

 <p><b>UNIVERSIDAD POLITÉCNICA DE MADRID</b></p> <p><i>E.T.S.I.S. Telecomunicación</i></p>				<b>APELLIDOS:</b>				
				<b>NOMBRE:</b>				<b>DNI:</b>
				<b>ASIGNATURA:</b> Diseño Digital 2				<b>Bloque 1</b>
<b>Fecha</b>			<b>Curso</b>	<b>Grupo</b>	<b>Notas Parciales</b>			<b>Nota Final</b>
19	04	2021	Tercero					

## ADVERTENCIAS PARA LA REALIZACIÓN DEL EXAMEN

- Rellene **AHORA** los datos personales que deben figurar en esta hoja.
- Mientras dure el examen deberá exponer su D.N.I. encima de la mesa.
- **NO SE ADMITIRÁN** exámenes escritos a lapicero ni con tinta roja o verde.
- **COMPRUEBE** que su ejemplar del examen consta de **3** ejercicios en **5** páginas numeradas.
- En este examen **NO PUEDEN UTILIZARSE CALCULADORAS, LIBROS, APUNTES NI DISPOSITIVOS DE TELECOMUNICACIÓN**. Retírelos ahora de la mesa.
- La duración del examen es de **90 minutos**.

Esta hoja se ha dejado en blanco intencionadamente

**Ejercicio 1 (3 puntos, 25 min).** El código del fichero *interfaz\_periferico.vhd*, ubicado en la carpeta *ejercicio\_1* corresponde al modelo del módulo *interfaz\_periferico* (uno de los bloques funcionales del Periférico I2C) realizado en el diseño del bloque 1.

Califique las siguientes afirmaciones, que se refieren a dicho módulo como verdaderas (V) o falsas (F). Utilice el cajetín al lado del número de cada pregunta; si se equivoca, táchelo y escriba la respuesta correcta a la izquierda. Las respuestas correctas se calificarán con +0.3 puntos, las incorrectas con -0.3 puntos y las no contestadas con 0 puntos.

**Nota:** En la carpeta *docu* dispone de un documento con el modelo de registros del Periférico I2C.

F 1.- Tras la ocurrencia de un reset asíncrono todas las salidas de la FIFO de lectura (*q*, *usedw*, *empty* y *full*) permanecen a '0'.

V 2.- Los datos se escriben en la FIFO de escritura utilizando la interfaz basada en registros y son leídos por el circuito *interfaz\_i2c*.

F 3.- El circuito entrega en su salida *dato\_out* el número de bytes almacenado en la FIFO de lectura cuando se realiza una lectura activando *rd* con *add* = 2.

F 4.- El circuito inicia una transacción en el bus I2C siempre que se realiza una escritura en el registro de control con el bit de menor peso de "dato\_in" a '1'.

V 5.- El circuito conoce la naturaleza (lectura o escritura) de la transacción I2C en proceso de configuración a partir del bit de menor peso del byte contenido en la primera posición de la FIFO de escritura.

V 6.- La señal auxiliar *ena\_wr\_byte* habilita que se escriba en la FIFO de lectura el byte que contiene la dirección I2C en operaciones de lectura cuando se utiliza el modo de "escritura de todos los bytes".

V 7.- La salida *ini* está modelada de manera que no puede tener *glitches*.

F 8.- Todas las entradas del circuito *interfaz\_periferico* deben estar libres de *glitches* para asegurar el correcto funcionamiento del mismo.

V 9.- Excluyendo a las memorias FIFO, el modelo del circuito puede realizarse empleando 15 flip-flops.

V 10.- Si al sintetizar el modelo, un analizador de tiempos indica que la frecuencia máxima de funcionamiento del circuito es de 125 MHz, es posible que el circuito pueda funcionar con una frecuencia de reloj mayor si las condiciones de funcionamiento no son las peores.

**Ejercicio 2 (3 puntos, 25 min).** La carpeta *docu* contiene el *datasheet* del sensor de temperatura y humedad HC1000 empleado en la realización del MEDTH, los esquemas de la tarjeta DECA MAX10 y el *device overview* de la familia MAX10. Consúltelos, si lo considera necesario, para indicar si las siguientes afirmaciones son verdaderas (V) o falsas (F). Utilice el cajetín al lado del número de cada pregunta; si se equivoca, táchelo y escriba la respuesta correcta a la izquierda. Las respuestas correctas se calificarán con +0.3 puntos, las incorrectas con -0.3 puntos y las no contestadas con 0 puntos.

F 1.- Una transferencia I2C compuesta por los bytes x81 x02 x15 x00 configuraría al sensor de temperatura en modo de captura secuencial de temperatura y humedad con una resolución de 11 bits para ambas magnitudes.

V 2.- Una transferencia sobre el sensor de temperatura compuesta por los bytes x80 x02 situaría el puntero del sensor de temperatura en la dirección correspondiente al registro de configuración.

F 3.- El mayor valor que puede obtenerse leyendo del registro de humedad del sensor es de xFFFF, correspondiendo a una humedad relativa del 100%.

F 4.- Las entradas *ADR0* y *ADR1* del sensor de temperatura y humedad pueden configurarse mediante *jumppers* en la tarjeta DECA para configurar una de entre cuatro posibles direcciones I2C.

V 5.- La tarjeta DECA posee dos conectores de expansión. Uno de ellos incluye, entre otras, la misma alimentación que se utiliza para los bancos 3 y 4 de la FPGA.

F 6.- En la tarjeta DECA, los condensadores C96 y C97 son condensadores de desacoplo de la alimentación, cuya función es reducir el ruido producido por la conmutación de los circuitos digitales.

F 7.- En la tarjeta DECA, la bobina L14 y el condensador C149 forman un filtro LC que protege a la alimentación del sensor de temperatura y humedad, VCC3P3\_HDC1000, de valores de VCC3P3 que ocasionalmente pudieran estar por encima de los *Absolute Maximum Ratings* definidos en la hoja de datos del sensor.

V 8.- En la tarjeta DECA, los diodos D16 y D17 *desconectan* la alimentación VCC5\_USB cuando a través de J9 se introduce una alimentación de 5 V.

V 9.- La FPGA de la tarjeta DECA dispone de una memoria FLASH para configuración, que permite la utilización de la tarjeta en aplicaciones en las que es necesario que no se pierda la configuración previa tras un ciclo de apagado/encendido de la alimentación.

F 10.- Con la FPGA de la tarjeta DECA pueden realizarse diseños que tengan, como máximo, 5000 flip-flops.

**Ejercicio 3 (4 puntos, 40 min).** La carpeta *ejercicio\_3* contiene el modelo de un *monoestable no redisparable* que debe generar una ventana *con una duración* de 1 ms empleando un reloj de 100 MHz y el código de un *test-bench* que dispone de un módulo que verifica el correcto funcionamiento del circuito y que emplea, además, otros dos módulos para generar el reloj, el reset del circuito, los estímulos y determinar la duración de la simulación.

**Nota:** Para resolver este ejercicio, cree un proyecto ModelSim e incluya en él los ficheros que se le proporcionan.

1.- Analice el código del fichero *monitor\_monoestable.vhd* para determinar la función de verificación de resultados que realiza el monitor que contiene el módulo. Después:

- Describa dicha función empleando comentarios VHDL (no emplee más de dos líneas) en el propio fichero, precediendo al código del monitor (1 punto).
- Escriba mensajes descriptivos para los errores potencialmente detectables (0.5 puntos).

2.- Complete el código del fichero *monitor\_monoestable.vhd* para que avise cada vez que se produzca la activación de la señal de disparo mientras se está generando el pulso por un disparo anterior. (1.5 puntos)

3.- Tras ejecutar la simulación revise los resultados de la misma para determinar si el modelo del *monoestable* es, o no, correcto. Justifique, aquí, su respuesta. En caso de detectar algún error en el modelo, corríjalo. (1 punto)

Justificación:

*No hay ningún error en el modelo del monoestable. Cumple perfectamente las especificaciones planteadas en el enunciado.*

Una vez haya terminado, comprima el proyecto ModelSim (llame al fichero comprimido *monoestable.zip*) y súbalo a Moodle.