



Instituto
Nacional
de Tecnología
Industrial



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



Congreso de Microelectrónica Aplicada (μEA 2010) Microprocesadores dentro de FPGAs

Autores:

Ing. Salvador E. Tropea (Softcores)

Ing. Diego J. Brengi (Leon y Grlib)

Electrónica e Informática

Desarrollo Electrónico con Software Libre (DESoL)

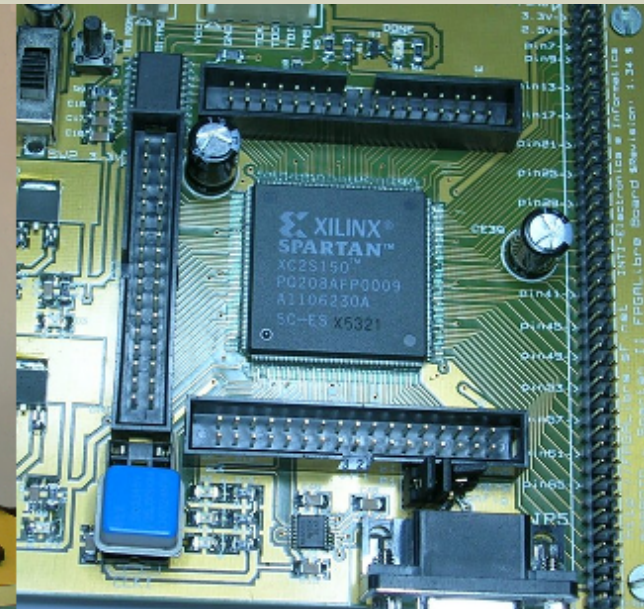




Instituto
Nacional
de Tecnología
Industrial



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



Microprocesadores dentro de FPGAs

Alternativas para sistemas de altas prestaciones y/o alto grado de integración





Agenda

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





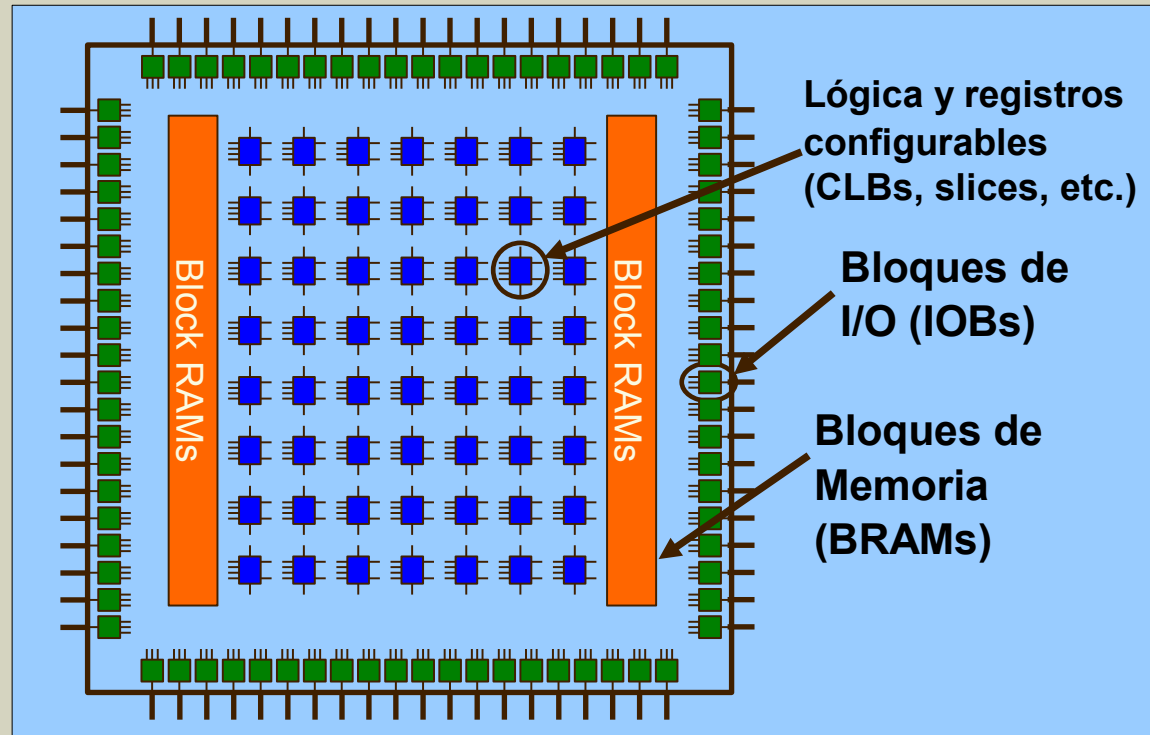
Agenda

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



Elementos básicos

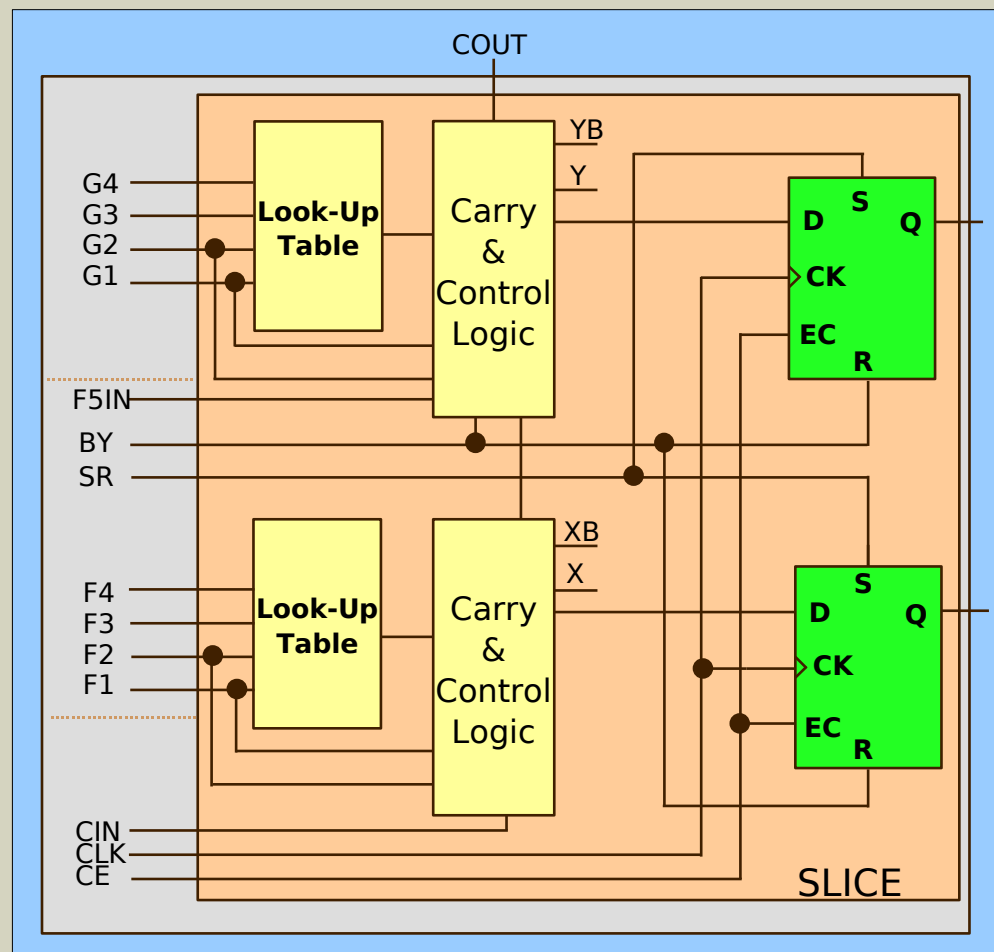
- Las FPGAs son circuitos integrados (re)configurables.
- Poseen lógica combinatorial (re)configurable.
- Poseen registros (flip/flops) (re)configurables.
- Poseen un sistema poderoso de interconexión (re)configurable.



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

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.





Agenda

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

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

Desventajas comparadas con un microcontrolador

- Las más baratas son más caras que los μP económicos
 - La brecha cae con los años
- 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

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
 - **Hard cores**
 - **Soft cores**
 - Del fabricante
 - De terceros
 - Desarrollados in-house
4. Casos de ejemplo
 - Clon PIC16F84
 - Clon AVR
 - ZPU



Opciones de CPUs

- Hard cores [en silicio]
 - Máximas prestaciones
 - Alto costo (por parte)
 - Muy dependiente del fabricante
 - Xilinx Power PC (405) [¿en transición a ARM?]
 - Altera ARM (v4 922T) [discont. ¿pasa a MIPS?]
- Soft cores [en la configuración]
 - Más flexible
 - Normalmente el costo no es por parte (no Actel ARM)
 - Puedo independizarme del fabricante



Opciones con FPGAs



Instituto
Nacional
de Tecnología
Industrial



Opciones de Soft Cores

- 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
 - Hard cores
 - Soft cores
 - **Del fabricante**
 - De terceros
 - Desarrollados in-house
4. Casos de ejemplo
 - Clon PIC16F84
 - Clon AVR
 - ZPU



Provistos por los fabricantes

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

Provistos por los fabricantes – Algunos ejemplos

- Xilinx Picoblaze (8 bits) [192 LUTs] free
 - Tiene un clon open source
- Xilinx Microblaze (32 bits) [1000-3000 LUTs 280 DMIPS] w/EDK
 - Compatibles open source en desarrollo (¡no clones!)
 - Corre Linux
- Altera NIOS II (32 bits)
 - NIOS II/e “economy” 700 LEs free (31 DMIPS)
 - NIOS II/s “standard” 1400 LEs licencia (127 DMIPS)
 - NIOS II/f “fast” 1800 a 3000 (c/MMU) LEs licencia (218 DMIPS)
 - Corre Linux
- LatticeMico8 (8 bits) [200 LUTs] (fuentes disponibles)
- LatticeMico32 (32 bits) [1984 LUTs] (fuentes disponibles)

Nota: área reportada por los fabricantes, suele ser engañosa



Agenda

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



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, ¿licencia?
 - Xilinx: sin datos, ARM dice que si
 - Sin código fuente, bajo costo, pero por parte
- En algunos casos con licencia dual:
 - LEON2 LGPL
 - LEON3 GPL (SPARC-v8 32 bits)
 - Suelen ser caros



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 103 CPUs, desde 1 bit
 - Compatibles: 6502, 6507, 6800/1, 68HC05, 68HC08, 68HC11, 68000, 8048/9, 8051, 8080, 80186, 90S1200, 90S2313, AVR, C54x, COP400, PIC16C5x, PIC16C84, PIC18xx, Microblaze, MIPS, MSP430, SAM87RI, SuperH-2, Z8, Z80, Z8001/2
 - OpenRisc 1200 es el más renombrado creado por el proyecto y corre Linux
 - Específicos: Forth, JOP (JVM), **ZPU (stack CPU)**, etc.



Ofrecidos por proveedores de IP

- 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
 - Hard cores
 - Soft cores
 - Del fabricante
 - De terceros
 - **Desarrollados in-house**
4. Casos de ejemplo
 - Clon PIC16F84
 - Clon AVR
 - ZPU



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, la validación más lenta.
- Es posible basarse en una versión ya existente.



Soft Cores



INTI

Instituto
Nacional
de Tecnología
Industrial



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.





Casos de ejemplo



Instituto
Nacional
de Tecnología
Industrial



Agenda

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





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.
- CPU muy popular.
- De la familia de PICs de 14 bits muy usados en nuestro laboratorio.



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 - Incluyendo

- Stack de 8 niveles.
- Sistema de interrupciones. Mascaras, etc.
- Watch Dog. Pero basado en el clock.
- Temporizador/Contador.
- Entrada externa de interrupciones.
- Interrupción por cambio en un grupo de pines.



Clon del PIC16F84 – No implementado

- EEPROM.
- Power-on Reset.
- Power-up Timer.
- SLEEP
- Programación serie de la memoria de programa.

Clon del PIC16F84 – Mejoras

- Número de entradas/salidas.
- Tamaño de la memoria de datos.
- Frecuencia de Clock.
- Interrupciones por nivel.
- Configurables los pines de interrupción.
- Bus de expansión WISHBONE
- Interfaz de debug (ICE)



Clon del PIC16F84 – Mejoras

Original:



- 10 MHz
- 36 bytes de memoria
- 13 pines de I/O
- Memoria de programa de 1 kWord

Sintentizado:

- 30 MHz (Spartan II)
- 50 MHz (Spartan 3)
- 464 bytes de memoria
- 24 pines de I/O
- Memoria de programa expandible hasta 64 kWords

Clon del PIC16F84 – Resultados

- Área mínima: 41 FFs+239 LUTs (prg 32 words, sólo 1 puerto I/O) 2 BRAMs (Spartan II)
- Área de un sistema completo: CPU + UART + I²C + Video 394 FFs+946 LUTs, 5 BRAMs (Spartan 3)
- Mismo sistema con interfaz de debug: 525 FFs+1055 LUTs, 8 BRAMs (Spartan II)
- VHDL93, independiente de la tecnología, sintetiza para FPGAs de Actel.
- Transferido a la industria aero-espacial.



Casos de ejemplo



Instituto
Nacional
de Tecnología
Industrial



Agenda

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





Casos de ejemplo



Instituto
Nacional
de Tecnología
Industrial



Clon AVR - ¿Por qué un AVR?

- Familia de CPUs muy popular.
- Actualmente 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 (muy bajo nivel)
- Compatible ciclo a ciclo
- También se evaluó otro proyecto que agregaba un pipeline.
- Poca documentación y ningún ejemplo sintetizable funcional.



Casos de ejemplo



Instituto
Nacional
de Tecnología
Industrial



Clon AVR – Soportado en el original

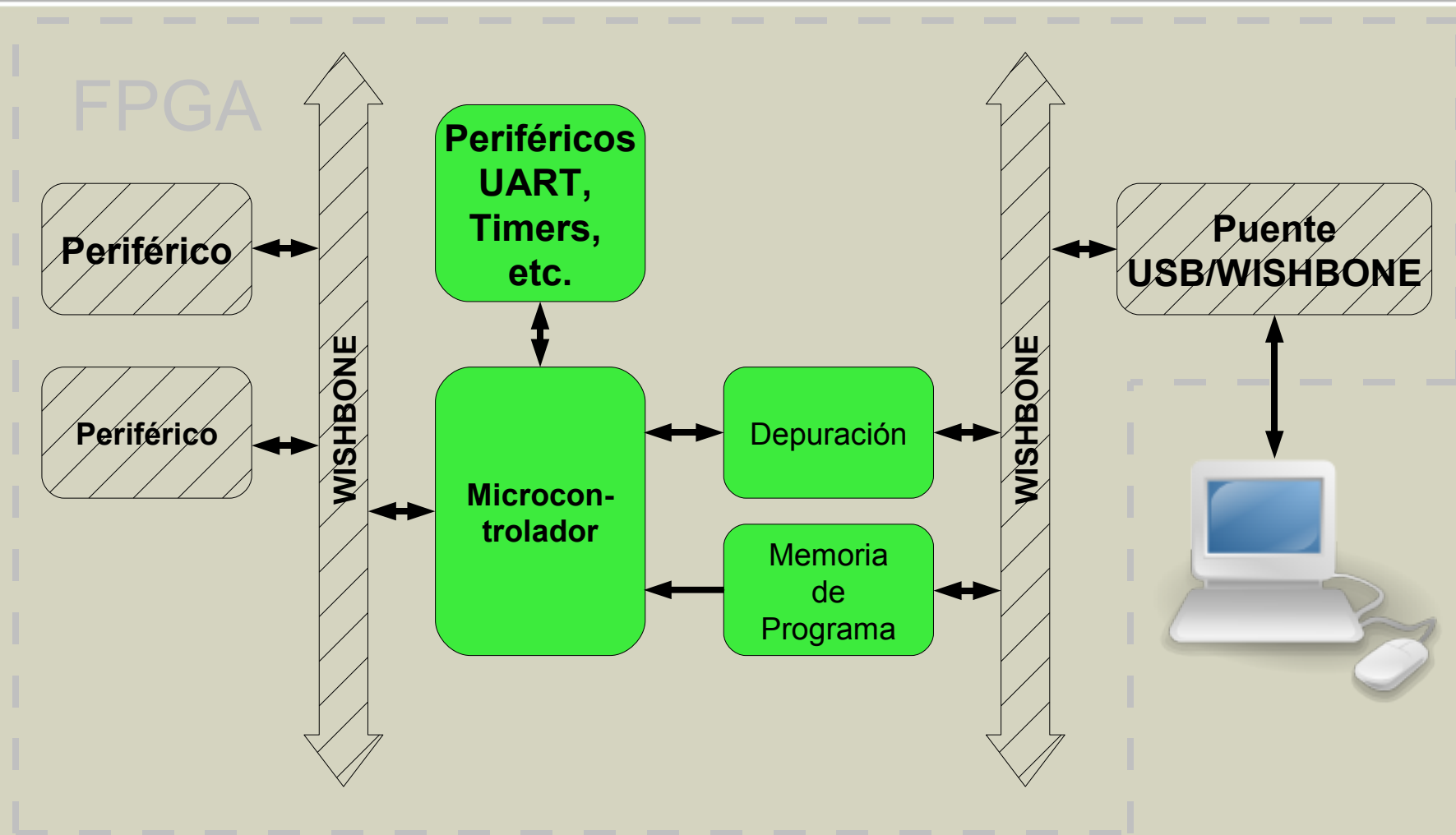
- Puertos de I/O (reescrito)
- Timers/Counters (0 y 2, no el 1) (parcialmente reescrito)
- UART (sin modificaciones)



Clon AVR – 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 – Agregado – Interfaz de depuración



Clon AVR – Resultados

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

Configuraciones de prueba

AVR	Datos	Progra ma	UART	I/ O	FFs	LUTs	BRAMs	Debug	Spartan	Fmax [MHz]
3	4k	4k	SI	0	302	1153	3	NO	3	31
3	4k	1k	WB	0	245	1124	3	NO	3A	33
4 ¹	1k	1k	WB	0	275	1244	2	NO	3A	30
2	256	1k	WB	0	237	1053	2	NO	3A	37
4 ¹	1k	1k	WB	8	902	2477	4	USB	3A	25 ²

1 Usa un multiplicador

2 Frecuencia fijada a 24 MHz por el USB

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: 95μs por iteración, 10443 Dhrystones/s => 5,94 DMIPS => 0,37 DMIPS/MHz
- `sizeof(int) == 2`, comparable a una AT 286



Casos de ejemplo



Instituto
Nacional
de Tecnología
Industrial



Agenda

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



ZPU - Introducción

- Estudiada como alternativa antes de desarrollar el AVR.
- Disponible en OpenCores.org como "the worlds smallest 32 bit CPU with GCC toolchain", la SmallCPU canadiense presentada en el SPL es más chica
- Escrita en VHDL de alto nivel, casi un programita.
- Autor: Øyvind Harboe
- Adaptó el gcc y las binutils para soportar a la ZPU.
- Agregó soporte al eCOS RTOS para la ZPU.
- Biblioteca de C: newlibc
- Bus de expansión: WISHBONE

ZPU - Arquitectura

- Es una *stack CPU*
- No posee registros, las operaciones se realizan en el *stack*: más simple, pero menor performance.
- Dos versiones disponibles: *small* y *medium*
- Estrategia para ahorrar área: no todas las instrucciones se implementan en hardware.
- La *small* implementa sólo las instrucciones más básicas, el resto son rutinas que ejecuta la CPU (tipo microcódigo, pero en RAM).
- La *medium* implementa casi todas en hard.
- Luego de varias pruebas se implementó una tercer alternativa.

ZPU – Resultados

CPUs sin ROM ni periféricos

Versión	FFs	LUTs	BRAMs ¹	Mult	Fmax ² [MHz]	DMIPS ³ / MHz	DMIPS máx.
small	143	501	3	0	91	0,010	0,91
medium	290	1612	3	3	70	0,055	3,85
alternativa	274	1239	3	0	80	0,026	2,08

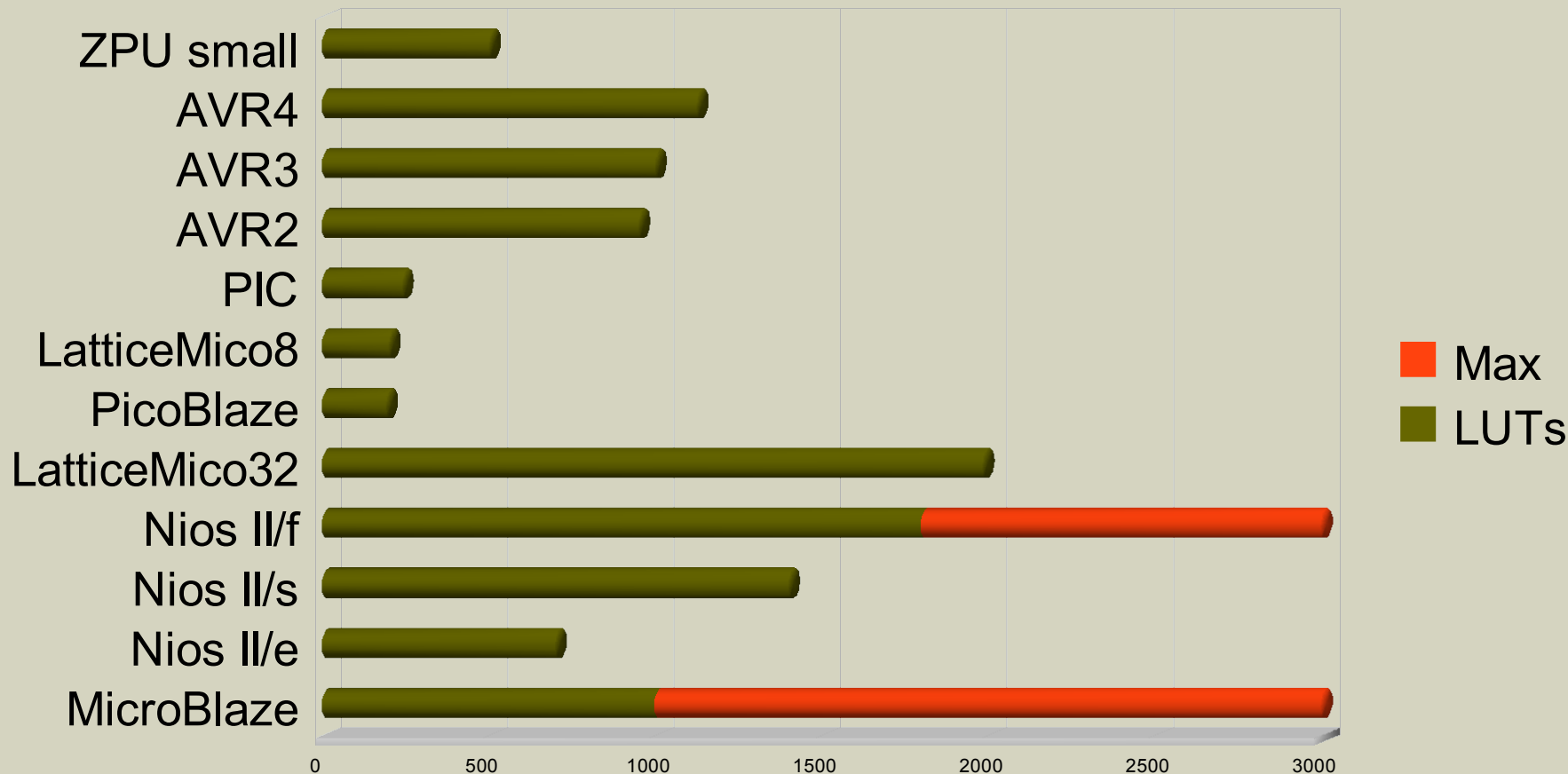
1 Espacio de memoria 8kB, programa, datos y *stack*. El programa más chico en C ocupa 4 kB

2 Spartan 3 -4

3 Dhrystone MIPS v2.1

ZPU - Ventajas

- La versión *small* es verdaderamente pequeña:



ZPU - Desventajas

- Muy baja performance
- Si se busca una CPU pequeña también se busca ahorrar en memoria y usar memoria embebida (BRAMs), pero el programa en C más chico ocupa 4 kB.
- El autor dice que el código es más compacto que el ARM thumb, pero un AVR necesita 70 bytes.
- Necesita modificaciones para poder colocar el código en ROM.
- La versión *medium* no implementa interrupciones.

ZPU – Comparada con el AVR

Resultados del Dhrystone MIPS v2.1

	AVR ATmega32	ZPU Small	Conclusión
Tamaño del código (bytes)	11.188	17.504	El AVR ahorra un 56%
Performance máxima (DMIPS) ¹	10,4	0,99	El AVR es 10 veces más rápido (y 37 por MHz)
Área [LUTs]	1375	704	El AVR es 85% más grande
Área [slices]	798	434	El AVR es 84% más grande

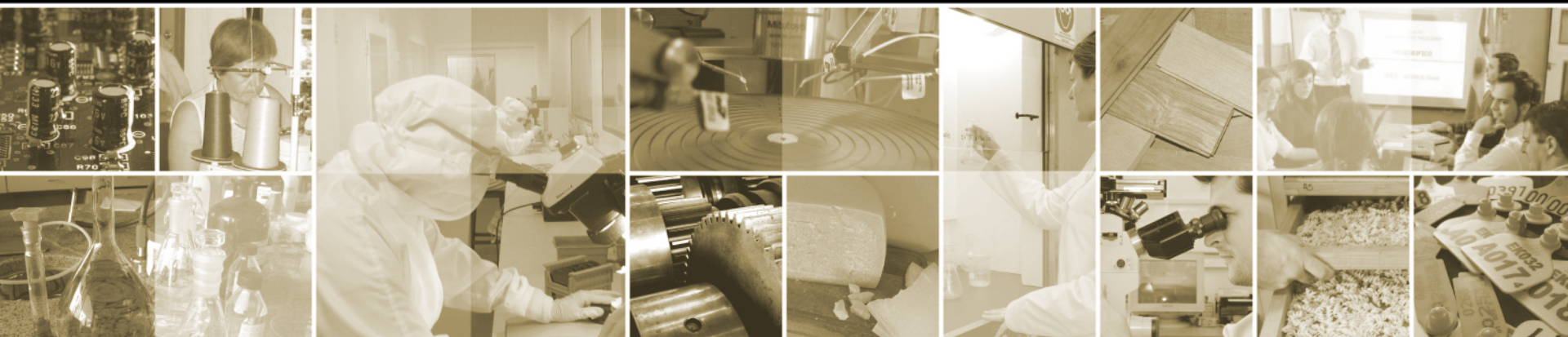
¹ El AVR usa enteros de 16 bits mientras que la ZPU usa enteros de 32 bits



Instituto
Nacional
de Tecnología
Industrial



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



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

julio 2010

