Advanced **L**ogic **S**ynthesis for **E**lectronics
http://www.alse-fr.com

April 2009, ver 2.1c

ALSE's **GEDEK** Gigabit Ethernet Data Exchange Kit

Introduction

Preamble

We (A.L.S.E.) have designed this **Gigabit Data Exchange Kit** to help our customers implement Data Transfers between an FPGA design and one or several PCs. Our “both ends” solution is strikingly **simple to use**, **extremely compact** and allows **unbeatable transfer speeds** !

Since we wrote (and own) the entire code of this Intellectual Property, we master its quality, we can port it to any type of FPGA or ASIC, and we can deliver the source code when this is mandatory.

This Kit has been carefully crafted, optimized, and it has already been used in many designs. We created and packaged GEDEK to make your life as simple as possible, while allowing to achieve ultimate performance when necessary. The interfaces are very simple so that integrating GEDEK in a design is a breeze and does not require any Ethernet skills.

In many applications, the “standard, off-the-shelf GEDEK” (with a set of standard options available), as described briefly here, can be used as is. Just keep in mind that, even if your application seems to require some different functions, interface(s) or protocols, this Kit will very likely be a perfect fit after proper customization. In this case, the customer defines the exact needs of his application and let us customize our Kit accordingly.

Note that, beyond customization, we can deliver this Kit under different schemes : from Netlist-only-single-FPGA family (cheapest) to full-RTL (most expensive).

Note also that, despite the name, GEDEK exists in three versions :**100 Mbits, 1Gb,** and dual speed **100/1000**. Obviously, the data rate achievable in 100 Mbits/s is ~1/10th Gigabit values i.e. around 10 Mbytes/s.

Last but not least, we have developed a number of low cost **Demo Kits** for Altera and Xilinx platforms (contact ALSE for Actel and Lattice), so you can actually try GEDEK before purchasing. These Kits are also great to develop FPGA communication and PC applications before having the custom board finalized.

Principle

This Kit is designed to implement very-high speed data exchanges between a hardware system (likely an FPGA-based application) and a host computer (PC under Windows or Linux Operating System for example) using a pervasive Interface: the Gigabit (or 100M) **Ethernet**.
Ethernet links are both cheap, robust, and extremely common. Most if not all laptops and desktop computers come natively with an Ethernet interface. Connectors and cables are found just everywhere and a lot of semiconductors vendors offer low-cost hardware interface chips (“Phys”). The physical link can easily reach over 50 meters without extra hardware, and extending it further is easy using a standard (and cheap) Ethernet switch.

As soon as it is necessary to move data between a hardware platform (FPGA) and a computer, GEDEK can be used.

GEDEK = **Data Rate Performance** + **Compactness** + **Easy Integration** + Simple **Win / *nix API**.

It is also possible to implement this communication system *between two FPGAs*. If you are interested by this variant, please contact ALSE.

Note: this Kit is not designed for use in Wide Area Networks, but only in the context of **short** and **local** links (preferably but not necessarily in a point-to-point configuration).

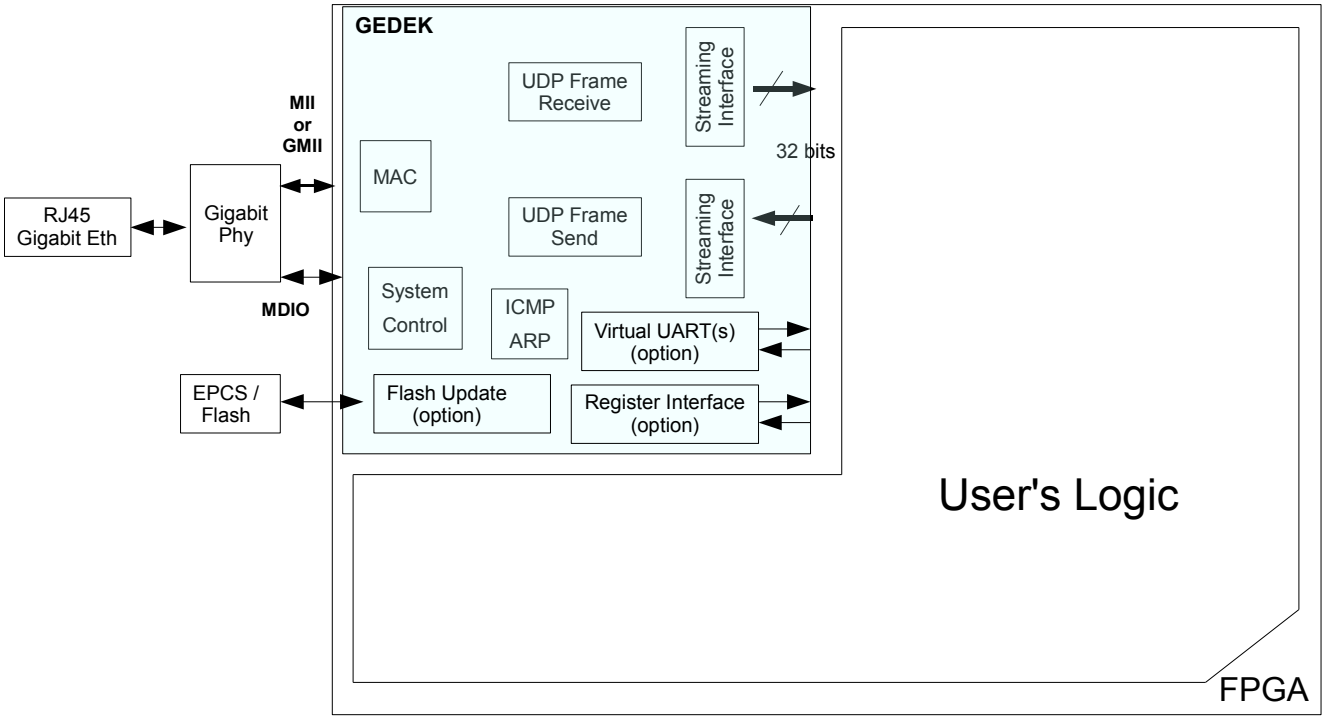
Typical application examples are :

- Computer peripherals, like industrial printers
- Video streaming and camera links
- Array of FPGA boards linked through an Ethernet network
- High Speed Data Acquisition,Transfer & Pattern Generation
- Remote Data Collection,
- Video Streaming,
- Multiple Virtual UART(s) over Ethernet
- Remote FPGA reconfiguration, remote serial or parallel Flash update, etc...

To implement the complete link, you need :

- An FPGA board
- A Gigabit or 100 Mbits (or dual speed) Ethernet PHY device connected to the FPGA. Note that ready-to-use Gigabit Ethernet extension boards exist.
- The ALSE GEDEK Demo Kit
- A host computer with a Gigabit or 100 Mbits Ethernet connection running under Linux or Windows. Note again that using GEDEK for inter-board communication (FPGA to FPGA) is perfectly possible, please contact ALSE.

Block Diagram



This is a simplified view of the FPGA side.

The RJ45 Ethernet connector is attached (through a transformer not represented here) to a physical interface device aka “PHY”. Several such devices exist, among which NS DP83865, Marvell 88e30xx, 88e1111, Sis196, Vitesse VSC86xx, etc...

The PHY device is connected to the FPGA through industry-standard interfaces :

- for Data interface, our GEDEK kit supports both **MII** (Media Independent Interface) and **GMII** (Gigabit Media Independent Interface). Other standards (SGMII, RGMII, SFP...) are available upon request, contact ALSE.
- For Control (access to the PHY's control and status registers), we implement and support the ubiquitous **MIIM** (aka **MDIO**) interface.

Internal Resources

The GEDEK solution is **very compact**.
On a Cyclone II target, the typical figures for Internal resource Usage are :
- **Logic Elements*** : typically around **2,000*** !
- Internal **Memory*** used : typically **8 to 10 x M4k** blocks (6 to 8 x M9k).
- Internal Multipliers : **none**, External Memory used : **none**

* : actual numbers may vary with device family & GEDEK options retained.

Zero CPU !

Note that GEDEK is based on a proprietary “IP stack” implemented entirely in hardware and does therefore not require any CPU. This leads to a gain in internal resources and to the ultimate performances that are not achievable with a CPU and a software stack.
Even if a CPU is present, GEDEK can be extremely useful in removing a heavy burden from the CPU, while leaving the processor application software taking care of TCP/IP transfers at the same time !

Data Exchange

The communication link relies on exchanging UDP/IP Frames containing the various kinds of user's Data (payload), secured by CRC.

For performance reasons, no further data integrity has been added in the payload due to the following assumptions about the context of use :

- The link will necessarily be a **local** link, if not point-to-point.
- The IP frame CRC is sufficient to guarantee the payload integrity in the above context.
- An extra payload CRC or checksum would require a significant amount of processing power on the PC side without any added value.
- There will not be out-of-order issues in the above context.
- Lost (or incorrect) frames can be handled by our protocol (option). In this case, the receiver side checks the frames received, detects missing or incorrect frames and issues a “re-send Frame X” command to the emitter to ask for a re-transmission of the wrong or missing frame. However, this scheme may require important storage memory resources in the FPGA to deal with the PC latency.
  In the absence of data corruption, the protocol takes advantage of all the hardware bandwidth (there is no positive acknowledge).

Indeed, this “default” protocol can be modified according to the customer's needs. We have for example used this Kit in the context of building a video flow for use by a video player (VLC), as in our demo kits.

Virtual UART(s)

An option is available to add one or several “*Virtual UARTs*” ! This allows to transparently establish streams of bytes interspersed with the main data flow. In practice, characters or strings are sent over dedicated Ethernet frames and dispatched / aggregated automatically.
This is ideal to replace old serial links like RS232, i²C, CAN, PS/2, SMBus...

Register Interface

An Optional **Read/Write Register Interface** is available. This is an excellent solution to enhance the controllability and the observability of the application.

Flash Update

Options are available to *Read*, *Erase*, *Program* and *Verify* part or all of an external **Flash Memory** (Serial or Parallel Flash) through Ethernet.
A ready-to-use Remote Flash Update utility is provided (with the source code) for the PC. This feature coexists with the other features (Data link, UARTs...). The FPGA configuration bitstream can be updated this way, but updating operating parameters, Firmware or ROM contents remotely is just as easy.

Performance

The GEDEK kit contains all the necessary logic to handle the Ethernet communication *in hardware*. Our (hardware) implementation ensures that frames can be processed and built *faster* than the physical link permits (even with Gigabit peripherals), thus guaranteeing absolute maximum performance. This is typically impossible with a software (processor-based) TCP/IP stack.

As a consequence, a board-to-board link with GEDEK on both ends can reach the maximum theoretical speed (~114 Mega Bytes per second). Even in the case of simultaneous Transmit and Receive (full duplex) ! This is also true for GEDEK linked to a "perfect" host.

When connected to a host computer, the effective throughput will be limited by the host capability to receive or produce Ethernet frames and accompanying data, and no general rule can be given.

By our experience, and for what is worth :

- For a given computer, Linux is usually faster than Windows (but this may not always be true).
- For a given (Computer H/W + O.S) combination, the physical interface and (even to a greater extent) the *software driver* can make a difference.
- If heavy data processing or storage is required on a continuous flow of data, then this may quickly become the bottleneck (like when a "standard" single disk is used for storage. Very high performance H/W is required to sustain the extremely high transfer rates achieved through GEDEK.
- We have measured actual throughput above 80 MB/s with a "standard" PC under Linux implementing a "processing" that merely checked the frames coming from the FPGA.
- Transfer rates in the 20 to 40 MB/s range are very easy to reach between GEDEK and a PC.

It is impossible to be more specific or to guarantee a given data rate under all possible conditions when a PC or workstation is involved. Fortunately, we deliver a sample design which can help assessing precisely and with few efforts the achievable data rate in given conditions. This design is helpful when optimizing the PC or Workstation setup.

Deliverables

The kit consists of several parts :

- **HDL IP**: the actual contents and format may vary according to the type of license purchased, from Netlist-only to Full-RTL.
  Note that all our code is written in VHDL, but this is rather transparent for customers purchasing the netlist license.
- **Host API**: always in source code form, this API contains simple-to-use functions implementing the actual data transfers. This API is available both under Windows and Linux.
- **Reference Design**: to help the customer in setting up and testing his complete system, we deliver a ready-to-use reference project that :
  - Generates frames inside the FPGA with a controlled payload & speed
  - Acknowledges these frames on the PC / API side.
  - Generate frames on the PC side
  - Acknowledges these frames inside the FPGA
  - Displays the actual transmission rates.

Ready to Use Hardware

An FPGA board with a SantaCruz connector or adapter can receive a DBGIG1 module to support Gigabit Ethernet (Demo or Reference Design). This enables "old" FPGA boards (like Cyclone 1C20 or 2C35...) to be used for evaluating this IP in Gigabit Ethernet mode.
Many commercial FPGA boards can be used for demos, contact ALSE.

Questions & Answers

What FPGA can be used ?

Practically any FPGA can take advantage our GEDEK kit. The presence of internal memory is necessary (so most CPLDs and MaxII devices for example can not be used). Relatively "old" FPGAs are perfectly suitable.

Is the Kit complex to integrate and use ?

No, and we can even make it easier for you : we can generate a sample project specifically for your board (commercial or custom), we can test and validate the Ethernet features of your custom board, etc...

What about the PC side ?

We made it simple for you on this end too. Our API (available for Linux or for Windows) is very simple to use and modify to suit any kind of need and development platform. The source code is always delivered.

Can MAC-Phy chips be used ? (DM9000A, LAN91C111, CS8900...)

No. These chips implement the MAC layer and a PC-type bus interconnect. Transfer performance is very limited through these chips and Gigabit is not supported. Our Kit requires just a PHY device, the MAC is included inside GEDEK.

Can SFP Modules be used (and Fiber links) ?

Yes ! This option is available, contact ALSE. Not all FPGAs will support this high-speed serial link though.

Can Fast (100 Mbits) Ethernet be used instead of Gigabit ?

Absolutely ! We have 100 only, 1000 only and 100+1000 versions available.

Can ALSE help me testing my new Gigabit FPGA board ?

Yes ! We help you test and validate your board with respect to Ethernet interface. We have developed many internal tools for this purpose.

Conclusion

ALSE's GEDEK Kit is probably the simplest, the most compact, the most reliable and the most efficient way to exchange a lot of data at a fast pace between an FPGA and a computer. No processor, no software stack no Operating System and no embedded software are required on the FPGA side. *With prices starting below 5k €uros, it is very likely the cheapest solution too !*

When a board with the proper PHY attached is available (see previous section), our Reference Design can be operational in just *a few hours*.
This has been verified on many FPGA platforms, and by many customers.

Both the hardware and the PC-side programming techniques are based on existing standards, and they are both stable and easy to implement under different Operating Systems. No special hardware or drivers are required (though a poor quality driver may impact negatively the PC performance).

Other high speed links are much more complex, expensive, and proprietary whereas GEDEK relies on multi-source providers (for the Phy).

Last but not least, GEDEK is a very "alive" product: we keep adding new exciting features and options. And if the "standard" kit does not match your application exactly, we encourage you to let us know your exact needs (your "dream system"): we can very quickly adapt GEDEK to make it fit perfectly. We can also help you select the physical interface chip, and we can share our experience with you.

Do not hesitate to contact us:

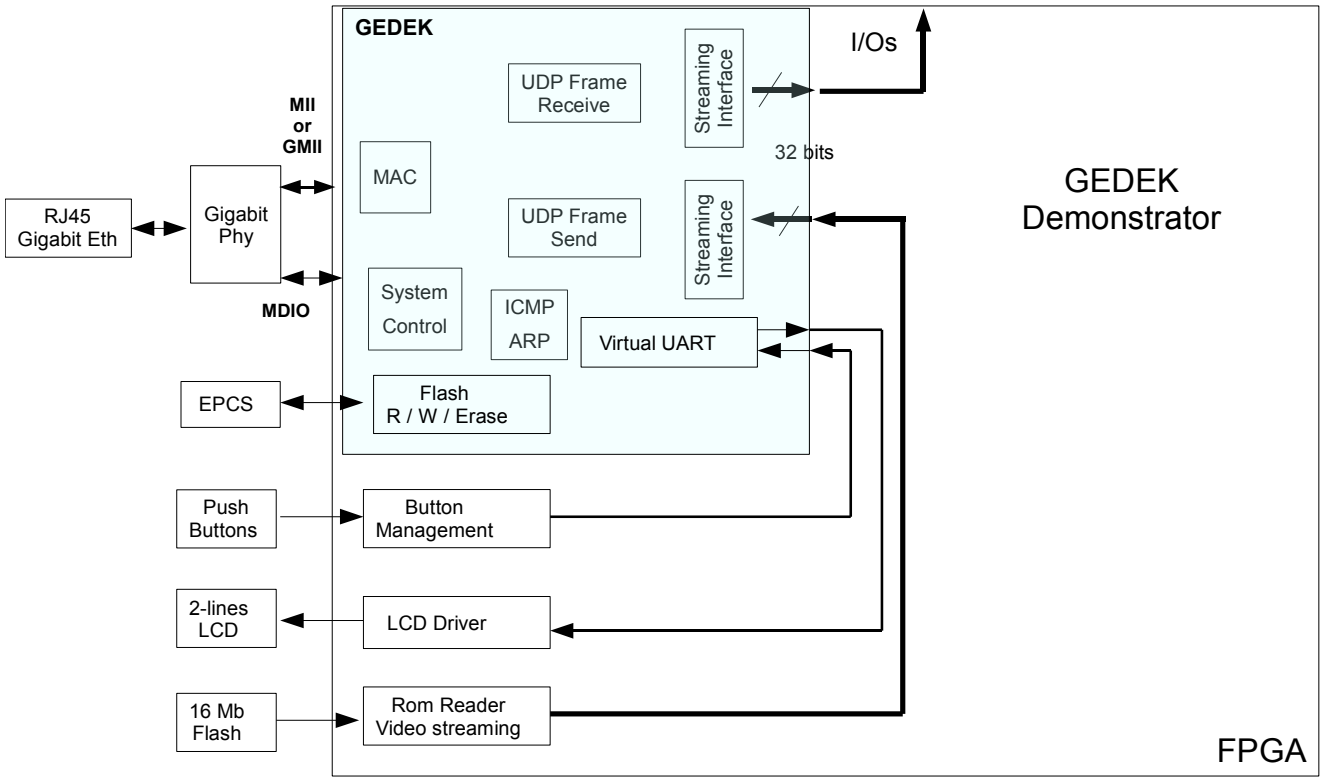
Bertrand CUZEAU
Technical Manager A.L.S.E.
Telephone: +33 1 4279 5138 – E-mail: info@alse-fr.com

GEDEK Demonstrators

Versions are available for : Altera CycloneII board, Altera NEEK kit, DBM module, StratixIII GX, Cyclone III DSP, Spartan3E, etc.. contact ALSE.

To help customers evaluate our GEDEK Intellectual Property and test our Technology, we have created several Demo Kits for different vendors and FPGA Boards. This Demonstrator implements the GEDEK core with two options : virtual UART, and Flash Programming.

Demo Kit Architecture :



Principles:

1. **Push-Buttons**.
   When pressed, they generate an ASCII message routed to the Virtual UART input.
2. **LCD driver**.
   A physical driver (available on ALSE's Website for free) is simply connected to the Virtual UART output.
3. **EPCS**. This option is transparent (integrated in GEDEK). Allows a proper PC application to Erase, Read, Write and Compare the contents of the EPCS (won't work if the design is booted from the Parallel Flash Loader).
4. **Flash – Video Streaming**
   This is a very simple block reading the Flash (compressed video) and sending the **video stream** to the GEDEK streaming port.
5. PC-side : **Virtual Terminal**

   A PC utility (Virtual Terminal) emulates a character terminal hooked to the Virtual UART.
   - Strings can be sent and will transit through the Virtual UART block, and will then appear on the 2-Lines LCD. The LCD is refreshed at (programmable) periodic intervals.
   - The Virtual terminal will display strings generated inside the FPGA (buttons)

There is no CPU, no RTOS, no S/W Stack : everything can happen concurrently.
Once GEDEK is installed in the FPGA, everything (including updating the Flash memory) can be done remotely through the Ethernet connection.