



INTI



SASE 2014

Softcores - Microprocesadores dentro de FPGAs

Disertantes: Salvador E. Tropea¹, Diego J. Brengi²,
Rodrigo A. Melo¹

Centro de Micro y Nanoelectrónica (CMNB)
1 Unidad Técnica Sistemas Inteligentes (UTSI)
2 Unidad Técnica Integración de Sistemas (UTIS)



Ministerio de
Industria
Presidencia de la Nación



Agenda

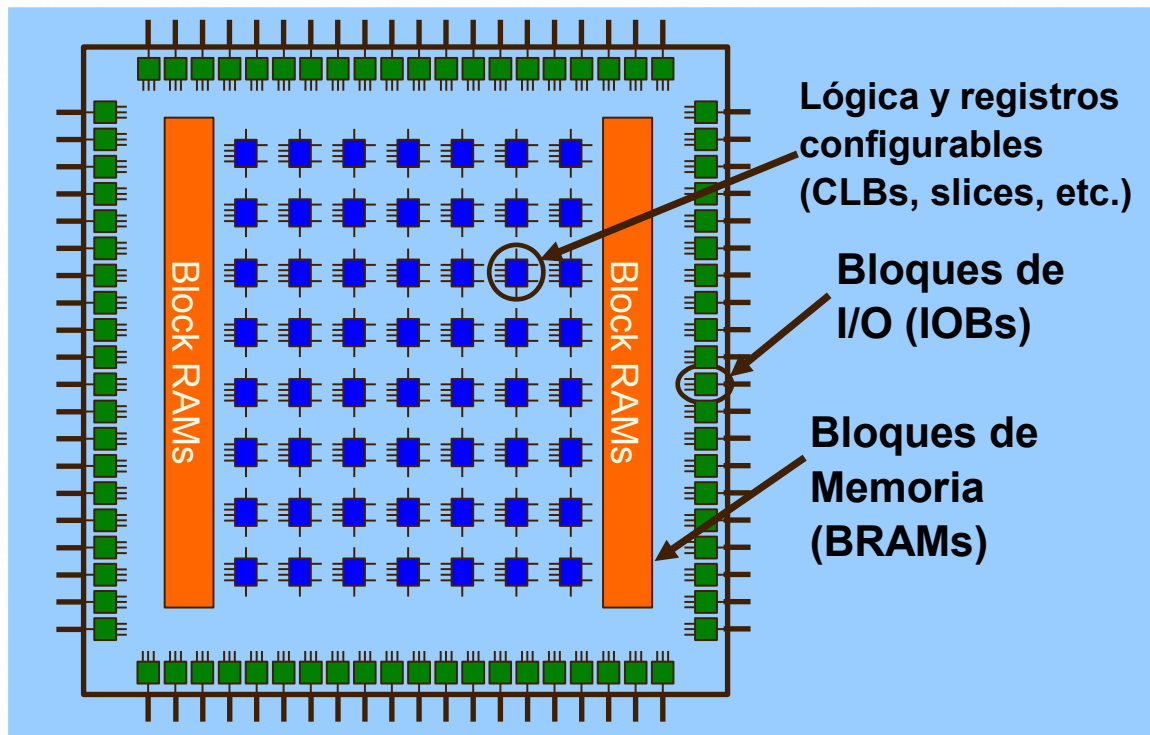
1. ¿Qué es una FPGA?
2. ¿Cuándo usar una FPGA?
3. CPUs disponibles para FPGAs
 - Hardcores
 - Softcores
 - Del fabricante
 - De terceros
 - Desarrollados in-house
4. Casos de ejemplo
 - Clon PIC16F84
 - Clon AVR

Agenda

1. **¿Qué es una FPGA?**
2. ¿Cuándo usar una FPGA?
3. CPUs disponibles para FPGAs
 - Hardcores
 - Softcores
 - Del fabricante
 - De terceros
 - Desarrollados in-house
4. Casos de ejemplo
 - Clon PIC16F84
 - Clon AVR

¿Qué es una FPGA? Elementos básicos

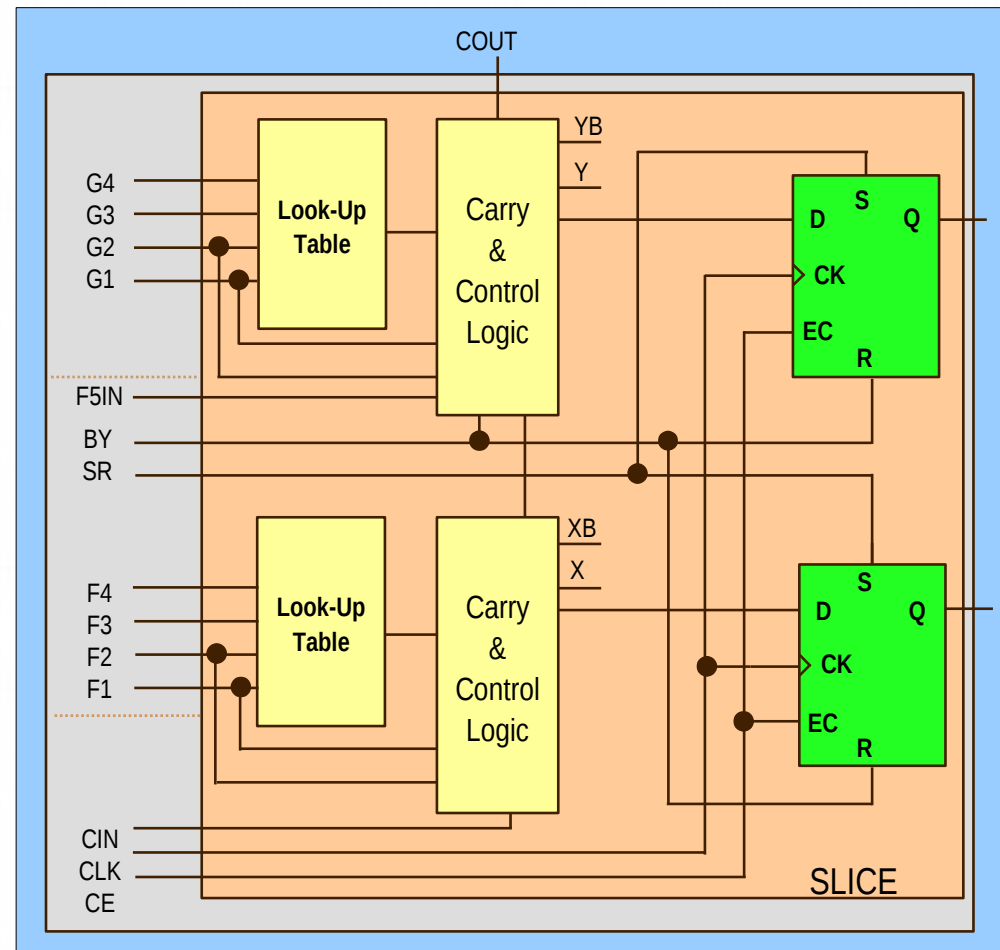
- Las FPGAs son circuitos integrados (re)configurables.
- Poseen lógica combinacional (re)configurable.
- Poseen registros (flip/flops) (re)configurables.
- Poseen un sistema poderoso de interconexión (re)configurable.



- Poseen entradas/salidas muy flexibles y (re)configurables.
- Agregan más y más funcionalidad: Bloques de memoria, multiplicadores, bloques “multiply/accumulate”, PLLs, **CPUs**, etc.

¿Qué es una FPGA? Celda básica – Terminología usada

- Celda básica: Lógica+Flip/Flop
- Casi todos los fabricantes usan lo mismo o equivalente
- Cada uno llama las cosas diferente, incluso cambia de un producto a otro.
- A los fines de esta charla usamos la terminología de Xilinx
- Lógica: LUT (look-up table), 4 entradas y 1 salida.
- Flip-flop: 1 flip-flop
- En la figura: 1 slice = 2 LUTs+2 FFs.
- Tendencia actual: grupos de 2 LUTs flexibles.
- La métrica sigue siendo esto (logic cells, etc.)



Agenda

1. ¿Qué es una FPGA?
2. **¿Cuándo usar una FPGA?**
3. CPUs disponibles para FPGAs
 - Hardcores
 - Softcores
 - Del fabricante
 - De terceros
 - Desarrollados in-house
4. Casos de ejemplo
 - Clon PIC16F84
 - Clon AVR

¿Cuándo usar una FPGA? Ventajas comparadas con un microcontrolador

- Alto grado de integración [SoC]
 - Necesidad de usar un número alto de periféricos de un mismo tipo.
Ejemplo: 100 PWMs.
 - Gran cantidad de pines disponibles
- Alta performance. Una FPGA puede superar ampliamente a un DSP o a un microprocesador de uso general (paralelismo)
 - Menor consumo en HP, sólo el hard que necesitamos implementar
- Gran flexibilidad de producto
 - Podemos actualizar no sólo el firmware ¡¡sino el hardware!!
 - Una misma placa puede servir para más de un producto, aun con hard muy distinto.
- ¡Una FPGA es un superset de un μ !
 - Arquitecturas experimentales

¿Cuándo usar una FPGA? Desventajas comparadas con un microcontrolador

- Las más baratas son más caras que los μP económicos
 - La brecha cae con los años (lentamente)
- Mayor complejidad de diseño
 - ¡Se diseña soft+hard!
 - Los μP actuales poseen muchísimos periféricos, en ocasiones no es tan simple obtenerlos en la FPGA
- Encapsulados más complejos
 - Los μP actuales también tienden a esto
- En aplicaciones de baja performance usualmente tienen mayor consumo

Opciones con FPGAs Posibles metodologías

- Desarrollo 100% en hardware
 - Máxima performance
 - Mínima flexibilidad
 - El hard **es** duro, > time to market
 - Menos gente sabe hacerlo
- Usar una o más CPUs
 - Más mano de obra disponible
 - No es necesario capacitar a gente que ya dominaba los μ Ps
 - Debo balancear que hago en hard y que en soft (co-design)

Agenda

1. ¿Qué es una FPGA?
2. ¿Cuándo usar una FPGA?
3. CPUs disponibles para FPGAs
 - **Hardcores**
 - Softcores
 - Del fabricante
 - De terceros
 - Desarrollados in-house
4. Casos de ejemplo
 - Clon PIC16F84
 - Clon AVR

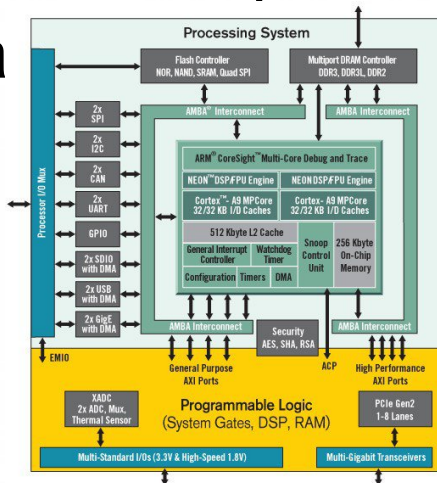


Hardcores

- Implementados en Silicio
- Máximas prestaciones
- Alto costo (por parte)
- Dependiente del fabricante
- Actuales:
 - Xilinx: Dual Cortex A9 (Zynq-7000)
 - Altera: Single/Dual Cortex A9 (Cyclone V y Arria V/10 [versiones Sx])
 - Microsemi: Cortex M3 (SmartFusion 1 y 2)
- Futuros:
 - Xilinx: 64 bits (Zynq UltraScale)
 - Altera: 64 bits - Quad Cortex A53 (Stratix 10)

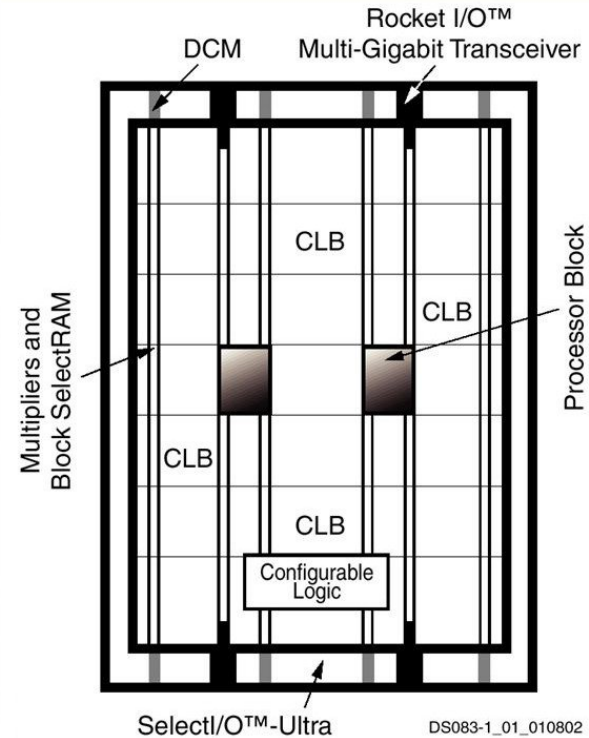
Hardcores Xilinx

- En el pasado, **FPGA con CPU**
- Basados en PowerPC 405 y 440
- Virtex II Pro (U\$S 56), Virtex 4 FX (U\$S 129), Virtex 5 FXT (U\$S 469), Virtex 6 (no) [DigiKey x 1 – mínimo]
- Mercado de alta performance y costo, dejando softcores para el resto
- Tendencia actual:



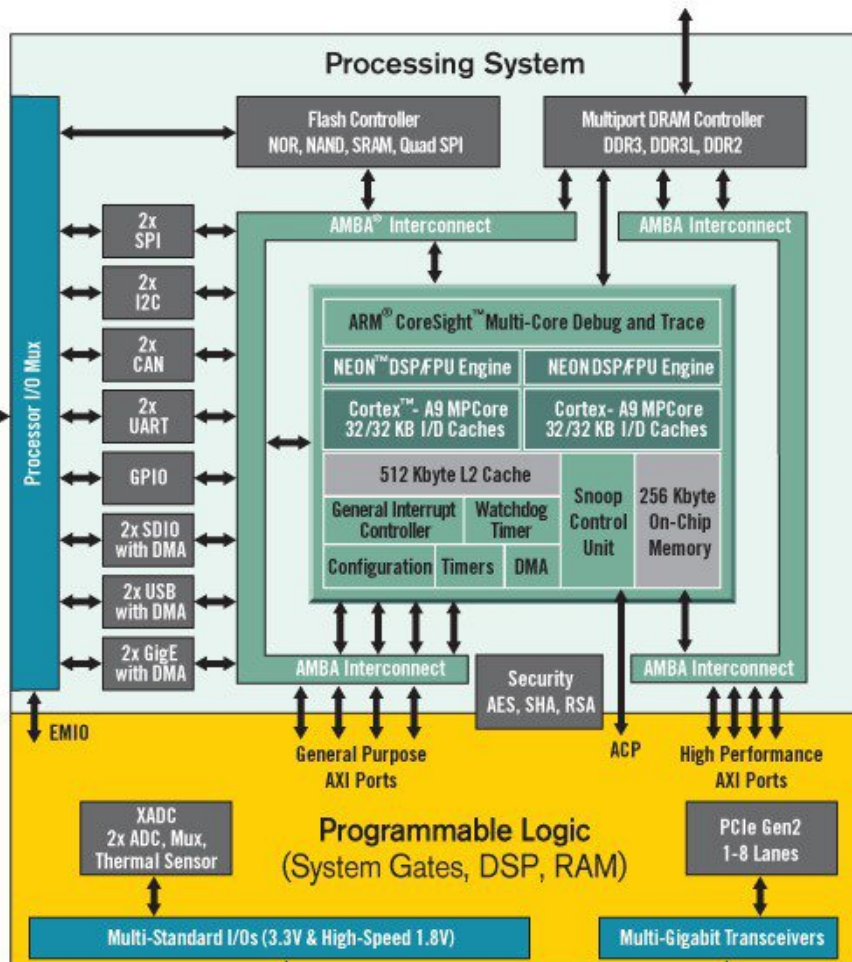
SoC

FPGA



Fuente: Virtex-II Pro Platform FPGAs: Functional Description (2002)

Hardcores – Xilinx – Zynq 7000



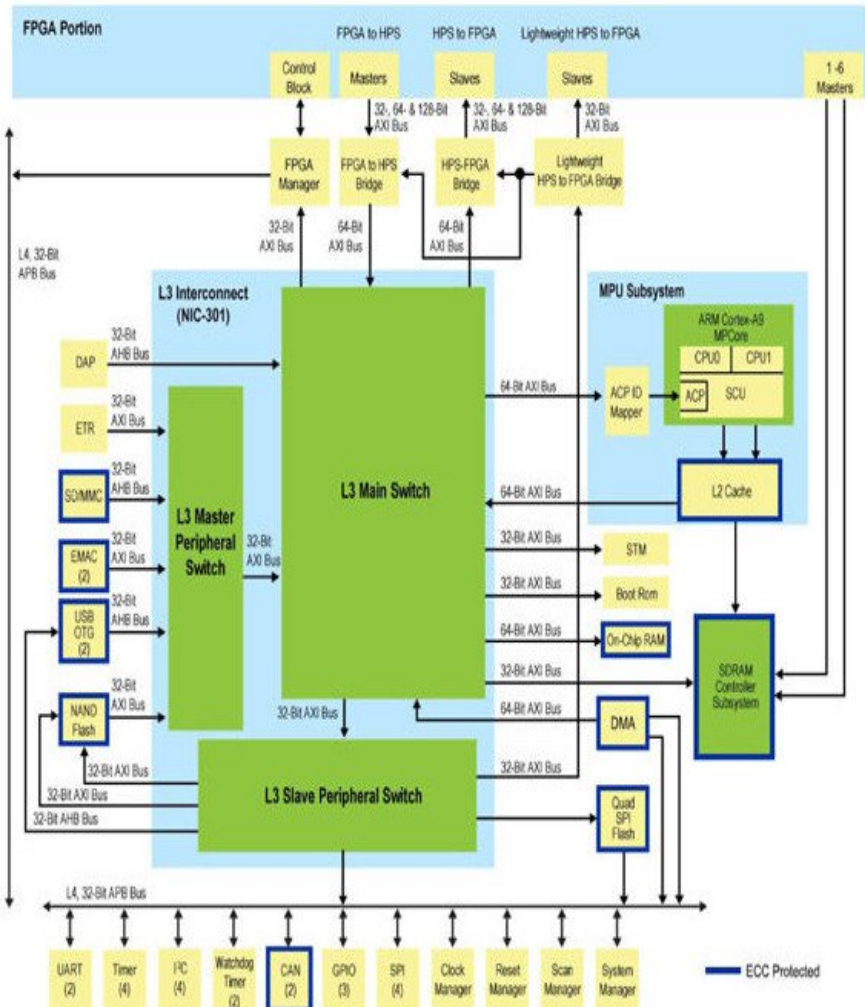
- Dual ARM Cortex-A9 MPCore
 - Up to 1 GHz
 - NEON and Single & Double Precision FPU
 - L1: 32+32kB
 - L2: 512kB (unified)
 - 256kB on-chip Memory
- DDR2, DDR3, DDR3L and LPDDR2 Dynamic Memory Controller
- QSPI, NAND/NOR Flash Memory Controller
- 2xUSB2.0 (OTG), 2xGbEthernet, 2xCAN2.0B, 2xSD/SDIO, 2xUART, 2xSPI, 2xI2C, 4x32b GPIO
- AES & SHA 256b encryption engine for secure boot and configuration
- 2xDual channel 12bit 1Msps ADC

Fuente: <http://www.xilinx.com/products/silicon-devices/soc/zynq-7000/index.htm>

Hardcores – Xilinx – Zynq 7000

- Advanced Programmable Logic:
 - 28k to 444k Logic Cells (approximately 430k to 6.6M of equivalent ASIC Gates) [Artix / Kintex]
 - 240kB to 3020kB of Extensible Block RAM
 - 80 to 2020 18x25 DSP Slices (100 to 2020 GMACS peak DSP performance)
- PCI Express® Gen2x8 (in largest devices)
- 139 to 400 User IOs (Multiplexed + SelectIO)
- Up to 16 12.5Gbps Transceivers (in largest devices)

Hardcores – ALTERA

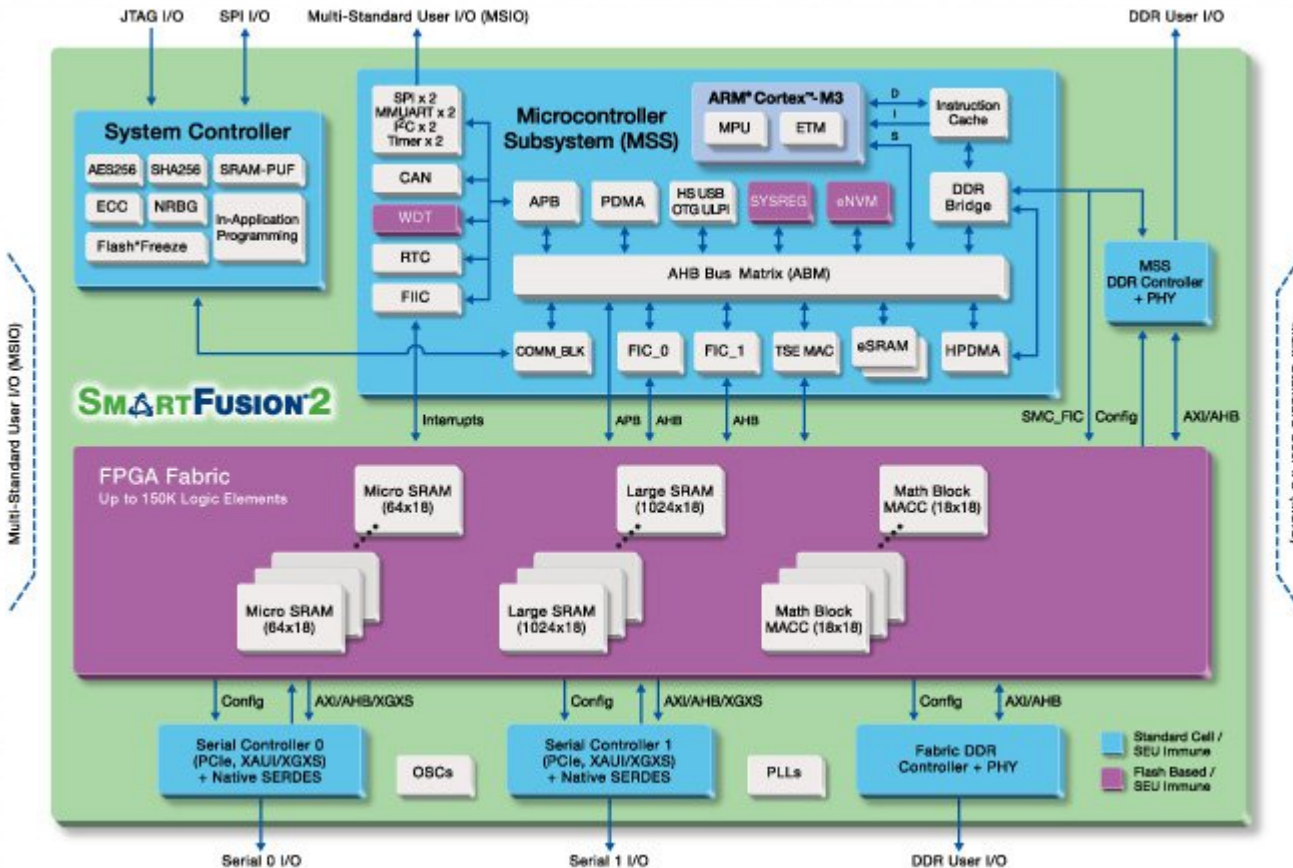


- Single/Dual ARM Cortex-A9 MPCore
 - Up to 1.05 GHz
 - NEON and FPU
 - L1: 32+32kB
 - L2: 512kB (unified)
 - 64kB on-chip Memory
- DDR2, DDR3, DDR3L and LPDDR2
- QSPI, NAND Flash Memory Controller
- 2xGbEthernet, 2xUSB 2.0 (OTG), 1xSD/MMC/SDIO, 2xUART, 4xI2C, 2xCAN, 2xSPI master, 2xSPI slave

Excalibur (ARM9) discontinuado, al igual que las familias de FPGA que lo incluían.

Fuente: Architecture Matters: Choosing the Right SoC FPGA for Your Application (ALTERA)

Hardcores – Microsemi



- ARM Cortex-M3
 - Up to 166 MHz
 - L1: 8K (I)
 - 64kB on-chip Memory
- LPDDR, DDR2, DDR3
- GbEthernet, 2xUSB 2.0 (OTG), 2xUART, 2xI2C, CAN, 2xSPI

Fuente: <http://www.microsemi.com/products/fpga-soc/soc-fpga/smartfusion2>

Hardcores – Comparación

Family	SoC	Frec (MHz)	DMIPS min	Logic (K)	BRAM (Kb)	U\$S
SmartFusion	M3	80 ~ 100	100	1.5~11.5	36~108	18~535
SmartFusion2	M3	166	207.5	6~146	191~4488	25~835
Cyclone V SE	A9	600~800	1500	25~110	1538~6191	50~228
Cyclone V SE	2xA9	600~925	3000	25~110	1538~6191	60~326
Arria V SX	2xA9	800~1050	4000	350~462	19304~25478	1247~5500
Zynq (Artix)	2xA9	667~866	3335	28~85	1920~4480	55~162
Zynq (Kintex)	2xA9	667~1000	3335	125~444	8480~24160	201~4774

- Cortex M3 equivalente por U\$S 5,6 (pero hay desde U\$S 1)
- Cortex A9 equivalentes por U\$S 18,4 (iMX6S) y U\$S 34,2 (iMX6D)
- FPGAs IGLOO2 6K (U\$S 21), Cyclone V 25K (U\$S 35), Artix7 35K (U\$S 38)
- Tenemos equivalente SmartFusion2 por U\$S 26.6, Cyclone V por U\$S 53,4, Zynq (Artix7) por U\$S 72,2. Más caros, pero con GPU!
- IMX6SL (U\$S 12,1) + Spartan6 4K (U\$S 11) por u\$s 23,1 (con GPU)



Agenda

1. ¿Qué es una FPGA?
2. ¿Cuándo usar una FPGA?
3. CPUs disponibles para FPGAs
 - Hardcores
 - **Softcores**
 - Del fabricante
 - De terceros
 - Desarrollados in-house
4. Casos de ejemplo
 - Clon PIC16F84
 - Clon AVR

Softcores

- La CPU se implementa utilizando los recursos reconfigurables de la FPGA.
- Más flexible que un hardcore
- En los casos más usados el costo no es por parte
 - En hardcores afecta al costo de la parte
 - Hay casos de licencias con royalty, ej: Cortex M1
 - Hay casos de partes que incluyen la licencia
- Puedo independizarme del fabricante
 - Usando un core independiente

Softcores Opciones disponibles

- Provistos por el fabricante de la FPGA
- Comprados a proveedores de IP
- Obtenidos de internet
- Desarrollados in-house

Agenda

1. ¿Qué es una FPGA?
2. ¿Cuándo usar una FPGA?
3. CPUs disponibles para FPGAs
 - Hardcores
 - Softcores
 - **Del fabricante**
 - De terceros
 - Desarrollados in-house
4. Casos de ejemplo
 - Clon PIC16F84
 - Clon AVR

Softcores Provistos por el fabricante de la FPGA

- Ventajas
 - Optimizados para la tecnología del fabricante
 - Bajo costo
 - Muchos periféricos disponibles
 - Entornos de desarrollo amistosos
- Desventajas (casi siempre)
 - No se dispone del “fuente”
 - Dependencia de un proveedor

Softcores Provistos por los fabricantes – Algunos ejemplos

- Xilinx Picoblaze
 - 8 bits
 - 192 LUTs
 - Licencia: **free**
 - Tiene un clon open source
- Xilinx Microblaze
 - 32 bits
 - 1000-3000 LUTs
 - 232 a 445 DMIPS (Artix-7 a Virtex-7)
 - Licencia: **con el EDK (anual)**
 - Compatibles open source
 - **Corre Linux**

Softcores Provistos por los fabricantes – Algunos ejemplos

- Altera NIOS II

- 32 bits

- **Corre Linux**

- NIOS II/e “economy”

- 650 LEs

- Licencia: **free**

- 22 a 51 DMIPS

- NIOS II/s “standard”

- 1400 LEs

- Licencia: **licencia**

- 64 a 192 DMIPS

- NIOS II/f “fast”

- 1800 a 3000 (c/MMU) LEs

- Licencia: **licencia**

- 153 a 350 DMIPS

Valores máx. para: Stratix V @ 300 MHz aprox.

Valores mín. Para: Cyclone III LS @ 120 MHz aprox. (-7)

Softcores Provistos por los fabricantes – Algunos ejemplos

- Lattice Mico 8
 - 8 bits
 - 200 LUTs
 - Licencia: ¿open source?
- Lattice Mico 32
 - 32 bits
 - 2400 LUTs
 - Licencia: ¿open source?
 - Fuentes generados por la herramienta (VERILOG)
 - 115 MHz (¿DMIPS?)
 - Corre μ C Linux

Agenda

1. ¿Qué es una FPGA?
2. ¿Cuándo usar una FPGA?
3. CPUs disponibles para FPGAs
 - Hardcores
 - Softcores
 - Del fabricante
 - **De terceros**
 - Desarrollados in-house
4. Casos de ejemplo
 - Clon PIC16F84
 - Clon AVR

Softcores Ofrecidos por proveedores de IP

- Gran oferta para la industria
- En algunos casos masificado a través de acuerdos con el fabricante, ej ARM Cortex-M1
 - Actel: 4353 LUTs, sólo para partes específicas
 - Altera: 2600 LUTs, licenciado, paquete por 1000.
 - Sin código fuente, bajo costo, pero por parte
- En el site del fabricante podemos encontrar información y que certificaciones posee (ALTERA: V1 Coldfire, 8051, BA22, etc.)
- Licenciamiento muy variable, CPU pequeña p/ASIC aprox. 100.000 U\$\$
- En algunos casos con licencia dual:
 - LEON2 LGPL
 - LEON3 GPL (SPARC-v8 32 bits) (20.000 €)
- Suelen ser caros



El procesador LEON y la GRLIB

- Procesador de 32 bits.
- Cumple con la arquitectura SPARC V8.
- Utiliza Bus AMBA 2.0 (AHB y APB) para interconectar periféricos.
- Lenguaje VHDL.
- Código fuente bajo licencia GPL/LGPL (según versión).
- Mantenido y comercializado por la empresa Aeroflex/Gaisler.



- Recursos en FPGA:
 - Síntesis de LEON3 en Spartan6
 - 15K7 LUTs (25K Logic Cells), 67 MHz
 - Spartan6 de 45K => U\$S 52
 - Artix7 de 35K => U\$S 38
 - 1.4 DMIPS/MHz => 94 DMIPS

Historia del procesador LEON

1997 – Nace el LEON

Proyecto interno de la "European Space Research and Technology Centre" (ESTEC) para estudiar y desarrollar un procesador de 32 bits de alto rendimiento y tolerante a fallas, para emplear en las misiones espaciales europeas posteriores al año 2000. 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.

1999 - LEON1

La ESA "Agencia Espacial Europea" libera el código del LEON1 bajo LGPL.

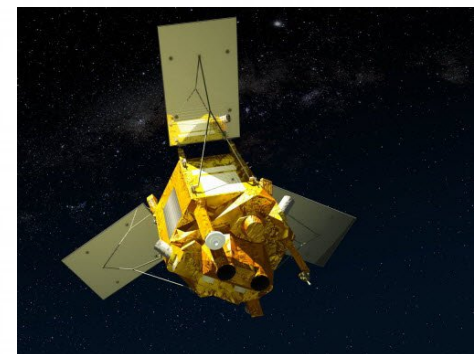
2002 – LEON2

El chip AT697 (LEON2-FT) vuela por primera vez en 2008, dentro de una computadora prototipo de payload llamada ERNObox.



2004 – LEON3 + GRLIB

El LEON3 es parte del satélite francés de observación Spot-6, lanzado en septiembre de 2012.



http://www.esa.int/Our_Activities/Space_Engineering/LEON_s_first_flights

El procesador LEON3



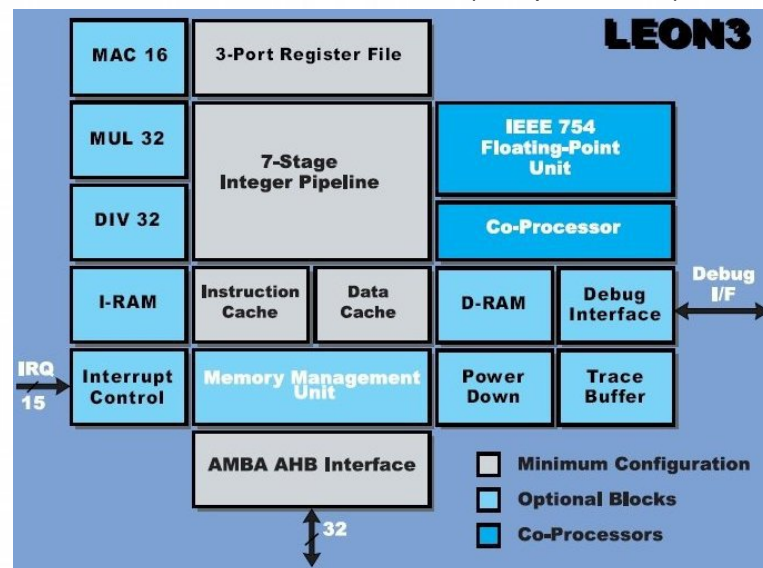
- Pipeline avanzado de 7 etapas.
- Unidades de hardware para multiplicación, división y MAC.
- Caches de datos e instrucciones.
- MMU (SRMMU, SPARC Reference Memory Management Unit).
- Bus AMBA-2.0 AHB.
- Debug y trace buffer para datos e instrucciones.
- Soporte para multiprocesadores simétricos (SMP).
- Compiladores, kernels, simuladores, monitores. Software libre como las GNU tools y el kernel Linux y otros privativos como el monitor y el simulador.
- Altamente configurable.

Opcionales no entregados bajo GPL (pagos):

- FPU IEEE-754 de alta performance .
- Versiones tolerantes a falla y a prueba de SEU (Single Event Upset), para aplicaciones espaciales.

Technology	MHz	Area
TSMC 65 nm ASIC (std-cell)	650 MHz	25 kgates
UMC 130 nm ASIC (std-cell)	400 MHz	25 kgates
Virtex-5 FPGA	140 MHz	3,500 LUT
Virtex-4 FPGA	125 MHz	3,500 LUT
Altera Stratix-3 FPGA	150 MHz	3,500 cells
Altera Stratix-2 FPGA	125 MHz	3,500 cells
Actel ProASIC3 FPGA	40 MHz	6,500 cells
Actel AX1000 FPGA	33 MHz	4,000 cells
Actel RTAX2000S FPGA	25 MHz	4,000 cells

**Datos del fabricante (solo procesador)*

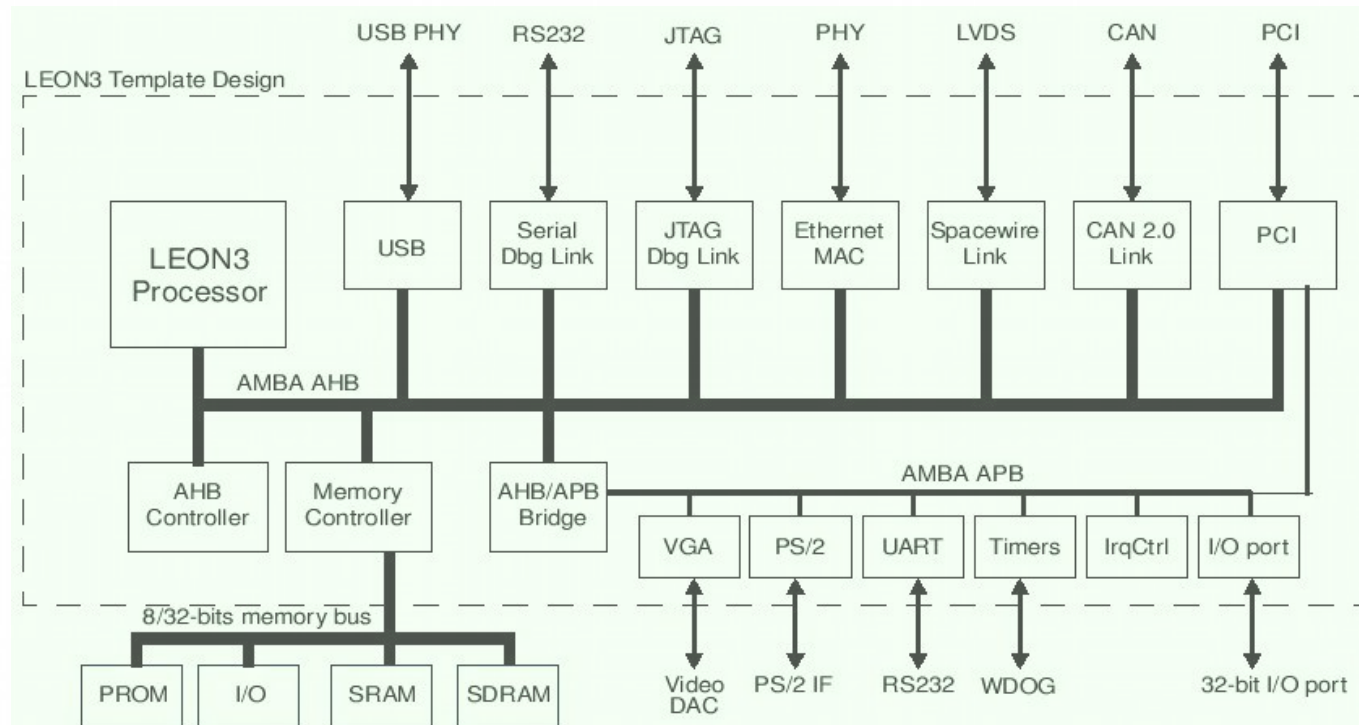


La GRLIB



- La GRLIB es un conjunto organizado y configurable de IP Cores.
- El núcleo central es el procesador LEON3.
- Conectado con buses AMBA AHB y AMBA APB.
- Altamente configurable, pudiendo seleccionar qué componentes incluir y las características de cada uno.

- **VHDL neutral** para compatibilidad con distintas tecnologías de FPGAs y ASICs.
- La GRLIB posee **licencia GPL**.
- Incluye IP-cores de terceros como por ejemplo OpenCores.



Sistemas operativos soportados por el LEON y la GRLIB

Libres/Abiertos y gratuitos:

Linux 2.6: <http://www.kernel.org/>

uClinux: <http://www.uclinux.org/>

RTEMS: <http://www.rtems.com/>

eCos: <http://ecos.sourceware.org/>



uClinux



Cerrados/pagos:

VxWorks (Wind River)

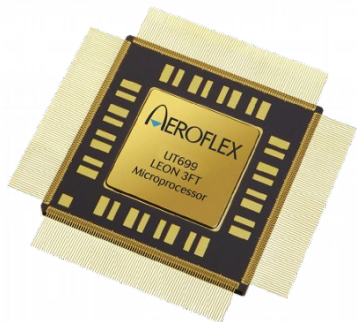
Nucleus (Mentor Graphics)

ThreadX (Express Logix)

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

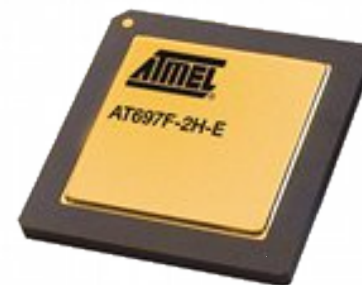
Implementaciones ASIC

Tolerantes a falla

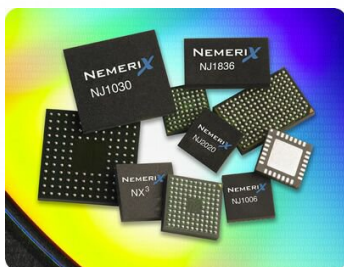


Aeroflex:
UT700, 130nm CMOS, 1MHz-166 MHz
UT699E, 130nm CMOS, 1MHz-100 MHz
UT699, 0.25um CMOS, 1MHz-66 MHz
GR712RC, 180 nm CMOS, 2-core, 100 MHz

AT697E/F, 180 nm, 0-100 MHz
AT7913E, 0-50 MHz



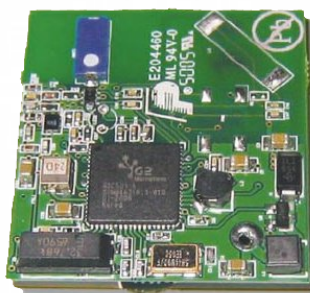
Algunos Productos



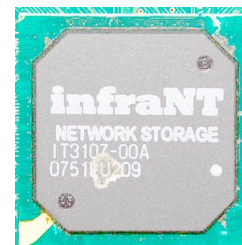
Nemerix NJ1030A para GPS



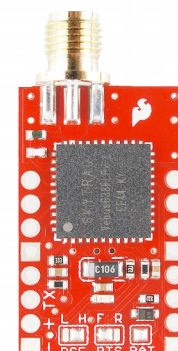
Zhuhai Orbita Control Engineering. S698. (PoS y cajas registradoras)



G2 Microsystem. G2C501 con 802.11 y RFID.



Infrant Technologies. IT3102/IT3107 Network Storage Processor.



SkyTraq Venus8. Chips GPS.



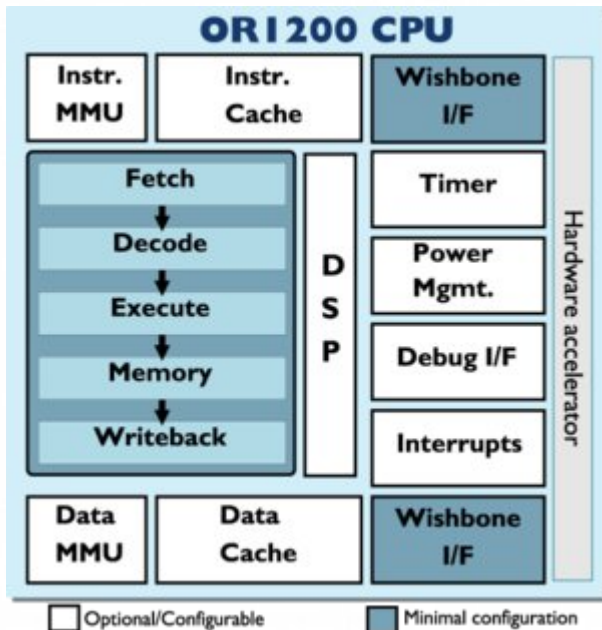
Softcores Ofrecidos por proveedores de IP

OpenCores.org

- De código abierto y/o libre, ejemplo: OpenCores.org
- La calidad y documentación es muy variable
- No siempre están completos o son usables
- Más de 170 CPUs, desde 1 bit
- Compatibles: 6502/7, 6800/1, 68HC05/08/09/11, 68000, 8048/9, 8051, 8080, 80186, 90S1200, 90S2313, ARM v2a, ARM7, AVR, C54x, COP400, PIC16C5x, PIC16C84, PIC18xx, Microblaze, MIPS, MSP430, SAM87RI, SuperH-2, Z8, Z80, Z8001/2 entre otros
- OpenRisc 1200 es el más renombrado creado por el proyecto y corre Linux
- Específicos: Forth, JOP (JVM), ZPU (stack CPU), etc.



Softcores Ofrecidos por proveedores de IP – OpenRISC



- **OpenRISC 1000 (OR1K)**: especificación de la arquitectura (32/64 bits)
- **OpenRISC 1200 (OR1200)**: **LGPL** implementación de OC (monocore, Verilog)
- **ORPSoC**: OpenRISC Reference Platform SoC: OR1200+Periféricos. Bus WISHBONE
- Otros basados en OR
 - MinSoC (recortado, más compacto)
 - Mor1kx (re escritura, más performante)
 - Munich University: multi-core
 - Bologna University: ultra-low power multi-core

Softcores Ofrecidos por proveedores de IP – OpenRISC

- Implementaciones en silicio:
 - Flextronics (2003) lo implementó (demo) en ASIC. En 0.18u 6LM process, 150 MHz, 150 DMIPS.
 - Core AR100 del Allwinner A31, manejo de ahorro de energía cuando el resto del SoC está apagado.
 - Parte del Multi Format Codec (MPEG-2/4, H.263/4, VC1) del Exynos 4210 (S5PV310).
- Recursos en FPGA:
 - Síntesis de ORPSoC en Spartan6
 - 13K ALUTs (21K Logic Cells), 45 MHz
 - Spartan6 de 24K => U\$S 34
 - Artix7 de 35K => U\$S 38
 - 1 DMIPS/MHz => 45 DMIPS



Softcores Ofrecidos por proveedores de IP – Licencias O.C.

- Las licencias abiertas y/o libres son variadas y pueden tener problemas asociados, entre las más frecuentes
- **Dominio Público:** Se le puede dar cualquier uso.
- **BSD:** Se le puede dar cualquier uso en tanto que se atribuya el mérito al autor original.
- **LGPL:** Cambios en la descripción deben ser ofrecidos al cliente.
- **GPL:** El diseño completo debe ser ofrecido al cliente.

- **De acuerdo con Richard M. Stallman, líder del movimiento del software libre, la licencia GPL no es correcta para descripciones de hardware y probablemente no sirva.**

Agenda

1. ¿Qué es una FPGA?
2. ¿Cuándo usar una FPGA?
3. CPUs disponibles para FPGAs
 - Hardcores
 - Softcores
 - Del fabricante
 - De terceros
 - **Desarrollados in-house**
4. Casos de ejemplo
 - Clon PIC16F84
 - Clon AVR

Softcores Desarrollados in-house y adaptados

- Por razones didácticas:
 - Se aprende mucho desarrollando una CPU
 - Se pueden estudiar arquitecturas poco frecuentes o especiales.
- Para tener mayor control y/o conocimiento del core
- Para ofrecerlo a terceros
- Para aplicaciones especiales:
 - Procesadores muy específicos o muy optimizados para alguna tarea en particular.
- El desarrollo de una CPU pequeña es rápido, aunque la validación más lenta.
- Es posible basarse en una versión ya existente.

Softcores Desarrollados in-house y adaptados

- En nuestro laboratorio tenemos dos ejemplos:
 - **Clon del PIC16F84**, compacto, útil para programar en assembler, muy popular en la región
 - **Clon del AVR**, apto para ser programado en lenguaje C.

Agenda

1. ¿Qué es una FPGA?
2. ¿Cuándo usar una FPGA?
3. CPUs disponibles para FPGAs
 - Hardcores
 - Softcores
 - Del fabricante
 - De terceros
 - Desarrollados in-house
4. Casos de ejemplo
 - **Clon PIC16F84**
 - Clon AVR

Clon del PIC16F84 ¿Por qué un PIC?

- Arquitectura simple, poco trabajo. Fue uno de los primeros cores desarrollados (2005).
- Es pequeño, apto para kits con Spartan 2 100 que teníamos.
 - Ver S2PROTO <http://fpgalibre.sf.net>
- CPU muy popular.
- De la familia de PICs de 14 bits muy usados en nuestro laboratorio (en esa época).

Clon del PIC16F84 Arquitectura

- Basada en el diagrama en bloques de la hoja de datos => compatible aún con trucos.
- RISC de 4 ciclos de reloj: Decodificación, Lectura de operandos, Ejecución y Escritura de resultados.
- Soporta RMW.
- Memoria de programa 14 bits.
- Memoria de datos 8 bits.

Clon del PIC16F84 ¿Qué se implementó?

Stack de 8 niveles	EEPROM
Sistema de interrupciones. Mascaras, etc.	Power-on Reset
Watch Dog. Pero basado en el clock	Power-up Timer
Temporizador/Contador	SLEEP
Entrada externa de interrupciones	Programación serie de la memoria de programa.
Interrupción por cambio en un grupo de pines	

Clon del PIC16F84 Mejoras/Comparación

Original	 MICROCHIP	Sintetizado
10 MHz		30 MHz (Spartan II) 50 MHz (Spartan 3)
36 bytes de memoria		464 bytes de memoria
13 pines de I/O		24 pines de I/O
Mem. programa 1 kWord		Expandible hasta 64 kWords
		+WISHBONE, ICE, INT x nivel



Clon del PIC16F84 Resultados

- VHDL93, independiente de la tecnología, sintetiza para FPGAs de Actel.
- Transferido a la industria aero-espacial.

Configuración	Recursos
Mínima: prg 32 words, sólo 1 puerto I/O	41 FFs+239 LUTs (2 BRAMs) Spartan II
Sistema completo: CPU + UART + I ² C + Video	394 FFs+946 LUTs (5 BRAMs) Spartan 3
Mismo sistema con interfaz de debug	525 FFs+1055 LUTs (8 BRAMs) Spartan II

Agenda

1. ¿Qué es una FPGA?
2. ¿Cuándo usar una FPGA?
3. CPUs disponibles para FPGAs
 - Hardcores
 - Softcores
 - Del fabricante
 - De terceros
 - Desarrollados in-house
4. Casos de ejemplo
 - Clon PIC16F84
 - **Clon AVR**

Clon AVR ¿Por qué un AVR?

- Familia de CPUs muy popular.
- Aún hoy en uso en proyectos de nuestro laboratorio.
- Apto para programar en C.

Clon AVR Metodología

- Basado en un core funcional de OpenCores.org (AVR Core de Ruslan Lepetenok)
- El original implementa un AVR de 3ra generación ATmega103
- Fuertemente modificado (era de muy bajo nivel)
- Compatible ciclo a ciclo
- También se evaluó otro proyecto que agregaba un pipeline.
- Poca documentación y ningún ejemplo sintetizable funcional.

Clon AVR ¿Qué soporta?

- Versión original de O.C.
 - Puertos de I/O (reescrito)
 - Timers/Counters (0 y 2, no el 1) (parcialmente reescrito)
 - UART (sin modificaciones)
- Agregado
 - Soporte para 2da y 4ta generación (ej: ATtiny22 y ATmega8)
 - Watchdog
 - Máscaras y flags de interrupciones externas.
 - Bus de expansión WISHBONE
 - Interfaz de depuración (ICE) [Debug a nivel de fuentes C usando USB]

Clon AVR Resultados - Área

CPUs sin ROM ni periféricos

AVR	Generación	Datos	FFs	LUTs	BRAMs	Mult	Fmax [MHz]
ATtiny22	2	256	175	948	1	0	36
ATmega103	3	4k	183	1009	2	0	38
ATmega8	4	1k	213	1128	1	1	29

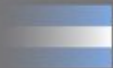
Espacio direccionable para la memoria de programa: 64 k words
Spartan 3

Clon AVR Resultados - Velocidad

- Benchmark: Dhrystone MIPS v2.1
 - Muy usado, no demasiado representativo.
- Clón ATmega32 (código 11k>8k)
 - Adaptado para 8kB de datos.
- Compilado con gcc 4.3.2.
- Clock: 16 MHz, 20.000 iteraciones
- Resultado: 95 μ s por iteración, 10443 Dhrystones/s → 5,94 DMIPS → 0,37 DMIPS/MHz
- sizeof(int) == 2, comparable a una AT 286



INTI



**INDUSTRIA
ARGENTINA**
ORGULLO NACIONAL

¡Muchas Gracias!

Av. Gral. Paz 5445
(1650) San Martín
Buenos Aires, Argentina
(11) 4724-6315
salvador@inti.gov.ar
<http://fpgalibre.sf.net>