Dear Contractor,

We are requesting hereby a quote for the following FPGA design.

Objective:

Create a basic prototype for the Netmory interconnect.

The Netmory interconnect is describbed in the document attached **0.pdf** on page 7. This is only a functional description and a non constraining example of implementation.

In addition, the interconnect can been defined as in the following figure 1:

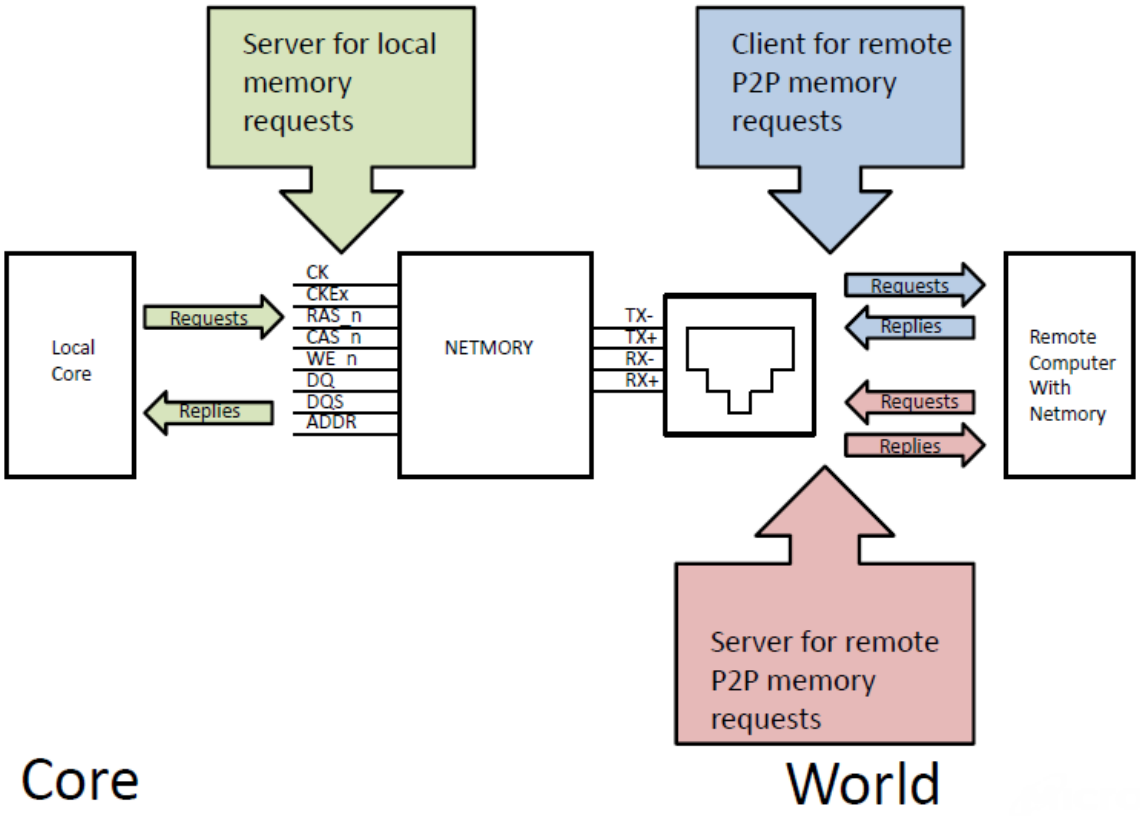


Figure 1

On the core side (left) the interconnect is a server. On the Ethernet side (right) the interconnect is a client and a server.

This implies that there is a new protocole on the Ethernet that enables the transfer of memory resource over the network.

In the figure 2 below there is an example of implementation for that protocole.

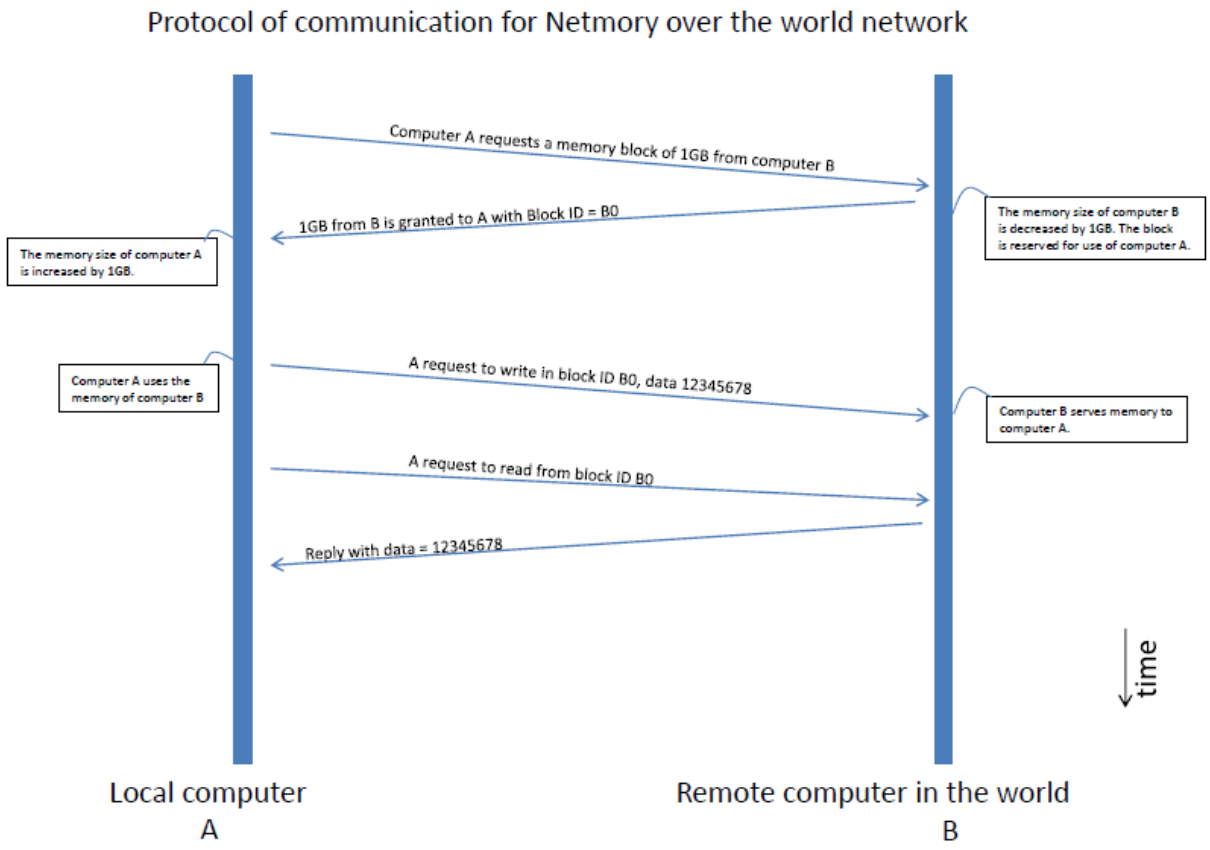
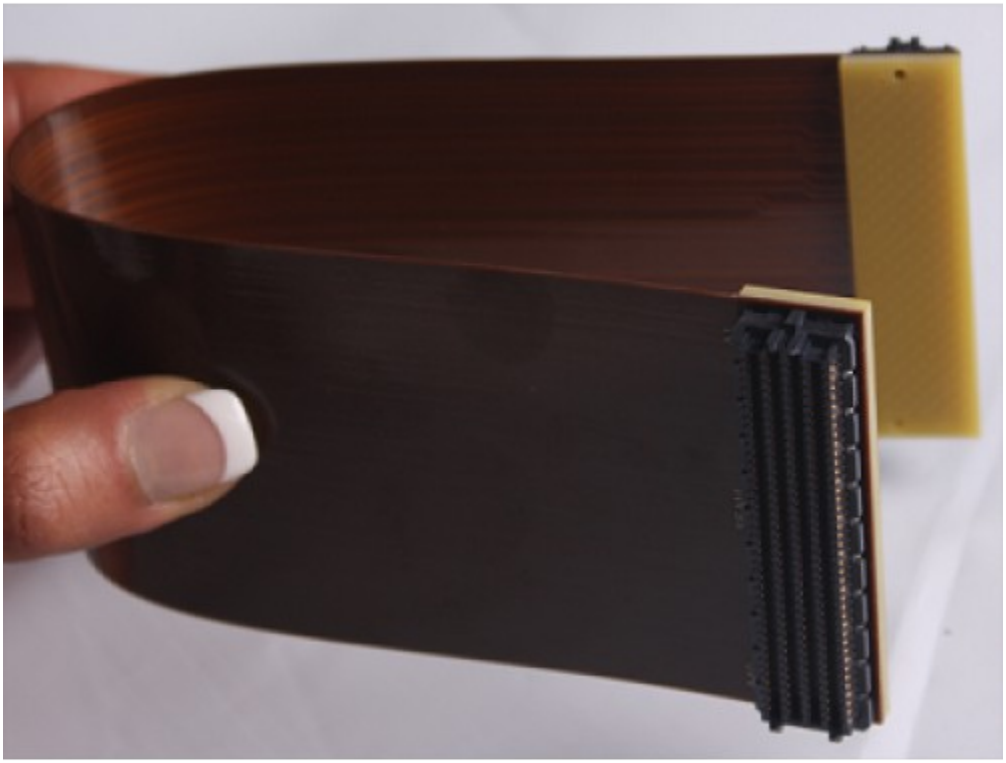


Figure 2

Deliverable:

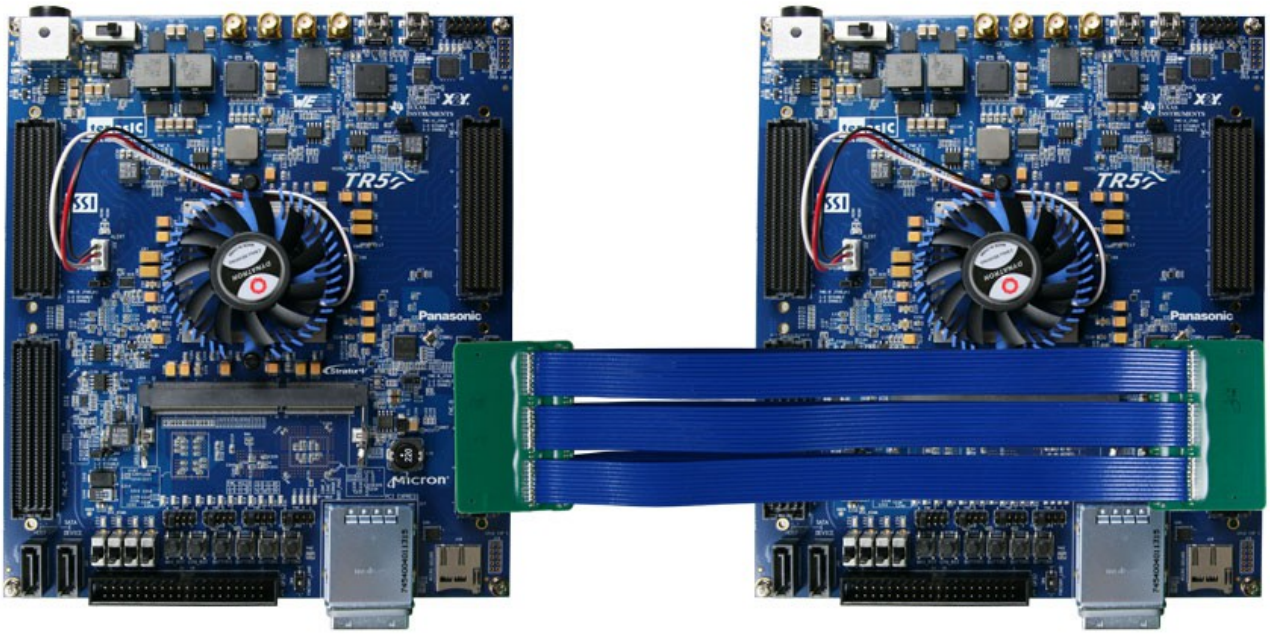
1) Hardware

2 pairs of setups: Pair A and pair B.
Each pair is composed from 2 FPGA boards interconnected via an FMC board to board cable.



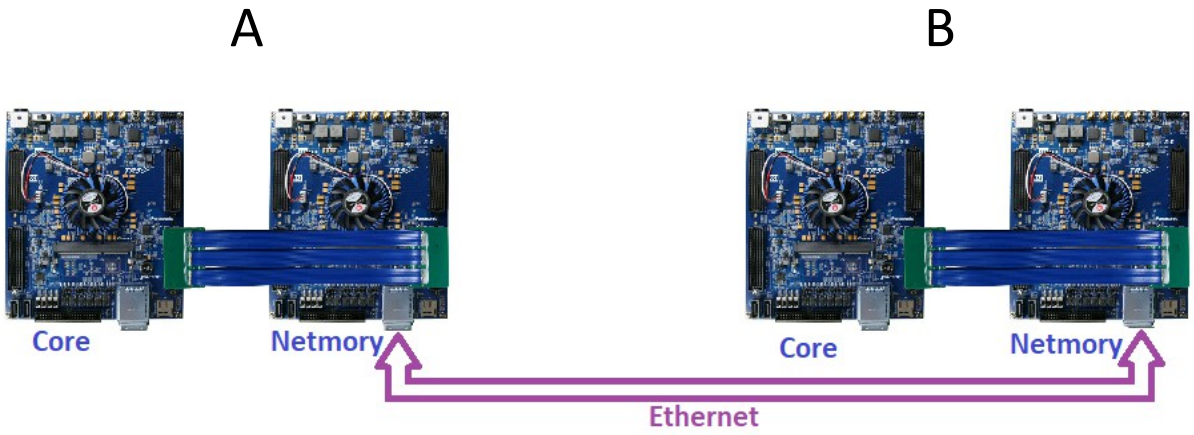
FMC to FMC cable (board to board) shown here as an example

A



Showing 2 FPGA boards with an FMC to FMC board to board connection example.

The 2 setups A and B communicate with an ethernet connection. This can be done directly or via the internet.



Showing an example of a setup of a network with 2 terminals incorporationg the Netmory interconnect

2) Software

hello_world program that display 3 strings :

- hello world form the interconnect immediat memory
- hello world from the interconnect stotage
- hello world from world

The program should use a pointer in the memory where the above strings are stored. The 3 strings are stored in 3 different address ranges corresponding to their physical location.

One address range corresponds to the immediat memory, the second address range to the interconnect storage and the last range for the world network.

We can then differenciate 3 memory ranges on which we can perform read and write operations. This adds up to 6 different transactions.

The program must be stable and should not hang.

3) Demo

Setup the Hardware of section 1) and run hello_world of section 2).

3) Environment

Use Xilinx Vivado and SDK. The program above can run in the first stage boot loader (FSBL). The hardware may be based on Zynq Ultrascale+ or more recent ARM based Xilinx device.

Attached with this document, a guide from Xilinx : ug1165-zynq-embedded-design-tutorial.pdf. In chapter 5 of this guide, there is an example of how another agent can talk to the memory with AXI (Advanced eXtensible Interface).

I was thinking that we can do something similar with 2 Zynq boards. One will be the core and one will be the interconnect. The 2 boards can communicate with the FMC (FPGA Mezzanine Connector).

4) General

deliverable to includes:

All source code and binaries and bit files.

A document and training showing how to generate the binaries and the bit files.

A working stable demo.

The setup must be self contained and be based on the deliverable. Another skilled in the art must be reasonably capable to produce the entire demo without writing additional source code or RTL code.

The delivrable includes source code of underlying software and tools. Such network drivers or utilities.

This contract authorizes the contractor to use the patent for the purpose of performing the scope of work described in this document. Thus this work does not constitute an infringement of the patent. The contractor is prohibited from using/re-using the design without a separate agreement with the customer.

5) Definitions

Customer:

Design:

Contractor:

Patent: