# Bloque I

1. ¿En qué consiste el Codiseño?
2. Es el “CoWorking” aplicado a la electrónica**.**
3. **En el diseño del hardware y el software de un sistema basado en procesadores**
4. Se basa en el uso de equipos multidisciplinares para a la realización de un diseño
5. ¿Qué tipo de procesador Hard suelen incluir las FPGAs actuales de IntelFPGA?

*A. Nios II*

**B. ARM Cortex A9**

C. Intel i9.3. En el mercado de las FPGAs.¿Cuál es la empresa mayoritaria?

1. **Xilinx.**

B. ARM.

C. IntelFPGA.

1. En el sistema de visualización de la pantalla MTL2 visto en las prácticas. ¿Qué significa que el direccionamiento de la memoria que contiene el fondo sea del tipo X-Y?**A. Que la dirección de la memoria donde se encuentra almacenado un píxel de la imagen, se obtiene concatenado la coordenada X de la posición del píxel en la imagen con la coordenada Y.**

B. Que la dirección de la memoria donde se encuentra almacenado un píxel de la imagen, se obtiene sumando la coordenada X de la posición del píxel en la imagen con la potencia de 2 de la coordenada Y.

C. Que la dirección de la memoria donde se encuentra almacenado un píxel de la imagen, se obtiene sumando la coordenada X de la posición del píxel en la imagen con la coordenada Y.

1. ¿Cuál de las siguientes afirmaciones sobre la interface Avalon Streaming de Nios II es correcta?A. Se trata de un enlace punto a punto entre un emisor y varios receptores con una alta tasa de transferencia de datos.
2. **Se trata de un enlace punto a punto entre un emisor y un receptor con una alta tasa de transferencia de datos.**

C. Se trata de un enlace serie punto a punto entre un emisor y un receptor con una alta tasa de transferencia utilizando un solo bit de datos

1. ¿Qué tipo de procesador es Nios II?**A. RISC**

B. CISC

C. PISC

1. En un procesador con cache de datos. ¿Cuál de las siguientes afirmaciones es correcta?

A. El acceso a memoria de datos debe realizarse directamente sin usar la cache

**B. El acceso a periféricos debe realizarse directamente sin usar la cache**

C. Es recomendable utilizar la cache para acceder a los periféricos.

1. En la función MTL\_text.¿Para qué se emplea la instrucción offset = (y << 6) + x;?
2. Para determinar la posición del carácter en el modo de direccionamiento X-Y
3. Para determinar el offset que se debe añadir a la posición del carácter con respecto a los pixeles de la pantalla.
4. Para determinar la posición del carácter en el modo de direccionamiento continuo.
5. En la IP del Interval Timer.¿Por qué se configura el valor del contador mediante dos operaciones de escritura?A**. Porque el bus de datos del Interval Timer es de 16 bits y el contador es de 32 bits**.
6. Porque el bus de datos del Nios II es de 16 bits y el Interval Timer es de 32 bits.

C. Para configurar el valor del contador y su modo de funcionamiento.

1. ¿Dónde se ubica y para que se utiliza el archivo system.h?
2. **El archivo system.h se encuentra en el proyecto BSP y se utiliza como “header” del proyecto de la Aplicación porque contiene la información del hardware.**

B. El archivo system.h se encuentra en el proyecto de la Aplicación y se utiliza como “header”.

C. El archivo system.h se encuentra en el proyecto BSP y solo se utiliza en el empleo del HAL.

1. En el flujo de diseño de un SoPC.¿Cuál es el vínculo entre el hardware y el software?A**. El diseño hardware debe proporcionar al diseño software un archivo de configuración del sistema procesador.**
2. El diseño software debe proporcionar al diseño hardware un archivo de requisitos del sistema procesador
3. Ninguno, son totalmente independientes
4. . ¿Cuál de las siguientes afirmaciones sobre el interface Avalon Streaming de Nios II es correcta?

A. Se trata de un enlace punto a punto entre un emisor y varios receptores con una alta tasa de transferencia de datos

1. **Se trata de un enlace punto a punto entre un emisor y un receptor con una alta tasa de transferencia de datos.**
2. Se trata de un enlace serie punto a punto entre un emisor y un receptor con una alta tasa de transferencia utilizando un solo bit de datos.
3. En el sistema de visualización de la pantalla MTL2 visto en las prácticas. ¿Dónde emplazamos el texto de caracteres ASCII para su visualización?
4. **En el Background.**
5. En el Foreground.
6. Indistintamente en el background o en el foreground.
7. En la gestión de los pulsadores por medio de las interrupciones. ¿Qué puerto del PIO se debe leer para determinar que pulsador se ha accionado?
8. **Edgecapture**
9. Interruptmask.
10. Data.
11. ¿Qué función tiene el registro “interruptmask” de un PIO?

A. Contiene el número de la petición de interrupción que corresponde al PIO.

B. Borra la petición de interrupción del PIO.

C. **Habilitar o deshabilitar la petición de interrupción de cada bit del puerto paralelo.**

1. ¿Cuál de las siguientes afirmaciones sobre el archivo .sopcinfo es correcta?

A. El archivo. sopcinfo contiene información general sobre los parámetros del proyecto.

**B. El archivo. sopcinfo contiene la información sobre el sistema necesaria para la creación del proyecto de Eclipse.**

C. El archivo. sopcinfo es generado por el Editor del BSP de Eclipse.

1. En la función MTL\_box, se utiliza la instrucción: offset = (row << 9) + col; ¿Por qué se desplaza la variable row, 9 bits?

A. Porque la memoria de la pantalla almacena 240 píxeles de alto (variable row) siendo la potencia de 2 superior más próxima 256.

**B. Porque la memoria de la pantalla almacena 400 píxeles de ancho (variable col) siendo la potencia de 2 superior más próxima 512.**

C. Porque se utiliza un direccionamiento continuo de la memoria de la pantalla siendo la manera de calcular la posición del píxel.

1. ¿Cuál es la ruta para acceder a los ficheros (name\_file) empaquetados en un archivo ZIP (zipfile.zip) almacenado en la memoria Flash?

A. /mnt/zipfile.zip/name\_file

**B. /mnt/rozipfs/name\_file**

C. /dev/mnt/rozipfs/name\_file

1. En el sistema de visualización de la pantalla MTL2 visto en las prácticas. ¿Cuánto espacio de memoria hace falta para almacenar una imagen del fondo o background?

**A. 512 x 240 posiciones de memoria.**

B. 400 x 240 posiciones de memoria.

C. 800 x 480 posiciones de memoria.

1. ¿Cuál de las siguientes afirmaciones sobre las versiones de Nios II es correcta?

A. La versión “Economy de Nios, siempre ocupa el mismo espacio en todos los dispositivos.

B. La velocidad y ocupación del procesador depende de la familia del dispositivo escogido.

**C. La versión “Fast” está optimizada para obtener la máxima velocidad**

1. En una FPGA. ¿Qué se entiende por procesador Soft?

**A. Se trata de un procesador que se sintetiza en la FPGA utilizando los elementos lógicos de la**

**misma.**

B. Cualquier procesador de una FPGA es del tipo Soft.

C. Aquel procesador de bajas prestaciones

1. ¿En que memoria se almacenan las imágenes del fondo o background que se visualizan en la pantalla MTL2?

A. SDRAM.

B. On-Chip.

**C. SRAM**

# Bloque II y III

1. ¿Cuál de los siguientes elementos NO corresponde a un elemento del interface Avalon Memory Mapped?

**A. Source**

B. Slave

C. Master

2. Las señales chipselect, read\_n, readdata, writedata, write\_n

**A. Son señales de un interface Avalon-MM**

B. Son señales de un interface Avalon-ST

C. No corresponde a ningún interface de los anteriores.

3. ¿Cuál de las siguientes afirmaciones es cierta? La señal waitrequest:

A. Es obligatoria para los interfaces de tipo Avalon-MM.

**B. Es necesaria en un interface Avalon-MM cuando un periférico no tiene un tiempo respuesta**

**conocido.**

C. Es obligatoria para los interfaces de tipo Avalon-ST

4. Queremos diseñar un periférico para poder conectarlo al Nios en Platform Designer, asignarle un rango de direcciones de memoria y desde el micro acceder a él. ¿Qué interface debemos definir para el periférico?:

A. Avalon MM Master

B. Avalon ST Sink

**C. Avalon MM Slave**

5. En un programa en C para un sistema basado en Nios tengo definida la siguiente sentencia volatile int \* interval\_timer\_ptr = (int \*) AVALON\_TIMER\_BASE; donde VALON\_TIMER\_BASE es una etiqueta con la dirección de memoria del componente AVALON\_TIMER. Si quiero escribir el registro 3 del AVALON\_TIMER con todo unos.¿Cuál de las siguientes soluciones es la correcta?:

A. \*(interval\_timer\_ptr + 0x3) = FFFFFFFF;

B. \*(interval\_timer\_ptr + 0x3) = 0x11111111;

**C. \*(interval\_timer\_ptr + 0x3) = 0xFFFFFFFF;**

6. ¿Qué solución tecnológica suele alcanzar mayores frecuencias de reloj?

**A. CPU**

B. GPU

C. FPGA

7. Queremos declarar una variable de tipo punto fijo sin signo con VHDL: variable x : unresolved\_ufixed(4 downto -3)¿Cuantos bits tiene la parte entera y cuantos bits la parte decimal?

A. 4 bits la parte entera y 3 bits la parte decimal

**B. 5 bits la parte entera y 3 bits la parte decimal**

C. 4 bits la parte entera y 4 bits la parte decimal

8. Indicad cuál de las siguientes afirmaciones es FALSA

A. En la técnica 1 utilizamos un interfaz de tipo Custom Instruction para conectar el acelerador con el host (nios2)

**B. En la técnica 2 utilizamos un interfaz de tipo Custom Instruction para conectar el acelerador con el host (nios2)**

C. En la técnica 3 utilizamos un interfaz de tipo Custom Instruction para conectar el acelerador con el host (nios2)

9. Indicad cuál de las siguientes afirmaciones es FALSA

A. La tendencia en el mundo de los aceleradores es poder diseñarlos directamente con lenguajes más propios del mundo software, como el OpenCL, y utilizar compiladores automáticos que generen el hardware acelerador

**B. Los aceleradores creados a partir de OpenCL son exclusivamente para FPGAs**

C. Los aceleradores creados a partir de OpenCL con FPGAs pueden trabajar en sistemas embebidos en donde el host y el kernel están bajo el mismo encapsulado de la FPGA o en sistemas en donde el host y el kernel están conectados via PCI-express , estando el kernel en la FPGA

10. Los puertos que salen de un periférico definido en Platform Designer que queremos conectar a los leds disponibles en la DE2-115 deben ser definidos como

A. Interface Avalon Streaming Source.

**B. Interface Conduit.**

C. Interface Interrupt Sender

11. ¿Cuál de las siguientes afirmaciones es cierta? Si nuestro diseño para Platform Designer es capaz de generar una señal de interrupción:

A. Nuestra señal de interrupción pertenecerá a un interface de tipo Interrupt Receiver y necesitaremos un MFB de tipo Avalon Interrupt Sink para verificarlo.

**B. Nuestra señal de interrupción pertenecerá a un interface de tipo Interrupt Sender y necesitaremos un MFB de tipo Avalon Interrupt Sink para verificarlo.**

C. Nuestra señal de interrupción pertenecerá a un interface de tipo Interrupt Receiver y necesitaremos un MFB de tipo Avalon Interrupt Source para verificarlo

12. Indicad cuál de las siguientes afirmaciones es FALSA

A. Los consumos de los aceleradores implementados con FPGAs suelen ser los más bajos frente a CPU y GPU

B. La mayoría de los aceleradores implementados con FPGAs basan su aceleración en base a una aproximación cuyo error pueda ser aceptable desde el punto de vista de la aplicación

**C. Arquitecturalmente los aceleradores de altas prestaciones tienden hacia el paralelismo y la homogeneidad**

13. Si medimos el rendimiento de un acelerador en términos exclusivamente de velocidad, cual es la tecnología electrónica imbatible hoy en día suponiendo que trabajamos en coma flotante precisión simple.

A. FPGA

B. CPU

**C. GPU**

14. ¿Cuál de las siguientes respuestas sería la mejor opción? Si mi DUV posee un interface Avalon-MM de tipo slave, un interface Avalon-ST sink y un interface Avalon-ST de tipo source. Para el banco de pruebas necesito los siguientes modelos funcionales de bus:

**A. Uno de tipo Avalon-ST source, uno de tipo Avalon-ST sink y un Avalon-MM de tipo Master.**

B. Uno de tipo Avalon-ST source, uno de tipo Avalon-ST sink y un Avalon-MM de tipo Slave.

C. Un Avalon-MM Master

15. Indicad cuál de las siguientes afirmaciones es FALSA

A. Las ROM síncronas trabajan con latencia 1

B. Implementamos las ROM síncronas para poder utilizar para su implementación las memorias embebidas de la FPGA

**C. La inicialización de la memoria ROM se realiza mediante una escritura síncrona**

16. De las 3 claves tecnológicas disponibles en los sistemas embebidos, cual es la que mayormente utilizaremos para nuestras tareas de aceleración en Codiseño Hardware-Software

A. La tecnología de Circuito Integrado

B. La tecnología de Diseño

**C. La tecnología de procesador**

17. ¿Cuál de las siguientes afirmaciones es falsa? En un banco de pruebas los modelos funcionales de bus nos permiten: **A. Tener un mayor control en cada una de las señales del interfaz de comunicaciones y de esta forma mejorar la verificación de dicho interfaz**

B. Encapsular toda la comunicación del interfaz de comunicaciones, permitiendo definir los casos de test con un nivel de abstracción mayor de forma independiente del interfaz usado.

C. Verificar de forma independiente el interface de comunicaciones del DUV y la funcionalidad de dicho DUV

18. Se desea hacer un acelerador hardware de la función Arco tangente

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Como se puede observar esta función tiene unos valores asintóticos de 90º y -90º y es simétrica. Con lo cual en nuestra aproximación decidimos las siguientes características:

* A partir de una x de 256, aproximar f(x) a 90º , con lo cual almacenar en la tabla ROM solo valores del arco tangente corespondientes a valores de x entre  0 y casi 256.
* Queremos utilizar una memoria ROM de organización 214x  8 y queremos aprovechar totalmente sus bits disponibles
* La entrada y salida del modulo son IEEE 754 precision simple y es importante señalar que los valores de salida se expresarán en grados

Para dicha implementación utilizamos el siguiente código verilog del top

**module**top\_arctan\_14\_8 (  
          input [31:0] num\_entrada,  
          output [31:0] num\_salida,  
          input clk  
          );

**localparam**pi\_medios=  **x**;      
**localparam**size\_entrada= **x**;  
**localparam**size\_salida= **x**;      
**localparam**punto\_entrada= **x**;  
**localparam**punto\_salida= **x**;       
**wire**[size\_entrada:0] salida\_fix;  
**reg**[size\_entrada-1:0] addr;  
**wire**[size\_salida-1:0] q;  
**wire**[size\_salida:0] valor\_fix;  
**wire**[31:0] salida\_float,num\_salida\_reg2 ;  
**reg**[31:0]  num\_entrada\_reg1, num\_salida\_reg3,num\_salida\_reg4;  
**wire**[31:0] entrada\_reg\_pos;  
**wire** idea;  
**wire** [5:0] otra\_idea;  
**wire** [30:0] valor\_float;  
**wire**signo=num\_entrada\_reg1[31];  
**reg**signo\_reg,over\_reg, signo\_reg2,over\_reg2;  
**assign**entrada\_reg\_pos[31]=1'b0;  
**assign**entrada\_reg\_pos[30:0] =num\_entrada\_reg1[30:0];  
**Float2Fixed**#(.FIXEDSIZE(size\_entrada+1)) float2fix (  
    .InFloat(entrada\_reg\_pos),  
    .InRadixPoint(punto\_entrada),  
    .OutFixed(salida\_fix),  
     .OutException(),  
     .OutOverflow(otra\_idea)

    );   
**memoria\_single\_port\_mejorado**#( .DATA\_WIDTH(size\_salida), .ADDR\_WIDTH(size\_entrada), .punto\_entrada(punto\_entrada), .punto\_salida(punto\_salida) ) my\_memory   
                        (     
                              .addr(addr),  
                              .clk(clk),  
                              .q(q));

**assign**valor\_fix[size\_salida]=1'b0;  
**assign**  valor\_fix[size\_salida-1:0]=q;                              
**Fixed2Float**#(.FIXEDSIZE(size\_salida+1)) fix2float (  
    .InFixed(valor\_fix),  
    .InRadixPoint(punto\_salida),  
    .OutFloat(salida\_float)  
    );

**always**@(posedge clk)  
begin  
signo\_reg<=signo;  
over\_reg<=otra\_idea>0;   
signo\_reg2<=signo\_reg;  
over\_reg2<=over\_reg;  
num\_entrada\_reg1<=num\_entrada;  
addr<=salida\_fix[size\_entrada-1:0];    
num\_salida\_reg3<=num\_salida\_reg2;  
end

**assign**valor\_float= over\_reg2?pi\_medios:salida\_float[30:0];   
**assign**num\_salida\_reg2[31] = signo\_reg2,

**assign**num\_salida\_reg2[30:0] =valor\_float;

**assign**num\_salida=num\_salida\_reg3;         
**endmodule**

y el siguiente código  VHDL de la ROM instanciada:

**LIBRARY**ieee;  
**USE**ieee.std\_logic\_1164.all;   
**USE**ieee.std\_logic\_arith.all;  
**USE**ieee.std\_logic\_unsigned.all;  
**USE**ieee.fixed\_pkg.all;

**USE**ieee.math\_real.all;  
**entity**memoria\_single\_port\_mejorado is  
     generic (DATA\_WIDTH: integer:=9;  
                 ADDR\_WIDTH: integer:=10;  
                   
                 punto\_entrada: integer:=4;  
                 punto\_salida: integer:=4  
                 );  
    port (clk:          in std\_logic;  
          addr:       in  std\_logic\_vector (ADDR\_WIDTH-1 downto 0);   
          q:               out std\_logic\_vector (DATA\_WIDTH-1 downto 0)  
          );  
**end**memoria\_single\_port\_mejorado;  
**ARCHITECTURE**inferencia OF memoria\_single\_port\_mejorado IS  
**CONSTANT**TAMANYO:integer :=2\*\*ADDR\_WIDTH-1;  
TYPE ROM IS ARRAY(0 TO TAMANYO) OF STD\_LOGIC\_VECTOR(DATA\_WIDTH-1 DOWNTO 0);  
**FUNCTION** INIT\_ROM RETURN ROM IS  
VARIABLE romvar: ROM;  
VARIABLE x: real;  
VARIABLE temp: real;  
constant idea: real :=57.2957;  
begin  
 for I in 0 TO TAMANYO loop  
         x:= ARCTAN(to\_real(to\_ufixed(CONV\_STD\_LOGIC\_VECTOR(i,ADDR\_WIDTH), **x,x** )));  
            temp:=x\*idea; --pasamos de radianes a  grados  
         romvar(i):=to\_slv(to\_ufixed(temp,**x** , **x**));            
  end loop;  
  return romvar;  
end;

**CONSTANT**memoria: ROM := INIT\_ROM;  
**BEGIN**  
memory: **process**(clk)  
begin  
    if clk'event and clk='1' then  
        q<=memoria(conv\_integer(addr));  
    end if;  
end **process**;  
**END**inferencia;

Rellenad los huecos incluidos en el codigo verilog  y del código VHDL y responded a la siguiente afirmación: la latencia de esta implemnetación e**s  4**

**Sol: 31'h42B40000, 14, 8, 6|6'd6, 1|6'd1, ADDR\_WIDTH-1-punto\_entrada|7, -punto\_entrada|-6, DATA\_WIDTH-1-punto\_salida|6, -punto\_salida|-1, 4**

# BLOQUE IV

1. Indique qué servicios son útiles para compartir información entre tareas en un RTOS como uCOS.

A. Semáforos, Punteros a enteros, Colas y Flags de Eventos

B. Mailbox, Colas, Variables globales o locales y Flags de Eventos

**C. Semáforos, Mutex, Mailbox.**

2. Respecto a los servicios de colas de mensajes en UCOS:

**A. Son útiles para trabajar con tareas productoras de datos con velocidades de generación distinta a la de consumo de datos de las tareas pendientes de los mensajes de la cola.**

B. Permiten crear colas de tipo FIFO pero no permite el uso de mensajes de tipo LIFO por parte de las ISR.

C. Crean colas de tamaño N mensajes y necesitan que el diseñador del sistema defina los punteros de la cola, tanto el de lectura como de escritura del buffer circular.

3. Un RTOS es un Sistema Operativo que (seleccione la correcta):

**A. Independientemente de la arquitectura del sistema satisface los requerimientos de tiempo real**

B. Permite alcanzar grandes velocidades de proceso, cercanas a los centenares de MHz.

C. Proporciona una gestión de procesos en batch mediante técnicas round-robin de asignación de recursos.

4. Respecto a estadísticas de tareas en UCOS:

**A. Se requiere de la llamada a OSStatInit() antes de la creación de las tareas sobre las que se quiera hacer estadísticas si vamos a usar OSTaskStatHook().**

B. Se requiere de la llamada a OSStatInit() si queremos usar el hook OSTaskStatHook() y antes de la creación de cualquier tarea del proyecto, aunque no se realicen estadísticas sobre todas ellas.

C. No es recomendable utilizar una tarea de inicialización que cree el resto de tareas y se duerma posteriormente, es mejor crear todas las tareas en la inicialización del sistema.

5. Un semáforo MUTEX:

A. Es un semáforo binario que no realiza un Post sobre el recurso mutex de manera que se excluye al resto de tareas del uso del mismo.

B. Es un semáforo de valor N que permite modificar la prioridad de la tarea que lo adquiere con el objetivo de tener más prioridad que el resto de las tareas que compiten por el recurso.

**C. Es un semáforo binario que permite modificar la prioridad de la tarea que lo adquiere con el objetivo de tener más prioridad que el resto de las tareas que compiten por el recurso.**

6. Respecto al uso de Flags de Eventos:

A. Se pueden definir de tipo AND u OR y se pueden activar a 1 los bits del flags desde una tarea pero no pueden activar a 0.

**B. Se pueden definir de tipo AND u OR y pueden tener un tamaño byte o 16bits o 32bits como flags .**

C. No pueden activarse (OSFlagPost()) los flags desde una ISR, solo desde las Tasks.

7. Una tarea procesa los datos recibidos por UART y genera un grupo de datos de información por paquete recibido, indicando número de bytes, checksum y código de error. Si quisiéramos compartir esta información generada por cada paquete recibido con otras tareas que se encarguen de procesar esos datos deberíamos:

A. Se debería usar semáforos para bloquear el acceso a memoria del resto de tareas.

**B. Se debería usar colas de mensajes.**

C. Se debería usar EFG (Flags de Eventos), un bit por cada datos que queramos indicar: checksum, tamaño, código de error.

8. Una característica destacada de UCOS II es:

**A. Que tiene un scheduler preemptive o expulsivo.**

B. Que utiliza round-robin para temporizar las tareas que se crean y asignar así los tiempos para

cada una de ellas.

C. Que tiene todos los servicios activos siempre y no permite desactivarlos para ocupar menos

espacio en FLASH, permitiendo así hacer uso de todos ellos sin problemas para el Arquitecto de Software

9. Respecto a la gestión de tareas en UCOS, podemos afirmar que:

**A. Tareas de menor prioridad pueden dormir (borrar) y cambiar la prioridad de tareas de mayor prioridad.**

B. Las tareas no pueden bloquear el scheduler ya que pertenece al kernel del RTOS.

C. Tareas de menor prioridad no pueden dormir tareas de mayor prioridad, solo pueden "resume" o reactivarlas si están en estado de espera.

10. Respecto al parámetro OS\_TASK\_OPT\_STK\_CHK|OS\_TASK\_OPT\_STK\_CLR

A. Es opcional pero no indispensable para hacer estadísticas y se incluye en el servicio

OSTaskCreateExt().

**B. Es necesario para hacer estadísticas de tareas y se incluye en el servicio OSTaskCreateExt().**

C. Se incluye en el servicio OSTaskCreate() y se debe llamar antes de OsStatInit().