LOCKHEED MARTIN

Advanced Technology Laboratories
1 Federal Street – A&E 2W
Camden, NJ 08102
(609) 338–4250

Application Notes – Hardware Synthesis Study of WSSPT SVI Interface Encapsulations

Date: July 24, 1995

Author: Greg Buchanan

Contact: Janet E. Wedgwood (janet.e.wedgwood@lmco.com)

1. Purpose

The purpose of the this initial SVI synthesis study was twofold: first, to gain experience in writing synthesizable VHDL code for SVI encapsulations, and second, to gain a better understanding of the hardware overhead introduced by a typical interconnect fabric encapsulation. The lessons learned from the experience of creating optimized synthesizable VHDL code have been documented as coding guidelines in *Standard Virtual Interface Specification, Version 0.5*. The results pertaining to the hardware overhead introduced by SVI encapsulation are the subject of this memo.

2. Interface Description

The hardware synthesis study was based on the design for the WSSPT (Wafer Scale) MCM Interface FPGA. This FPGA is used in the WSSPT system to interface the processing element nodes to the common interconnect fabric. The processing elements (up to 16 on a node) are tied together with a proprietary bus known as IOBUS, and the interconnect fabric, known as PCI, is similar in protocol to the PCI standard. The functional block diagram of the MCM interface FPGA is shown in Figure 1. The MCM Interface was realized in an AT&T ORCA 2C15 FPGA, and utilized approximately 340 of 400 available Programmable Function Units (PFU's).

The SVI implementation of the the WSSPT MCM Interface logic, which amounts to the encapsulation of the PE and interconnect fabric elements, was undertaken as part of a separate verification study in SVI encapsulation. This encapsulation was done in such a way as to demonstrate SVI utilization in conjunction with a COTS interface controller, since in practice a COTS controller will often be used to interface to the standard interconnect fabric – in this case, PCI. A functional block diagram of this implementation is shown in Figure 2. The PCI FIFO function shown in Figure 1 is now included in COTS_PCI. (Note that in the case of the original WSSPT MCM Interface design, a COTS PCI controller is not actually used. For the SVI encapsulation study, the control logic was partitioned out of the PCI interface logic [labelled COTS_PCI in Fig. 2] and treated as a COTS controller for demonstration purposes.)

The logic of the MCM Interface was partitioned for synthesis as shown in the block SVI_BLOCK. The block MCM_ARBIT, which contains the arbiter function for the IOBUS, was left out of the synthesis target in order to constrain the scope of the synthesis experiments to SVI encapsulations of "pure" interface logic. The COTS_PCI block was left out since it was being treated as a COTS part. In order to make equal comparisons of the MCM Interface FPGA with and without SVI encapsulation however, estimates were made for the required FPGA resources required for both of these blocks plus a separate IOBUS FIFO.

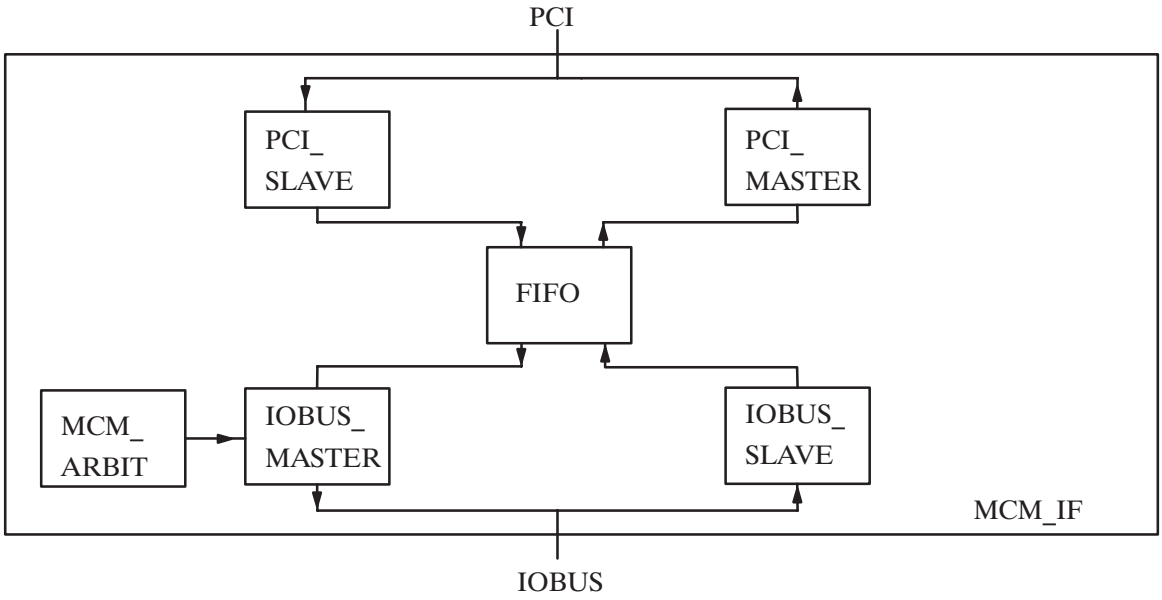

FIG. 1 – WSSPT MCM INTERFACE FPGA

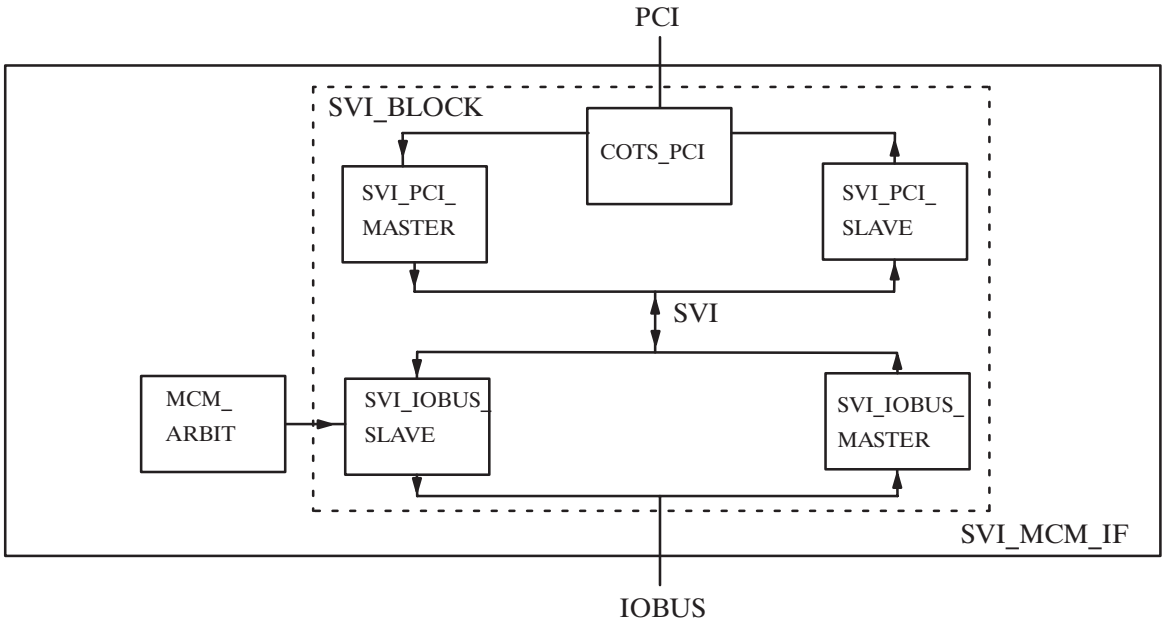

FIG. 2 – SVI MCM INTERFACE FPGA

3. Results

All synthesis runs were performed under the following tool versions and libraries: Synopsys v3.3a, NeoCad v7.0, AT&T ORCA Library v3.1. Total PFU count for the combined blocks SVI_IOBUS_SLAVE, SVI_IOBUS_MASTER, SVI_PCI_SLAVE, and SVI_PCI_MASTER was 388. This represented a utilization of 97% of available PFU's, and this design was fully placed and routed into an ORCA 2C15 FPGA. Greatest packing density was achieved with the standard ORCA cell library (ie. designware and lookup tables libraries not used), with the Pack Logic Blocks option in NeoCad. *No performance data was obtained.*

In order to estimate the total equivalent gates for a direct comparison to the original MCM Interface FPGA, the results from the synthesis runs were modified to reflect the integration of the COTS_PCI and FIFO functions into the interface logic, and the addition of MCM_ARBIT. The

following procedure was used to estimate the equivalent gate count:

1) The original individual interface logic blocks (two masters, two slaves) were synthesized independently
2) The "optimization ratio" was calculated between the synthesis of the original interface blocks and the synthesis of the combined blocks (calculated to be 75%), and all interface logic was scaled down by this amount
3) The WSSPT MCM Interface FIFO (16x73) and arbiter were synthesized and gate counts obtained
4) Total equivalent gates were calculated as follows, with scaling factors applied to interface logic blocks to reflect the integration of COTS_PCI functions (scaling factors are estimates provided by logic designer). Refer to Figure 3.

LOGIC	GATE COUNT
SVI_IO_MASTER	same as WSSPT
SVI_IO_SLAVE	remove (1) 64−bit reg., scale by 0.8
IO_MASTER	same as SVI_SLAVE
SVI_PCI_MASTER	add (1) 64−bit reg., scale by 1.25
SVI_PCI_SLAVE	same as WSSPT
PCI_MASTER	same as SVI_PCI_SLAVE
FIFO's (2x64x4)	scale WSSPT FIFO by 0.44
MCM_ARBIT	same as WSSPT MCM_ARBIT

Total equivalent cells for the SVI version of the MCM Interface FPGA was estimated to be 635 PFU's. This design would not fit into one FPGA (the ORCA 2C26 contains 675 PFU's), but could be realized in two ORCA 2C15 FPGA's.

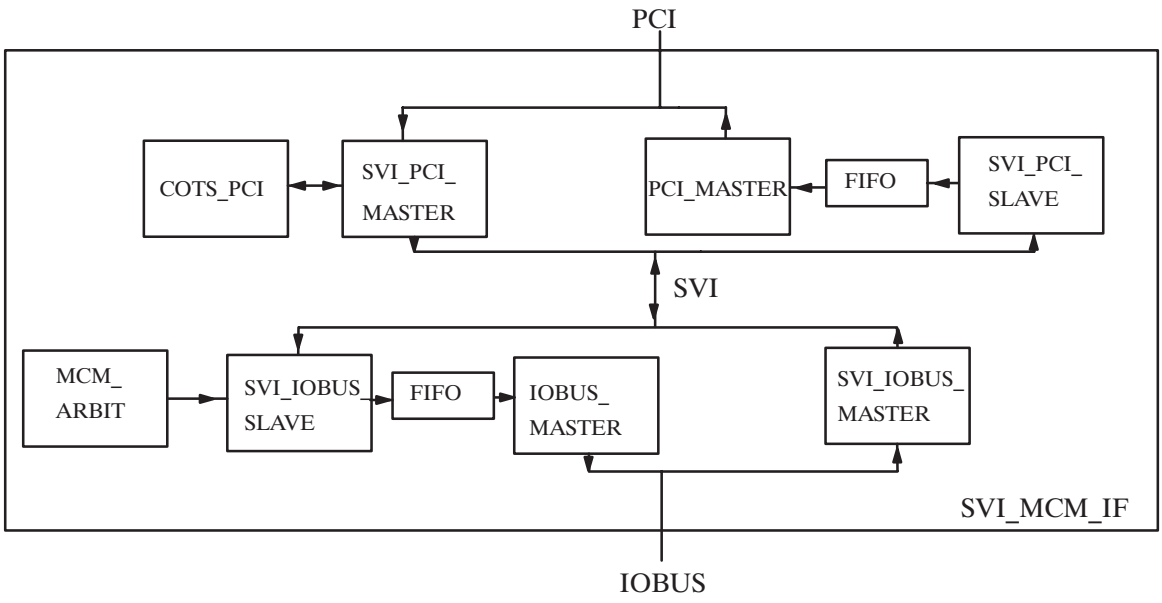

FIG. 3 − LOGIC BLOCKS FOR EQUIVALENT GATE ESTIMATION

4. Conclusion

The amount of hardware overhead introduced by the SVI approach will certainly be dependent upon the specifics of a particular application. From the example of the WSSPT PE encapsulation however, it can generally be said that an SVI PE encapsulation will probably add one additional FPGA worth of logic to a typical MCM or board processor application.