Reconfigurable High-speed Asynchronous I/O Ports for Flexible Protocol Support

Suleiman Abu Kharmeh and Simon Hollis

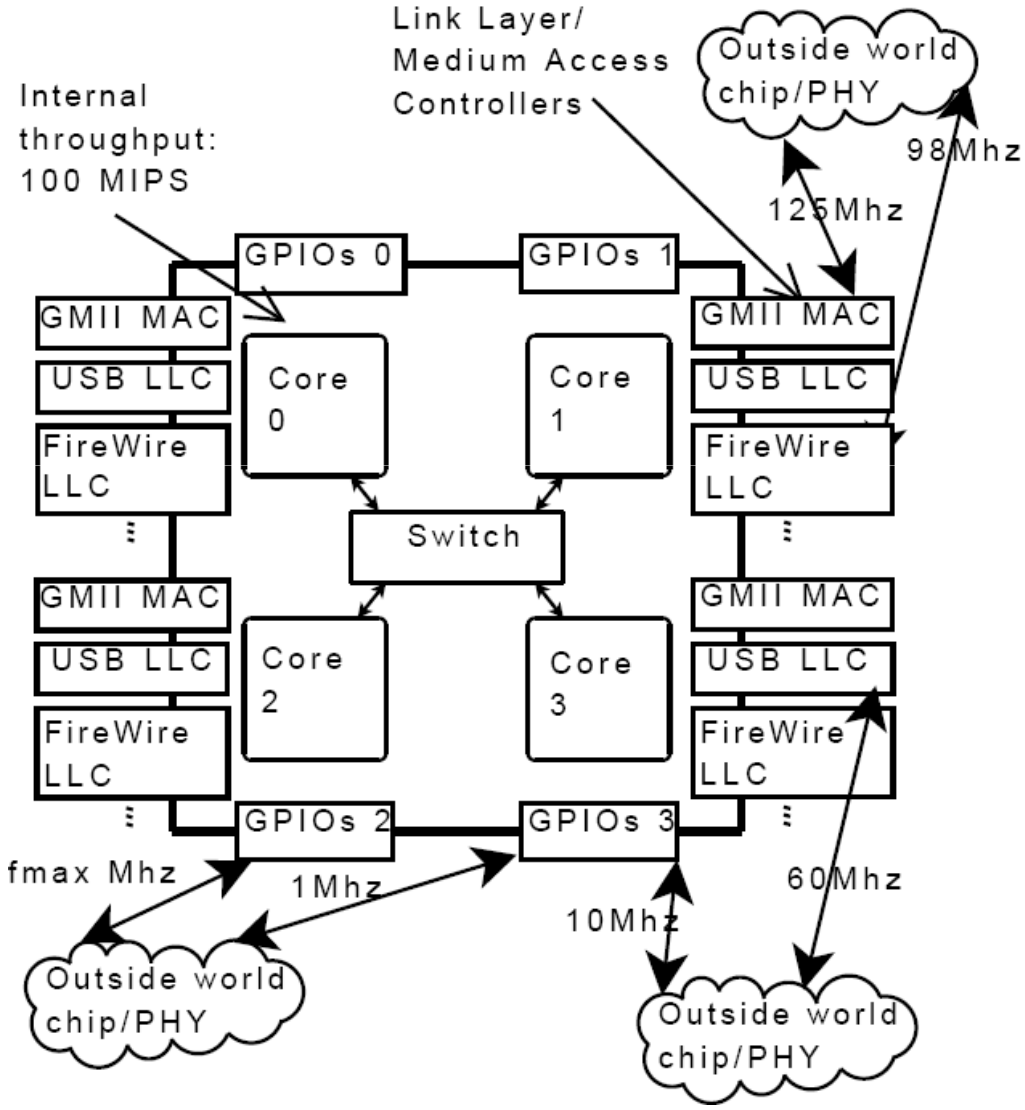
Department of Computer Science, University of Bristol;
XMOS Semiconductor Ltd.

{kharmeh,simon}@cs.bris.ac.uk

Introduction

- Digital Communication systems/protocols becoming extremely serial
- Current speed range: 100’s Mbps to 10’s Gbps
- A growing number of standards that share many properties: All based on the OSI model
- Our focus: MAC/LLC layer to get data in and out of the chip

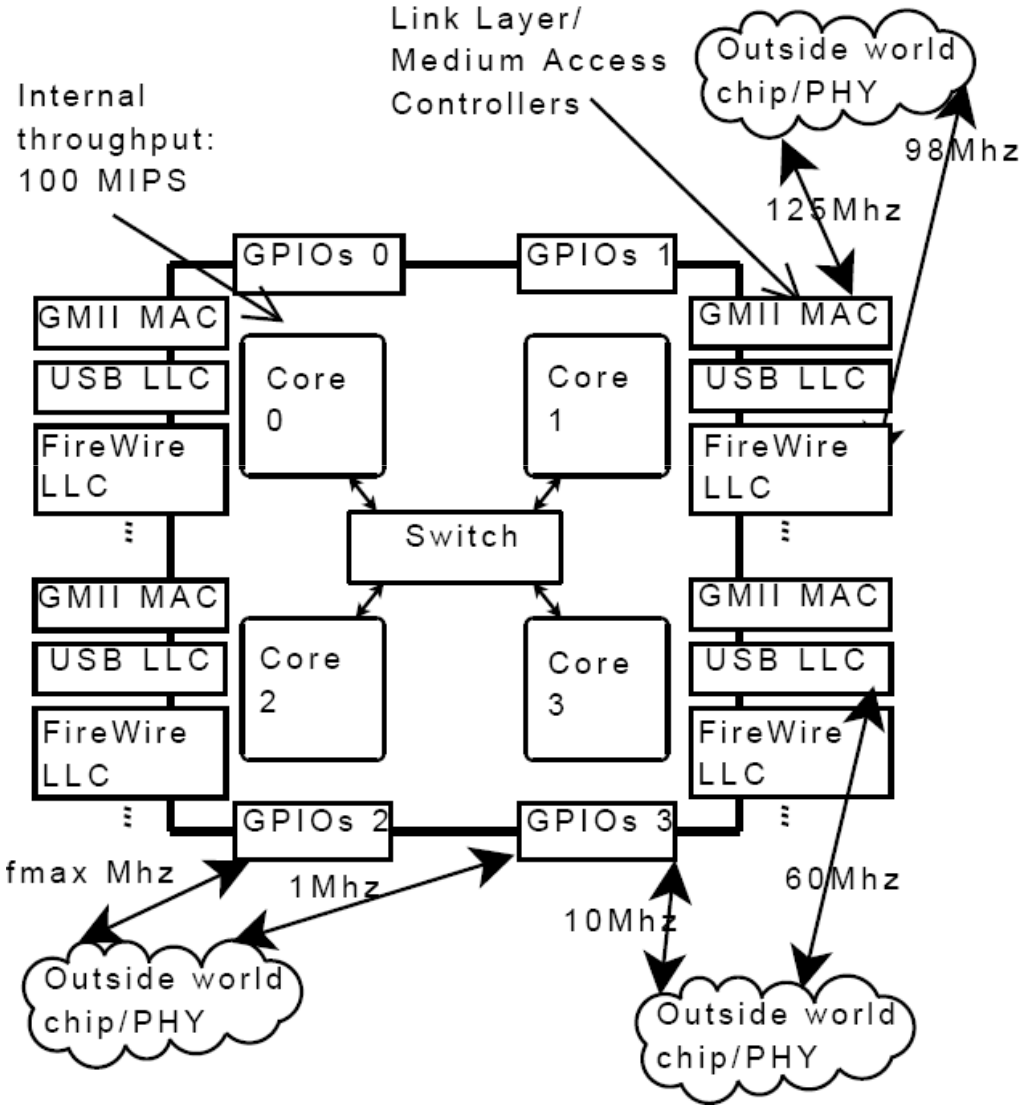
Problem: one fixed controller per supported protocol per core



Problem Explained

- Integration of MAC/LLC controller per protocol
- Not flexible to incorporate specification changes
- Not able to support new standards
- Separate hardware block for each protocol supported
- Not practical in a multi-core chip

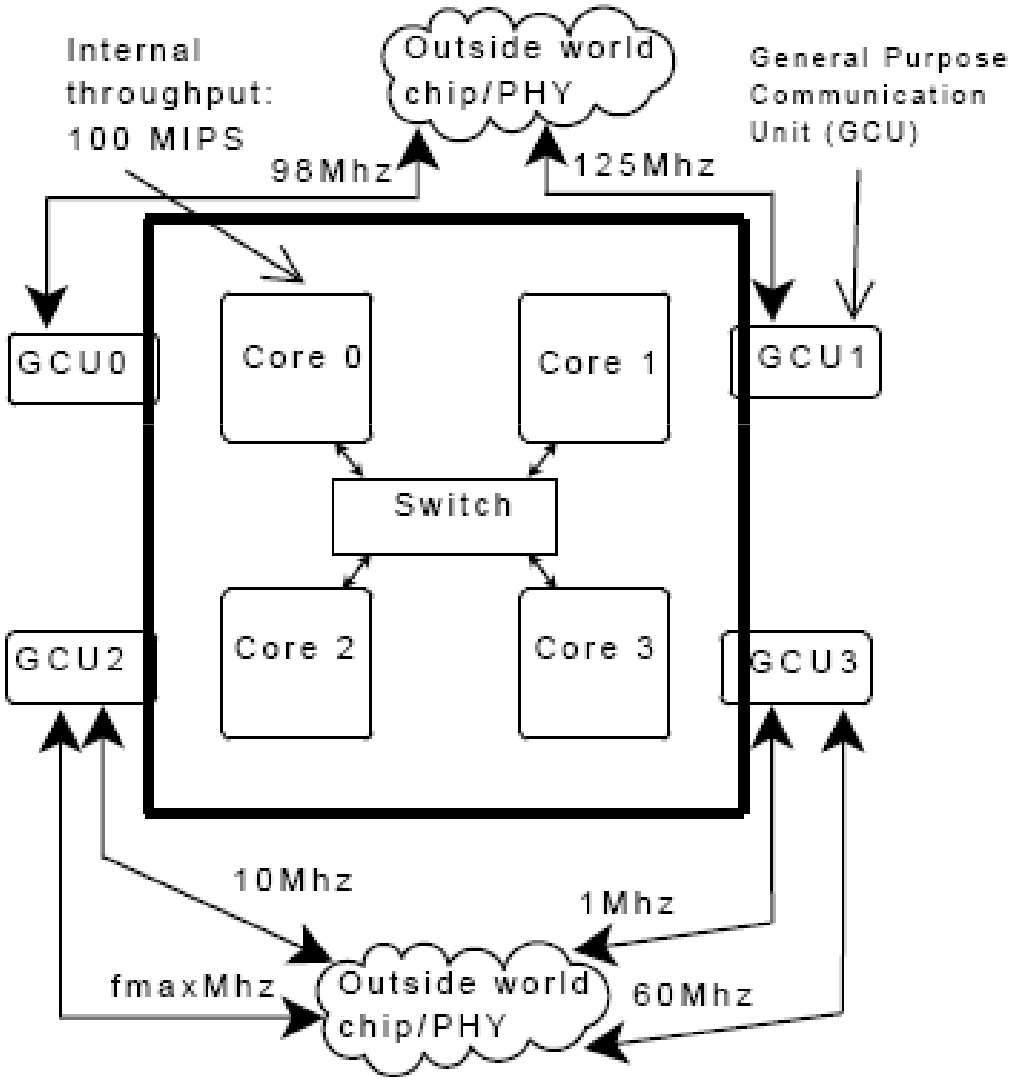
System view before



Our Approach

- Highly programmable General purpose Communication Unit (GCU)
- Tightly coupled with each processing node on the chip
- Will handle all LLC and MAC specific functions:
  - Initially for the benchmark protocols
- The final design will be able to support a new protocol out-of-the-box

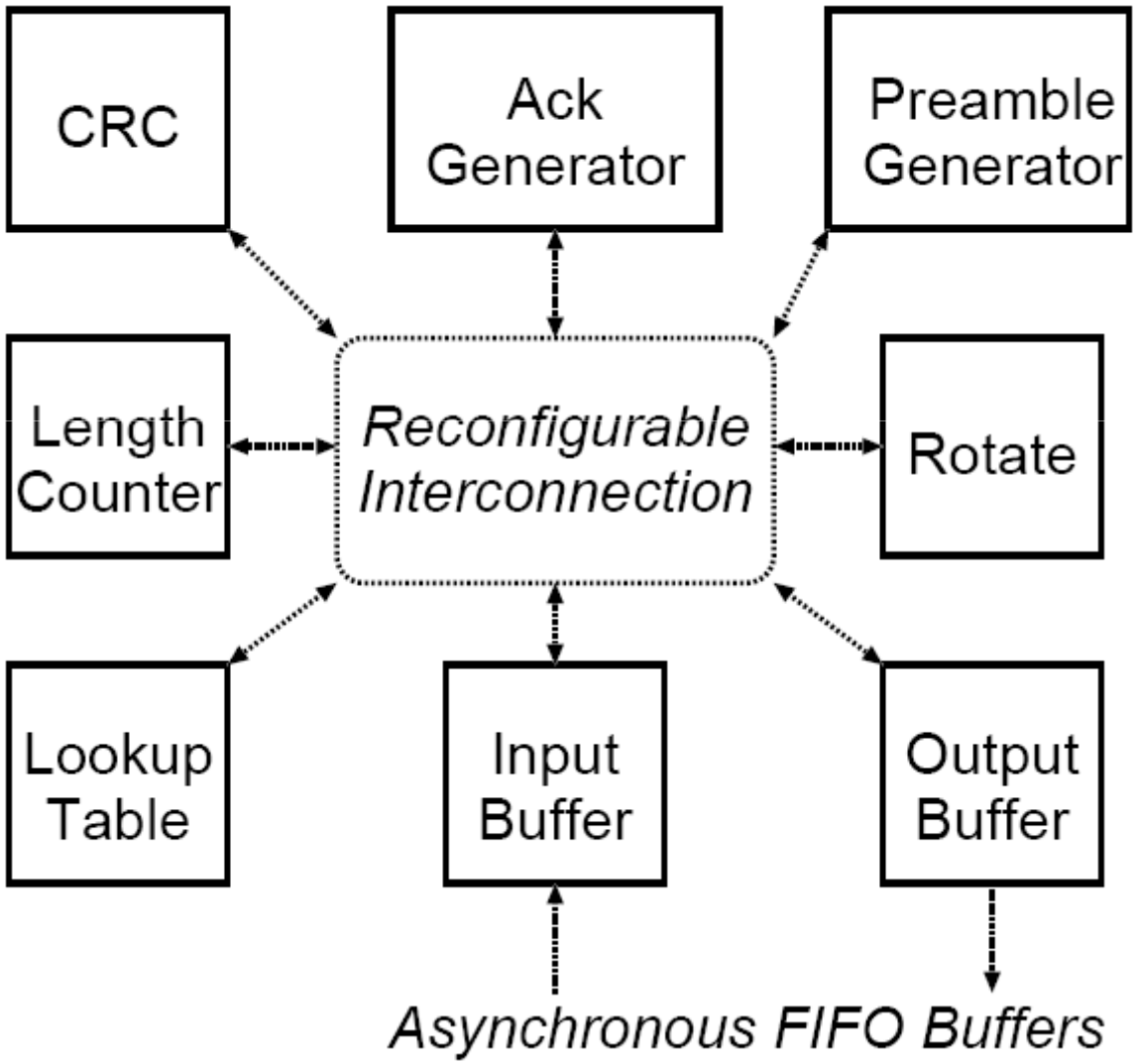
System view after



Functional Building Blocks

- Properties shared amongst many protocols (MII, GMII, SATA, USB, ...etc):
  - Packet creation in output buffer
  - Header generation, preamble, and synch. words
  - Generation of CRC, possibly ECC
  - Serialisation of the packet onto the I/O pins
- Basic operations in parameterisable dedicated hardware blocks (F-Blocks)
- Look-up table/ flash memory for arbitrary functionality

F-Blocks composing the GCU



Asynchronous Composition of F-Blocks

- Dynamically combinable to synthesise the protocol in question
- Inter-block synchronous timing verification not practical:
  - 12 blocks = 1.3×10^{9} possible combinations
- Asynchronous interface to the F-Blocks:
  - Implementing delay sensitive encoding
  - Verification problem is solved
- Worst case delay becomes: $\sum_{i=1}^{n}\frac{1}{f_i}$ versus $n\times\frac{1}{f_{\min}}$

Asynchronous I/O FIFOs

- Conventional FIFOs introduce high latency
  - Also very limited flexibility in throughput
- Asynchronous FIFOs are better suited for such task:
  - Accommodates different I/O rates
  - Able to support very high data rates on both sides: the pin-side and the GCU side
  - Can handle bursty traffic efficiently: Low latency

Conclusions

- A General purpose high speed Communication Unit is an essential building block in a modern embedded system
- Flexibility is a key factor in designing such unit
- A combination of techniques:
  - Fixed and dynamic hardware blocks
  - Asynchronous and synchronous techniques
  - Software configurable hardware

Conclusions (ctd)

- Accommodate a wide range of data rates, and unpredictable and bursty traffic
- Provide space efficiency for the ASIC designer
- Effective software implementation for all desired LLC/MAC controllers
- Flexibility to adopt new protocols:
  - Implemented by software configuration of the GCU

Questions?