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

February 2009, ver 2.1a

ALSE's GEDEK Gigabit Ethernet Data Exchange Kit

Product Brief

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.

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.

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. The easiest (and probably cheapest) for a customer is clearly to define with us the exact needs of his application and let us customize this Kit to his exact needs.

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, this Kit has a **100 Mbits** a **1Gb** version**,** and a **100/1000** version. Obviously, the data rate achievable in 100 Mbits/s is ~1/10th Gigabit values i.e. ~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 very standard and common Interface : the Gigabit 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 many 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) 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).

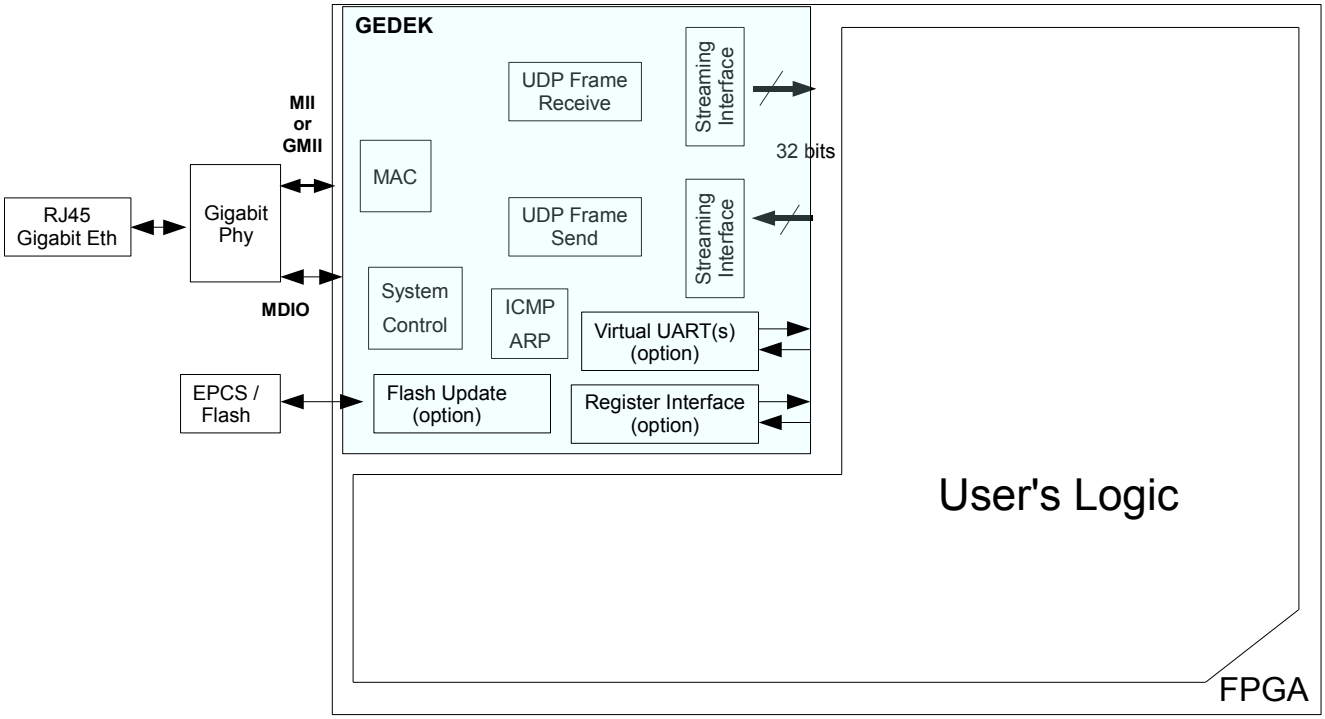
Typical application examples are :

- Computer peripherals, like industrial printer
- High Speed Data Acquisition,Transfer & Pattern Generation
- Remote Data Collection,
- Video Streaming,
- Multiple Virtual UART(s) over Ethernet
- Remote FPGA reconfiguration and serial Flash update, etc...

To implement the complete link, you need :

- An FPGA board
- A Gigabit or 100 Mbits 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 applciation 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 the usual CRCs.

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)

Upon request, we can 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 observability of the design.

Flash Update

We offer an option 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 parameter, 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 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/Worksation 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 the DBGIG1 module available at http://www.devboards.de and thus be ready to run the Gigabit Demo or the Reference Design. "Old" FPGA boards (Cyclone 1C20 or 2C35...) can 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...

How 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.

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, most compact, and most efficient way to exchange a lot of data at a fast pace between an FPGA and a computer. With prices starting below 5k €uros, it is very likely the cheapest 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 performance).

Another alternative might be USB 2.0 but this solution is more complex would be dependent on some specific USB2 chip 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.
Tel +33 1 4279 5138
info@alse-fr.com

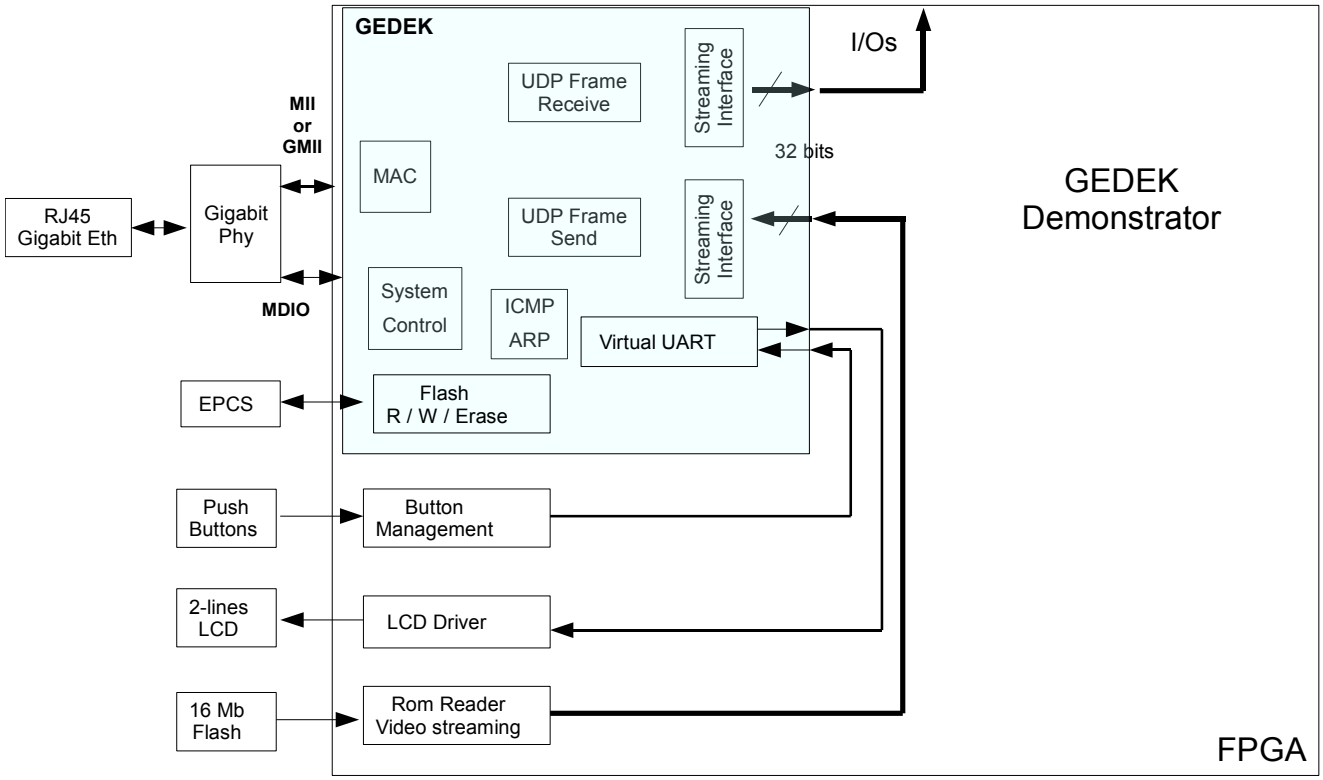
GEDEK Demonstrators

- NEEK version available, contact ALSE
- DBM version available, contact ALSE
- Spartan3E version available, contact ALSE

To help customers evaluate our GEDEK Intellectual Property and test our Technology, we have created several Demo Kits, including the one described below, based on an Altera Cyclone II FPGA board equipped with a Gigabit PHY Extension connected to one SantaCruz expansion connector.

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 byte stream to the GEDEK streaming input.
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.

Building a GEDEK Demonstrator with the 2C35 Board

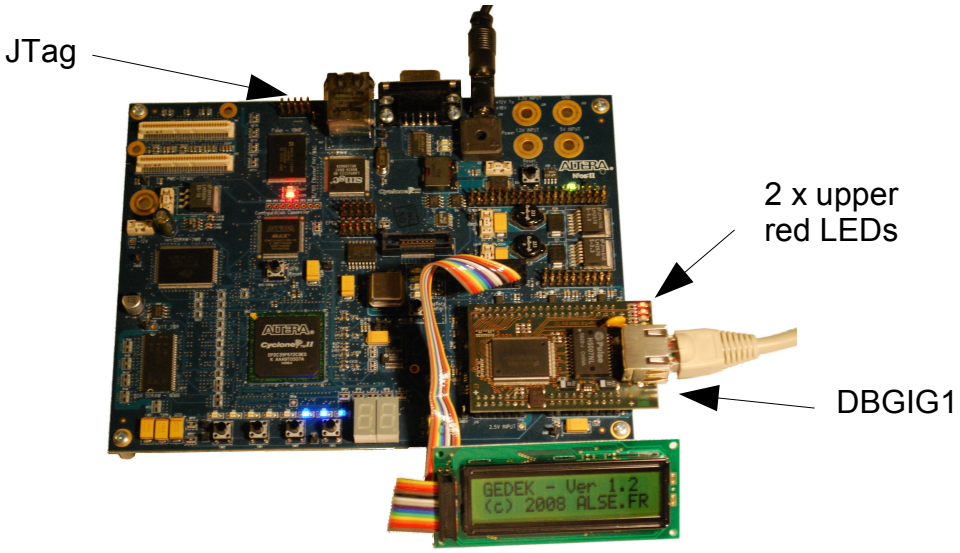
What you'll need to set up the Demonstrator

+ A working Nios II **Cyclone II 2C35 FPGA board** with its **2-lines LCD** module,
+ a **DBGIG1** Ethernet Triple Speed Phy module,
+ a **PC** under Windows (for Linux, contact ALSE) with either a Gigabit or a FAST (100M), Ethernet Network port and administration rights,
+ Quartus II v8.1 and Nios II IDE v8.1 installed and licensed, Web Edition ok,
+ a cat6/5e **Ethernet RJ45 cable** (a Gigabit or 100M switch can be used) ,
+ the free software VideoLAN **VLC media player** installed,
+ **GEDEK_Demo files** version 1.5 from ALSE.

Preparing the Board

- Install the demo files : run **GEDEK_Demo.exe** and **install under C: or D:**.
  (Do not even try to install in a path with space characters !)
  The installation program will create automatically a sub-directory named GEDEK_Demo.
- Set up the Nios II **Cyclone II 2C35 FPGA board** :
  Make sure the **LCD** is attached to **J12**, with pin 1 on the Cyclone side.
  Plug a **DBGIG1 Phy module** on the second Santa Cruz connectors (**J15-16-17**).
  Connect a **USB Blaster** on the **JTag** U62 connector.
  Apply the **Power**.

  Note : Your USB Blaster should be [USB-0].
  Otherwise, you'll have to modify the shell script accordingly.



Preparing the PC

Note that GEDEK works very well with PCs under Linux, and our API as also available under these platforms. However, this demo is only targeted to PCs under Windows.

- **Modify** your **Gigabit or Fast Ethernet** Network property to **fixed IP = 192.168.1.100**.
  (in fact, you can use between 192.168.1.2 and 192.168.1.250, but **not** 192.168.1.18 !)
  Note: though this is often preferable, you do not have to connect directly the PC to the board (point-to-point). You can stay on a generic LAN with switches etc, and keep using DHCP for example, but you'll need to be in the address range above.
  Being on a LAN is fine unless you try to transport very heavy loads of data across the network ! The amount of data involved by this demo remains reasonable anyway.

  This version of GEDEK (1.5) includes a **Speed Auto-Detect** option to adapt itself automatically to the connection speed. Note however that a **Full-Duplex** connection is mandatory.

- Connect an **RJ45 Cat6** or Cat5e **Ethernet** patch (straight) **cable** between the **DBGIG1** module and your PC **directly** (cross cable unnecessary: the Phy will auto-adapt).
  Do NOT try to use the RJ45 connector located on the FPGA board : it is not a Gigabit connector, and it is not connected to a Phy: no hope on this connector.
  Avoid staying on your usual network : this could lead to some addressing conflicts. You can try this later when the demo has worked point-to-point.

  On the DBGIG module, the two upper LEDs (external side) must immediately turn on, and the Phy chip will get very hot ! If this doesn't happen, you've not achieved a physical Gigabit connection.
  If you achieved a 100M bits connection, the demo (version 1.5) will also work.
  If you want to test in Gigabit Ethernet mode, you can try with another cable, another PC, another switch... until the two LEDs turn on.

Programming the Parallel Flash and the EPCS – Automatic Script

- Open a **Nios II Command Shell**
  (Program > Altera > Nios II EDS > Nios II Command Shell)
- Change directory (cd) to `GEDEK_Demo/2c35`
- Run the "Magic Script" : `./Flash_2c35.sh`

  Be patient ! There are two Flash memories to program, one is 16 megabytes large (for the video data), and we start by reading it (through JTag cable) to save the contents in a 40MB file. Reading/Saving the Flash takes about 7 minutes. Programming takes longer (~20 minutes).

  If you see a message from your firewall asking to unlock some program (likely epcs_prog), please comply.

If something goes wrong

If you see errors during the execution of the script, analyze the messages and make sure the requirements are respected (Quartus and Nios II IDE v8.1 installed, Demo files installed as above, USB Blaster to USB-0...).
Make sure you are connected directly between the PC and hhe Nios board.
Note that we have provided the files to program the Flash memories manually. Simply look at the script to see the steps involved. Note that you can program the EPCS memory directly with the JIC file included in the archive.
If you still do not succeed, contact ALSE (ip_support@alse-fr.com).

- If the script works and only fails while trying to "Program the EPCS", this is probably a network issue (see the steps above).
  You can program the EPCS with the JIC file and solve your network concerns later.
- Power Off the board, remove the JTag cable : **you're done, your System is READY !**
  You can go to the Demo itself, described next page.

Remarks

- During the setup script, the original and complete Flash contents is saved in a File named *OriginalFlashContent.flash*. If you want to restore this original contents (and lose the GEDEK demo) you can use the following command :
  nios2-flash-programmer --base=0x0 OriginalFlashContent.flash.flash
- Beware in Gigabit Ethernet mode ! You need good cables and ad hoc equipment to build a reliable GBE connection. In case of doubt, you can try running this demo with a Fast (100M) Ethernet connection.
- The script does verify the Flash contents : if there is no need (if the Flash is already programmed correctly), no erasing/programming will occur.

This version 1.5 is quite new, so please send feedback to ip_support @ alse-fr.com.

GEDEK Demonstrator - User's Manual for 2C35

We suppose the Board has been prepared as above.

Please respect carefully the sequence below.

- Apply the **Power** to the 2C35 board.
- Connect an **RJ45 Cat6** or Cat5e **Ethernet** patch (straight) **cable** between the **DBGIG1** module and your PC **directly** (cross cable unnecessary: the Phy will auto-adapt).
  Do NOT try to use the RJ45 connector located on the FPGA board : it is not a Gigabit connector, and it is not connected to a Phy: no hope on this connector.
  Avoid staying on your usual network : this could lead to some addressing conflicts. You can try this operation mode later, when the demo has worked point-to-point.

  On the DBGIG module, the two upper LEDs (external side) must immediately turn on, and the Phy chip will get very hot ! If this doesn't happen, you've not achieved a physical Gigabit connection.
  If you achieved a 100M bits connection, the demo (version 1.5) will also work.
  If you want to test in Gigabit Ethernet mode, you can try with another cable, another PC, another switch... until the two LEDs turn on.
- From the Windows Start menu, launch **VLC media player** (to display the video) :
  File > Open Network Stream : Port = **1234**
- On the PC, open a **Nios II Command Shell**. (Program > Altera > Nios II EDS > Nios II Command Shell)
- Change to the `GEDEK_demo` directory (cd).
- Ping the board which is by default at ".18" : **ping 192.168.1.18**
  The ping should be successful, otherwise, check your network settings/connections.
- Launch the *gedek* Register Access utility: `./gedek.exe`
  If your firewall asks, unlock (authorize) this application.
- Open a second (other) **Nios II Command Shell** and change to the GEDEK_Demo directory.
- In this second command shell, run the Virtual Terminal: `./uart_api.exe`
  If your firewall asks, unlock (authorize) this application.
  Every ~4 seconds a Welcome Message should be displayed & scroll.
- In the Gedek.exe utility application, activate the Video and the UART flows :
  W <Enter> 0 <Enter> 84 <Enter> 8 (hex) is the video stream bit, 4 is UART stream
  Enjoy the video !
- In the Virtual Terminal, **type** some **string** terminated by the <**Enter**> key : your text should be displayed on the LCD within a few seconds.
- You can modify the LCD **refresh rate**: for example, **W**rite at address **2** the value **1800000**.
- Press the four **Push-Buttons** and look at the Virtual terminal window.
- You can modify the displayed **version numbers** by **w**riting major/minor at addresses **3** & **4**.
- Let's now **Remote-Program the EPCS Memory through Ethernet !**
  Note : you can leave the video streaming on while programming, no problem !

  You must **exit** the GEDEK tool : type **Q** then *exit* or close the window.

  In a new Nios II Command Shell, launch ALSE's EPCS Programmer utility :
  `./epcs_prog.exe ./2c35/archives/top_2c35_def.rbf`

  The Erase / Program / Verify occurs while all the rest of the system works (video included).

  Beware: if you program a wrong bitstream and power down the board, you will (indeed) have to use the blaster to reconfigure the FPGA and/or reprogram the EPCS. We deliver two versions of the Demonstrator's bitstream: one displays "default" and the other "Version 1.5".

This sequence should demonstrate the main concepts and the beauty of the GEDEK solution.
The demonstrator was designed in just a few hours around the GEDEK module, the HDL code involved is pretty straightforward, and the complete design fits in ~5k LEs.

We hope you enjoyed the Demo !

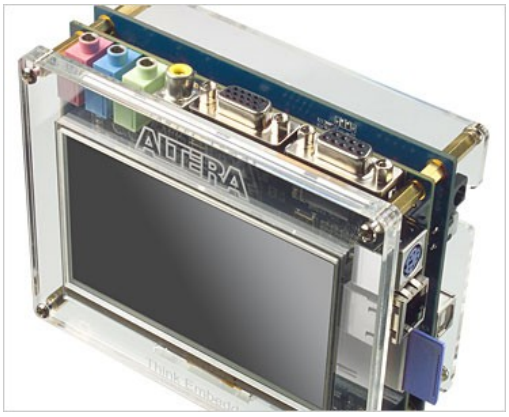
The A.L.S.E. Team

Building a GEDEK Demonstrator with NEEK
(GEDEK version 1.51)

Please note that, since the NEEK board does not have a GbE (Gigabit Ethernet) PHY, the demo will only run at 100 Mbps. Please use another kit to test the GigaBit Ethernet mode !

What you'll need to set up the Demonstrator:

- A working **Nios II Cyclone III 3C25 Evaluation Kit " NEEK "**. See picture →
- A **PC** under Windows with a 100 Mbps (or Gigabit) **Ethernet** Network port, with administration rights and with **Altera** Quartus II + **Nios II IDE** installed.
- An **Ethernet RJ45 "patch" cable**.
- Off course, the **GEDEK_Demo files** from ALSE.
- *The free VideoLAN **VLC media player** installed.*



- Connect the **RJ45** (patch) cable into the Ethernet port of the NEEK (on the right in the above picture) on one side, and connect the other side to your PC. Prefer a direct connection (but a switch can be used). Note : a cross-over cable is not necessary since the PHY on the Kit can automatically adapts itself.
- Connect a **USB** cable from your PC into the USB Port (also on the right in the above picture), then apply the **Power**. Make sure the driver for the USB-Blaster is installed and the embedded USB-Blaster is recognized.
- **Configure the Ethernet connection of your PC under Windows (XP or 2000)** :
  Modify your Ethernet Network property to **fixed IP = 192.168.1.100**.
  (in fact, you can use between 192.168.1.2 and 192.168.1.250, but **not** 192.168.1.18 !)
  Note: you can stay on a generic LAN with switches etc, and keep using DHCP for example, but you'll need to be in the address range above. This is fine unless you try to transport very heavy loads of data over the network ! It's certainly better to avoid doing this over a company network!

  Note that our APIs are compatible with Linux Operating Systems. However, these demos are delivered for use with Windows exclusively.

Important !

The Flash configuration process will erase the on-board Flash memory ! If all works well, the script will save the original content in a file named `OriginalFlashContent.flash`.
It's probably a good idea to save the original contents first anyway.

- Open a **Nios II Command shell** and change directory (cd) to the folder where the Gedek demo files have been unpacked. Then change to the **NEEK sub-directory**.
  Execute the configuration script with the following command :
  `./Gedek_demo_3c25.sh`

Be patient. The flash programming process will take about 10-15 minutes to complete.

- When the programming is complete, Power Off the NEEK, then remove the USB cable:

You're done, your NEEK kit is now Ethernet-READY !

GEDEK Demonstrator - User's Manual for NEEK

for Gedek version 1.51

We suppose the Board has been prepared as above.

Note that the 2-lines LCD display used with other demonstrators has been replaced by a simple text terminal displayed on the hi-res Graphic LCD display.

- Apply the **Power.**
- Connect an **RJ45 Ethernet** patch (straight) **cable** between the Ethernet port of the NEEK and your Ethernet switch or PC directly (cross cable unnecessary: the PHY will auto-adapt).
  On the NEEK, the two RJ45 connector's LEDs must immediately turn on.
- On the computer, open a **Nios II Command Shell**.
- Change to the **gedek_demo** directory (cd).
- Try to ping the board which is by default at the address ending with "192.168.1.18" :
  `ping 192.168.1.18`
  The ping should be successful, otherwise, check your network settings/connection.
  Note : this pinging is necessary ! The FPGA "remembers" this request and uses the sender information to automatically establish the route to send the data flows.
- Launch our *gedek* Register Access utility:
  `./gedek.exe`
- Open a second (other) **Nios II Command Shell**.
- In the second command shell, run our Virtual Terminal:
  `./uart_api.exe`
- From the Start menu, launch **VLC media player** (to display the video)
  File > Open Network Stream : Port = **1234**

  Note: VLC is not always stable on all computers (especially on some laptops). If VLC crashes, just try to launch it again, it usually solves the issue.
- In the Gedek utility window, activate the Video and the UART flows :
  W <Enter> 0 <Enter> 84 <Enter> 8 (hex) is the video stream bit, 4 is UART stream
  Enjoy the video !
- In the Virtual Terminal, **type** some **string** terminated by the <**Enter**> key : your text should be sent back within a few seconds and be displayed on the NEEK LCD screen.
- You can modify the Information **refresh rate**: for example, **W**rite at address **2** the value **1800000**.
- Press the four **Push-Buttons** and look at the Virtual terminal window (PC).
- You can modify the displayed **Version numbers** by **W**riting major/minor at addresses **3** & **4**.
- Using the NEEK board, you can also **Remote Read/Erase/Program/Verify the Parallel Flash**.
  You can reprogram the Configuration or the Video, or both !
  Note that our GEDEK has an option to reprogram the EPCS (serial) memory, but the NEEK board is not equipped with an EPCS. Try the demo with the 2C35 board or contact ALSE for further informations about **Remote EPCS Programming** !
  Note: if you reprogram the FPGA with a design that does not include a GEDEK, you won't be able to Remotely Program the Flash through Ethernet any more (you could re-load GEDEK in the FPGA using the USB Blaster, then use Ethernet again).

  To **verify the bitstream** programmed into the Parallel Flash:
- Stop the video first ! : **W <Enter> 0 <Enter> 04 <Enter>**
- `./flash_prog.exe ./3c25/archives/top_3c25_1.44.rbf 20000 check`

Voilà !

This sequence should demonstrate the main concepts and the beauty of our GEDEK kit.

The demonstrator was designed in just a few hours around the GEDEK IP, the HDL code involved being pretty straightforward. The complete design fits in less than 4k LEs.

We hope you enjoyed the Demo !

The A.L.S.E. Team

Building a GEDEK Demonstrator with 3SL150 Board

for Gedek version 1.45

All you'll need to set up the Demonstrator is :

- A working **Stratix III Developpement Kit FPGA board** with the **2-lines LCD** module connected →
- A **PC** under Windows with a **Gigabit** Ethernet Network port and administration rights
- A **cat6**/5e **Ethernet RJ45 cable** (a Gigabit switch can be used)
- The free VideoLAN **VLC media player** installed (available at www.videolan.org)
- The **GEDEK_Demo files** from ALSE.

- Set up the **Stratix III FPGA board**.
  Make sure the **LCD** is connected to the board.
- Connect an **USB** Cable the the USB connector. Apply the **Power**.
- Connect the **RJ45** cable into the Ethernet port of the Stratix III Dev Board.
- **Configure your PC under Windows (XP/2000)** :
  Modify your Ethernet Network property to **fixed IP = 192.168.1.100**.
  (in fact, you can use between 192.168.1.2 and 192.168.1.250, but **not** 192.168.1.18 !)
  Note: you can stay on a generic LAN with switches etc, and keep using DHCP for example, but you'll need to be in the address range above. This is fine unless you try to transport very heavy loads of data over the network ! It's certainly better to avoid doing this over a company network!
  Note that our APIs are compatible with Linux Operating Systems. However, these demos are delivered for use with Windows exclusively.

```
The flash configuration process will erase the original flash content !
The configuration script will create a backup, however you may also do your
own backup before processing !
```

- Open a **Nios II Command shell** and get into the folder where are located the Gedek demo files. Execute the configuration script with the following command :

  ```
  ./Gedek_demo_3sl150.sh
  ```

  The flash update process will take about 10-15 minutes.
- Once the programming is complete, you can launch the Gedek tool :

You're done, your Stratix III board is READY for the Gedek !

GEDEK Demonstrator - User's Manual for 3SL150 Board

for Gedek version 1.45

We suppose the Board has been prepared as above.

- Do not power off the board as the Gedek is not currently programmed into the programming device !
- Connect an **RJ45 Ethernet** patch (straight) **cable** between the Ethernet port of the SIII Board and your Ethernet switch or PC directly (cross cable unnecessary: the Phy will auto-adapt).
  On the SIII Board, the two RJ45 connector's LEDs must immediately turn on.
- On the computer, open a **Nios II Command Shell**.
- Change to the gedek_demo directory (cd).
- Try to ping the board which is by default at ".18" :
  `ping 192.168.1.18`
  The ping should be successful, otherwise, check your network settings/connection.
- Launch the *gedek* register access utility:
  ./gedek.exe
- Open a second (other) **Nios II Command Shell**.
- In the second command shell, run the Virtual Terminal:
  `./uart_api.exe`
- From the Start menu, launch **VLC media player** (to display the video)
  File > Open Network Stream : Port = **1234**
- In the Gedek utility application, activate the Video and the UART flows :
  W <Enter> 0 <Enter> 84 <Enter> 8 (hex) is the video stream bit, 4 is UART stream
  Enjoy the video !
- In the Virtual Terminal, **type** some **string** terminated by the <**Enter**> key : your text should be displayed on the LCD screen and sent back to the virtual uart tool within a few seconds.
- You can modify the Uart **refresh rate**: for example, **W**rite at address **2** the value **1800000**.
- Press the four **Push-Buttons** and look at the Virtual terminal window.
- You can modify the displayed **version numbers** by **w**riting major/minor at addresses **3** & **4**.
- Using the SIII Board board, you can also **Remote program the Parallel Flash** (Configuration + User's Data Flash device). You can reprogram the Configuration or the Video, or both !
  Note that our GEDEK has the option to reprogram the EPCS memory, but the Stratix III board is not equipped with an EPCS. Try the demo with the 2C35 board or contact ALSE for further informations about **EPCS Remote Programming** !
  Note: if you reprogram a configuration that does not include a Gedek, you won't be able to remote program the Flash through Ethernet any more (until you load the right design using the USB Blaster)
  To check the parallel flash content, reset the board then simply run the following command :

  ./**flash_prog**.exe ./3sl150/video.dat 100000 check

  This sequence should demonstrate the main concepts and the beauty of the GEDEK solution.
  The demonstrator was designed in just a few hours around the GEDEK module, the HDL code involved is pretty straightforward, and the complete design fits in <4k Les.

  We hope you enjoyed the Demo !

  The A.L.S.E. Team