Multiprocessor interconnection based on DMA for FPGA

Introduction

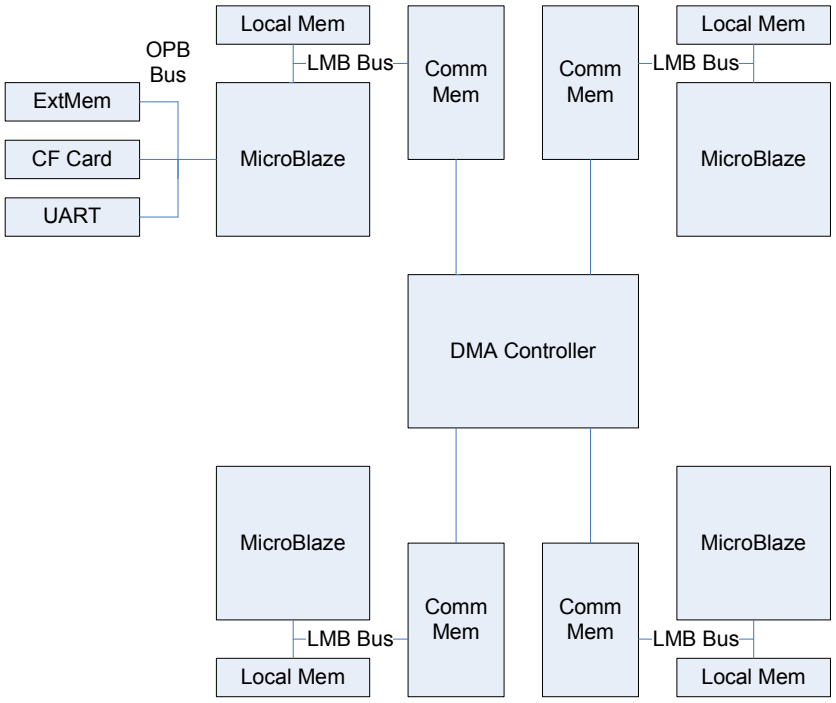
DMA is becoming popular for communication between multiprocessors on SoC or FPGA. This design is a template consisting of four Xilinx microblaze processors and external memory controller connected by a central DMA engine. Compared to fixed bus, FIFO or Dual port memory interconnection, it is supposed to achieve high predictability, scalability, flexibility and reusability with little cost in area and development efforts. It is the second part of my master project and finally some video compression algorithm can be mapped on this template.

The design can be synthesized by Xilinx EDK7.1 and ISE7.1. The simulator is ModelSim 6.1 starter version. The design consists of two parts: a DMA controller library and a testbench. The current testbench is based on JPEG encoder from another project on Opencores. See http://www.opencores.org/projects.cgi/web/mb-jpeg/overview.

I would implement the system by bus, FIFO, dual port memory and DMA controller. Then we can compare and see the advantage.

It can also be used as an IP core or library for other computing intensive applications for FPGA with a little modification of bus interface.

Architecture



Roadmap

1. Setup testbench

1.1 One processor system with external memory and JPEG encoder running *

1.2 Four processors connected by FIFOs *

1.3 Four processors connected by Dual Port memory

1.4 Four processors connected by bus

2. Design one channel DMA controller and replace one FIFO connection

2.1 one channel DMA controller for two-processor system (<- Current)

2.2 one channel DMA controller for four-processor system

3. Design two channel DMA controller with arbitration and replace two FIFO connections

3.1 two channel DMA controller for three-processor system

3.2 two channel DMA controller for four-processor system

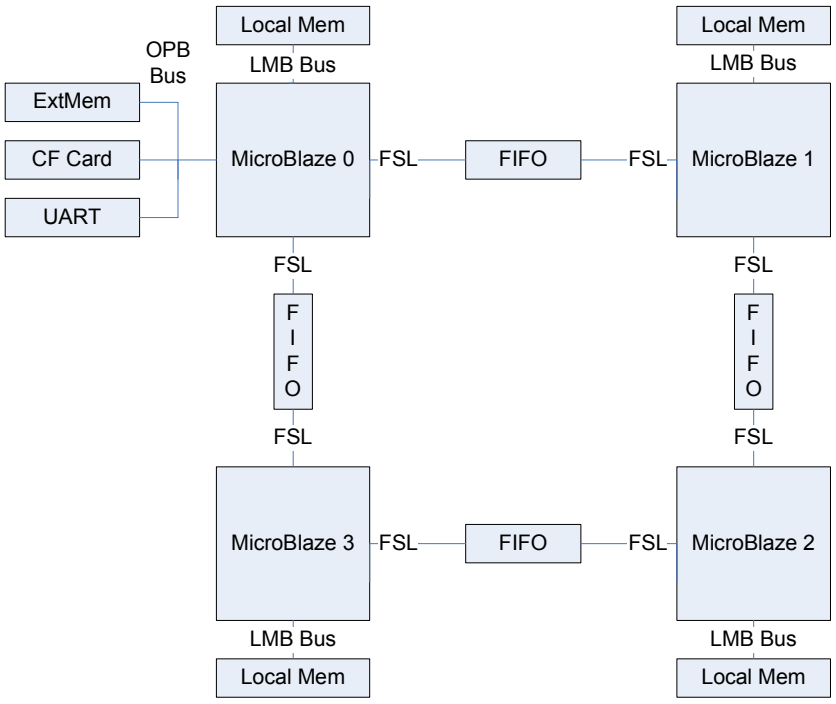
4. Design four channel DMA controller and replace all FIFO connections

5. Map MPEG application onto it and set up testbench for more processors

6. Design a global profiler to profile communication between multi processors
7. Let's see what we can do further... :)

Milestones

1. 2006/11/04, Step 1.2, Setup four-processor testbench connected by FIFO and map JPEG encoder onto it.



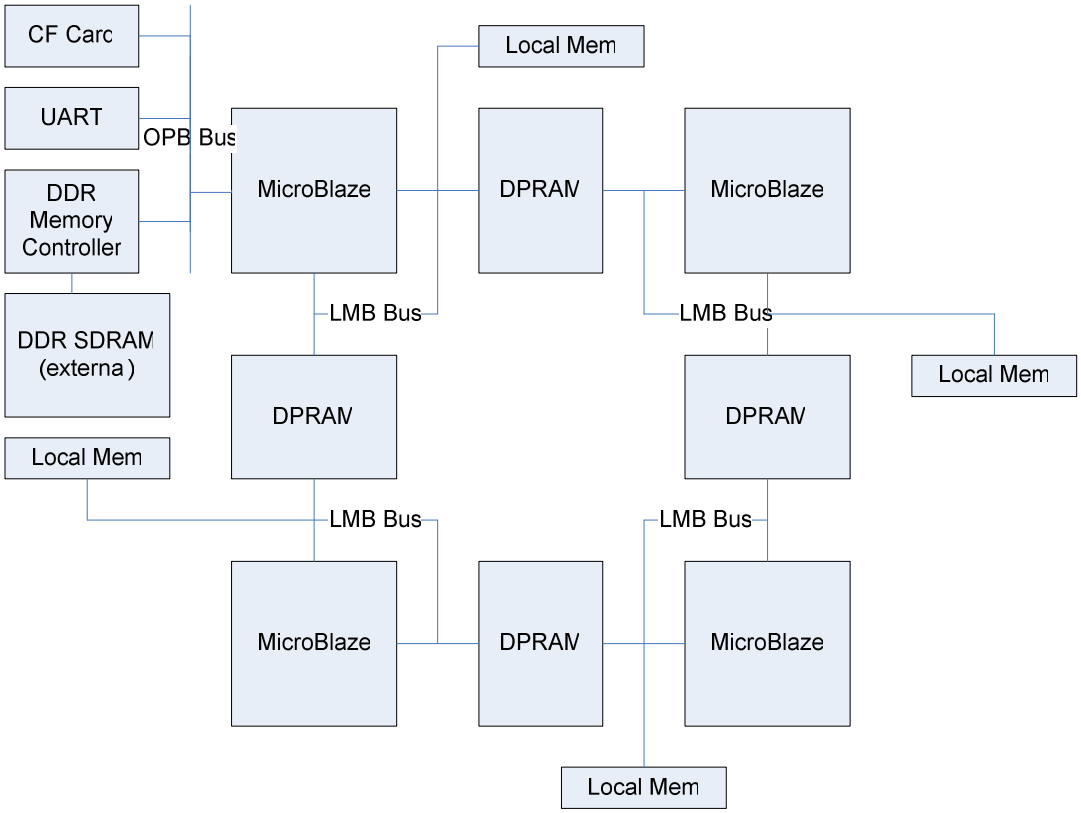
In this implementation, four microblaze processors are connected by fixed FIFO interconnection. After bitstream is downloaded to a Xilinx XUPV2Pro board, processor 0 can read a BMP file, “image01.bmp” from CF card, processor 1 does color conversion, processor 2 does DCT transform and processor 3 does variable-length encoding. After that, encoded data is send back to processor 0 and write back to CF card with filename “image01.jpg”

The component fifo_link, which you can find in hardware description file, MHS file, is simply used to simply connect FSL master and slave. FIFO function is included in FSL bus and I simply use it. There is no FIFO in fifo_link component.

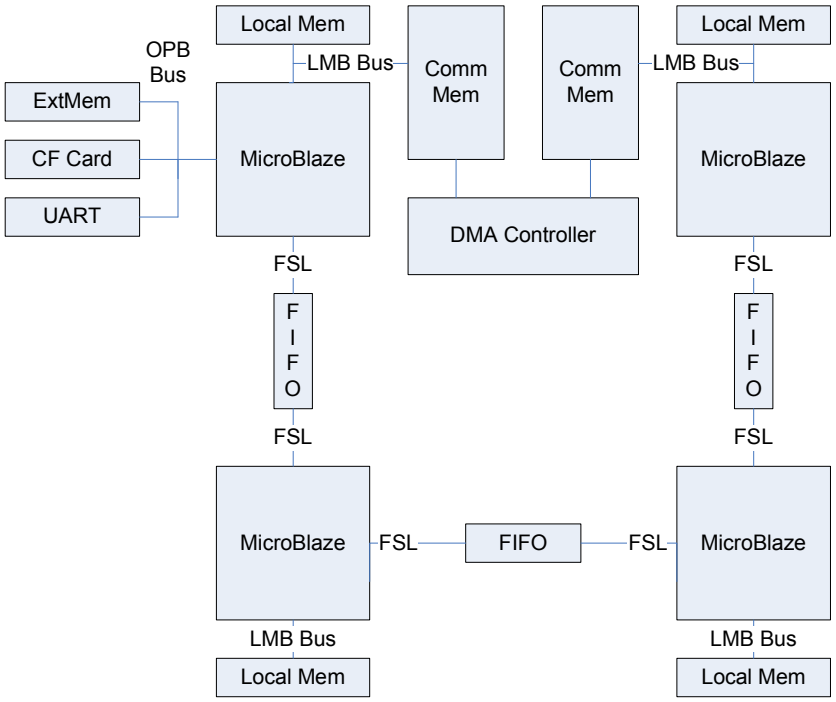
Software need to be modified as well to fit on this architecture. Tasks are distributed between four processors, I/O for processor 0, Color conversion for processor 1, DCT for processor 2 and VLC for processor 3. First, code need to be written in stream mode, which means that processor 0 only talk with processor 1, processor 1 only talk with processor 2 and etc. For message from processor 0 to processor 2, it requires forwarding from processor 1. Second, all communication need to be message oriented instead of RPC. It needs to specify which type of message it is. Third, for processor 0, it needs to wait message from processor 3 and wait processor 1 to accept message. These two events are asynchronous. Some special tricks have to be applied here.

One interesting issue is that I need to set FIFO depth to 128 instead of the default value, 8. Otherwise the system would stop. It looks some kind of deadlock. But I haven't found the exact reason yet.

1.3 Four processors connected by Dual Port memory



2.2 Design a two-port DMA controller and replace one FIFO



Please refer to http://www.opencores.org/projects.cgi/web/mpdma/overview for further information or email to me Sun Wei sunwei388@gmail.com

Reference

1. http://www.opencores.org/projects.cgi/web/mb-jpeg/overview "jpeg codec library based on microblaze", Sun Wei, Joris van Emden, Marcel Lauwerijssen, Cristian Tena
2. http://sourceforge.net/projects/mb-jpeg "embedded JPEG codec library", Sun Wei, Joris van Emden, Marcel Lauwerijssen, Cristian Tena