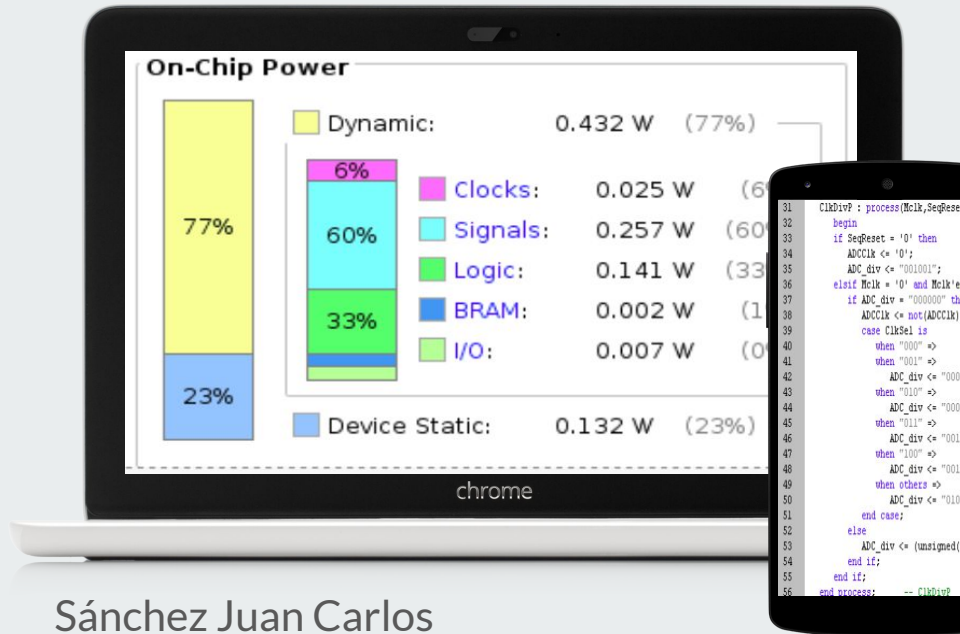


Estudio comparativo de simulaciones físicas en GPU y FPGA

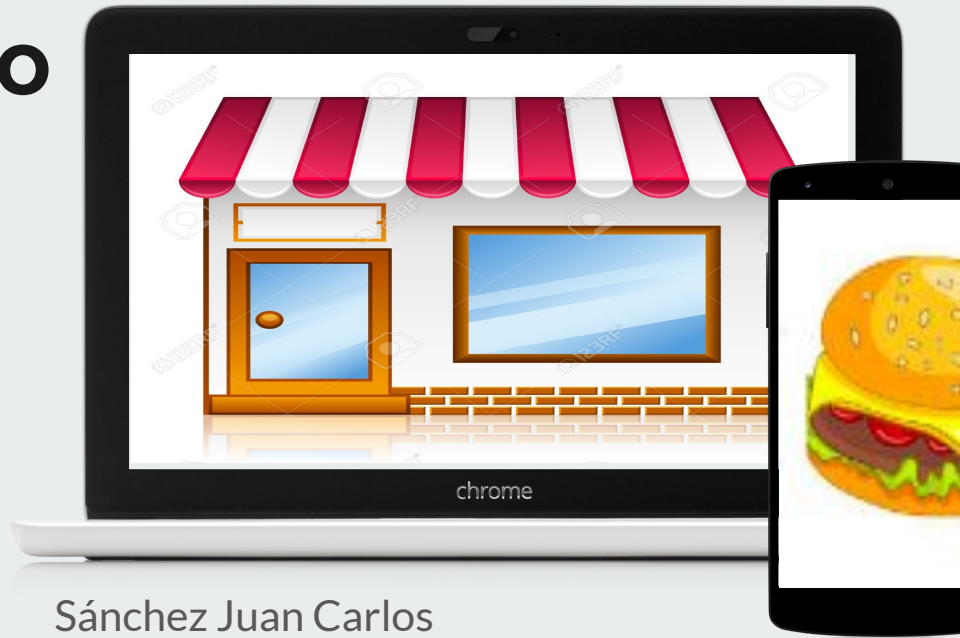
TRIBUNAL



Sánchez Juan Carlos
Director > Gonzalo Vodanovic

Estudio comparativo de simulaciones de **EMPLEADOS** en el comercio

¡INVITADOS



Sánchez Juan Carlos
Director > Gonzalo Vodanovic

Conceptos

CPU + Paralelismo + FPGA + GPU + Híbrido

Dueño

Concepto de trabajo

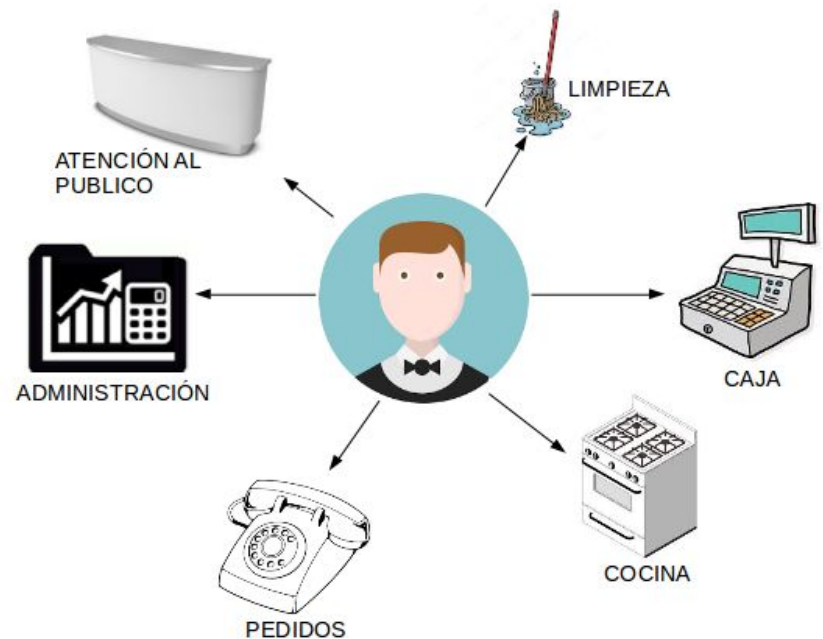
Tipos de empleado (blanco)

CPU

Unidad Central de Procesamiento, es el hardware dentro de un ordenador u otros dispositivos programables, que interpreta las instrucciones de un programa informático mediante la realización de las operaciones básicas aritméticas, lógicas y de entrada/salida del sistema.

Carlos Pascual Uriol

Llamaremos por su siglas al DUEÑO del local **CPU**...



PARALELISMO

El paralelismo es una forma de computación en la cual varios cálculos pueden realizarse simultáneamente, basado en el principio de dividir las grandes tareas para obtener varias tareas pequeñas

Como atender el NEGOCIO - CONCEPTO



Sr. CPU puede atender el NEGOCIO el mismo



Carlitos CPU,
puede ayudarse
de otras
PERSONAS para
un mejor
NEGOCIO

**DIVISIÓN DE TAREAS CON LA
AYUDA DE OTRAS PERSONAS**

Grupo de Personas hUmanas = GPU

E1

GPU

Unidad de procesamiento gráfico o GPU (Graphics Processing Unit) es un coprocesador dedicado al procesamiento de gráficos u operaciones de coma flotante, para aligerar la carga de trabajo del procesador central en aplicaciones

Mientras el CPU se ocupa de hacer nuevos convenios el GPU se ocupa del local (limpieza, cobro, etc.)

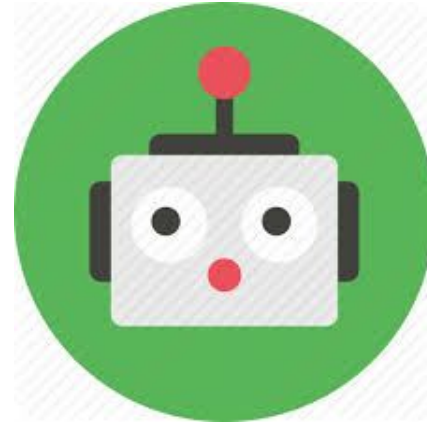


designed by freepik.com

FPGA

FPGA o matriz de puertas programables (del inglés field-programmable gate array) es un dispositivo programable que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada en el momento mediante un lenguaje de descripción especializado.

FPGA



INDUSTRIA S.A. - Creamos la herramienta a la medida del NEGOCIO

La máquina de hacer HAMBURGUESAS, el robot que limpia, el robot que da vuelta, lo que necesite...
La versión robótica lo espera

Hibrido

Implementación en FPGA a partir de código OpenCL.

ROBOT HIBRIDO

E3

TERMINATOR



Creemos la herramienta a la medida de su NEGOCIO, pero con más HUMANIDAD que nunca

El terminator de hacer HAMBURGUESAS, el robocop de la seguridad, C-3PO star wars de venta... Su versión robohumanidad lo espera

Comparativas => ELECCIÓN DE UN PROBLEMA

Comparativa de las diferentes arquitecturas bajo una simulación física... ¿Cuál?

EMPLEADO 1 **vs** EMPLEADO 2 **vs** EMPLEADO 3

Comparativa de los diferentes empleados bajo una simulación comercial... ¿Cuál?

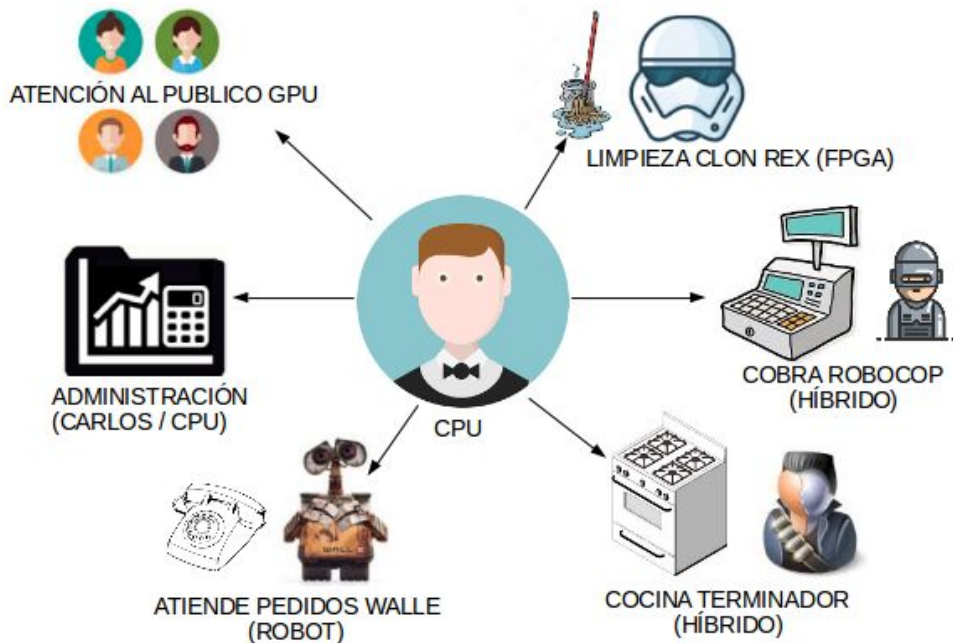
PROBLEMA

Para comparar necesitamos un resolver un problema y comparar los resultados

- Implementación de un problema de aplicación real en diferentes arquitecturas: GPU, FPGA y HÍBRIDA.
- Las arquitecturas tienen en común la resolución de problemas altamente paralelizables
- Involucrar operaciones con valores de punto fijo

DISTRIBUCIÓN DE CALOR

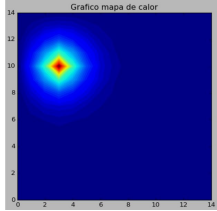
¿A quién contrato para hacer cierta tarea?



Son muchos los problemas que se presentan.
Comparar todas las posibilidades requiere mucho tiempo => SELECCIONAMOS UN PROBLEMA

PREPARACIÓN DE HAMBURGUESAS

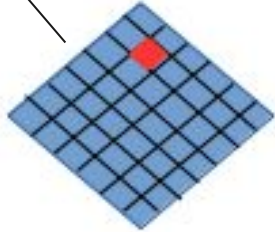
GRAFICO



$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

DISTRIBUCIÓN DE
TEMPERATURA
(Laplace)

Método
numérico



$T(i) = \text{Suma}(\text{vecinos}) / 4$
PARALLELIZABLE

Mediciones **ARQUITECTURAS**

- Tiempo de cómputo
- Potencia



tribunal



EMPLEADOS
ARQUITECTURAS



GPU



HÍBRIDO



FPGA

Mediciones **EMPLEADOS**

- Tiempo de preparación
- Gasto en \$\$

invitado

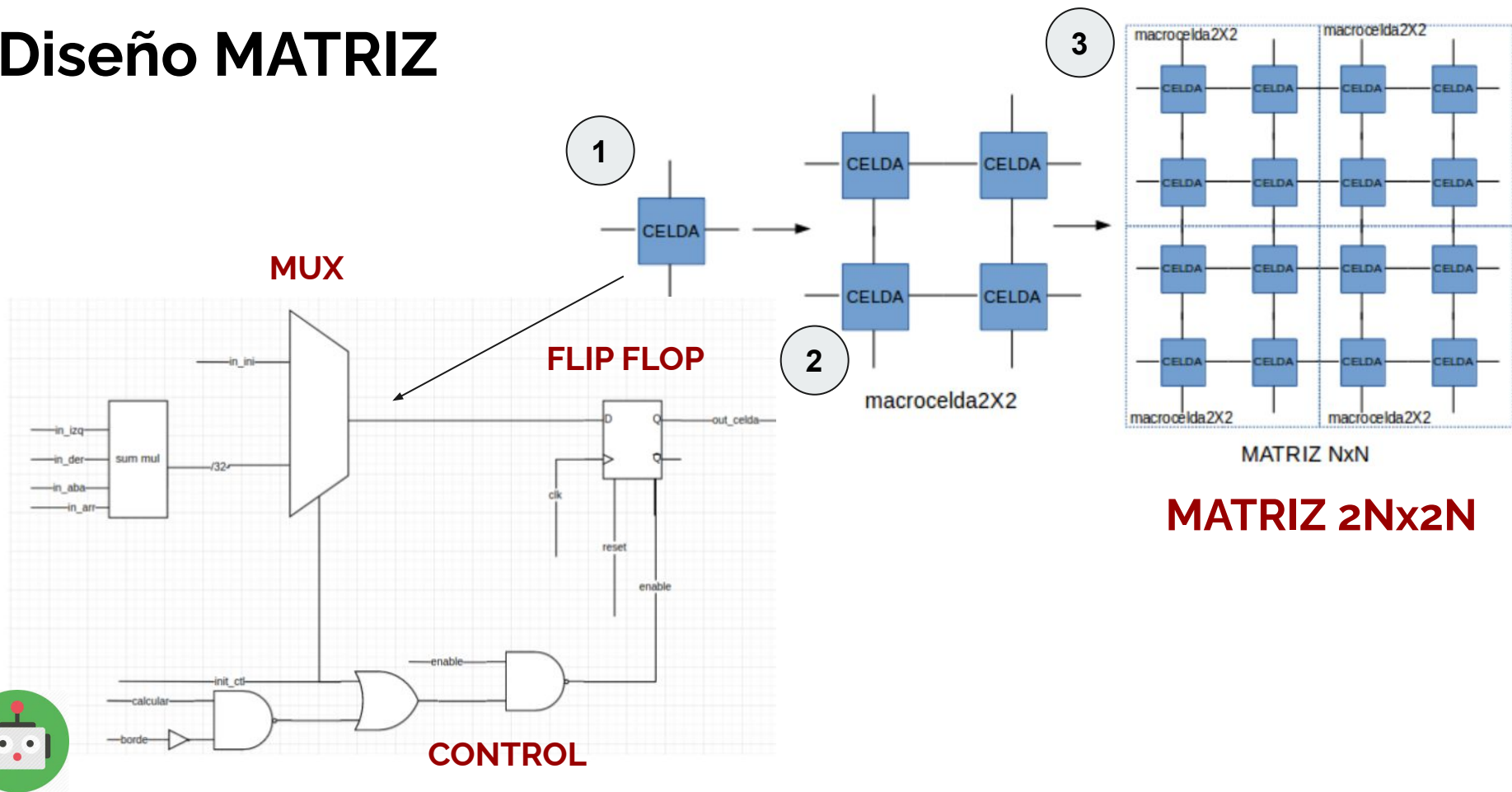


iMPLEEntación

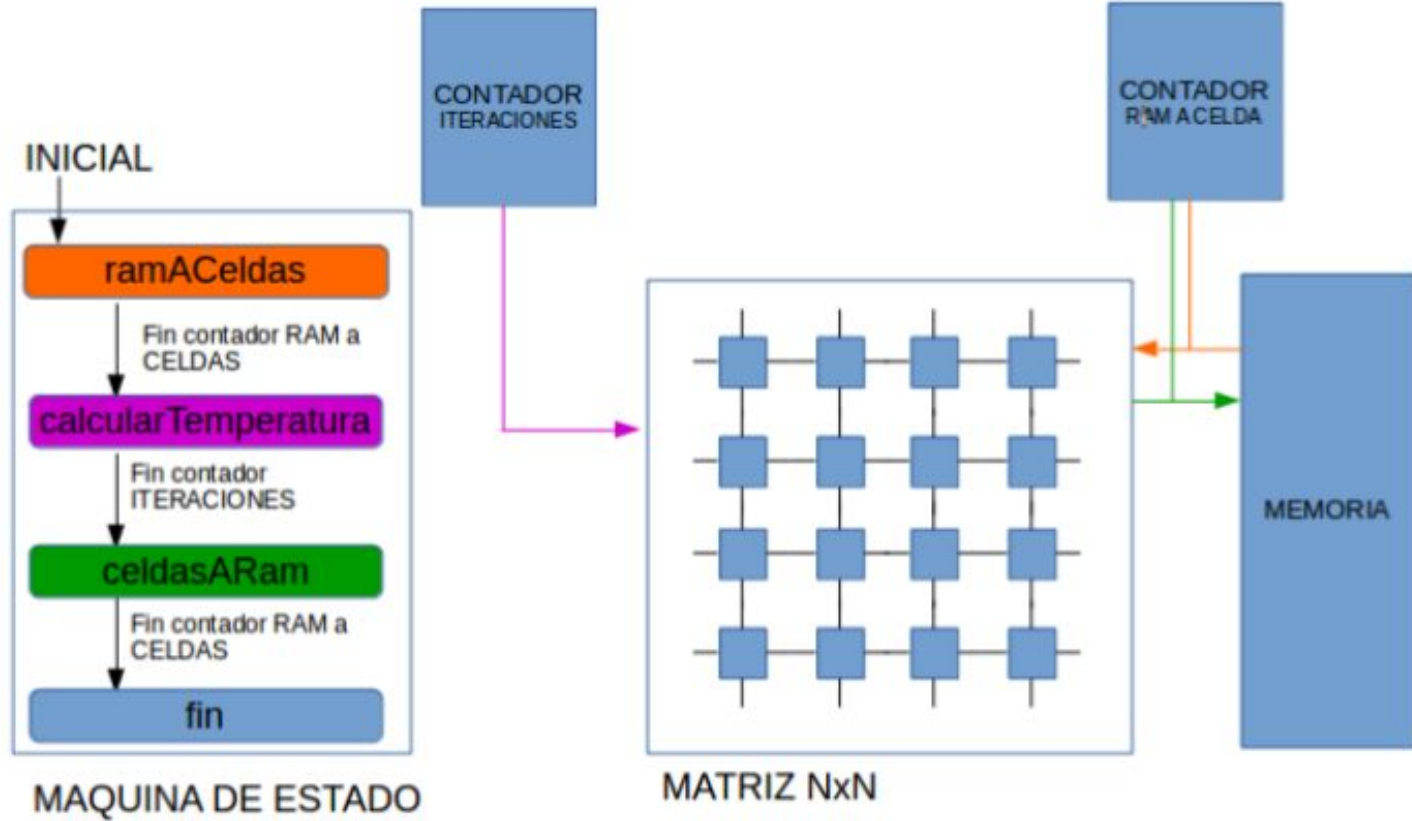
¿ Cómo se implementó en cada arquitectura?

iNVITADos => sigan solos!

Diseño MATRIZ



MATRIZ 2Nx2N



PLACA DISPONIBLE XILINX ARTIX 7



Código Kernel OpenCL

2

Kernel local

```
1  __kernel void calculo_calor(  
2      __global float * matrix,  
3      __global bool * bordes,  
4      int iteraciones)  
5  {  
6      __local float current[24*24];  
7      int p = get_global_id(0);  
8      current[p] = (float)matrix[p];  
9      barrier(CLK_LOCAL_MEM_FENCE);  
10     while(iteraciones--){  
11         current[p] = bordes[p]?current[p]:  
12             (current[p+1]+current[p-1]+current[p+N]+current[p-N]) *0.25;  
13         barrier(CLK_LOCAL_MEM_FENCE);  
14     }  
15     matrix[p] = (float)current[p];  
16 }
```

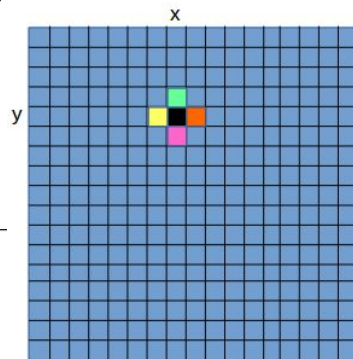
Todos los WI deben pertenecer al mismo WG ya que es necesaria la sincronización en cada iteración.

Cada CELDA = WI

PLACA DISPONIBLE GEFORCE GTX 980

1

Matriz NxN



- Vecino superior
- Vecino inferior
- Vecino derecho
- Vecino izquierdo

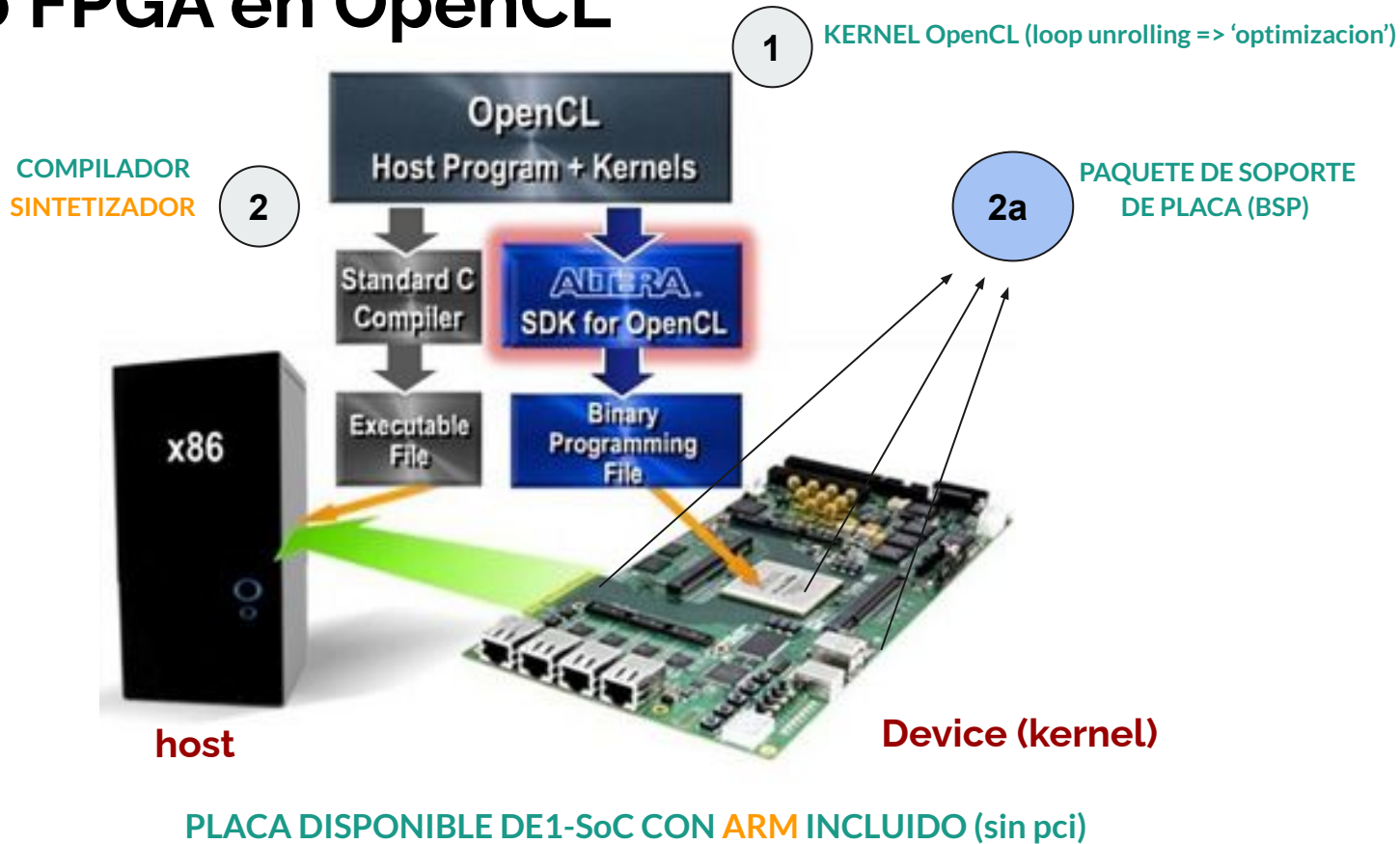
Vector de valores (float)



Vector de bordes (boolean)



Proceso FPGA en OpenCL

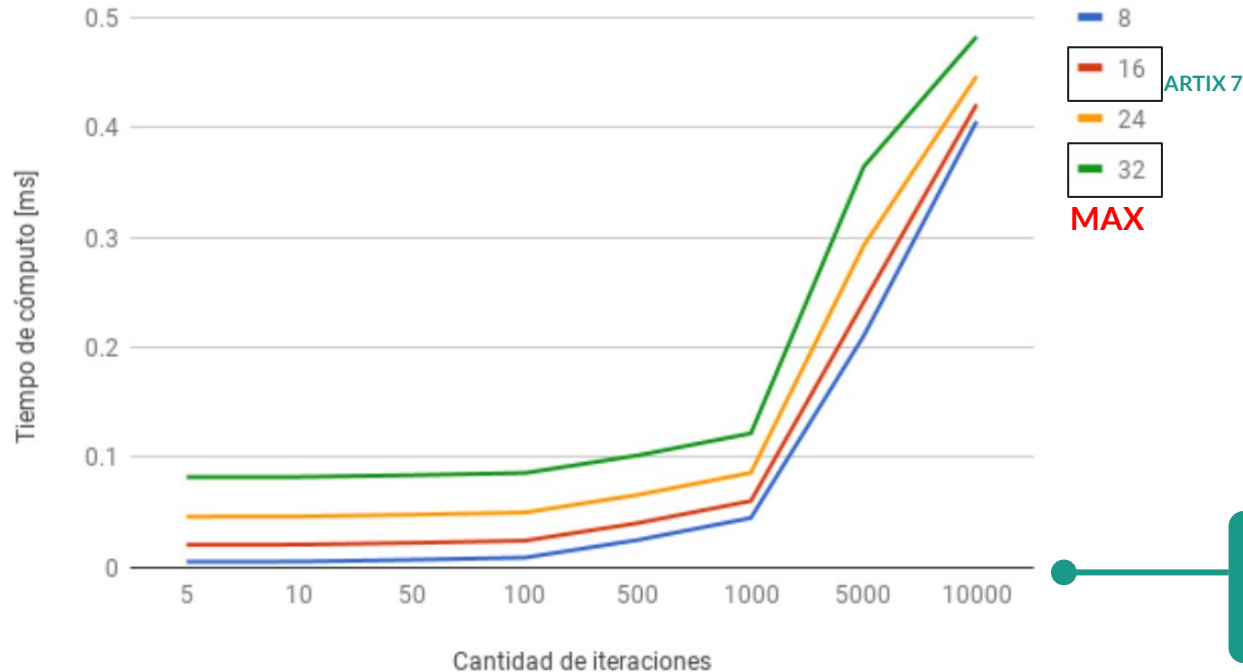


Mediciones obtenidas

- Tiempo en ms
 - Se tomaron diferentes iteraciones: 10...10.000
 - Diferentes dimensiones de la matriz: 8x8, 16x16, 32x32, 44x44
 - Valores en punto fijo
 - Para la comparativa NO se contemplaron los tiempo de L/E
-

Tiempo total de cómputo

Tiempo de Lectura/Escritura no fueron contemplados



Dimensión	Potencia (W)
8	0.099
16	0.11
24	0.364
32	0.564

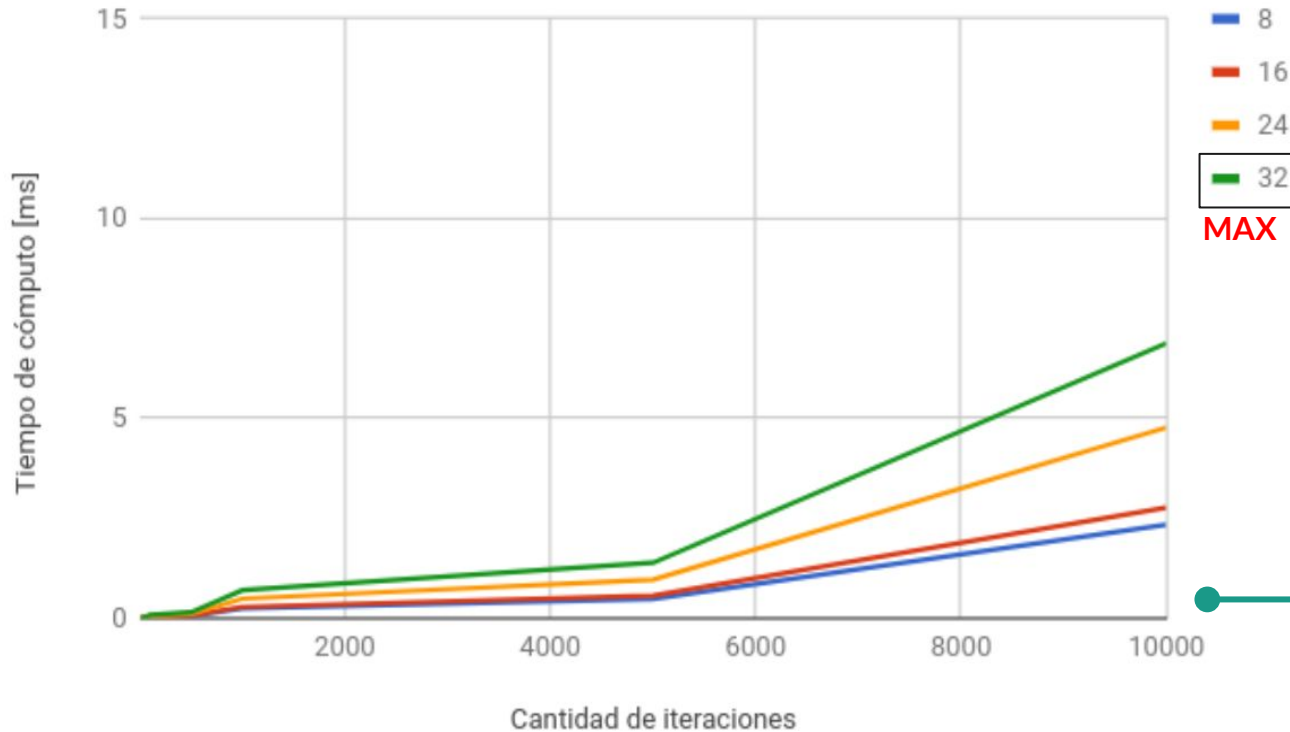
El consumo de potencia se ve afectado por el tamaño

Tiempo de cómputo no es muy afectando por la dimensión pero si por la cantidad de iteraciones

PLACA DISPONIBLE XILINX ARTIX 7 + ARTIX 7 > gama



Tiempo total de cómputo



Matriz	Potencia (W)
32	62.02
24	59.9
16	57.87
8	57.26

El consumo de potencia aumenta un 8 % entre la matriz de 8x8 y 32x32 celdas.

Tiempo de cómputo incrementa de forma no proporcional

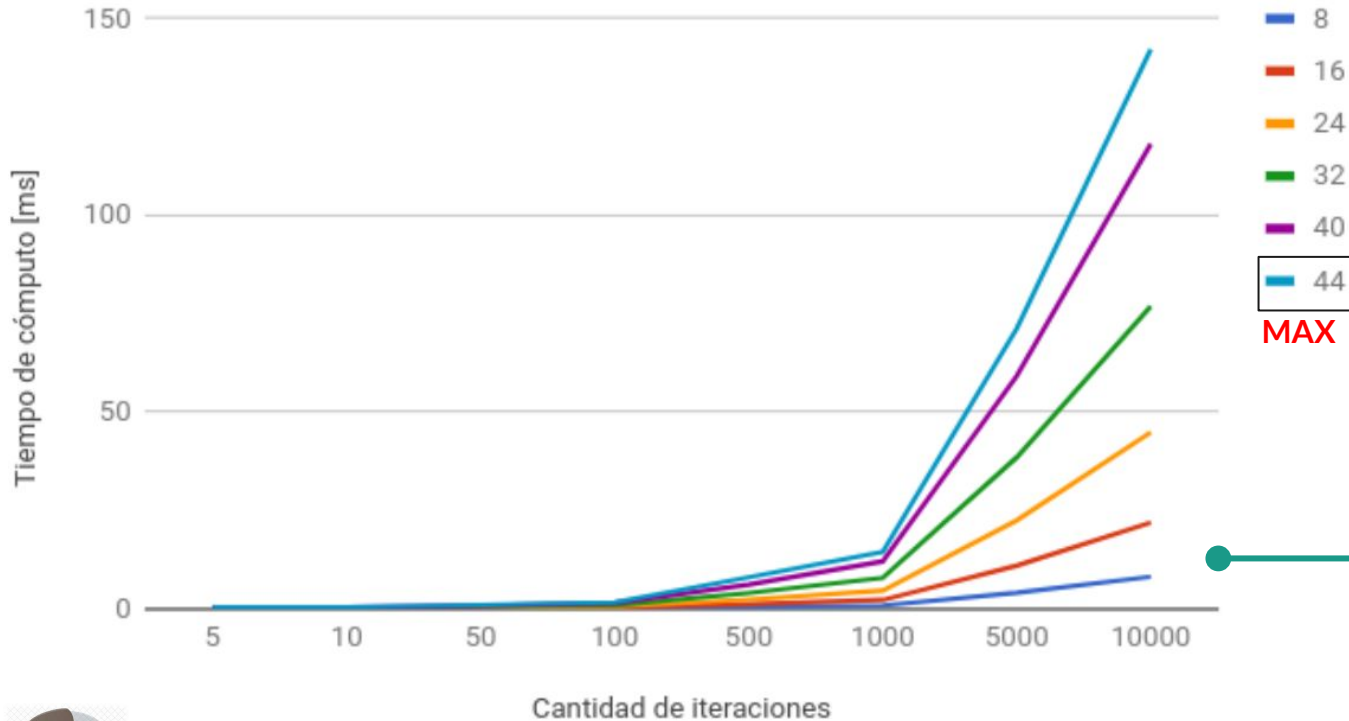
Tiempo L/E depende levemente del tamaño de la matriz

PLACA DISPONIBLE GEFORCE GTX 980



Tiempo total de cómputo

Requiere mayor comprensión del flujo del diseño



Matriz	Potencia (W)
44	1.03
32	1.06
24	1.05
16	0.8
8	0.74

El consumo de potencia aumenta un 40 % entre la matriz de 8x8 y 32x32 celdas.

Tiempo de cómputo dependiente de la dimensión de la matriz

Tiempo L/E depende levemente del tamaño de la matriz

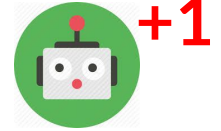
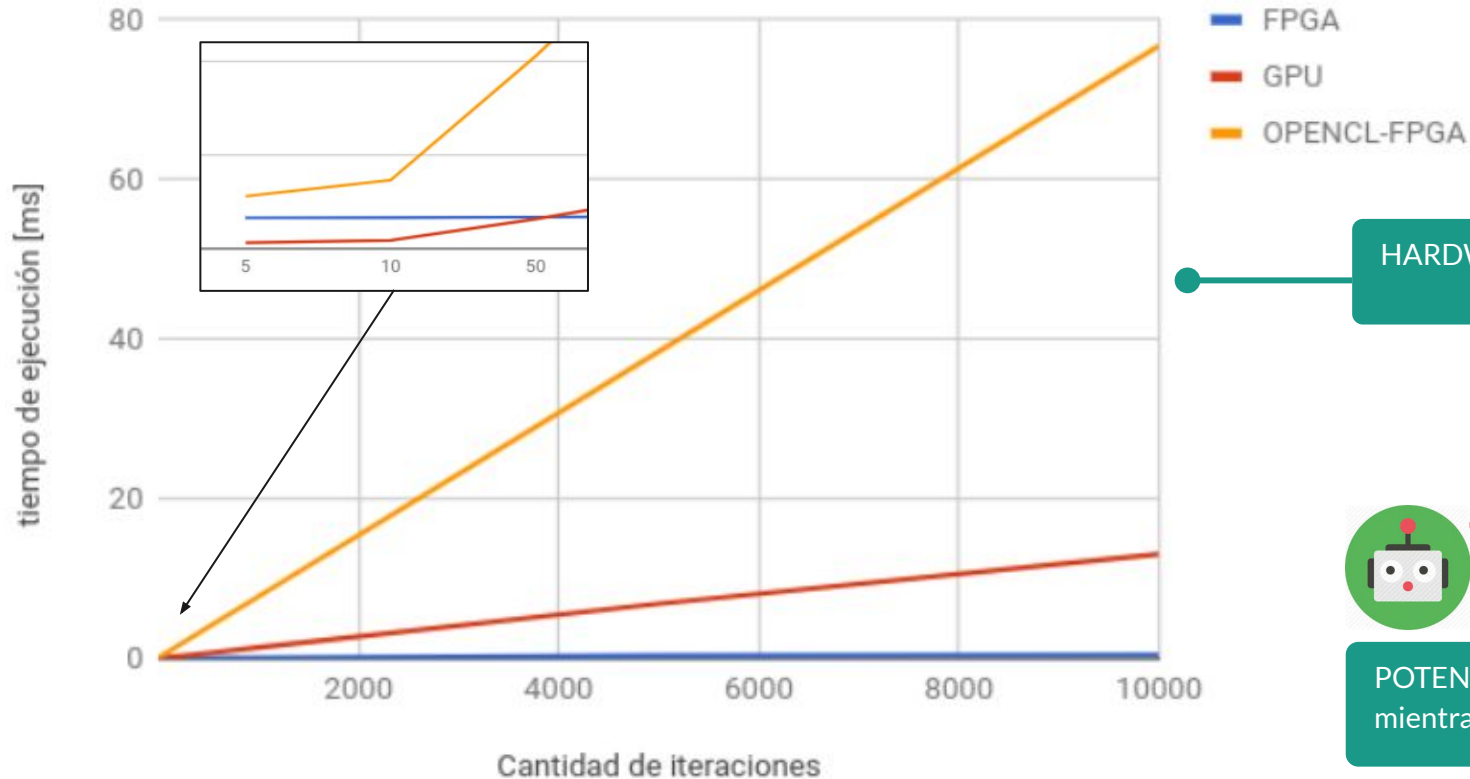
PLACA DISPONIBLE DE1-SoC CON ARM INCLUIDO



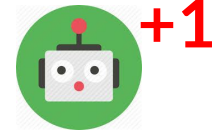
Comparativas

Tiempo total de cómputo

N=32x32



HARDWARE A MEDIDA



POTENCIA: GPU en el orden 60w
mientras FPGA no supera 2W

Conclusión



En crecimiento y mejorando



Mejor performance (tiempo de ejecución y potencia)



Menor tiempo de implementación

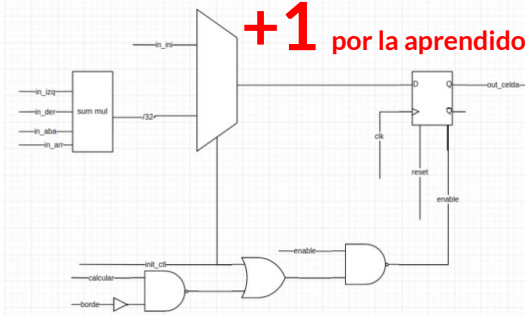
Opinión del Tesista

Me pareció...

→ EXCELENTE EXPERIENCIA



+1 por la aprendido



OpenCL

+1 por la aprendido



FPGA
1000x1000

Constraints + Optimización del sintetizador y el implementador (P&R) para obtener un diseño óptimo + Diseño de arquitectura a nivel electrónica + Simulaciones a varios niveles + Calculo de potencia +

Tiempo de síntesis e implementación + Tiempo de diseño



Carlos CPU trabajarán conjuntamente



Cálculo de proposito general



+ Futuro

GRACIAS

