

Módulo IP del μ C 8051 de Oregano

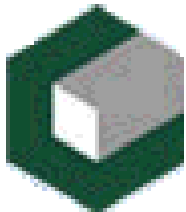
Maestría en Sistemas Digitales

Alejandro J. Cabrera Sarmiento

Dpto. de Automática y Computación

Universidad Tecnológica de La Habana “José Antonio Echeverría”
CUJAE

alex@automatica.cujae.edu.cu



Sumario

- **Características** fundamentales del módulo IP **mc8051 Oregano**
- **Componentes** del módulo IP de **Oregano**
- **Proceso de desarrollo** de un sistema basado en **Oregano**
- **Implementación** de un sistema 8051 **Oregano** sobre una placa **Spartan3E/3A Starter Kit**

Oregano mc8051

- **Soft CORE** compatible con **MCS-51**
- Arquitectura **optimizada**
- **Mayor velocidad** que 8051 original
 - **La mayoría de las instrucciones en uno o dos pulsos de reloj**
- Número **configurable** de temporizadores y puerto serie
 - **Relacionados**
- Implementación **opcional** de hardware
 - **MUL, DIV, DAA**
- Puertos de entrada/salida **independientes**
- **Buses independientes para cada memoria**

Oregano mc8051 *(cont.)*

- Señales **independientes** de **P3**
- Posibilidades de **128 bytes** de RAM interna y hasta **64 kBytes** de ROM y **64 kBytes** de RAM **“externa”**, **independientes**
- Independiente **tecnológicamente**
- Disponible en **código fuente VHDL** sin costo alguno (**licencia GNU LGPL**)
- Código fuente estructurado y comentado

<https://www.oreganosystems.at>

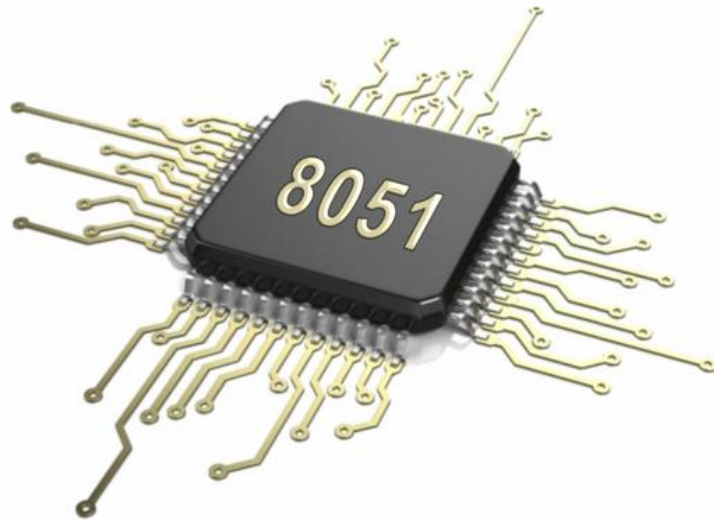
Oregano mc8051



8051 IP Core

[Home](#) / [Products](#) / [IP Cores](#) / 8051 IP Core

Search ...



The 8051 IP Core had been developed in cooperation with the Vienna University of Technology. This IP core is binary compatible to the well known 8051 processor from [Intel](#). The Oregano Systems 8051 IP core is available as a parameterizable, synthesizable circuit description (VHDL).

The Oregano Systems 8051 IP core offers faster program execution compared to the original 8051 devices due to an optimized processor's architecture. Additionally, the Oregano Systems 8051 IP core can be parametrized. The Oregano Systems 8051 IP Core is available **free of charge even for industrial applications** under the LGPL (Lesser General Public License).

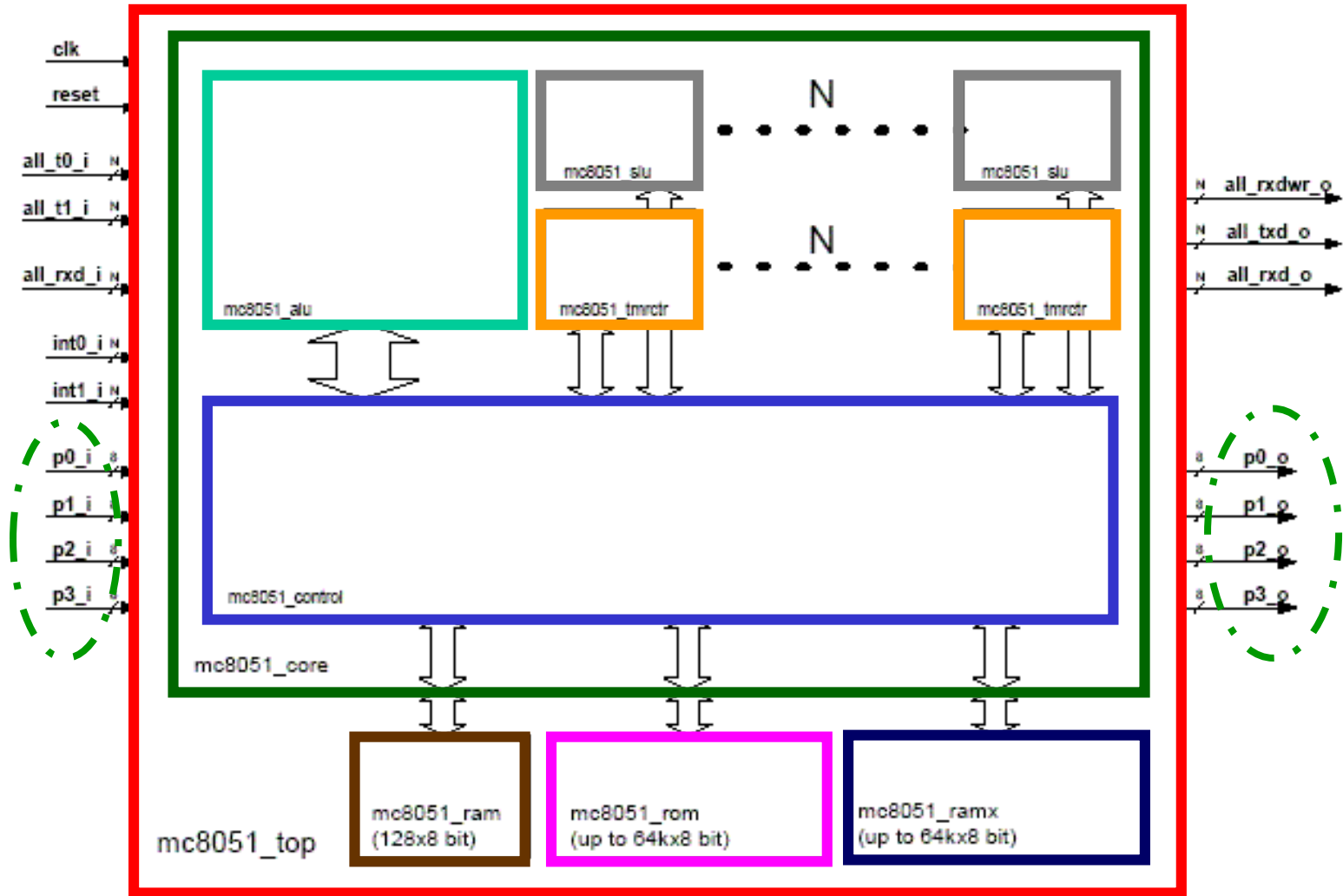
We kindly ask you to send us a brief feedback about your successful implementation of the Oregano Systems 8051 IP core in your FPGA or ASIC design. Please send us a brief email description or use the [8051 IP core feedback form \(PDF-File\)](#).

[Back to overview](#)

General Description

Oregano Systems 8051 IP Core is a 8051 compatible IP core. The design has

Diagrama en bloques de Oregano



Ficheros fuentes de IP de 8051

mc8051_p.vhd

mc8051_top.vhd

mc8051_core.vhd

mc8051_alu.vhd

addsub_core.vhd

addsub_cy.vhd

addsub_ovcy.vhd

alucore.vhd

alumux.vhd

comb_divider.vhd

comb_mltpr.vhd

dcmal_adjust.vhd

mc8051_control.vhd

control_mem.vhd

control_fsm.vhd

mc8051_siu.vhd

mc8051_tmrctr.vhd

Memorias RAM y ROM

Mediante *CORE Generator*

mc8051_top.ucf

Restricciones

Opciones de configuración

■ Mediante *constantes* en fichero *mc8051_p.vhd*

-- Select whether to implement (1) or skip (0) the multiplier

-- Default: 1

constant C_IMPL_MUL : integer := 0;

-- Select how many timer/counter units should be implemented

-- Default: 1

constant C_IMPL_N_TMR : integer := 1;

-- Select how many serial interface units should be implemented

-- Default: C_IMPL_N_TMR ---**(DO NOT CHANGE!)**---

constant C_IMPL_N_SIU : integer := C_IMPL_N_TMR;

Núcleo de mc8051

Buses y puertos
de E/S
independientes

entity *mc8051_core* is

port (clk : in std_logic;

reset : in std_logic;

rom_adr_o : out std_logic_vector (15 downto 0);

ram_adr_o : out std_logic_vector (6 downto 0);

adrx_o : out std_logic_vector (15 downto 0);

rom_data_i : in std_logic_vector (7 downto 0);

ram_data_i : in std_logic_vector (7 downto 0);

ram_data_o : out std_logic_vector (7 downto 0);

datax_i : in std_logic_vector (7 downto 0);

datax_o : out std_logic_vector (7 downto 0);

...

p0_i : in std_logic_vector (7 downto 0);

p1_i : in std_logic_vector (7 downto 0);

p2_i : in std_logic_vector (7 downto 0);

p3_i : in std_logic_vector (7 downto 0);

p0_o : out std_logic_vector (7 downto 0);

p1_o : out std_logic_vector (7 downto 0);

p2_o : out std_logic_vector (7 downto 0);

p3_o : out std_logic_vector (7 downto 0));

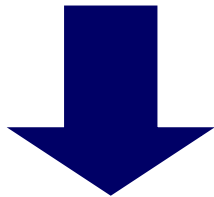
end *mc8051_core*;

Proyecto con Oregano en ISE

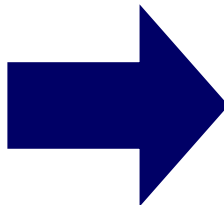
- Generación del **fichero de coeficientes (.coe)**
 - **Ensamble y enlace** del programa MCS-51 (**.hex**)
 - Conversión a **.coe** (**hex2coe**)
- Colocar ficheros fuente **.vhd**
- Generación de los bloques de **memoria RAM y ROM** internas
 - Mediante **CORE Generator**
- Declaración de los **componentes de memoria** en los ficheros **mc8051_top.vhd** y **mc8051_p.vhd**
- [Configurar las **opciones de síntesis** (**área/velocidad**)]
- Anadir fichero de restricciones (**.ucf**)

Conversión de .hex a .coe

```
:1000000075A0007580047455F590D2A0D2A1C2A14C  
:10001000C2A012001964FF80EF780579007A00DA37  
:06002000FED9FAD8F62219  
:00000001FF
```

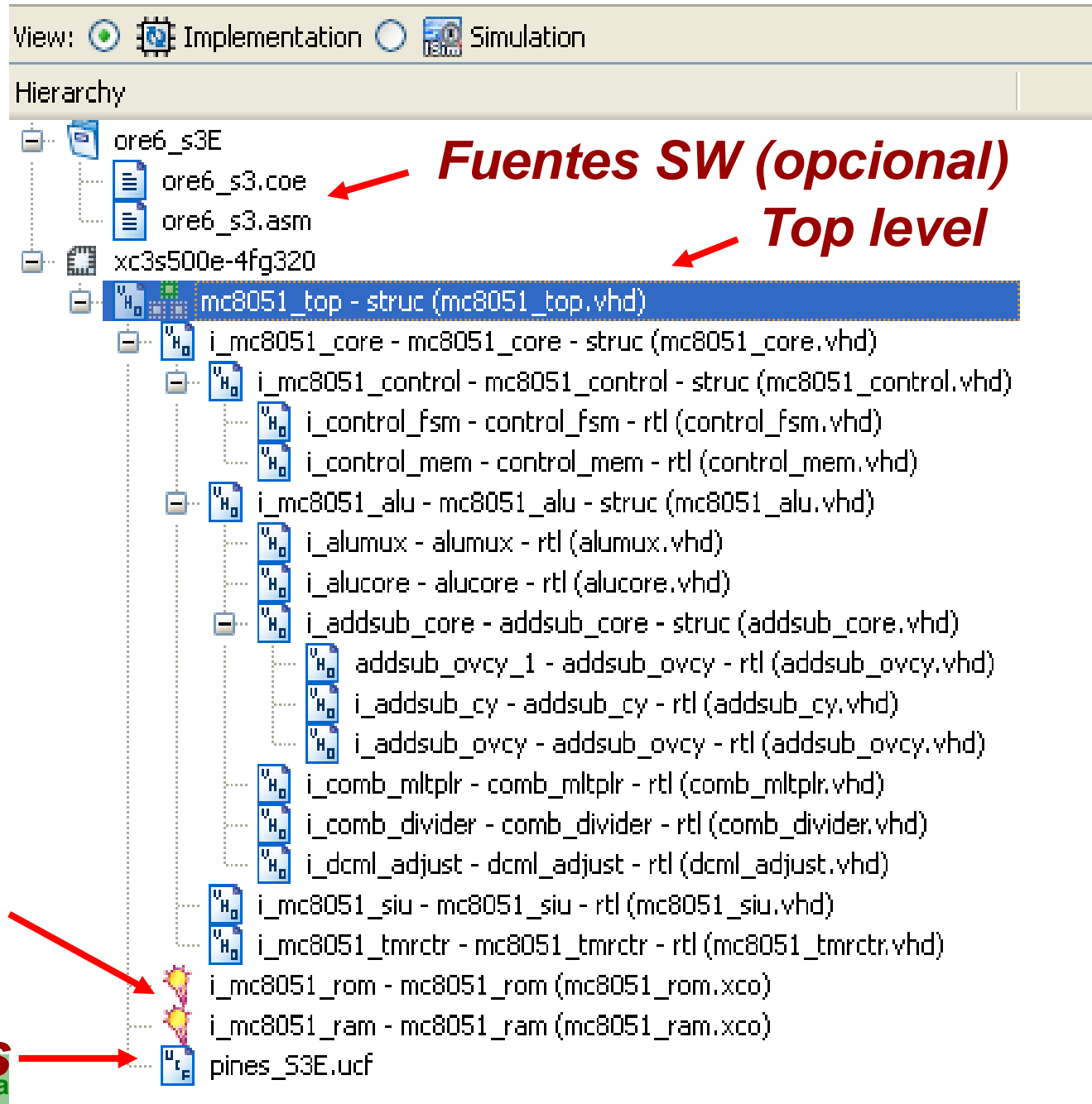


hex2coe



```
memory_initialization_radix = 16;  
memory_initialization_vector =  
75,  
A0,  
...  
F6,  
22;
```

Proyecto ISE con Oregano



ROM en *CORE Generator*

Block Memory Generator

Documents View

IP Symbol

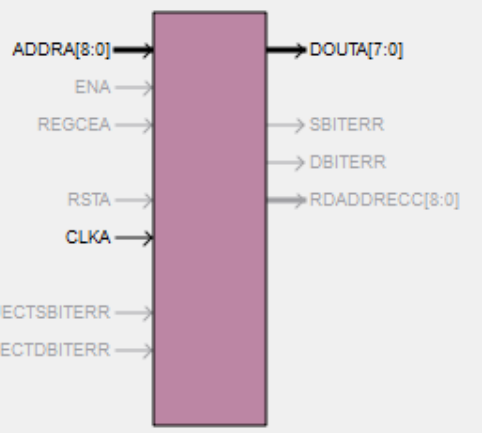


Diagram illustrating the Block Memory Generator IP block. The block is a vertical purple rectangle. Inputs on the left: ADDRA[8:0], ENA, REGCEA, RSTA, CLKA, INJECTSBITERR, and INJECTDBITERR. Outputs on the right: DOUTA[7:0], SBITERR, DBITERR, and RDADDRECC[8:0].

Nombres de las señales

LogiCORE Block Memory Generator 4.3

Optional Output Registers

Port A

- ☐ Register Port A Output of Memory Primitives
- ☐ Register Port A Output of Memory Core
- ☐ Register Port A Input of SoftECC logic
- ☐ Use REGCEA Pin (separate enable pin for Port A output registers)

Pipeline Stages within Mux: 0 Mux Size: 1225896x1

Latency added by output register(s):

Port A: 0 Clock Cycle(s)

For Spartan-6 Latency information may not be accurate

Memory Initialization

☒ Load Init File

Coe File: G:\uC_II\LABS\LAB1\LAB1\ore6_s3.coe **Fichero .coe**

☐ Fill Remaining Memory Locations

Remaining Memory Locations (Hex): 0

Datasheet < Back Page 3 of 5 Next > Generate Cancel Help

Señales de ROM interna

component **mc8051_rom**

port (**addra**: IN std_logic_VECTOR (**8 downto 0**);

clka: IN std_logic;

douta: OUT std_logic_VECTOR (**7 downto 0**);

rsta: IN std_logic);

end component;

attribute box_type : string;

attribute box_type of **mc8051_rom**: component is "black_box";

Depende de la capacidad de ROM requerida

Declararlas en CORE Generator

Señales de RAM interna

component **mc8051_ram**

port (addra: IN std_logic_VECTOR (**6 downto 0**);

clka: IN std_logic;

dina: IN std_logic_VECTOR (**7 downto 0**);

douta: OUT std_logic_VECTOR (**7 downto 0**);

ena: IN std_logic;

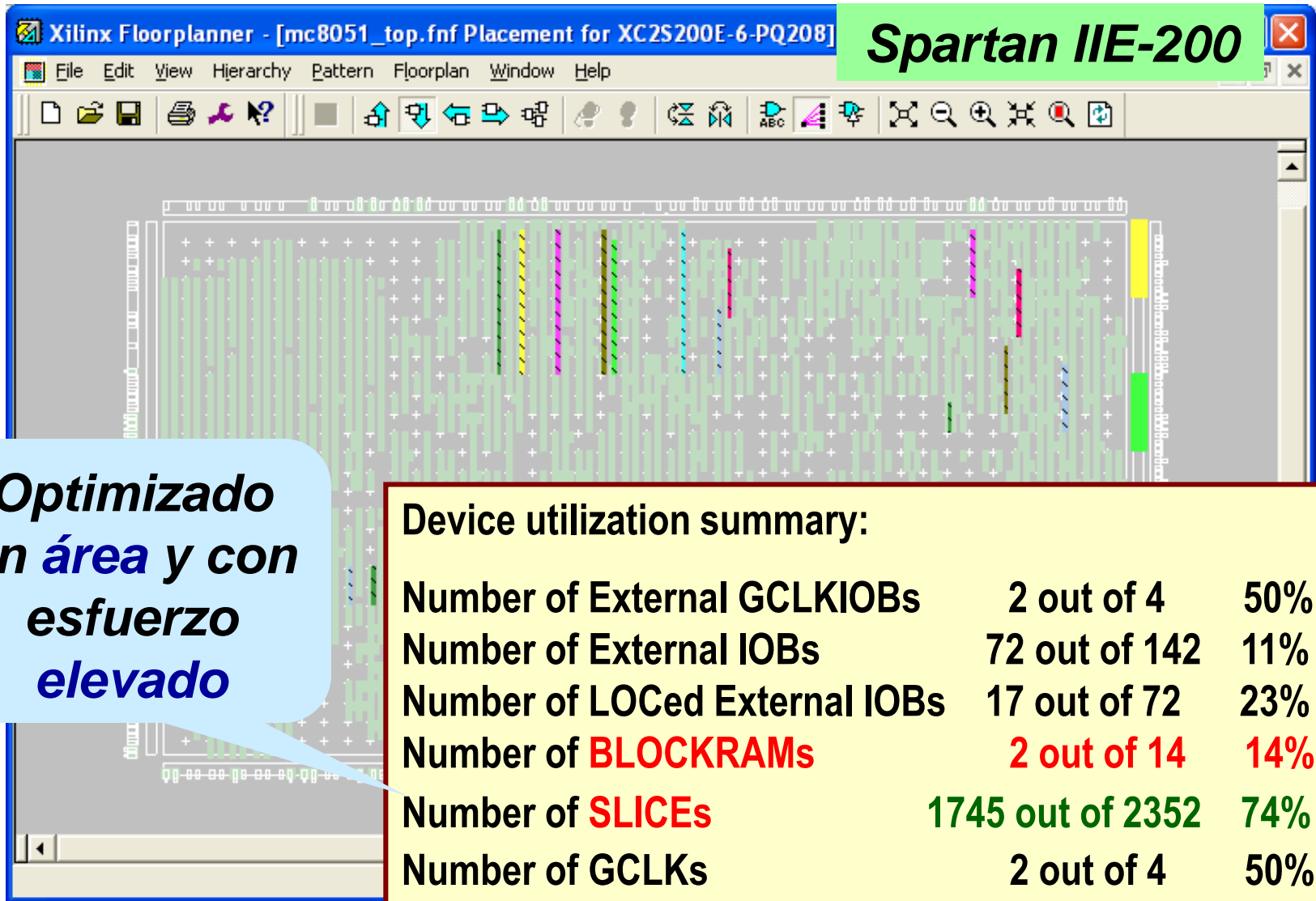
rsta: IN std_logic;

wea: IN std_logic_VECTOR (**0 downto 0**);

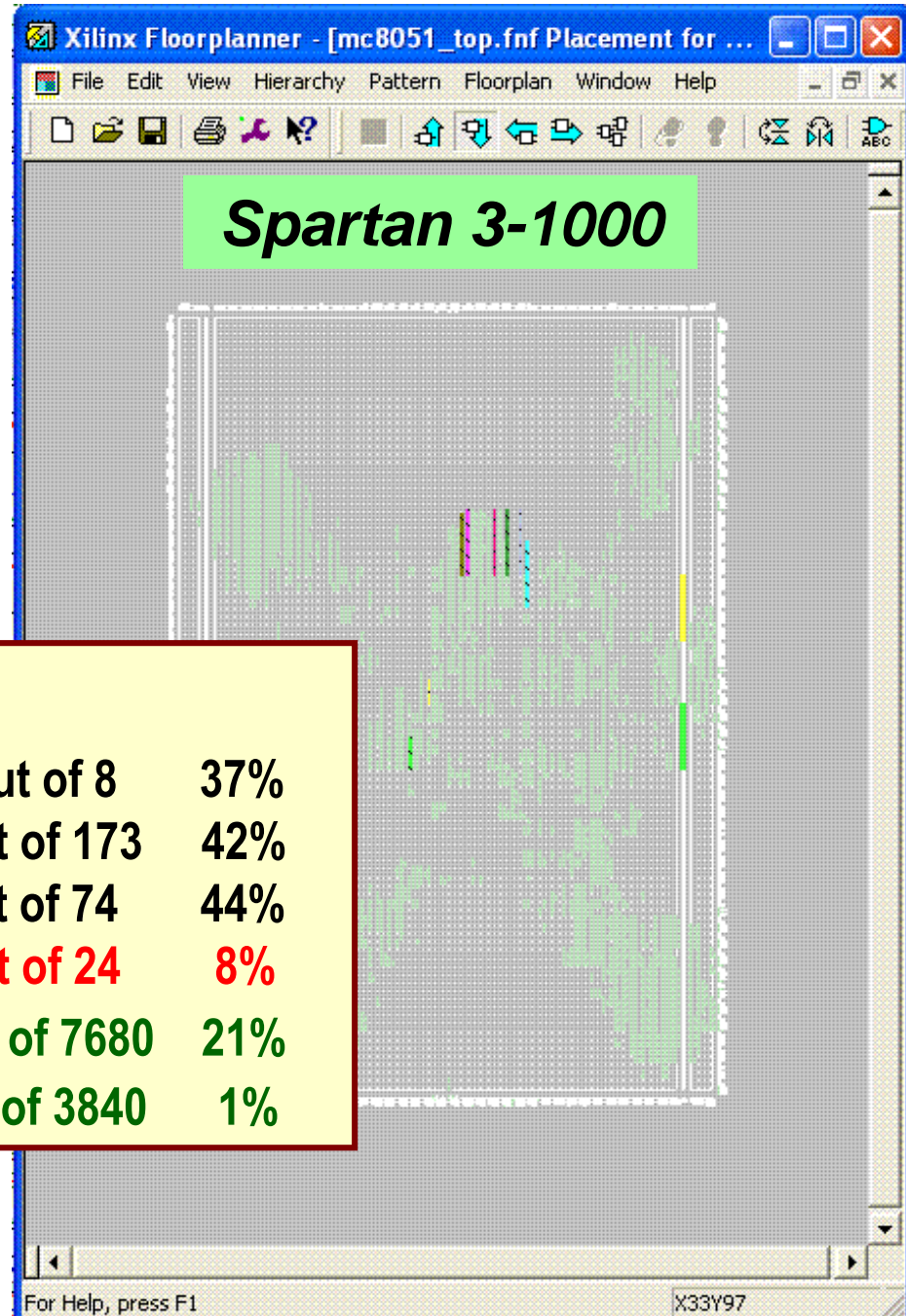
end component;

*Declararlas
en **CORE**
Generator*

Resultados de implementación



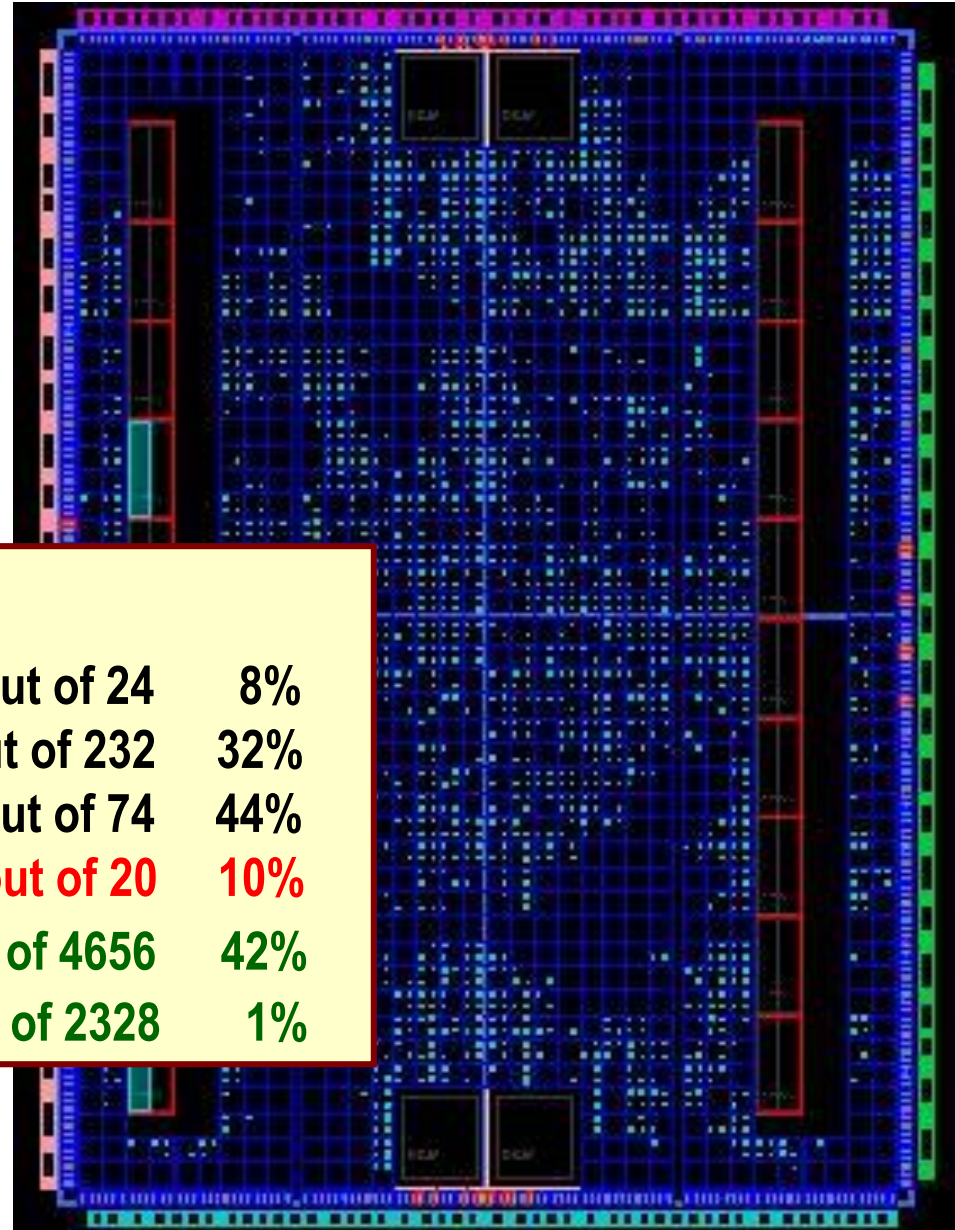
Resultados de implementación



Device utilization summary:

Number of BUFGMUXs	3 out of 8	37%
Number of External IOBs	74 out of 173	42%
Number of LOCed External IOBs	33 out of 74	44%
Number of RAMB16s	2 out of 24	8%
Number of SLICEs	1626 out of 7680	21%
Number of SLICEMs	32 out of 3840	1%

Resultados de implementación



Device utilization summary:

Number of BUFGMUXs	2 out of 24	8%
Number of External IOBs	75 out of 232	32%
Number of LOCed External IOBs	33 out of 74	44%
Number of RAMB16s	2 out of 20	10%
Number of SLICEs	1983 out of 4656	42%
Number of SLICEMs	32 out of 2328	1%

Resultados de implementación (*Spartan IIE200*)

Instruc	Unidades timers	Optimizado en	Esfuerzo en la síntesis	Esfuerzo en PAR	Utilización de slices
MUL, DIV, DAA	1	Velocidad	Normal	Normal	89%
MUL, DIV, DAA	1	Área	Alto	Alto	82%
-	1	Área	Normal	Normal	74%
-	1	Área	Alto	Alto	74%
MUL	1	Área	Alto	Alto	76%
MUL, DIV	1	Área	Alto	Alto	81%
-	2	Área	Alto	Alto	90%

Resultados de implementación (*Spartan-3A 700*)

Instruc	Unidades timers	Optimizado en	Esfuerzo en la síntesis	Esfuerzo en PAR	Utilización de slices
MUL, DIV, DAA	1	Área	Normal	Normal	31%
MUL, DIV, DAA	1	Velocidad	Normal	Normal	37%
MUL, DIV, DAA	2	Velocidad	Normal	Normal	47%
-	1	Velocidad	Normal	Normal	32%
DAA	1	Área	Normal	Normal	28%
MUL, DIV DIV -	1	Área	Normal	Normal	29%
MUL, DIV, DAA	2	Área	Normal	Normal	38%

Proyecto ISE de ore6

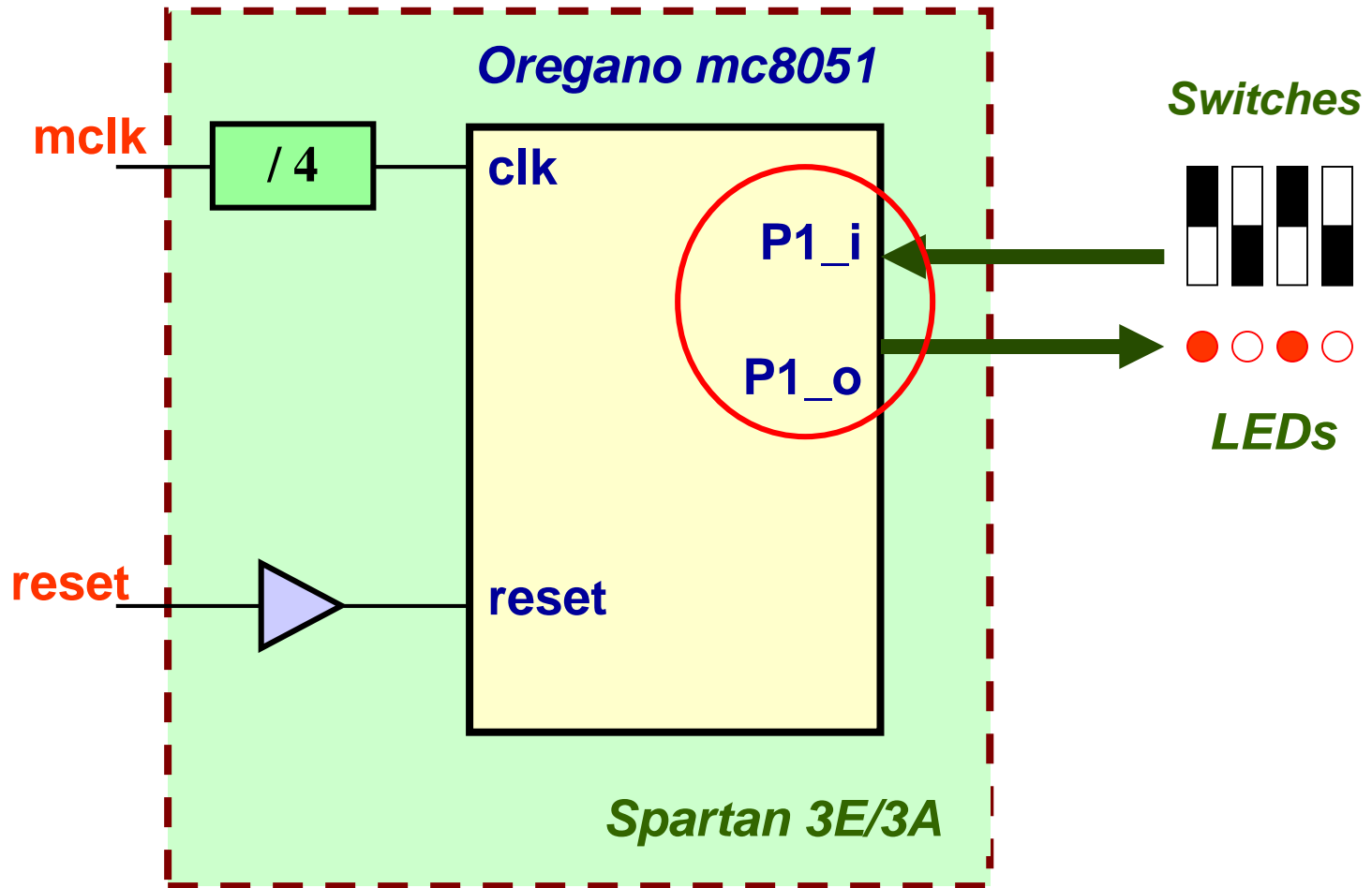
– Ficheros asociados

- *ore6_s3.asm*
→ *ore6_s3.coe*
- *mc8051_top.vhd*
- *mc8051_top.ucf*
- ...*restantes files .vhd*

– Proyecto ore6_s3E

- *ore6_s3E.xise*

Proyecto ore6_s3E



Proyecto ore1_EDIF

Obtención *preliminar* del *netlist* de **mc8051_core.vhd**

Proyecto *temp*

- Sólo sintetizar
- **NO** insertar *IOBUF*
- Netlist **mc8051_core.ngc**

Reducción
del tiempo de
desarrollo

➤ Ficheros asociados

- **ore1.asm**
→ **ore1.coe**
- **mc8051_top.vhd**
- **mc8051_top.ucf**
- **mc8051_core.ngc**

➤ Proyecto ore1_EDIF

- **ore1_EDIF.npl**

Limitaciones de Oregano

- **Alto consumo de recursos**
 - Muy **dependiente** de las **herramientas de síntesis**
 - Diferencias entre **XST** y **FPGA Express**
- **Configurabilidad limitada**
 - **Timer y UART relacionados**
 - **No permite añadir/eliminar puertos**
- **Diseño y configuración manual**
 - **No posee entorno de desarrollo**
- **Soporte técnico limitado**

Limitaciones de Oregano

- **Alto consumo de recursos**
 - Muy **dependiente** de las **herramientas de síntesis**
 - Diferencias entre *XST* y *FPGA Express*
- **Diseño y configuración manual**
 - **No posee entorno de desarrollo**
- **Soporte técnico limitado**

Trabajo independiente

- **Confeccione un programa para un 8051 con reloj de 12 MHz que cada dos segundos, medidos con ayuda de un temporizador atendido por interrupción, lea el estado de ocho interruptores conectados al puerto P1 y lo envíe a ocho LED conectados al puerto P1.**