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

ALSE's GEDEK Gigabit Ethernet Data Exchange Kit

Introduction

Preamble – why ALSE invented this concept

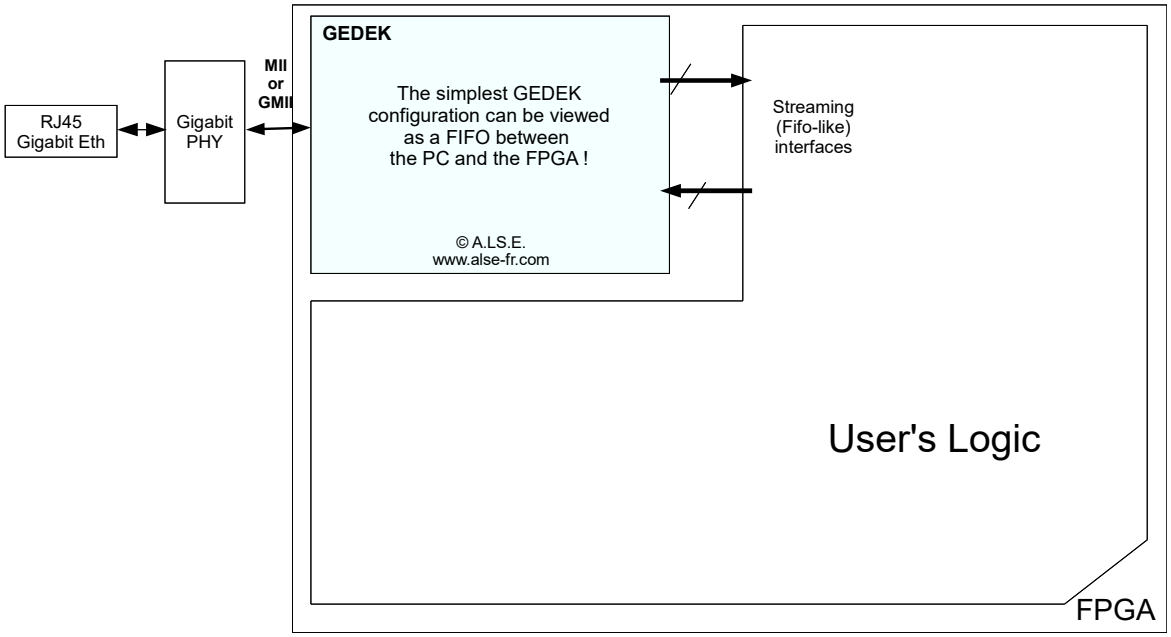
Back in 2007, we (A.L.S.E.) invented the concept and started the design of this **Gigabit Data Exchange Kit** to help customers implement **Data Transfers** between an **FPGA** and one or several **PCs** through **Ethernet**. Our "both ends" sophisticated solution is strikingly **simple to use**, **extremely compact** and allows **unbeatable transfer speeds with no effort!**

After all these years of growing success, we only start seeing the competition trying to catch up, but our experience and continued development keeps our solution ahead and unequaled.
Furthermore***, GEDEK is now available on 10G Ethernet ! (version now named 10GEDEK)***

Let us see the reasons behind this success.

Concept

We'll see later that GEDEK has now a lot of useful options, but in it's **simplest form**, one can view **GEDEK** as a kind of ***Magical Ethernet FIFO (Streaming Ports):***



Once the FPGA is programmed with GEDEK in it, you can **plug** it to your network **and** simply **PING** it ! *It is now ready to exchange data with your PC*. No programming or extra code is needed.
Just push data into the GEDEK input streaming port and they will end up in the PC. And data sent by the PC will show up on GEDEK's main streaming output port.

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 integrating GEDEK in a design is a breeze and does not require any Ethernet skills. This is why building a system to exchange data at very high speed between a PC and an FPGA usually takes *only a few hours* using GEDEK. The streaming protocol can be as simple as Data + DavaValid, with optional refinements when useful.

On the **PC-side**, we deliver a very simple API with functions to send and to receive a block of data (of any size). Again, no Ethernet programming skills are required! (Note however that experienced programmers familiar with Ethernet may use their own API if they prefer).

Since we designed (and own) the entire code of this Intellectual Property (GEDEK is 100% ALSE Technology), we stand behind its high quality, extreme performance and reliability. Our code has been ported to all kinds of FPGAs (or ASIC), and we can even deliver the source code when this is required. This IP has been carefully crafted, and is extremely optimized. It has already been used in a lot of designs & applications since 2007 ! We know that much more than 100,000 GEDEKs are at work everyday throughout the world, in all kinds of applications.

In most situations, GEDEK can be used with practically no control nor configuration ("**Plug and Ping mode**"). But GEDEK can also be used in "**Expert mode**" (providing full control over many functions included in the GEDEK core) as detailed in the IP documentation.

Licensing: Our IPs are available under different **licensing** schemes: from "Single Project" (cheapest) to full-RTL source code (the most expensive license, indeed).

Communication Speed: note also that, in spite of the "G" in the name, GEDEK exists in three versions : **100 Mbits** (aka **FAST** Ethernet), **1Gb** (aka **Gigabit** Ethernet)**, dual speed 100/1000** and even now **10G Ethernet !** Note: **10 Mbits** (obsolete format) is still supported for special applications. Obviously, the data rate achievable in 100 Mbits/s is ~1/10th the Gigabit Ethernet throughput. A 10Gbits version also exists.

Options: along the years, we have developed a lot of complementary features and options that add value to the solution and extend the usefulness of the core.

Demo Kits: Since it takes a few hours to port GEDEK to any FPGA and Board, we have developed a large number of **Demos** for Altera, Xilinx, Lattice and Actel/Microsemi, so you can actually see GEDEK in action before purchasing. These Kits are also great to develop the FPGA application and the PC software before having the custom board finalized.

Principle

The GEDEK IP is designed to connect a hardware system (an FPGA-based application) to Ethernet and allow ultimately efficient and high speed bi-directional communication.

The **other end** can be one or several *Computers* (running under Windows or Linux or another Operating System), but it can also be other GEDEK(s): you can see here a practical example of GEDEK used for FPGA-to-FPGA communication sold in an end-user equipment.

The **link** can be just a simple cable, or a complex Ethernet Infrastructure including switches and various media, using a pervasive Interface: the Gigabit (or 100M = FAST) **Ethernet**, over distances ranging between a few centimeters and thousands of kilometers.

Ethernet links are both cheap, robust, and extremely common. Most -if not all- laptops and desktop computers come with a native Ethernet interface. Connectors and cables are found just everywhere and a lot of semiconductors vendors offer low-cost hardware interface chips ("PHYs").
A physical link segment can easily extend to ~100 meters (300') without extra hardware, and extending it further is easy using standard (and cheap) Ethernet equipment (like switches).
Note that Ethernet is not limited to copper lines, it can go through radio (Wifi), Optical Fibers, or Power lines, and GEDEK can be used in all these contexts.

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

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

Note: even if nothing prevents other uses, this IP is best used in the context of **short** and **local** links, and for **data exchange efficiency.**

Don't get confused!

GEDEK is certainly not (only) a MAC nor even just an UDP engine! In fact, there is an extremely optimized MAC inside GEDEK, but it's only a small (and comparatively very simple) fraction of our IP.
We offer several choices for the MAC inside GEDEK: **Gigabit** only (for applications that *require* the bandwidth, it's pointless to fall back to 100M), **FAST** (100M) only (when the PHY is not Gigabit, it's useless to waste resources for a mode that is not available), and **Dual-speed** (for the less likely case when both speeds must be supported within the same application). **10G** is now also available.

GEDEK typically replaces -at least-: a MAC + an SGDMA + External Memory + a Processor + a Real Time Operating System + a Software Ethernet Stack + an Embedded Software application, all this while providing much higher performance and reliability.

Don't get fooled!

There are a lot of urban legends and misconception floating around Ethernet communication.
Long story short, the reality is that:

- TCP/IP is clearly and absolutely not the right protocol for data streaming or massive data transfers !
- UDP/IP packets do not get lost or mixed up (certainly less than TCP packets) ! Otherwise you would not be able to see television, listen to radio, use a telephone or Skype to your family and friends. Quite the contrary! Since UDP is used by all these critical applications (for a reason), all the network infrastructures treat UDP packets with care and provide a high quality of service.
  On the other hand, since it is acceptable to lose TCP packets and there is normally no streaming over TCP, these packets are more easily dropped.

All kinds of FPGAs and Applications

GEDEK's code is extremely optimized and portable, and it fits efficiently in all kinds of FPGAs, even the smallest and cheapest ones. This is important to keep in mind since it enables applications that were usually considered as impossible. Low Cost Remote Sensors can use GEDEK for example.

GEDEK is available for Altera, Xilinx, Lattice and Actel but can be easily ported to practically any existing or future FPGA.

When combined with other ALSE IP like JPEG compression and decompression engines (as in the example cited), cheap and small FPGAs can suffice to build complete video streaming solutions.

Typical GEDEK applications

- Video (and Audio) Streaming (see an example of Consumer application using GEDEK).
- Ethernet Camera links.
- Computer peripherals like industrial printers, medical equipment, imaging systems etc
- Array of FPGA boards linked through an Ethernet network
- High Speed Data Acquisition, Transfer & Pattern Generation
- Remote Data Collection, Remote sensors and actuators, Remote control
- High precision Ethernet Data Acquisition module (an ALSE product)
- Digital Recorders (for high speed data dump).
- Multiple Virtual UART(s) over Ethernet
- Remote FPGA reconfiguration, Remote Serial or Parallel Flash Programming through Ethernet
- etc...

Many Options

GEDEK has been used by a lot of customers, and for many different applications so we listened and developed many options like:

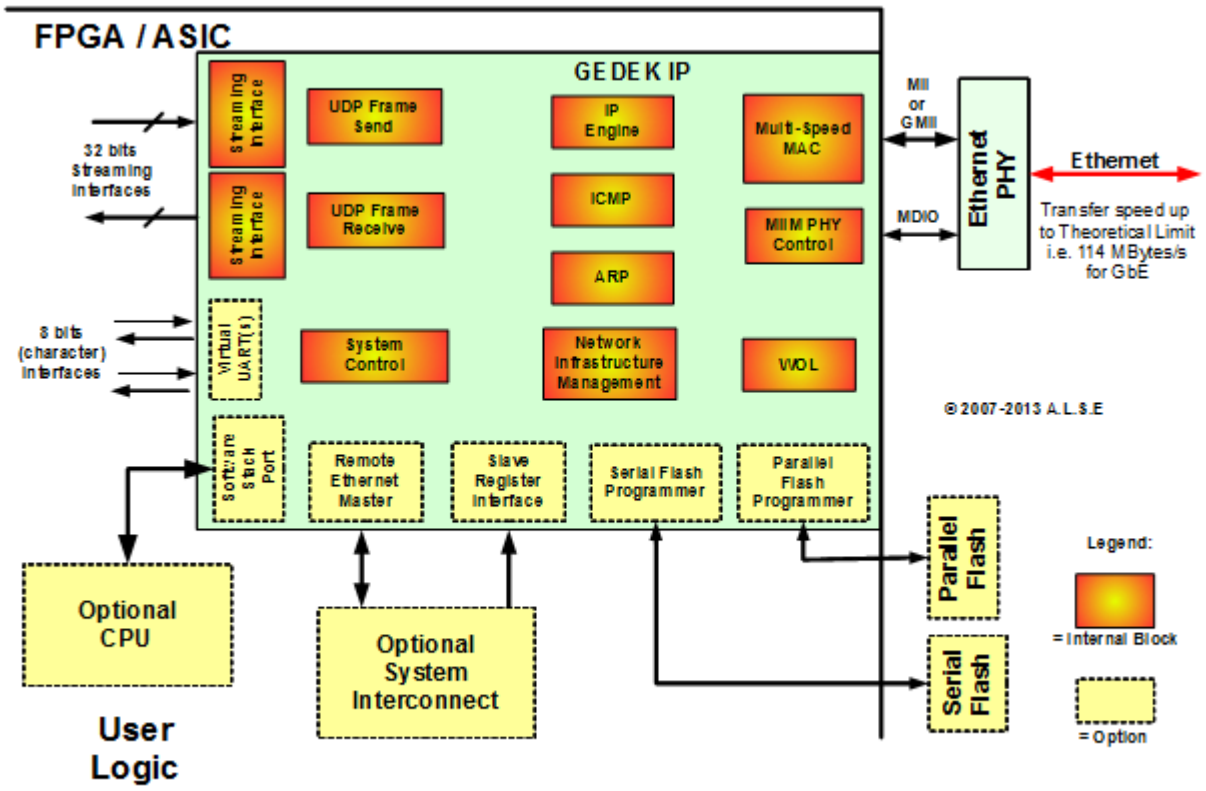
- **Remote Flash Programming** through Ethernet,
  for either Serial or Parallel Flash (FPGA remote programming and configuration).
- **Remote Ethernet Master port** (aka “Virtual Registers”), perfect for FPGA remote control.
  This popular option allows the PC to read and write FPGA internal registers or memory through Ethernet (a memory map master is driven by specific Ethernet frames).
- **Virtual UART channels**. Great to replace character links (RS232, RS485 etc).
- **SGMII / SFP**. To use SFP modules (fiber or RJ45) instead of PHY chips.
- **Secured-UDP** option. Any data lost will be automatically resent.
- **Jumbo Frames** (at no extra cost).
- **Wake On Lan** (at no extra cost).
- **Software Stack Port**. For users who need to implement also a traditional Embedded architecture with no limitation of protocols and take advantage of GEDEK as an UDP Offload Engine.

For other options, contact ALSE.

Hardware needed to use GEDEK

- An FPGA board that includes a Gigabit or 100 Mbits (or dual speed) Ethernet PHY.
- The ALSE GEDEK Intellectual Property.
- A host Computer with an Ethernet connection under Linux or Windows.

Simplified Block Diagram



Note: this is a *simplified* view of the GEDEK core.

The RJ45 Ethernet connector is attached (through a transformer not represented here) to a physical interface device aka “PHY”. A lot of 100M or Gigabit PHYs are available from several vendors, among which for example Ti/NS DP83865, Marvell 88e30xx, 88e1111, 88e1310, Micrel KSZ9021, Microchip LAN8710A, Lantiq Phy11G, Sis196, Vitesse VSC86xx, to name just a few.

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), but also **RMII**, **RGMII**, **SGMII** / **SFP** (as an option)...
- For PHY Control (access to the PHY's control and status registers), we implement and support the ubiquitous **MIIM** (aka **MDIO**) interface when this is necessary (option).

Plug and Ping!

GEDEK is a full-blown *completely standard* Ethernet solution! You can insert any switch or router, stay connected to your local network, use any type of OS and sockets programming on the PC, you can send data to standard UDP software (like VLC), **GEDEK is entirely standard and compliant**.

It takes just a few minutes to insert a GEDEK IP in an FPGA design. Compile it, program the FPGA, connect an RJ45 cable and *ping* your FPGA: it's live! You can exchange data both ways.

Internal Resources

The GEDEK IP is extremely optimized and therefore **very compact**.
On a Cyclone IV target, the typical figures for Internal Resource Usage are:
- **Logic Elements***: typically around **2,000*** !
- Internal **Memory*** used: typically **6 to 8 x M9k** blocks.
- Internal Multipliers: **none**, External Memory used: **none**

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

No External Memory!

Apart from a few internal memory blocks GEDEK doesn't need external memory, thus reducing considerably the BOM (Bill Of Materials) cost.

Zero CPU!

GEDEK is based on a proprietary “IP stack” implemented entirely in hardware and does therefore not require any CPU. This leads to an extremely compact IP and to transfer speeds not achievable with a CPU and 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 and other slower 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 a CRC. This base protocol is very efficient and universally retained by all streaming schemes.

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 is usually a **local** link but GEDEK allows to establish and maintain logical links through any standard network architecture,
- 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.
- Lost (or incorrect) frames can be handled by our “Secure-UDP” 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 frames. However, this scheme may require important storage memory resources on the FPGA side 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 to slow down the transfer).

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

Expert Mode Register Interface

GEDEK can be used with no programming in “automatic” mode which is suitable in many cases. For more challenging applications, complete control can be obtained using a memory mapped Read/Write Register (slave) Interface. This “Expert mode” allows full access in the FPGA to the internal features of GEDEK, for taking control of all default behavior, dynamically. Ethernet MAC & IP addresses, ARP table, network infrastructure management and many other parameters can be controlled by the application.

Remote Ethernet Master Port

This popular option (previously known as “Virtual Registers”) is perfect for FPGA remote control. It allows the PC to read and write in the FPGA through Ethernet (a memory map Master port is driven by specific Ethernet frames and generate Read and Writes transactions in the FPGA).
This option is popular because it allows simply to write programs that use the remote FPGA through Ethernet just as a PC peripheral.

Remote Flash Programming

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. Updating operating parameters, Firmware or ROM contents remotely is just as easy.
Safe (Factory default) and User mode are provided for some FPGAs.

Performance

The GEDEK IP 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 in Gigabit mode), thus **guaranteeing the 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 sustain 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.

However, when connected to a host computer with a standard NIC (Network Interface Card), the effective throughput will be limited by the host capability to receive or produce Ethernet frames and accompanying data. 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. We deliver a reference 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. By our experience, and for what is worth :

- 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.
- 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 MBytes/s range are very easy to reach by almost any kind of PC. Transfer rates between 30 and 50 Mbytes/s require a good architecture, an optimized OS and a lot of care, Transfers above 70 Mbytes/s can't be reliable with simple and common solutions.

Jumbo Frames

We have added the option to support Jumbo frames, but we ask customers to be careful about it. Jumbo frames are not widely used and can cause problems with the network infrastructure, the Computer driver or the OS... There is no justification of using Jumbo frames to minimize the protocol overhead. But in some carefully prepared cases, (depending essentially on the type of data exchanged and the Operating System) Jumbo frames can be advantageous. In any case, Jumbo Frames are supported by GEDEK.

Software Stack Port

This option (aka “Bypass Port”) is reserved to users wanting to also implement an Ethernet *Software* Stack besides the GEDEK communication Engine. One motivation is to implement protocols that are not part of GEDEK's internal Hardware Stack, like TCP/IP, IGMP, DHCP, Web server etc.. Our solution includes a complete Embedded system with FreeRTOS and a Nios Processor.
With this option, one can view GEDEK as an UDP Offload Engine.

Deliverables

The GEDEK IP consists of several parts:

- **HDL IP**: the actual contents and format may vary according to the type of license purchased, from Netlist-only to Encrypted source to Full-RTL, time-limited, permanent...
- **Host API**: this API contains simple-to-use functions implementing the actual data transfers. This API is available under Windows and Linux..
  Users familiar with sockets programming do not have to use this API and can write their own code integrated with their application.
- **Reference Designs**: in order to help the customers, we deliver some ready-to-use Reference Projects including one that can be used to test and optimize the reception (PC) side. This design does :
  - 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.

Behavioral Simulation Model

We have developed an extremely **powerful behavioral model**! This will allow you to simulate easily your application including Ethernet transfers to and from the outside, using simple text files for the frames contents. Otherwise, it would be nearly impossible to simulate the FPGA and the Ethernet transfers (you would need a model for the PHY and for the network, and this would be extremely difficult and tedious, if possible at all).

Loss-less (Secured) UDP Data Exchange

When the sender wants to ensure that the receiver has not lost any data (and if it has, resend missing data) this option can be used, but this implies several constraints on the FPGA side and usable bandwidth. Another possibility to secure the data transfers is to use our next option below.

Dedicated Network Interface Card

As an option, we are developing a specific Gigabit Ethernet PCI-Express Card (with an FPGA on it) to implement guaranteed loss-less transfers at 100+ MBytes/s ! Coupled with GEDEK, it is the perfect and final solution to secure UDP transfers at ultimate speeds with no processor overhead (a plus for sustained transfers typically at or above 80 Mbytes per second).
Indeed, this card contains a GEDEK IP and the data is deposited in the PC memory through the PCI-Express port without any processor intervention, and therefore without overhead. Moreover, the data exchange with the remote GEDEK is completely secured without requiring significant extra bandwidth nor latency.

Ready to Use Hardware

Most commercial FPGA boards and Design Kits can be used with GEDEK.

If you design your own FPGA board, we can help with the Gigabit PHY section, since there are a few things to be careful with and we have acquired a lot of experience with our own boards and helping customers.

Questions & Answers

What FPGA can be used ?

Practically any FPGA can take advantage our GEDEK kit. The presence of internal memory is necessary (so many old CPLDs like MaxII or MaxV devices for example can not be used). Relatively "old" FPGAs are still perfectly suitable. Recent CPLDs are also usable (Max10, MachXO, Igloo...).

Is the GEDEK IP 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. Their transfer performance is very limited and Gigabit is not supported.
GEDEK requires just a PHY device, the MAC part of the IP.

Can SFP Modules be used (and Fiber links)?

Yes! This option is available, contact ALSE. Not all FPGAs will support this high-speed serial links though (2xSERDES are required).

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 has been adopted by a lot of customers, but it is is still 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:

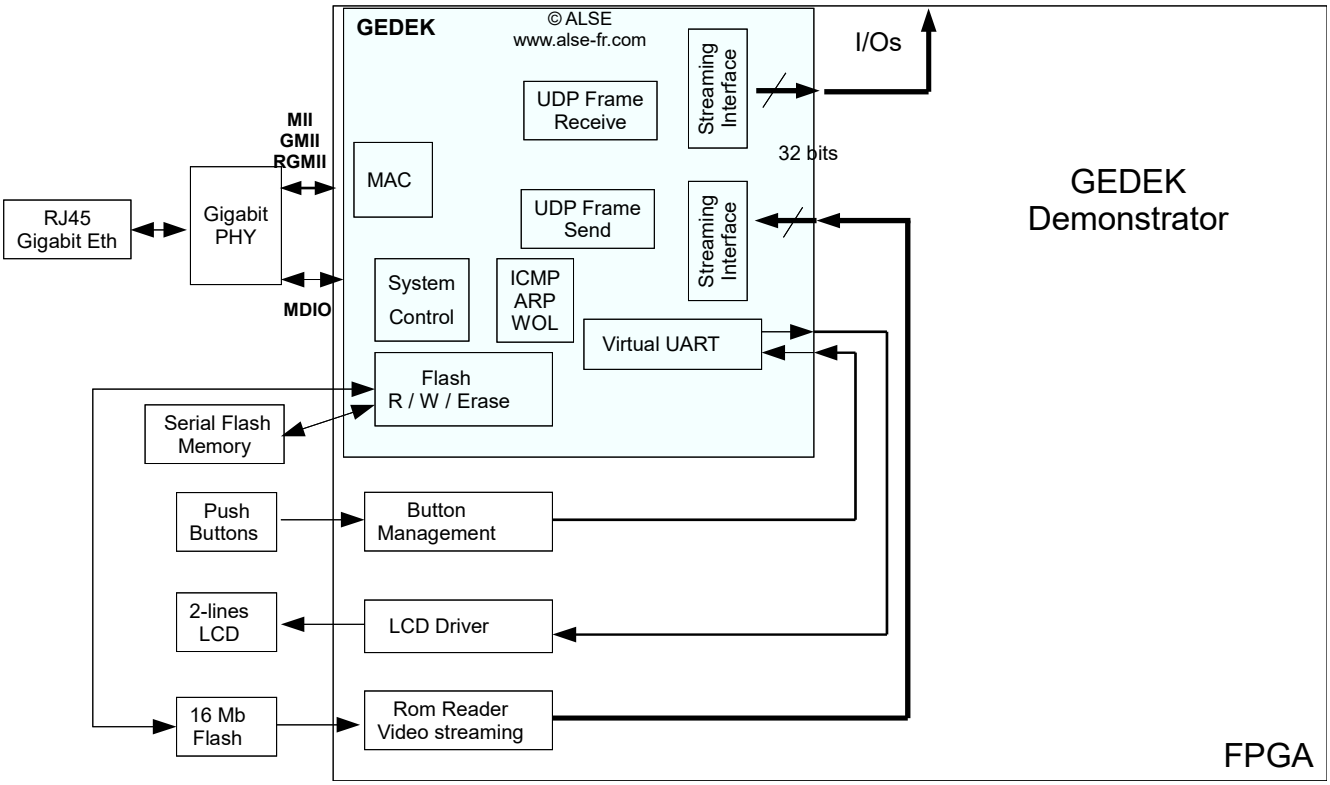
Bertrand CUZEAU
Technical Manager A.L.S.E.
Telephone: +33 1 84 16 32 32 – E-mail: info@alse-fr.com

GEDEK Demonstrators

Demos are available for: Altera Cyclone, Cyclone II, CycloneIII, DBM modules, StratixIII, StratixIV, StratixV, Cyclone IV, Cyclone V, ArriaII, ArriaV, Xilinx Spartan3E, Spartan6, Virtex6, Actel SmartFusion, etc...

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:



(note: this demo is available as source code for GEDEK customers)

Principles:

1. **Push-Buttons**.
   When pressed, they generate an ASCII message routed to the Virtual UART input.
2. **LCD driver**.
   A 2x16 LCD driver (available on ALSE's Website for free) is simply connected to the Virtual UART output. On kits where it is not available, a 80x25 CRT or a true UART port can be used.
3. **Serial** or **Parallel Flash**. This option is transparent (integrated in GEDEK). Allows a proper PC application to Erase, Read, Write and Compare the contents of the external Flash.
4. **Rom Video Streaming**
   This is a very simple block reading the Flash (compressed video) and sending the **video stream** to the GEDEK streaming port. This memory can be programmed through Ethernet !
5. PC-side: **Virtual Terminal + Video Player**

   A PC utility (Virtual Terminal) emulates a character terminal hooked to the Virtual UART.
   Strings can be typed in the PC, transit through the Virtual UART block, and appear on the LCD (or the screen for some Kits). The Virtual terminal also displays strings generated inside the FPGA (when buttons are pressed). The video stream (Cartoon) located in the Flash memory can be displayed on the PC (using VLC eg).

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