Simple switched GALS interconnect

Simon Moore, Robert Mullins, George Taylor

Computer Laboratory, University of Cambridge

George.Taylor@cl.cam.ac.uk

As part of an investigation into self-timed micro-controllers a means of sharing a common on-chip memory and addressable I/O system was required. A 1-of-4 encoded interconnect was chosen to avoid top-level timing closure problems and reduce the amount of data dependent power consumption. Arbitration between processors wishing to send requests to the memory system is performed by a lazy token-ring arbiter. The arbiter is optimised for the common case when only one processor is running.

The Springbank test chip contains five 16-bit XAP processors. Of these one mostly uses a 1-of-4 data encoding, two are dual-rail, one is bundled data and the remaining one uses synchronous DFFs. The latter is clocked from an on-chip ring oscillator which can be paused (i.e. one clock cycle is stretched) when a memory access is required.

The overall system is shown in Figure 1. To perform a memory operation one of the processors (P0-P4) first requests a token from the ring arbiter and sends the request (including data if a write) to the memory controller. Due to the use of 1-of-4 encoding a simple OR like function can be used to combine requests from the multiple processors. The memory controller will buffer this request and acknowledge the processor. At this point the processor is free to either release the token or to send further requests. On a read request the memory controller will produce a result which is steered to the appropriate processor. Whilst waiting for this result the processor is free to continue operation if possible and to issue further requests. All reads and writes are performed in the order requested.

The memory controller essentially consists of two full 1-of-4 pipeline buffers with a delay matched binary circuit in the middle. The output buffer is only used for read requests. In addition to address and data the memory controller is informed if the request is a read, in which case the processor number is recorded to later steer the result correctly, or a write, for which the processor number is not required. A further option permits the on-chip SRAM to be used in a balanced way for security against differential power analysis (DPA) attacks. If requested, the two on-chip SRAM blocks are either both read (but only one result is output from the memory controller) or both written one with complementary binary data from the other. This aims to hide the data dependency in the power used to drive the wires between the 1-of-4 to binary conversion logic and the standard SRAM blocks. Inside the memory controller different programmable delays are used to match on-chip SRAM, on-chip I/O or off-chip accesses. Because the SRAMs latch the address and data internally when clocked, the SRAM clock cycle for a write can be performed concurrently with setup of the next address/data.

There are two motivations for pipelining the memory interface even though the processors themselves have minimal internal pipelining (and stall upon memory reads). Firstly the XAP processor normally expects to see separate memories for program and data for which concurrent access may occur. In the Springbank chip the program and data interfaces are combined in the 1-of-4 interface logic attached to each processor. Secondly the intent of permitting multiple processors to execute is to use one processor to inject random memory traffic to further increase the difficulty of DPA. A lazy token ring arbiter is used to avoid spinning a token around in the common case of only one processor being active.

The use of a delay insensitive interconnect with uniform interface to each processor permitted a very rapid assembly of the top-level design which turned out to be critical in meeting the chip submission deadline.
Figure 1: (a) Memory system overview, (b) Memory controller, (c) Lazy token ring arbiter