U.S.A
560 Winchester Blvd, Sun Jose, CA 95128
sales: (408) -559-1802 support: 1877-830-1647
Email: sales_us@gidel.com

World wide
48 Harimon, Ein Ayyala, Israel 30825
Tel: +972-4-6391708
Email sales_eu@gidel.com

How to accomplish a 13 Weeks prototyping project in just 1 week, for less then $2K

This application was based on a project carried out at Philips.

The goal of the project was to build custom piece test equipment (system simulator) for a complex system.

System architecture

A PC with PCI card including:

- Physical interfaces to the system such as RS422 – LVTTL converter, OPTO Couplers, etc'.
- System communication protocol based on an FPGA.
- Data acquisition with / without input processing & including transfer to a computer via the PCI.

Software:

- A simple GUI for control and monitoring of the testing process.
- Hardware control and operation in the desired mode.
- Test timing
- Saving data / results on a disk

Typical critical system time line

Board & HW / SW interface specifications.	1 day
Board Design	3 weeks
Board layout	2 weeks
PCB Production	2 weeks
Assembly	1 week
Board debug	1 week
FPGA debug (with simple memory controller)	3 weeks
integration	1 week
Total	13 weeks

- The time used for each development stage depends on the chosen system architecture.

U.S.A
560 Winchester Blvd, Sun Jose, CA 95128
sales: (408) -559-1802 support: 1877-830-1647
Email: sales_us@gidel.com

World wide
48 Harimon, Ein Ayyala, Israel 30825
Tel: +972-4-6391708
Email sales_eu@gidel.com

Design challenges and risk

The major challenge in the design was building data storage and synchronization mechanism to be used in the data transfer to the PCI. This mechanism had to be deep enough to hold the incoming data while the PCI was used for other tasks. The conventional solution is using two memory blocks. One block stores the input data, while at the same time, the previous information is being copied to the PC. At a specific moment, the two bank switch rolls. This kind of solution requires perfect timing between the input of the data, its discharge to the PC, and switching the Banks function.
There are two basic implementations, the first is using SRAM. This way, the size of the banks is smaller and fast response time to the PC has to be verified. This may be accomplished, for example, by writing a code in Kernel mode. The second application is using DRAM memory. This requires designing a controller, a design which complexity may exceeds the rest of the system.

GIDEL's Solution.

System architecture:

- Using a turnkey board from the *PROC* series with a PCI interface to carry out the logic and data storage.
- Using a *PROCProto* daughter-card for rapid implementation of a system interface.
- Using the on board DRAM memory and the PROC*MultiPort* (IP) for simple and effective data storage and transfer to the PC.
- Automatic creation of a driver and equivalent code in FPGA for complete integration between hardware and software.
- Automatic instantiation of the desired memory controller instance & connecting it to user's acquisition system on one hand and the software on the other hand.
- Writing a FPGA system communication protocol.
- Making the GUI and a software system timing mechanism.

The PROC boards include dynamic memories and FPGA devices. PROCStar II boards include ALTERA STRATIX II devices with an option of up to 2.5GB memory or PROCSpark II boards that are extremely cost-effective and include an optimal development kit that makes them highly suitable for large quantity systems.

PROCMultiPort is an innovative memory controller utilizing a user-friendly interface designed to meet specific application requirements. It enables simultaneous access to the memory from a number of ports. Each port may be assigned a different clock and bus width.

In the following case, the SDRAM memory utilization was as a cyclic buffer, resembling a giant FIFO with two ports. One port was used to input from sizable bus width and system clock timing. The other port was used for outputting data towards the PCI bus according to the timing and local bus width.
Further applications may require additional port designations which will have access to data while it is transferred from the system on the PC.
The described usage decreased the input delay due to the fact it was operated through a virtual giant FIFO and not by a double buffer mechanism. Furthermore, the size of the buffer enabled a simple usage of the user's software. The system connections was carried out by interface components (RS422 etc'..) which were installed on the *PROCProto* daughter card using the Wire Wrap.
The HW / SW interface has been accomplished automatically by the *PROCWizard*. The *PROCWizard* software produced an application driver, and the interface and deciphering units for each FPGA device.

U.S.A
560 Winchester Blvd, Sun Jose, CA 95128
sales: (408) -559-1802 support: 1877-830-1647
Email: sales_us@gidel.com

World wide
48 Harimon, Ein Ayyala, Israel 30825
Tel: +972-4-6391708
Email sales_eu@gidel.com

System time line

• HW / SW interface specifications • Application driver production • Preparation of FPGA pin list • Writing the top level infrastructure • Building a memory wrapper including Two port for different bus components.	30 minutes
assembling system interfaces	Day 1
FPGA content design	Days 2, 3
Preparing the GUI and operation	Day 4
Integration	Day 5