



Instituto
Nacional
de Tecnología
Industrial



Ministerio de Industria y Turismo
Secretaría de Industria, Comercio
y de la Pequeña y Mediana Empresa



Simposio Argentino de Sistemas Embebidos (SASE) 2010 FPGA softcores

Disertantes:

Ing. Salvador E. Tropea (Softcores)

Ing. Diego J. Brengi (Leon y Grlib)

Electrónica e Informática

Desarrollo Electrónico con Software Libre (DESoL)

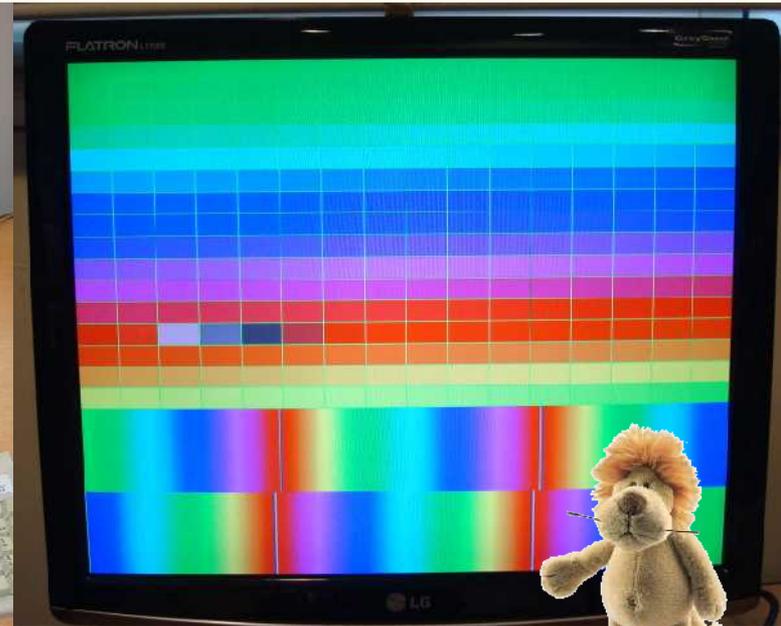
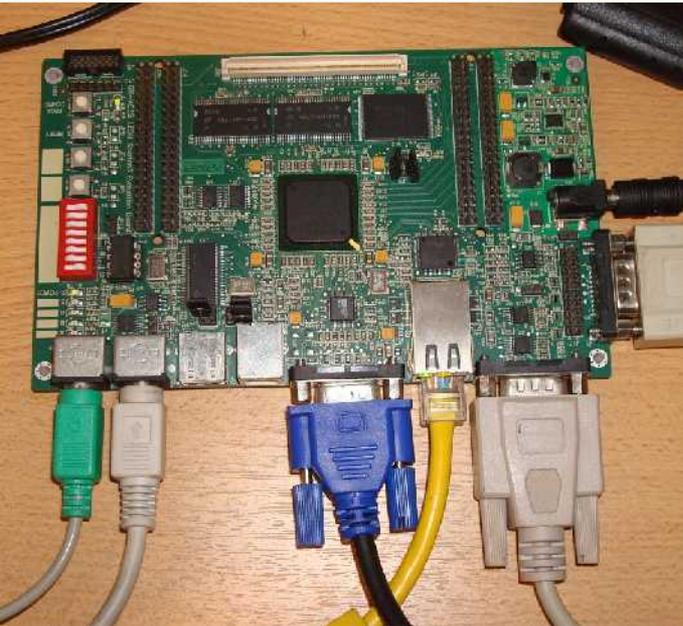




Instituto
Nacional
de Tecnología
Industrial



Ministerio de Industria y Turismo
Secretaría de Industria, Comercio
y de la Pequeña y Mediana Empresa



El procesador LEON y la GRLIB Introducción y ejemplo de uso con GNU/Linux





Agenda

PRIMERA PARTE

“Introducción al procesador LEON y la GRLIB”

1. Motivación
2. Alternativas investigadas
3. Introducción al procesador LEON, la arquitectura SPARC y el bus AMBA
4. El procesador LEON
5. La GRLIB
6. Casos de aplicación

SEGUNDA PARTE

“Ejemplo de uso de la GRLIB con GNU/Linux”

7. Hardware utilizado
8. Fuentes y software para la GRLIB
9. Configurando la GRLIB
10. Verificando la GRLIB
11. Síntesis y Place & Route
12. Software y toolchain para el S.O. embebido
13. Configurando Snapgear Linux
14. Obtener y transferir la imagen Linux
15. Pruebas del sistema





PARTE 1

El LEON y la GRLIB





Motivación

Objetivo de la búsqueda y criterios:

- Investigar y trabajar con una plataforma basada en FPGA, que permita implementar un sistema GNU/Linux embebido.
- De uso y comercialización libre.
- Independiente del vendedor de FPGA (no atarse a uno solo).
- Disponer de los fuentes HDL del procesador (preferencia por VHDL).
- Apto para sistemas embebidos.



El procesador LEON y la GRLIB

Introducción



Instituto
Nacional
de Tecnología
Industrial



Algunas alternativas investigadas

NOMBRE	BITS	CODIGO	PERMISOS	LENGUAJE	GCC	LINUX	FPGA
PICOBLAZE	8	NO	Gratis		NO	NO	Xilinx
PACOBLAZE	8	SI	Tipo BSD	Verilog	NO	NO	Independiente
Simil AVR en OC	8	SI	GPL	VHDL	SI	NO	Independiente
ARM/Cortex M1	32	NO?	Uso pago.		SI	SI	Actel, Xilinx, Altera
MICROBLAZE	32	NO	Viene comprando Entorno.		SI	SI	Xilinx
NIOS II	32	NO	Viene comprando Entorno.		SI	SI	Altera
Open SPARC T1 (Niagara)	64	SI	GPL	Verilog	SI	SI	Silicio.
Open SPARC S1 (Sirocco)	64	SI	GPL	Verilog	SI	SI	Xilinx (LUTS>35000). Altera? Actel?
OpenRisk 1200	32	SI	LGPL	Verilog	SI	SI	Actel, Xilinx, Altera
ZPU	32	SI	BSD	VHDL	SI	NO	Actel, Xilinx, Altera
LEON	32	SI	GPL	VHDL	SI	SI	Actel, Xilinx, Altera y Lattice

Ver también: <http://www.1-core.com/library/digital/soft-cpu-cores/>





El procesador LEON

- **Procesador de 32 bits.**
- **Cumple con la arquitectura SPARC V8 (RISC). Especificación publicada por SUN y de libre uso.**
- **Utiliza Bus AMBA 2.0 (AHB y APB) para interconectar periféricos. Especificación de ARM, de libre uso incluso para cosas que no sean de ARM.**
- **Lenguaje VHDL.**
- **Código fuente bajo licencia GPL/LGPL (según versión).**
- **Mantenido y comercializado por la empresa Aeroflex/Gaisler.**

<http://www.sparc.org/standards/V8.pdf>

El procesador LEON y la GRLIB

El procesador LEON



Instituto
Nacional
de Tecnología
Industrial



Historia del procesador LEON

1997

Se inició un proyecto interno del "European Space Research and Technology Centre" (ESTEC) para desarrollar un procesador de 32 bits tolerante a fallas, para emplear en las misiones espaciales. Uno de los objetivos era **lograr independencia tecnológica con respecto a los procesadores tolerantes a falla provenientes de Estados Unidos, que comenzaban a plantear problemas legales de exportación y limitaciones para su aplicación y redistribución.**

2001

Se funda Gaisler Research, una empresa privada, para desarrollar y dar soporte para el LEON.



El procesador LEON y la GRLIB

El procesador LEON



Instituto
Nacional
de Tecnología
Industrial



Licencias del procesador LEON/GRLIB

- Los procesadores LEON1 y LEON2 están bajo la licencia LGPL
- El LEON3 y la GRLIB se ofrecen bajo GPL, salvo algunos pocos casos LGPL.



El procesador LEON y la GRLIB

El procesador LEON

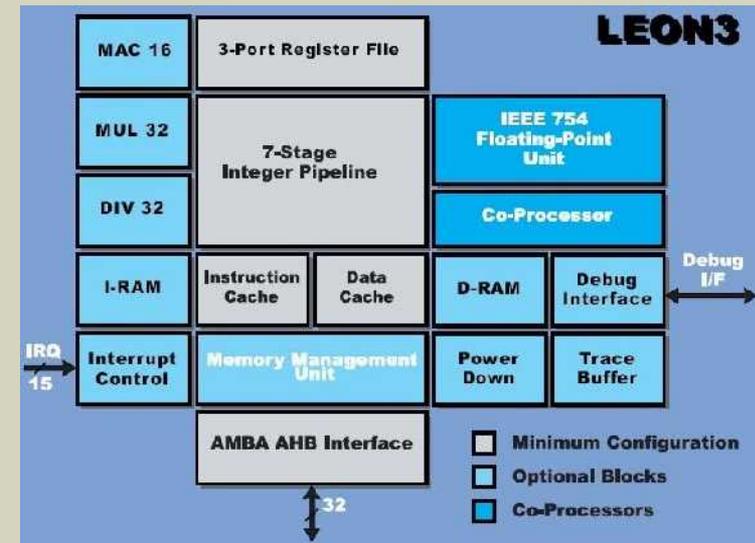


Instituto
Nacional
de Tecnología
Industrial



El procesador LEON3

- **Pipeline avanzado de 7 etapas.**
- Unidades de hardware para multiplicación, división y MAC.
- Caches de datos e instrucciones.
- Caches configurables: 1 - 4 sets, 1 - 256 kbytes/set.
- **MMU opcional.**
- Interfase principal conforme al bus AMBA-2.0 AHB.
- Soporte avanzado para debug y trace buffer para datos e instrucciones.
- **Soporte para multiprocesadores simétricos (SMP).**
- Diseño completamente sincrónico con flanco de reloj simple.
- Frecuencias hasta 125 MHz en FPGAs y 400 MHz en ASIC de 0,13 μm .



El procesador LEON y la GRLIB

La GRLIB

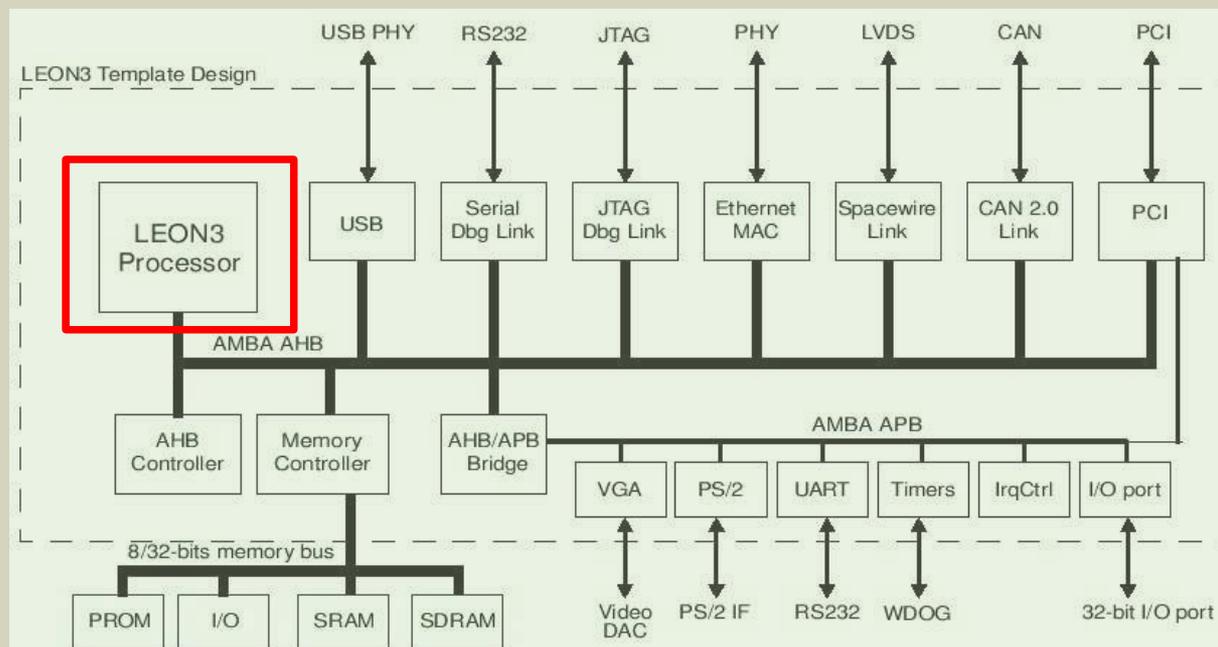


Instituto
Nacional
de Tecnología
Industrial



Introducción a la GRLIB

- La GRLIB es un conjunto organizado y configurable de IP Cores.
- El núcleo central es el procesador LEON3.
- Busca siempre utilizar un VHDL neutral para mantener compatibilidad con distintas tecnologías de FPGAs y ASICs.



El procesador LEON y la GRLIB

La GRLIB



Instituto
Nacional
de Tecnología
Industrial



IP-cores de la GRLIB

IP-Cores Bajo GPL

- **Procesador LEON-3 y su unidad de debug.**
- **PCI Master y Target, con DMA.**
- **Ethernet MAC 10/100.**
- **Controladores de memorias RAM y ROM, SDRAM PC133, DDR2, DDR266.**
- Controlador de interrupciones y multiplicador de 32x32 bits.
- Unidades de debug por JTAG, Ethernet y serie.
- Trace buffers y Analizador lógico dentro del chip.
- Puente entre los buses AHB y APB.
- Teclado PS/2, UART, ATA, CAN (OpenCores), temporizadores, FIFO y E/S de propósitos generales. Controlador VGA y SVGA.
- I²C maestro y esclavo, SPI y JTAG.
- PacketWire.

Solo bajo licencia paga

- **Versión FT del procesador.**
- **Unidad de punto flotante FPU IEEE-754.**
- **Controlador de HOST USB 2.0.**
- **Debug USB.**
- **Gigabit Ethernet.**
- Monitor de bus AMBA.
- Manejo de memorias EDAC (Error Detection and Correction).
- Cores de criptografía.
- Comunicación SpiceWire.
- Controlador de bus CAN con DMA.
- Interfase ADC y DAC combinadas.
- IP cores para telemetría y telecomando espacial.



El procesador LEON y la GRLIB

La GRLIB



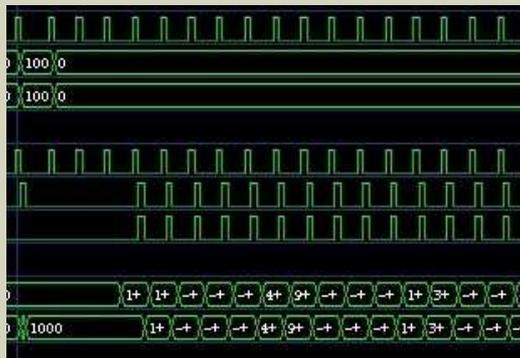
Instituto
Nacional
de Tecnología
Industrial



Simulación, Síntesis y P&R

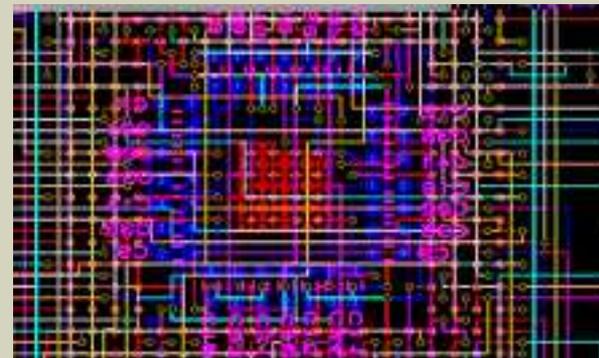
Simulación

- **GNU VHDL Simulator (GHDL)**
- Cadence NCsim
- Modelsim VHDL and Verilog simulator
- Active-HDL VHDL and Verilog simulator
- Riviera VHDL and Verilog simulator
- Symphony-EDA Sonata VHDL simulator



Síntesis y Place & Route

- Actel Designer/Libero
- Altera Quartus
- Cadence RTLIC
- Lattice ispLEVER
- Mentor Leonardo Precision
- Synopsys DC
- Synplify
- Xilinx ISE/XST
- eASIC eTools



El procesador LEON y la GRLIB

La GRLIB



Instituto
Nacional
de Tecnología
Industrial



Estimación de área

Block	Virtex2				RTAX		Proasic3		ASIC
	Pins	LUT	RAMB 16	MULT18	Cells	RAM 64K36	Tiles	RAM 4K9	Gates
AHBCTRL	2	500			1000		1500		2000
AHBJTAG	4	120			350		550		200
AHBUART (DSU UART)	2	450			800		1200		2000
APBCTRL		150			200		300		800
APBPS2	4	450			800		1300		2000
APBUART	4	200			300		450		1000
APBVGA	30	250	5						1400
CAN_OC (CAN-2.0 core with APB I/F)	4	1600	2		2800	2	4000	2	5000
DIV32 (64/32-bit iterative divider)		400			500		1400		2000
EDCL (Ethernet DSU + 4 Kbyte buffer)	15	5200	5						25000
ETH_OC (10/100 bit ethernet MAC)	15	2900							14000
GPTIMER (16-bit scaler + 2x32-bit timers)	2	250			400		600		1300
GRFPU-Lite including LEON3 controller		4000	6		7000	4	12000	4	35000
GRFPU IEEE-754 floating-point unit		9000		16					100000
GRFPC for LEON3		5000	4						25000
GRGPIO, 16-bit configuration	16	100			150		150		800
IRQMP (1 processor)		300			350		600		1500
LEON3, 8 + 8 Kbyte cache	1	4300	12		6500	40	10500	40	20000
LEON3, 8 + 4 Kbyte cache	1	4300	10		6500	31	10500	31	20000
DSU3 debug unit	3	700			1000		1000		5000
DSU3 Instruction trace buffer		150	4		300	4	400	4	600
DSU3 AHB trace buffer		100	4		200	4	300	4	500
LOGAN, 32 chan. 1024 traces, 1 trigger		300	2		500	8	1000	8	
MCTRL	64	350			1000		800		1500
MCTRL including SDRAM support	70	600			1400		1800		2000
MUL32 (32x32 multiplier, 4-cycle iterative)		200		1	1400		1900		5500
PCI_TARGET, simple PCI target	50	150			500		600		800
PCI_MTF, master/target PCI with FIFO	54	1100	4		2000	4	3100	8	6000
PCIDMA, master/target PCI with FIFO/DMA	54	1800	4		3000	4	4500	8	9000
PCITRACE	54	300	2		600	2	1000	3	1400
SRCTRL	64	100			200		300		500
SDCTRL	60	300			600		800		1200

FPU: 9000 LUTs
Ethernet EDCL: 5200 LUTs
LEON3 : 4300 LUTs
PCI DMA Master: 1800 LUTs
CAN_OC: 1600 LUTs

**) Información provista por el proveedor en su sitio web.
 **) Se marcan en rojo los ip-cores no disponibles bajo GPL.*



El procesador LEON y la GRLIB

La GRLIB



Instituto
Nacional
de Tecnología
Industrial



Sistemas operativos soportados por la GRLIB

Varios sistemas han sido portados al LEON/Grlib:

Libres/Abiertos y gratuitos:

- Linux 2.6: <http://www.kernel.org/>
uClinux (2.0): <http://www.uclinux.org/>
RTEMS: <http://www.rtems.com/>
ecOS: <http://ecos.sourceware.org/>



Cerrados/pagos:

- VxWorks (Wind River)
- Nucleus (Mentor Graphics)
- ThreadX (Express Logix)

*Las URLs se dan como referencias, bajar siempre las versiones suministradas por Aeroflex/Gaisler.



El procesador LEON y la GRLIB

La GRLIB



Instituto
Nacional
de Tecnología
Industrial



Templates para tarjetas con FPGA

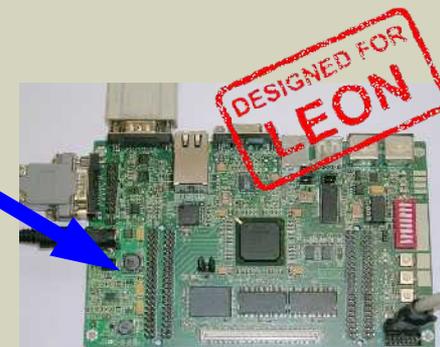
La GRLIB viene con diseños ya adaptados a varias placas de desarrollo con FPGA, listos para usar:

- actel-coremp7
- altera-ep1c20
- altera-ep2s60-ddr
- altera-ep2s60-sdr
- altera-ep3c25
- avnet-3s1500
- avnet-eval-xc4v
- avnet-ml401
- avnet-xc2v1500
- digilent-xc3s1000
- digilent-xc3s1600e
- digilent-xup
- ge-compact-ep1s40
- ge-compact-ep2s180
- ge-compact-ep2s60
- ge-hpe-midi-ep2s180



**Virtex
XC4VLX25.**

- ge-hpe-mini
- ge-hpe-mini-lattice
- gr-cpci-ax
- gr-cpci-xc2v6000
- gr-cpci-xc4v
- gr-pci-xc2v3000
- gr-pci-xc5v
- gr-xc3s-1500
- hardi-haps31
- hardi-haps51
- hardi-haps52
- hardi-haps54
- jopdesign-ep1c12
- memec-v2mb1000
- nuhorizons-3s1500
- xilinx-ml501
- xilinx-ml505



Spartan XC3S1500

El procesador LEON y la GRLIB

La GRLIB

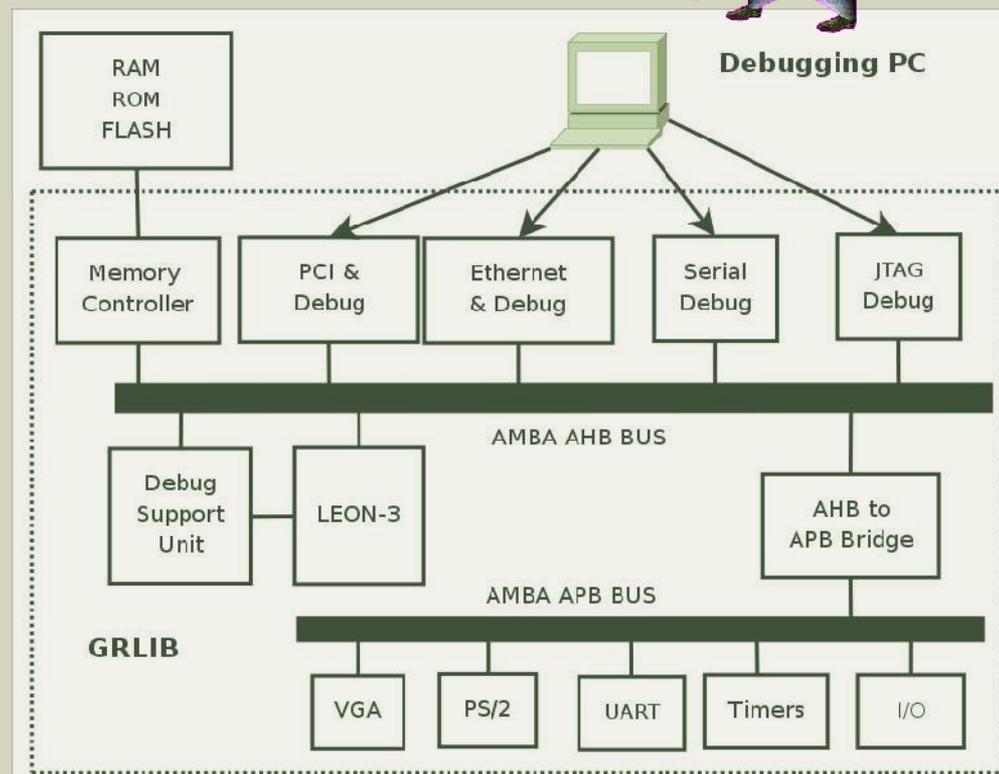


Instituto
Nacional
de Tecnología
Industrial



Depuración de la GRLIB

- La GRLIB tiene IP-cores especiales para debug mediante USB, Serial, Ethernet, PCI, JTAG y más. (no todos son libres).
- Estos IP-cores dan acceso al bus AHB para realizar operaciones de lectura y escritura.



El procesador LEON y la GRLIB

La GRLIB



Instituto
Nacional
de Tecnología
Industrial

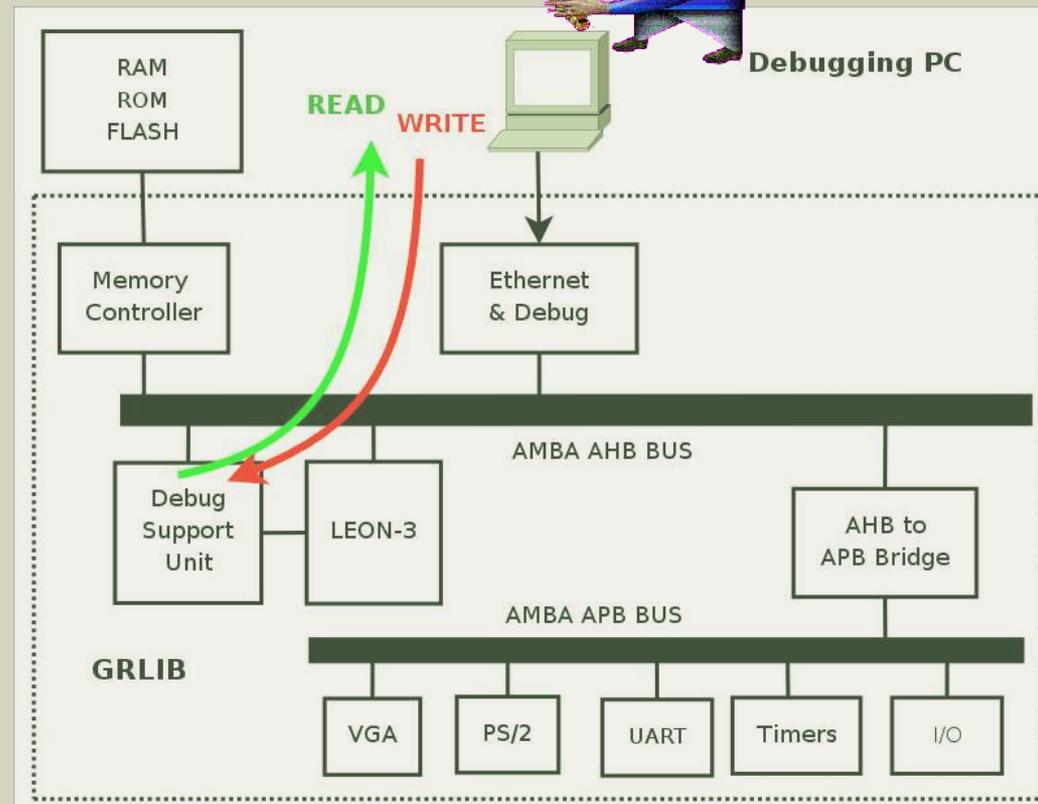


Depuración de la GRLIB

- Se puede depurar el LEON-3 a través de la DSU (Debug Support Unit).



Debugging PC



Ejemplo: Debug del procesador a través del core Ethernet



El procesador LEON y la GRLIB

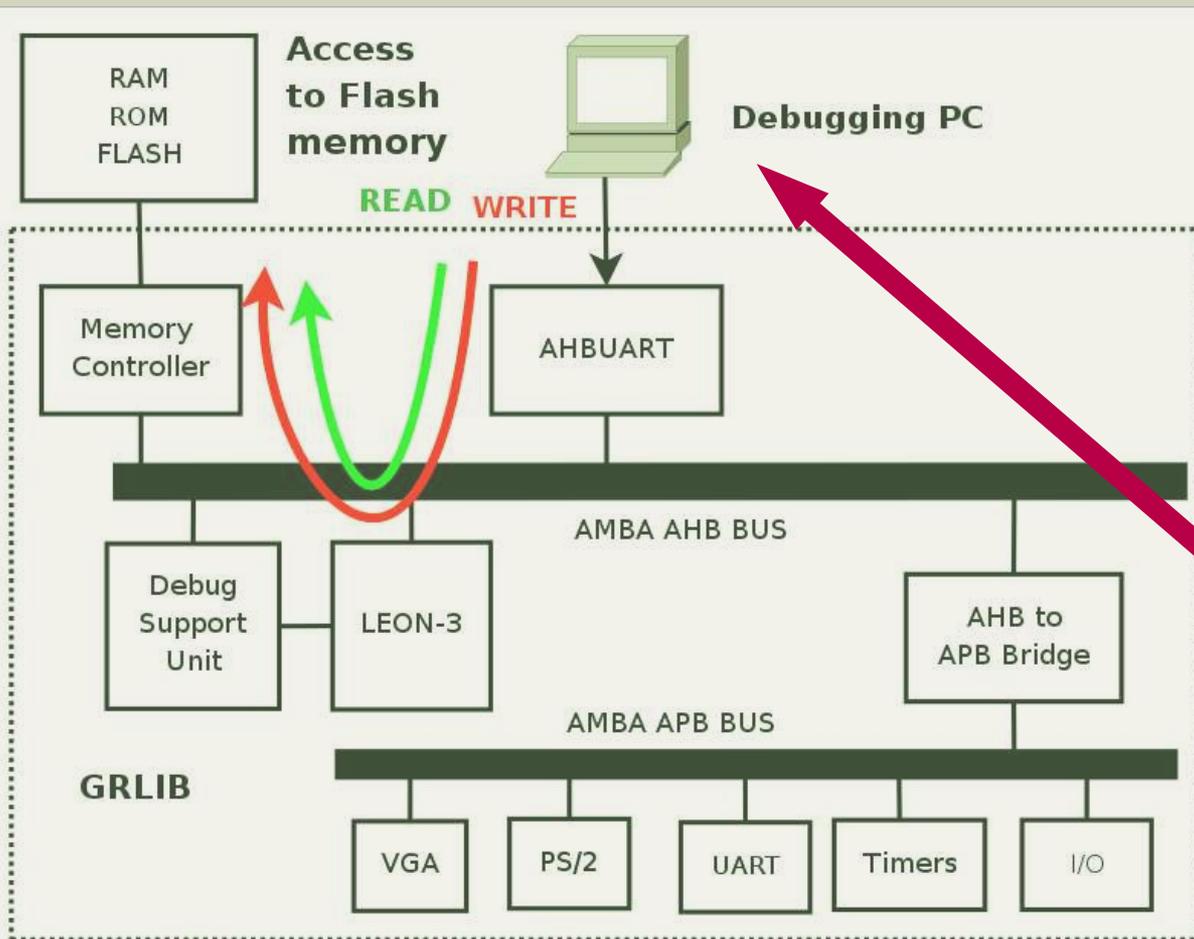
La GRLIB



Instituto
Nacional
de Tecnología
Industrial



Software para depuración de la GRLIB



- Este mecanismo es de utilidad para grabar una imagen ejecutable de un sistema GNU/Linux en la FLASH.

Software disponible:

- **GRMON**
- **FPGALibre LEON Monitor**



El procesador LEON y la GRLIB

La GRLIB

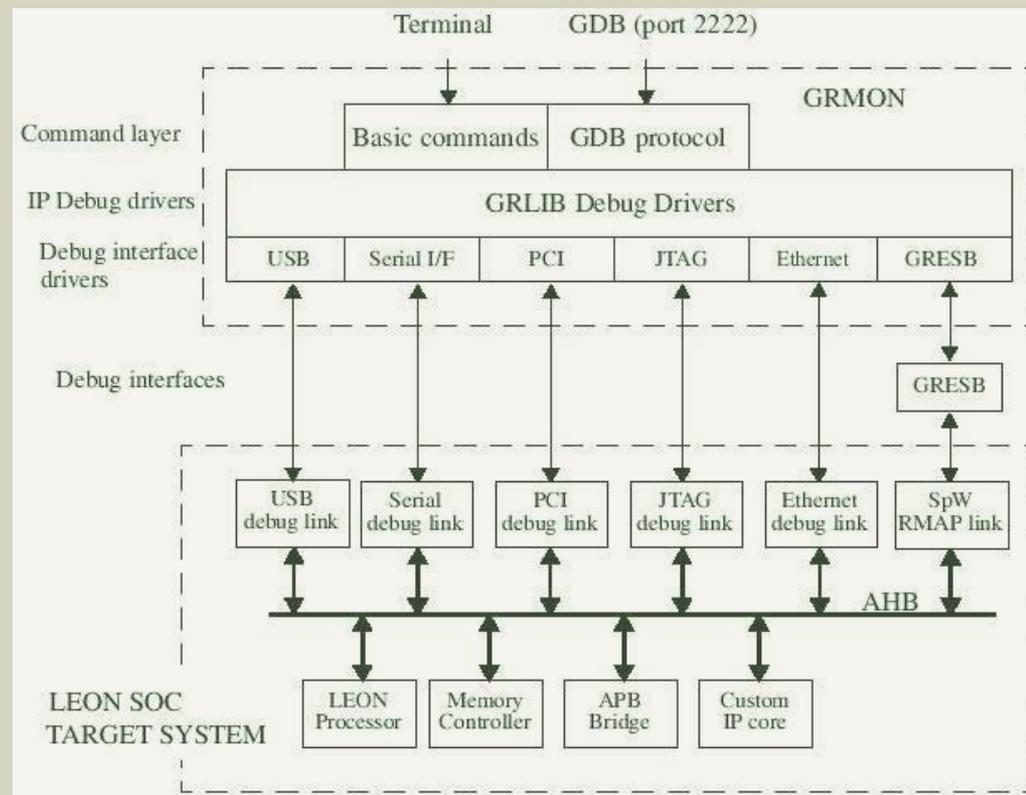


Instituto
Nacional
de Tecnología
Industrial



Software GRMON para depuración

- GRMON es un software que permite comunicarse con el bus AHB y realizar funciones de alto nivel con los IP-cores de la GRLIB.
- Todos los cores de Debug.
- Debug del procesador.
- Leer y escribir RAM y FLASH.
- Mucho mas...
- **ES PROPIETARIO. Se brindan versiones gratuitas de evaluación, para fines no-comerciales y solamente por 21 días.**



El procesador LEON y la GRLIB

La GRLIB

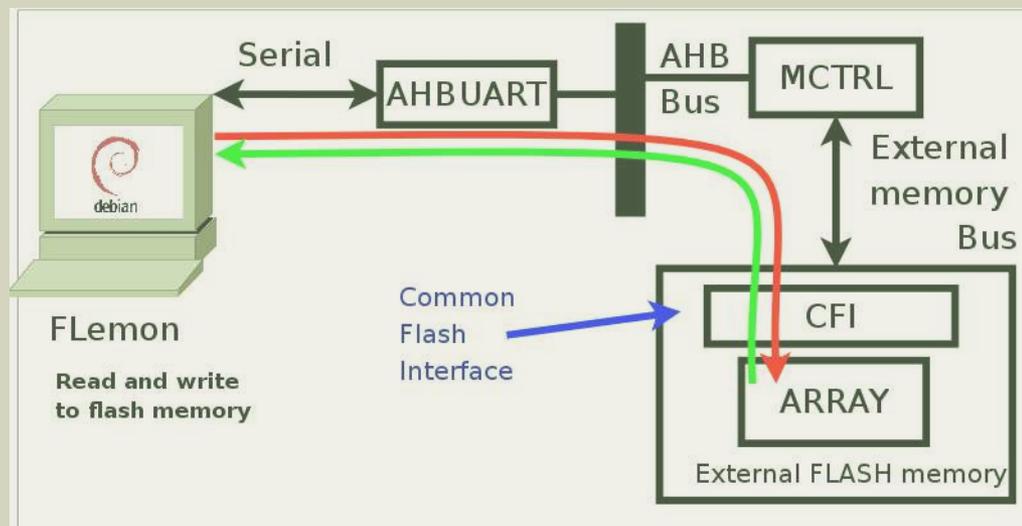


Instituto
Nacional
de Tecnología
Industrial



FPGALibre LEON Monitor

- FLeMon fue desarrollado como parte del proyecto FPGALibre debido a los inconvenientes de GRMON en su versión gratuita.
- El primer objetivo fue poder grabar una imagen del sistema operativo GNU/Linux en la memoria Flash.
- Licencia GPL.



<http://fpgalibre.sf.net>

FPGA Libre

Capacidades actuales:

- Solo puerto serie (AHBUART)
- Lectura, borrado, escritura y verificación de la memoria FLASH (Intel CFI).





CASOS DE APLICACIÓN



El procesador LEON y la GRLIB

Casos de aplicación

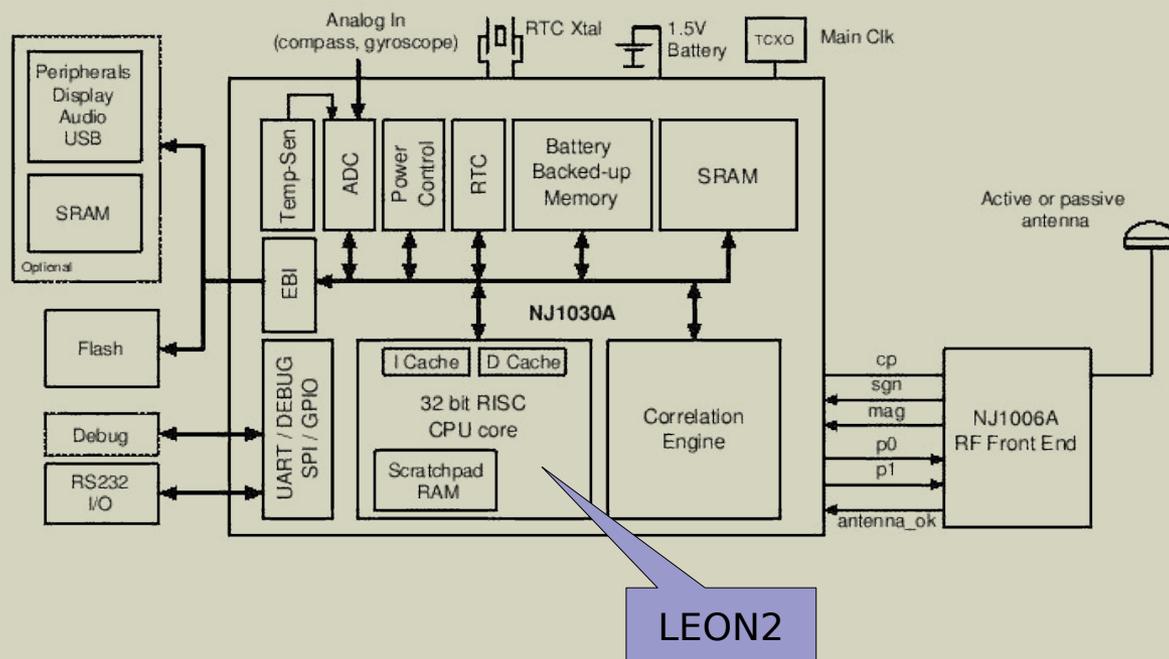
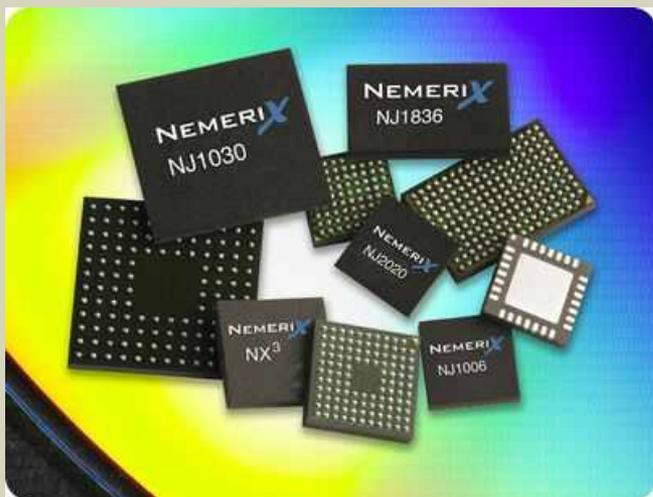


Instituto
Nacional
de Tecnología
Industrial



Procesador para GPS de NEMERIX

- El Nemerix NJ1030A es circuito integrado para procesar señales de banda base de GPS y entregar mensajes NMEA. Utiliza internamente un procesador LEON2.



El procesador LEON y la GRLIB

Casos de aplicación



Instituto
Nacional
de Tecnología
Industrial



Otros casos

Procesadores rad hard de ATMEL. (LEON2).



Procesador para discos de red de INFRANT (LEON1).



Chips para cajas registradoras y puntos de venta de la empresa Orbita Control Engineering (LEON2).





PARTE 2

Ejemplo de uso de la GRLIB con GNU/Linux



El procesador LEON y la GRLIB

Ejemplo de uso con GNU/Linux



Instituto
Nacional
de Tecnología
Industrial



Hardware utilizado

FPGA:

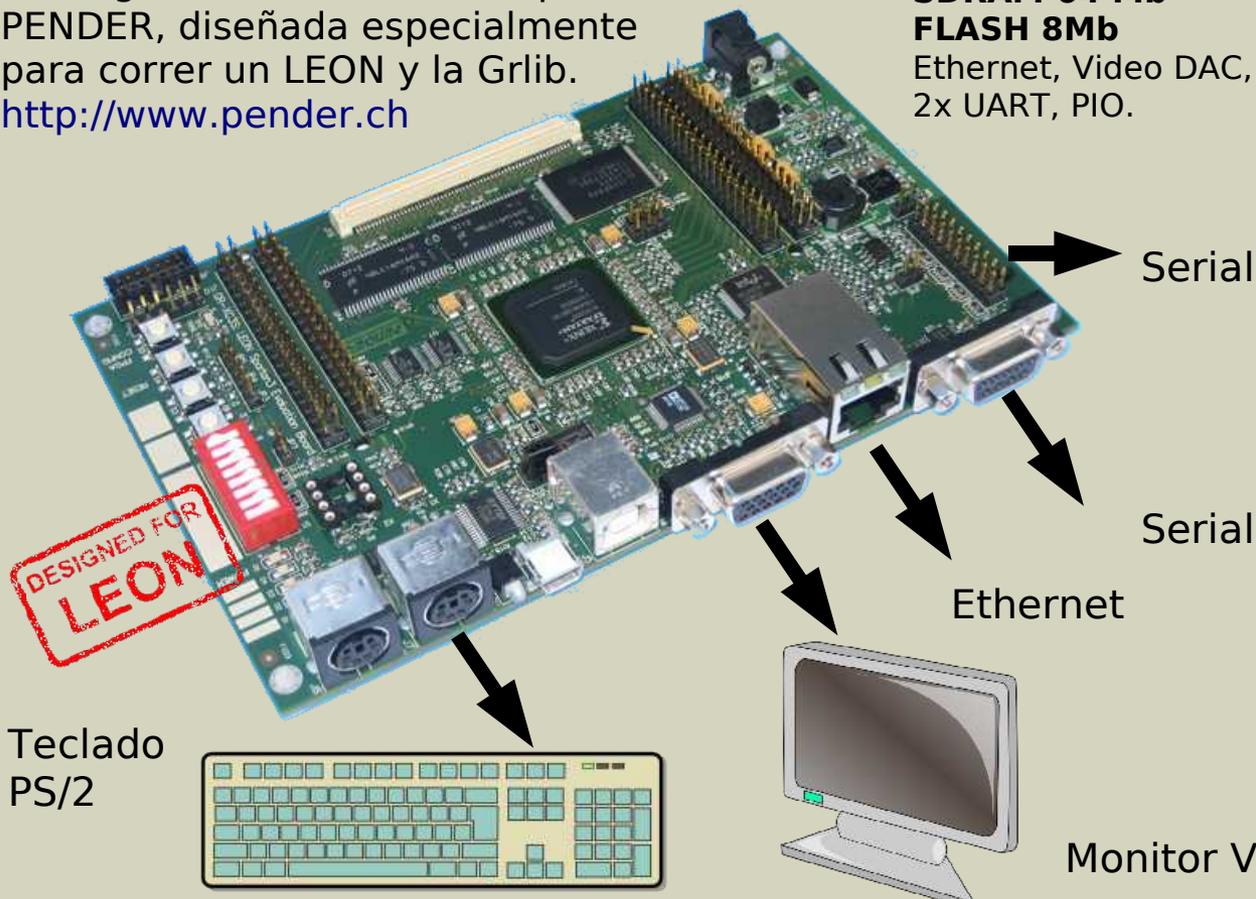
Placa gr-xc3s-1500 de la empresa PENDER, diseñada especialmente para correr un LEON y la Grlib.
<http://www.pender.ch>

Xilinx Spartan3, XC3S1500 FPGA.

SDRAM 64 Mb

FLASH 8Mb

Ethernet, Video DAC, USB, 2x PS/2,
2x UART, PIO.



Estación de trabajo con:

- Puerto paralelo (JTAG).
- Puerto serie 1 (Debug).
- Puerto serie 2 (Consola Linux Embebido).
- Ethernet.



El procesador LEON y la GRLIB

Ejemplo de uso con GNU/Linux



Instituto
Nacional
de Tecnología
Industrial



Fuentes y software para la GRLIB



- Sistema operativo Debian GNU/Linux
- Xilinx ISE Webpack
- Fuentes de la Grlib.
- GHDL (Si queremos correr el testbench).
- BASH, GNU Make, GCC, TCL/tk

- Descargar la Grlib (código VHDL)
- Descomprimir con unzip.
- Entrar al subdirectorio de diseño según nuestra placa.

```
$ wget  
  http://www.gaisler.com/products/grlib/grlib-gpl-1.0.22-b4075.zip  
  
$ unzip grlib-gpl-1.0.22-b4075.zip  
  
$ cd grlib-gpl-1.0.22-b4075/designs/leon3-gr-xc3s-1500/
```



El procesador LEON y la GRLIB

Ejemplo de uso con GNU/Linux



Instituto
Nacional
de Tecnología
Industrial



Configurando la GRLIB

Configurar la Grlib significa elegir que ip-cores incluimos o no, y seleccionar algunos parámetros de cada uno.

The image shows two windows from the LEON3MP Design Configuration tool. The top window is the 'LEON3MP Design Configuration' main menu, which has a yellow callout bubble pointing to it with the text 'Ventana principal'. The bottom window is the 'Clock generation' sub-window, which is open and shows various configuration options for the clock generator. A red arrow points from the 'Clock generation' option in the main menu to the sub-window. Below the main menu, there is a terminal window with the command '\$ make xconfig'.

Ventana principal

```
$ make xconfig
```



El procesador LEON y la GRLIB

Ejemplo de uso con GNU/Linux



Instituto
Nacional
de Tecnología
Industrial



Verificación de la GRLIB

Paso 1: Compilando el testbench con GHDL

```
$ make ghdl-clean  
$ make ghdl
```

Esto nos crea un binario ejecutable llamado “tesbench” que realizará la simulación.

Paso 2: Ejecutando el testbench generado.

Error normal para finalizar la simulación.

```
$ ./testbench  
...  
**** GRLIB system test starting ****  
Leon3 SPARC V8 Processor  
CPU#0 register file  
CPU#0 multiplier  
CPU#0 radix-2 divider  
CPU#0 cache system  
CPU#0 memory management unit  
Multi-processor Interrupt Ctrl.  
Modular Timer Unit  
timer 1  
timer 2  
chain mode  
Generic UART  
GR Ethernet MAC  
Test passed, halting with IU error mode  
  
testbench.vhd:375:6:@2357898ns:  
(assertion failure): *** IU in error  
mode, simulation halted ***  
./testbench:error: assertion failed
```

PASO!



El procesador LEON y la GRLIB

Ejemplo de uso con GNU/Linux



Instituto
Nacional
de Tecnología
Industrial



Síntesis y Place & Route

En nuestro caso se usa XST del ISE Webpack 8

```
$ make ise
...
Release 8.2.02i - Bitgen I.33
Copyright (c) 1995-2006 Xilinx, Inc. All rights reserved.
Loading device for application Rf_Device from file '3s1500.nph' in environment
/usr/local/ISEWb8.
    "leon3mp" is an NCD, version 3.1, device xc3s1500, package fg456, speed -4
Opened constraints file leon3mp.pcf.

Fri Feb 26 10:37:12 2010

Saving ll file in "leon3mp.ll".
Creating bit map...
Saving bit stream in "leon3mp.bit".
Saving bit stream in "leon3mp.rbt".
Saving Readback bit file leon3mp.rbb.
Saving Readback bit file leon3mp.rba.
Saving Readback golden data file leon3mp.rbd.
Saving mask data in "leon3mp.msd".
Creating bit mask...
Saving mask bit stream in "leon3mp.msk".
Bitstream generation is complete.
```

Bitstream listo para transferir por JTAG!



El procesador LEON y la GRLIB

Ejemplo de uso con GNU/Linux



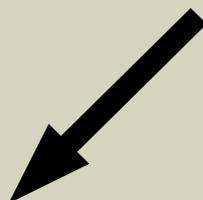
Instituto
Nacional
de Tecnología
Industrial



Software para compilar el sistema operativo embebido

En nuestro ejemplo, usaremos:

- Un sistema con MMU.
- Un sistema sin FPU.
- Un kernel 2.6 (2.6.21.1) con Gnu Libc.
- Snapgear Linux.



Software adicional necesario en la estación de trabajo:

- Toolchain para cross-compilar (Versión Gaisler)
- Fuentes de Snapgear Linux (Versión Gaisler).
- TCL/tk.
- Libqt3.

Para kernel 2.6 + GNU LibC (con MMU):

Toolchain: sparc-linux-3.4.4-2.6.21-1.0.1.tar.bz2

Snapgear: snapgear-2.6-p41.tar.bz2

La "p" es de Parches aplicados por Gaisler 

El procesador LEON y la GRLIB

Ejemplo de uso con GNU/Linux



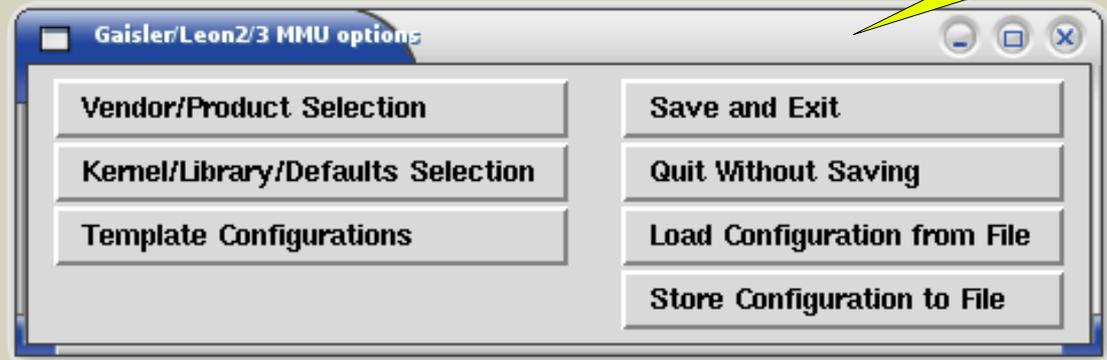
Instituto
Nacional
de Tecnología
Industrial



Configurando Snapgear Linux

- Ingresar al directorio principal y ejecutar “make xconfig”.

```
$ make xconfig
```



El procesador LEON y la GRLIB

Ejemplo de uso con GNU/Linux



Instituto
Nacional
de Tecnología
Industrial



Configurando Snapgear Linux

Ejemplo:

Para usar la consola Framebuffer, el bootloader debe informar al kernel al pasarle al control.

```
console=tty0 video=grvga:640x480@60,8,307200
```

Para booteo con consola serie:

```
console=ttyS0,38400
```



El procesador LEON y la GRLIB

Ejemplo de uso con GNU/Linux



Instituto
Nacional
de Tecnología
Industrial

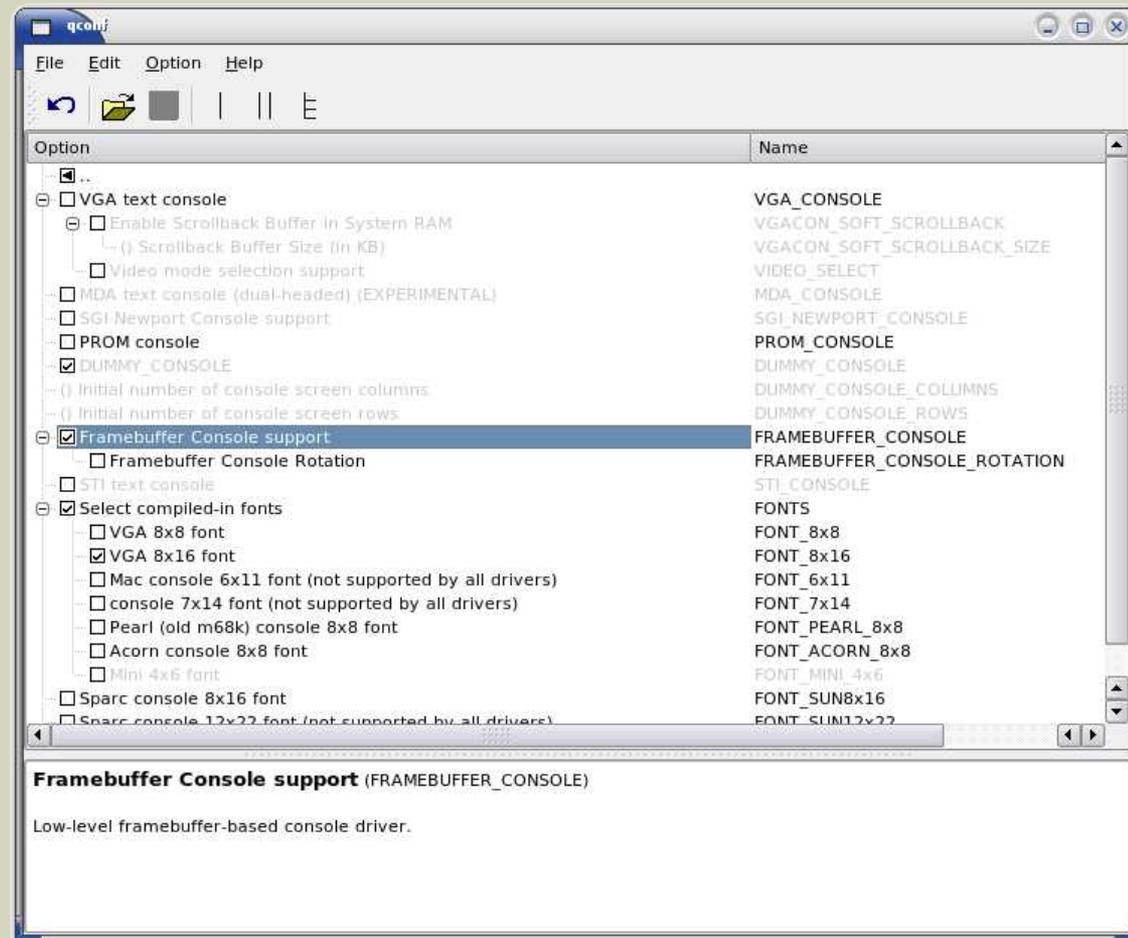


Configurando Snapgear Linux - Kernel

Si elegimos configurar el kernel, al salir de la primer ventana, aparecerá una nueva ventana para tal propósito.

Ejemplo:

Se habilita el soporte para Framebuffer y las fuentes de 8x16.



El procesador LEON y la GRLIB

Ejemplo de uso con GNU/Linux



Instituto
Nacional
de Tecnología
Industrial



Configurando Snapgear Linux - BusyBox

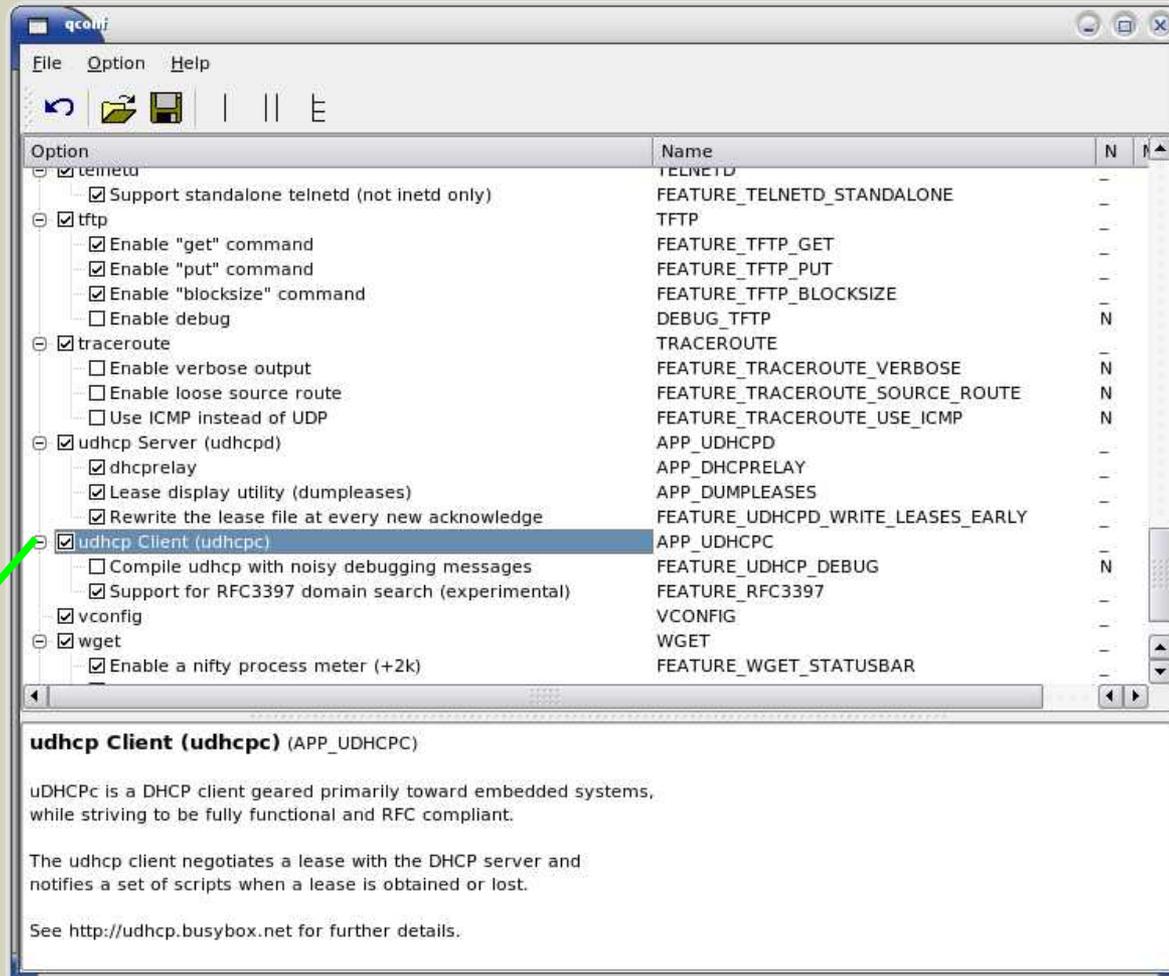
La mayoría de las aplicaciones habilitadas por defecto pertenecen a Busy-box.

<http://busybox.net/>

Si elegimos configurar BusyBox, al salir de la ventana anterior, accedemos a esta nueva configuración.

Ejemplo:

Habilitamos el cliente de DHCP (udhcp).



El procesador LEON y la GRLIB

Ejemplo de uso con GNU/Linux



Instituto
Nacional
de Tecnología
Industrial



Resumen de algunas configuraciones modificadas

- **Framebuffer, teclado y consola.**
- **IP obtenida por DHCP.**
- **Incluir un programa en C propio.**
- **Servidor Web + páginas propias.**
- **Ciente NFS.**
- **Sincronización de hora.**



El procesador LEON y la GRLIB

Ejemplo de uso con GNU/Linux



Instituto
Nacional
de Tecnología
Industrial



Compilando Snapgear Linux

Finalizada la configuración, compilamos el kernel, las aplicaciones y generamos la imagen del filesystem.

```
$ make
...
$ ls -lh images/image.flashbz
-rwxr-xr-x 1 diego diego 3,1M feb 26 16:13 images/image.flashbz
```

Obtenemos un archivo ELF con la imagen del filesystem comprimido lista para grabarla a la memoria FLASH de nuestro sistema.



El procesador LEON y la GRLIB

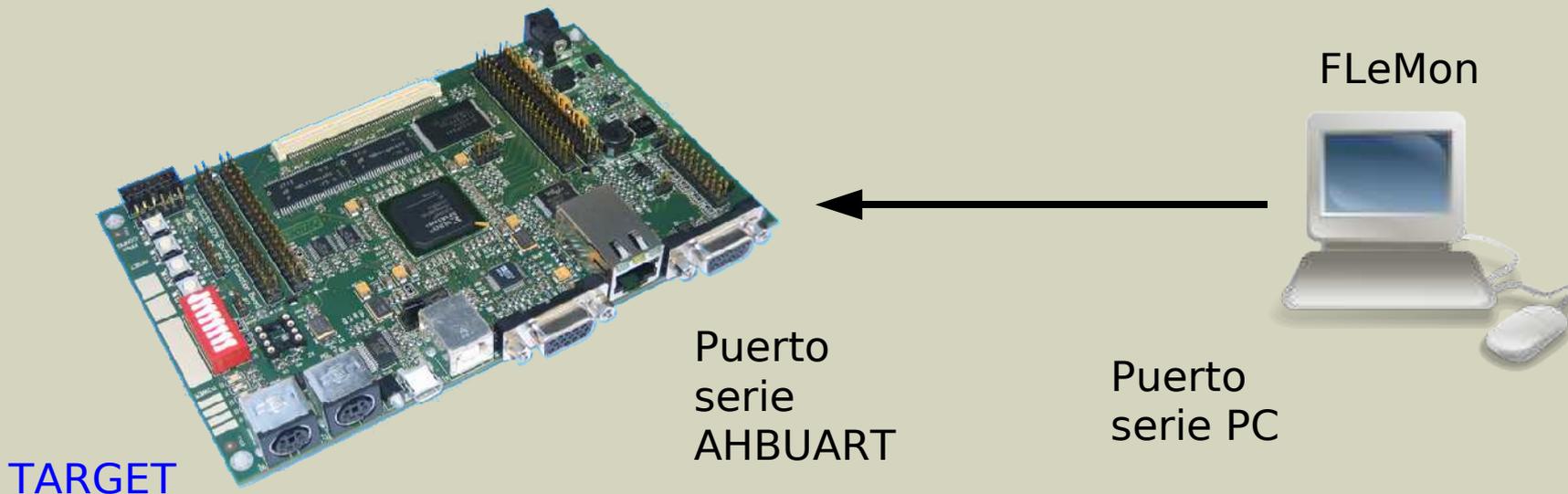
Ejemplo de uso con GNU/Linux



Instituto
Nacional
de Tecnología
Industrial



Transfiriendo la imagen Linux a la flash



En nuestro ejemplo, para transferir la imagen Linux, utilizaremos la unidad de debug por línea serie (AHBUART), un puerto serie de la PC y el software libre “FPGALibre LEon MONitor”.



El procesador LEON y la GRLIB

Ejemplo de uso con GNU/Linux



Instituto
Nacional
de Tecnología
Industrial



Transfiriendo la imagen Linux - FPGALibre LEON Monitor

Paso1: Establecer comunicación con el target.

Paso2: Listar ip-cores presentes en el sistema (solo para verificación).

Paso3: Borrar la memoria flash.

Paso4: Grabar la memoria FLASH con la imagen.

```
diego@Demeter:~$ flemon -D /dev/ttyS0
```

```
FLeMon$ detect
```

```
FLeMon$ ls
```

```
FLeMon$ fxerase images/image.flashbz
```

```
FLeMon$ fxload images/image.flashbz
```



El procesador LEON y la GRLIB

Ejemplo de uso con GNU/Linux



Instituto
Nacional
de Tecnología
Industrial



Resumen de pasos realizados

- 1) Fuentes de la Grlib + herramientas de síntesis.
- 2) Configuramos la Grlib.
- 3) Testbench.
- 4) Síntesis + P&R.
- 5) Cargamos el bitstream (JTAG).
- 6) Fuentes Snapgear Linux + Toolchain.
- 7) Agregamos archivos propios si hacen falta.
- 8) Configuramos el kernel + BusyBox.
- 9) Compilamos y generamos una imagen del sistema de archivos.
- 10) Transferimos a la memoria flash (FLeMon).
- 11) Reseteamos la FPGA y arranca nuestro sistema!





PRUEBAS



El procesador LEON y la GRLIB

Ejemplo de uso con GNU/Linux



Instituto
Nacional
de Tecnología
Industrial



Pruebas del sistema - Arranque

- Los mensajes de arranque del kernel salen por el dispositivo Frame-Buffer, según lo configuramos.
- Se ejecuta el cliente DHCP y ntpdate.
- Se monta el servicio remoto NFS.
- Arranca el servidor web.



El procesador LEON y la GRLIB

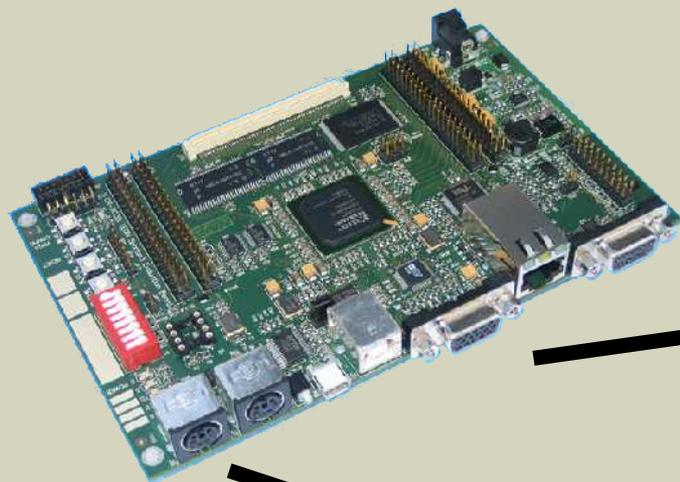
Ejemplo de uso con GNU/Linux



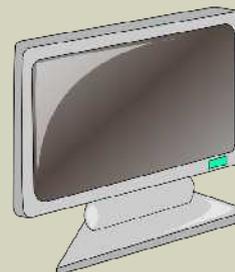
Instituto
Nacional
de Tecnología
Industrial



Pruebas del sistema - Consola Local



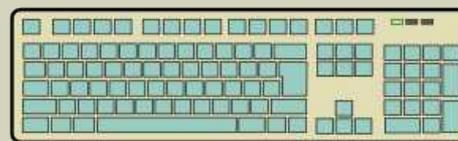
Monitor VGA



- Y aparece una línea de comandos para nuestro Linux embebido!



Usuario local



Teclado PS/2 y

El procesador LEON y la GRLIB

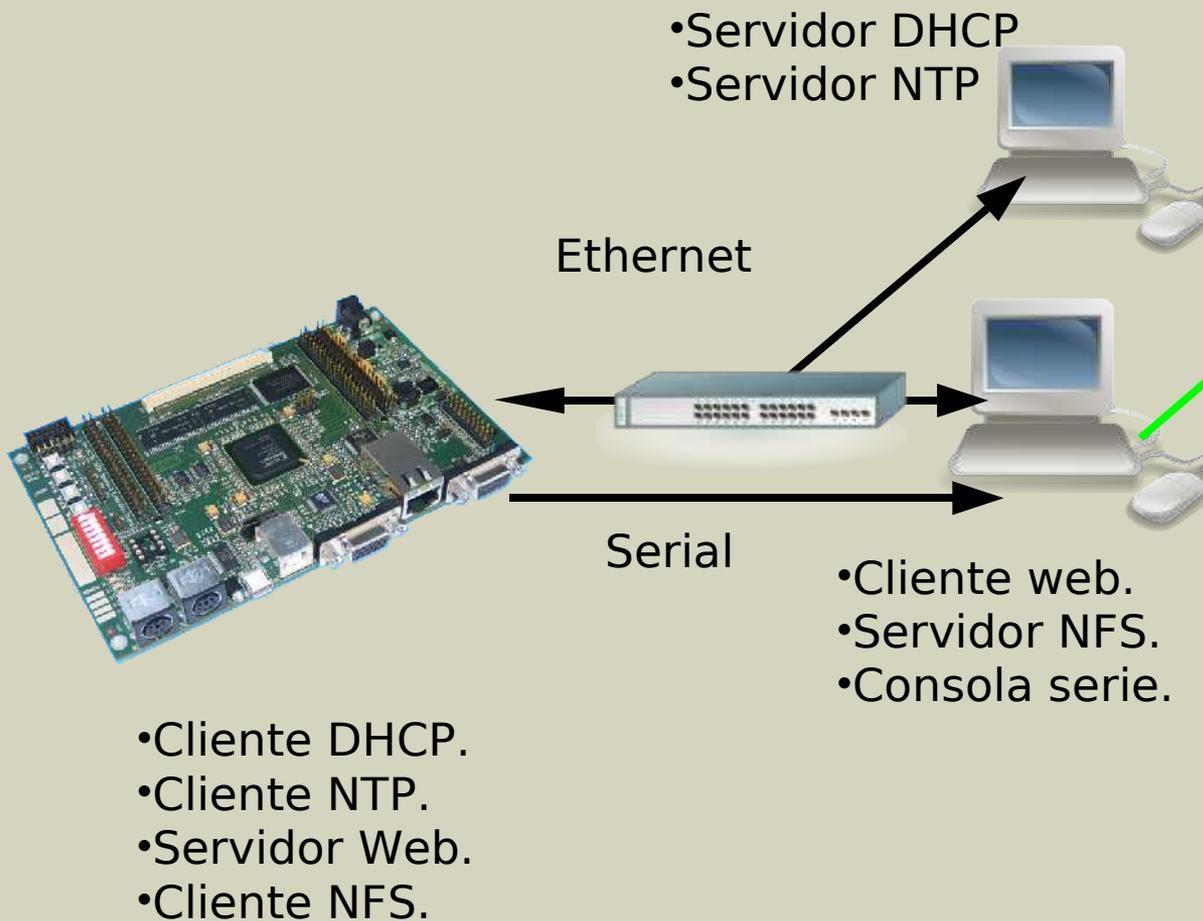
Ejemplo de uso con GNU/Linux



Instituto
Nacional
de Tecnología
Industrial



Pruebas del sistema - Servicios de red



El procesador LEON y la GRLIB

Ejemplo de uso con GNU/Linux



Instituto
Nacional
de Tecnología
Industrial



Pruebas del sistema - FrameBuffer

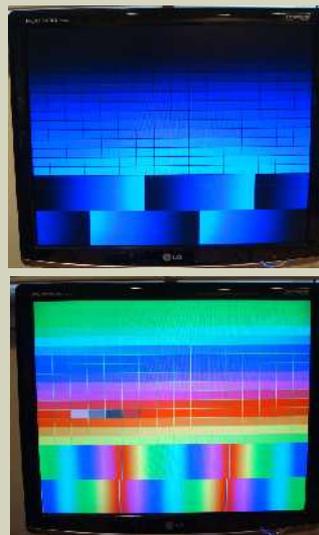
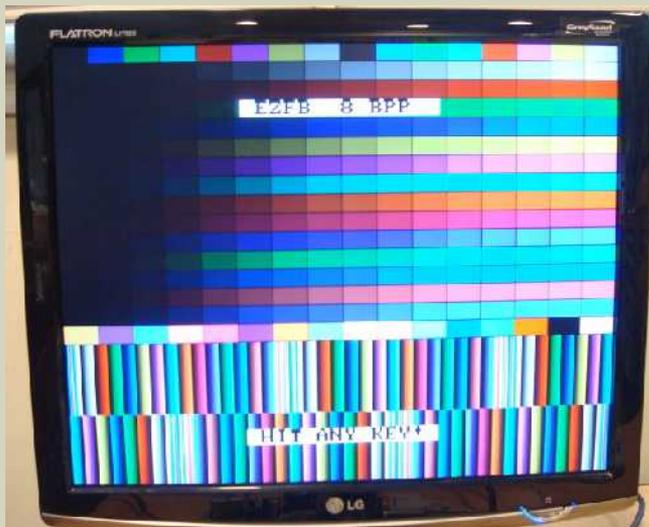
Para probar rutinas que utilicen Framebuffer, se usó:

EZFB ~ Linux Frame Buffer API

<http://www.akrobiz.com/ezfb/>



- Posee varias aplicaciones demostrativas.
- No está particularmente pensado para SPARC o para el LEON/Grlib.
- Se cross-compilan (modificando el Makefile), se colocan los binarios en el servidor NFS y se ejecutan mediante la consola local o la consola serie.

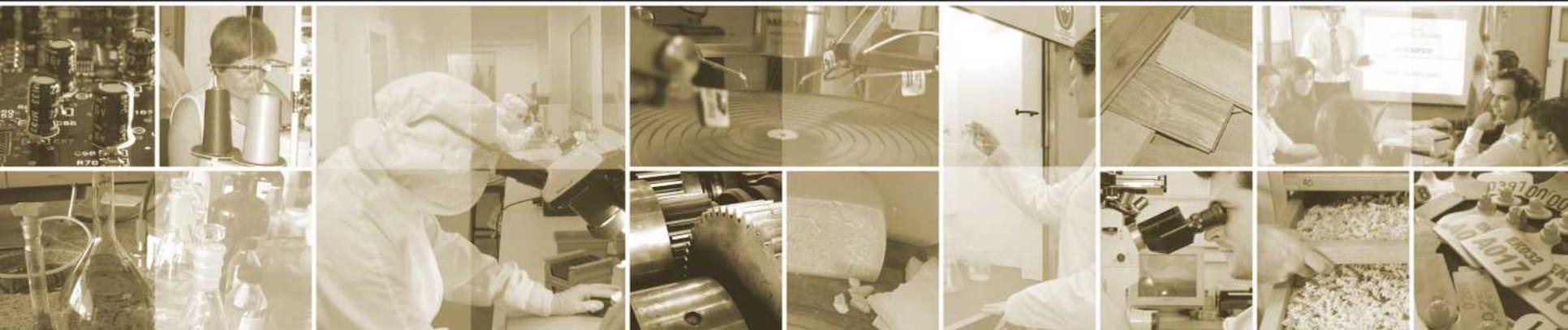




Instituto
Nacional
de Tecnología
Industrial



Ministerio de Industria y Turismo
Secretaría de Industria, Comercio
y de la Pequeña y Mediana Empresa



GRACIAS!

Av. General Paz 5445
(1650) San Martín
Buenos Aires, Argentina
(5411) 4724-6369
brengi@inti.gov.ar
salvador@inti.gov.ar
<http://utic.inti.gov.ar>
<http://fpgalibre.sf.net>

Copyright © 2010 Instituto Nacional de Tecnología Industrial
Copyright © 2010 Diego J. Brengi

Marzo de 2010

