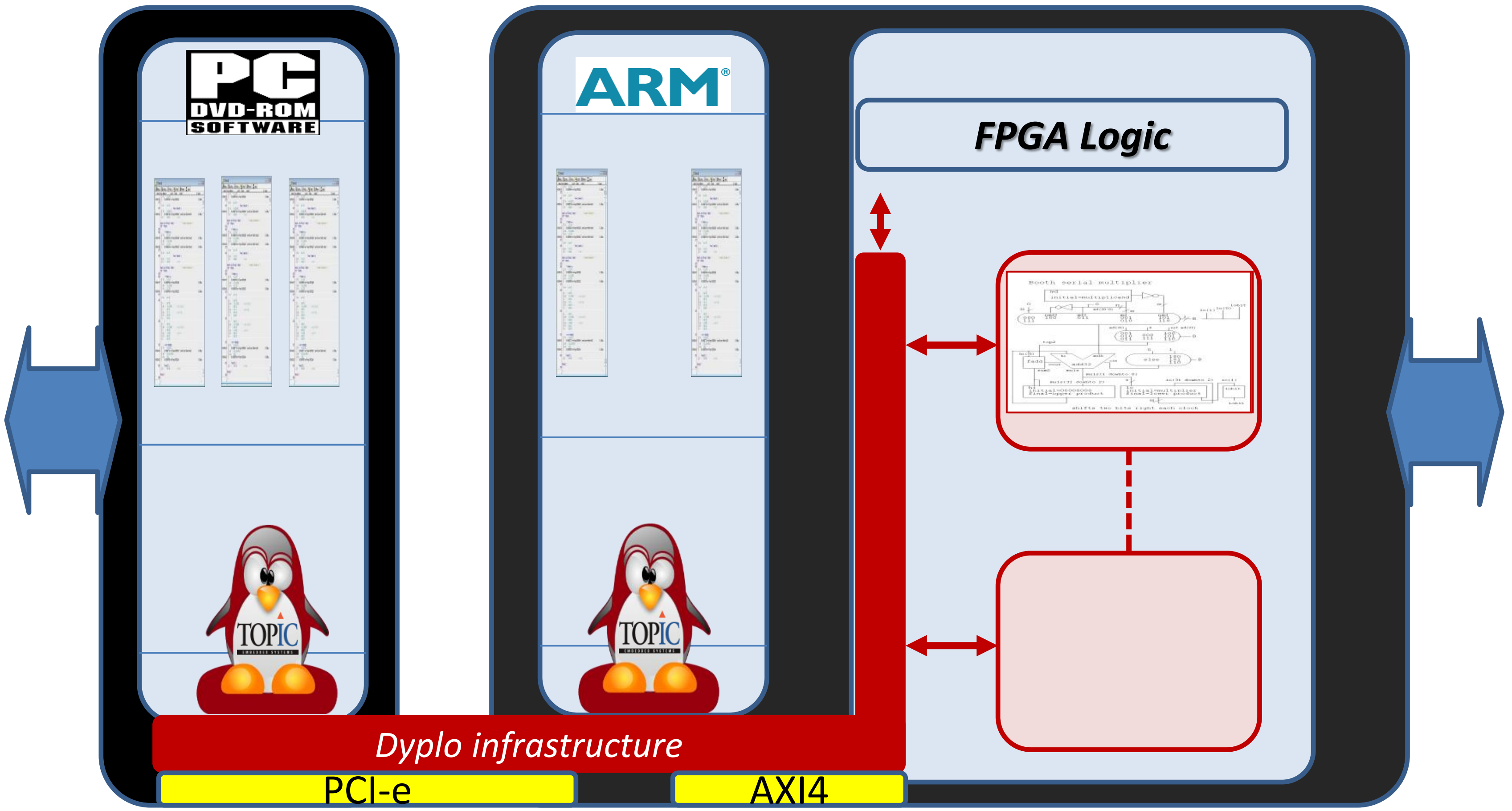


OS support for:
•Threading
•Memory management
•synchronization

•Fixed functionality
•Complex interfaces
•Complex integration with SW

What brings Dyplo?

- ✓ Simplified heterogeneous platform development
- ✓ Out-of-the-box integration of processor & FPGA
- ✓ Faster “Time-to-Space”
- ✓ FPGA programming made software-friendly
- ✓ Runtime re-use of FPGA fabric
- ✓ Architectural exploration

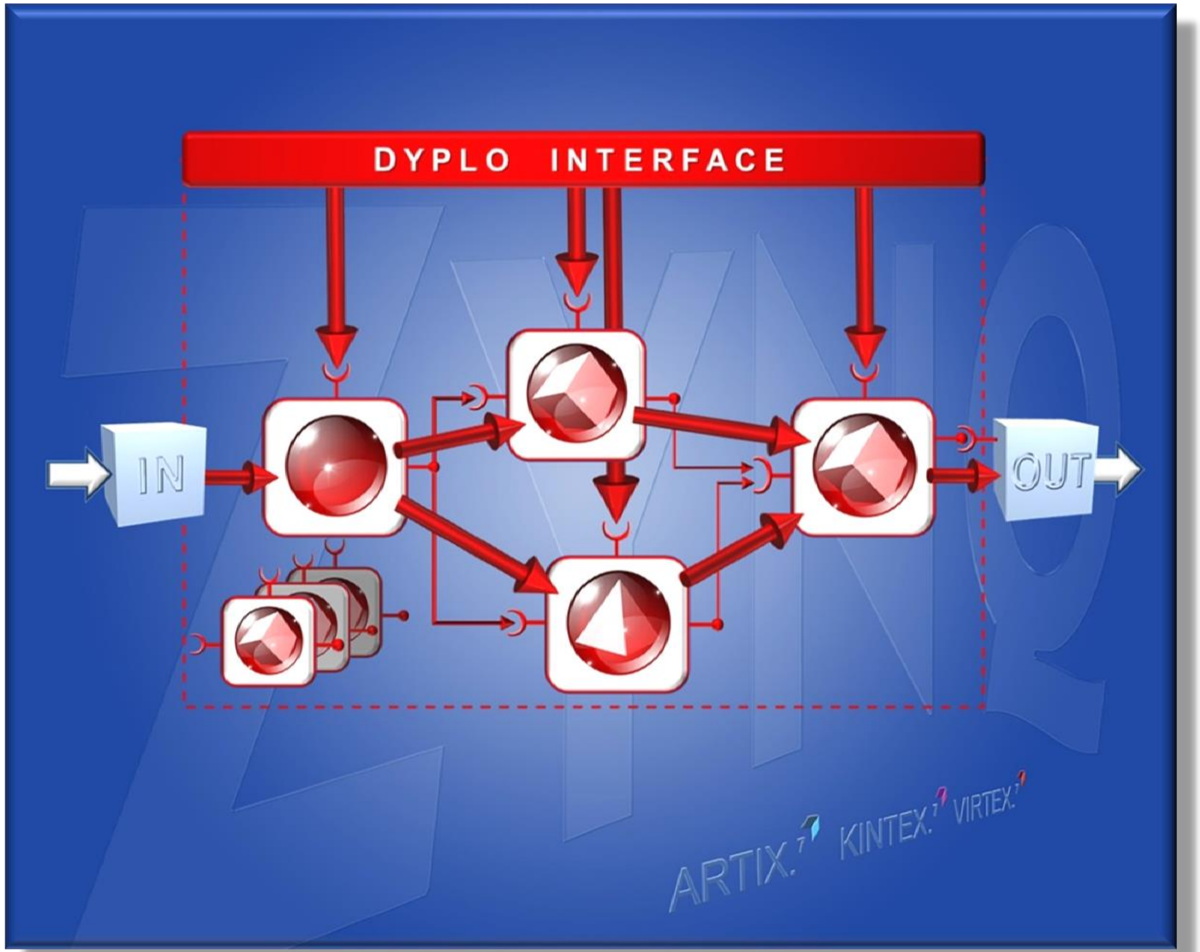


TOPIC EMBEDDED PRODUCTS

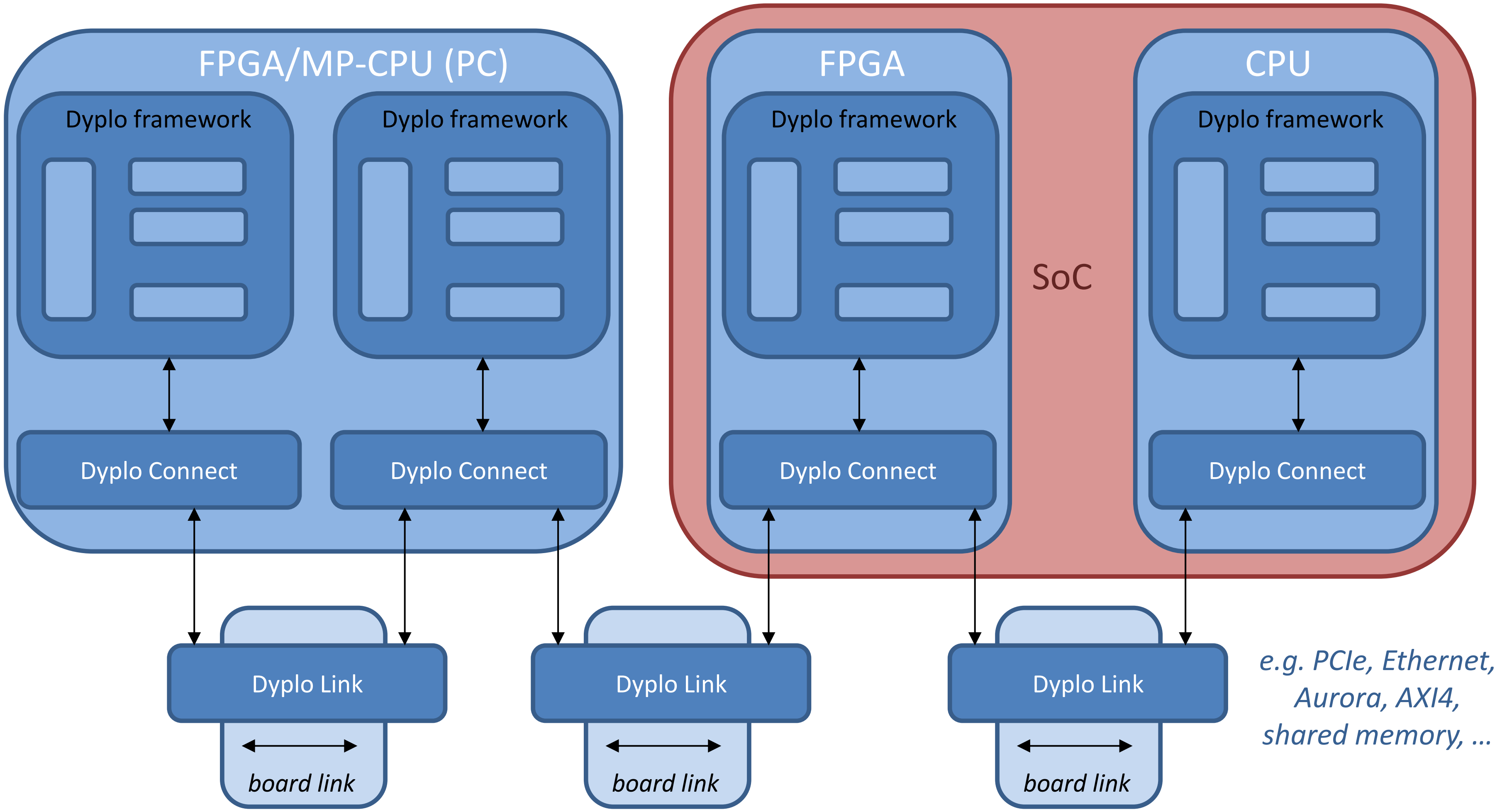
DYPLO Powered by TOPIC

Embedded in your future

Process Networks



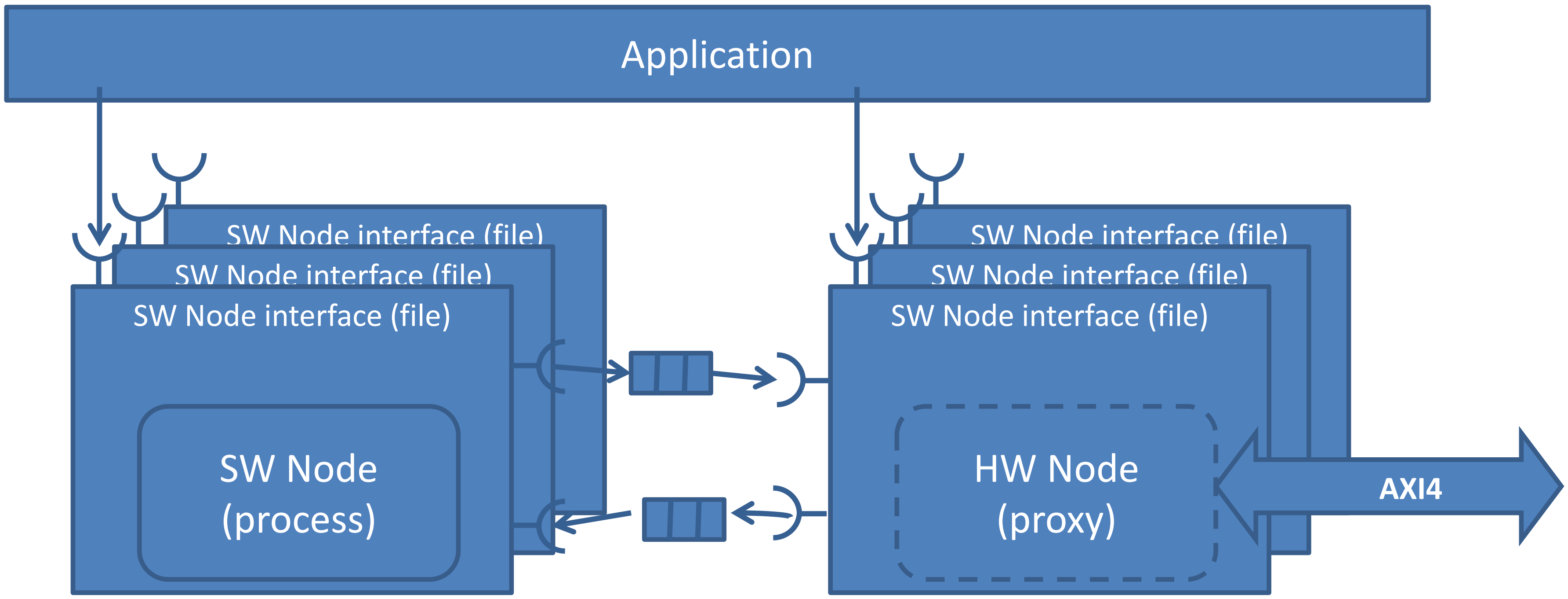
Unified programming infrastructure



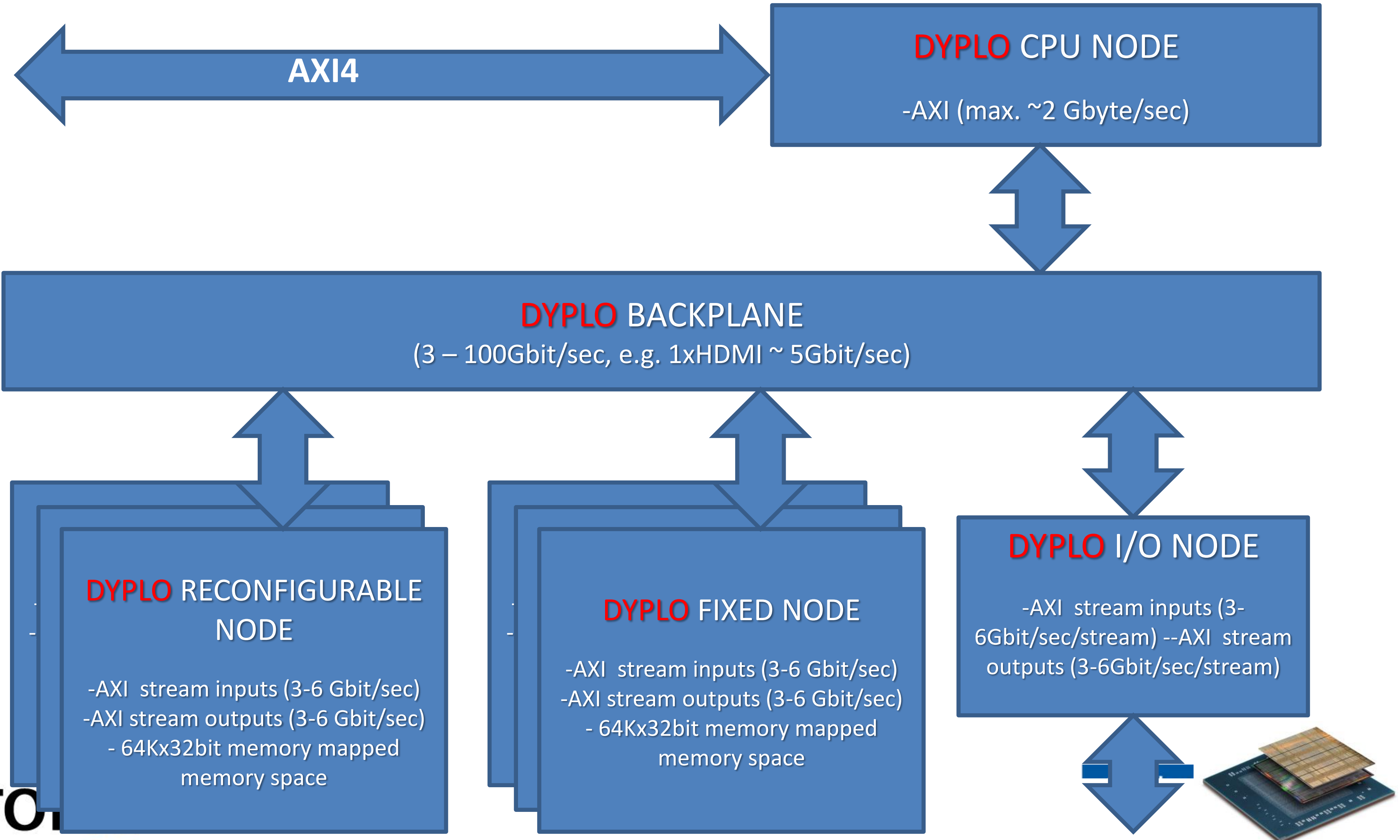
e.g. PCIe, Ethernet, Aurora, AXI4, shared memory, …

Embedded in your future

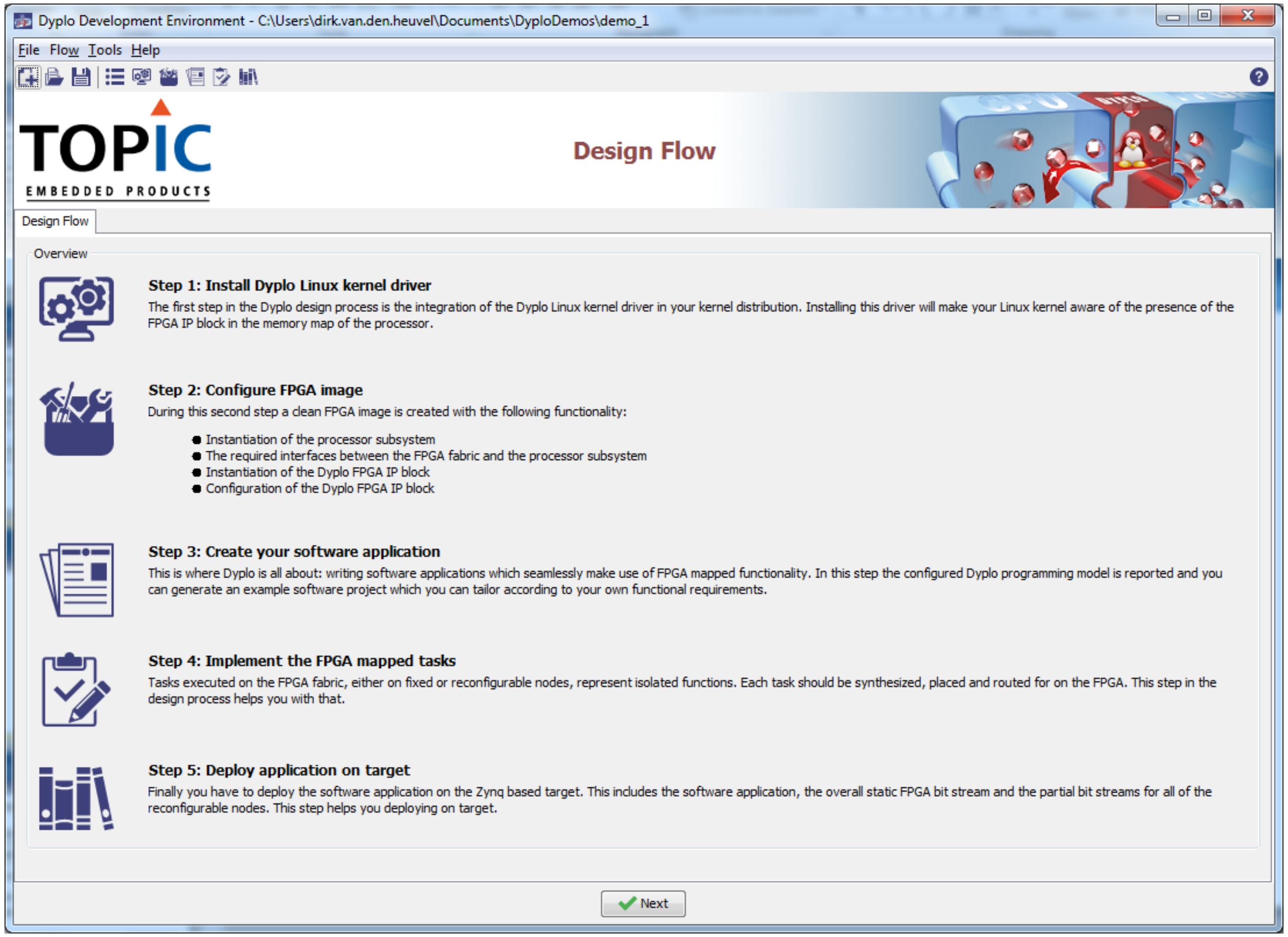
Dyplo Software Infrastructure



Dyplo Programmable Logic Infrastructure


AXI4
DYPLO CPU NODE
-AXI (max. ~2 Gbyte/sec)
DYPLO BACKPLANE
(3 – 100Gbit/sec, e.g. 1xHDMI ~ 5Gbit/sec)
DYPLO RECONFIGURABLE NODE
-AXI stream inputs (3-6 Gbit/sec)
-AXI stream outputs (3-6 Gbit/sec)
- 64Kx32bit memory mapped memory space
DYPLO FIXED NODE
-AXI stream inputs (3-6 Gbit/sec)
-AXI stream outputs (3-6 Gbit/sec)
- 64Kx32bit memory mapped memory space
DYPLO I/O NODE
-AXI stream inputs (3-6Gbit/sec/stream) --AXI stream outputs (3-6Gbit/sec/stream)

Dyplo Development Environment


Dyplo Development Environment - C:\Users\dirk.van.den.heuvel\Documents\DyploDemos\demo_1
File Flow Tools Help
TOPIC EMBEDDED PRODUCTS
Design Flow
Design Flow
Overview

Step 1: Install Dyplo Linux kernel driver

The first step in the Dyplo design process is the integration of the Dyplo Linux kernel driver in your kernel distribution. Installing this driver will make your Linux kernel aware of the presence of the FPGA IP block in the memory map of the processor.

Step 2: Configure FPGA image

During this second step a clean FPGA image is created with the following functionality:

- Instantiation of the processor subsystem
- The required interfaces between the FPGA fabric and the processor subsystem
- Instantiation of the Dyplo FPGA IP block
- Configuration of the Dyplo FPGA IP block

Step 3: Create your software application

This is where Dyplo is all about: writing software applications which seamlessly make use of FPGA mapped functionality. In this step the configured Dyplo programming model is reported and you can generate an example software project which you can tailor according to your own functional requirements.

Step 4: Implement the FPGA mapped tasks

Tasks executed on the FPGA fabric, either on fixed or reconfigurable nodes, represent isolated functions. Each task should be synthesized, placed and routed for on the FPGA. This step in the design process helps you with that.

Step 5: Deploy application on target

Finally you have to deploy the software application on the Zynq based target. This includes the software application, the overall static FPGA bit stream and the partial bit streams for all of the reconfigurable nodes. This step helps you deploying on target.

TOPIC EMBEDDED PRODUCTS

DYPLO® Powered by TOPIC

Embedded in your future

Software driven threaded FPGA development

- Treaded FPGA design using Dyplo
  - Extensive use of partial reconfiguration techniques
  - Seamless integration in the software design flow
- Dyplo Development Environment (DDE)
  - Frame-work → Development → Deployment
- Time/cost-to-Space
  - Dyplo certifiable framework
  - Re-use of FPGA fabric
  - In-flight reprogramming of functionality

TOPIC EMBEDDED PRODUCTS

Embedded in your future

Want to know more?

Eindhovenseweg 32-C, 5683 KH BEST, The Netherlands
P.O. Box 440, 5680 AK BEST, The Netherlands

Phone: +31 499 336969 | Fax: +31 499 336970

www.TopicProducts.com | info@TopicProducts.com

Embedded
in your future