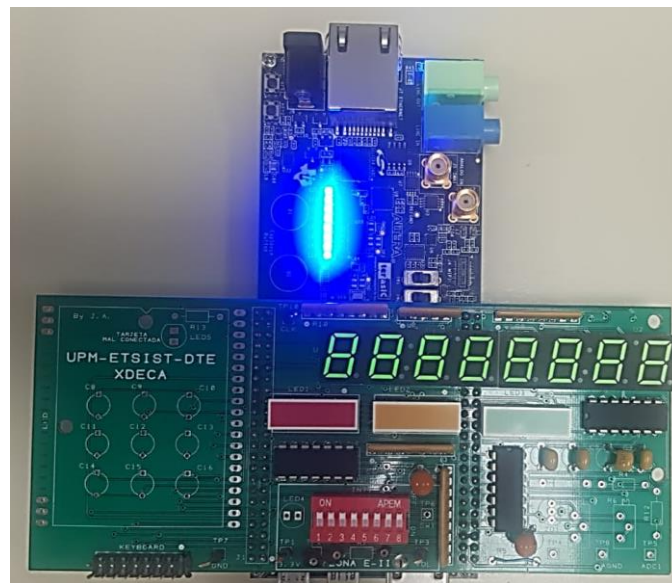




DISEÑO DIGITAL 2

MEMORIA DEL DISEÑO:

NIVEL



Autor: **Ainhoa Sainz Gordo**
Volker Rivera Alvarez
Joselyn Selena Velasco Lucero
Leticia Marcos Avendaño

Curso 2021-2022.

Control de versiones

Versión	Fecha	Autor	Cambios realizados
0.0	19/04/2022	DTE	Inicial
1.0	03/05/2022		Fase 1 : Diseño de Máster-SPI a cuatro hilos.
2.0	09/05/2022		Fase 2 : Diseño de un controlador del Máster-SPI..
3.0	13/05/2022		Fase 3 : Calculo inclinación y representación de leds y display.
3.0	18/05/2022		Fase 4 : Implementación en Quartus y placa final.

Tabla de contenido

Tabla de contenido.....	3
1 Especificación del diseño.	4
Introducción.....	4
Interfaces	4
Interfaz con el acelerómetro	4
Interfaz con la barra de displays de 7 segmentos	5
Interfaz con la barra de LEDs.....	5
Especificaciones	6
2 Diseño jerárquico.....	8
Bloque INTERFAZ_SPI	10
2.1.1 Bloque AGENTE_SPI.....	10
Bloque CONTROLADOR_SPI.....	11
Bloque CALC_OFFSET	11
Bloque ESTIMADOR	11
Bloque AUXILIAR	11
3 Diseño detallado	12
4 Pruebas de verificación funcional de NIVEL.....	13
Test INTERFAZ_SPI	13
Test CONTROLADOR_SPI	14
Test REPRESENTACION	15
5 Diseño físico.....	16
Asignación de pines.....	16
Restricciones de la síntesis	18
Recursos utilizados	19
Frecuencia máxima de reloj.....	20
6 Bibliografía.....	21

1 Especificación del diseño.

Introducción

El nivel electrónico (NIVEL) permite detectar la inclinación de la tarjeta DECA en dos ejes de movimiento. La información de la inclinación de la tarjeta se obtiene utilizando el acelerómetro incluido en la DECA. La visualización de la inclinación se realiza utilizando los LEDs de la DECA para uno de los ejes y los displays de 7 segmentos de la XDECA para el otro.

Interfaces

El diagrama de bloques general del diseño se muestra en la siguiente figura.

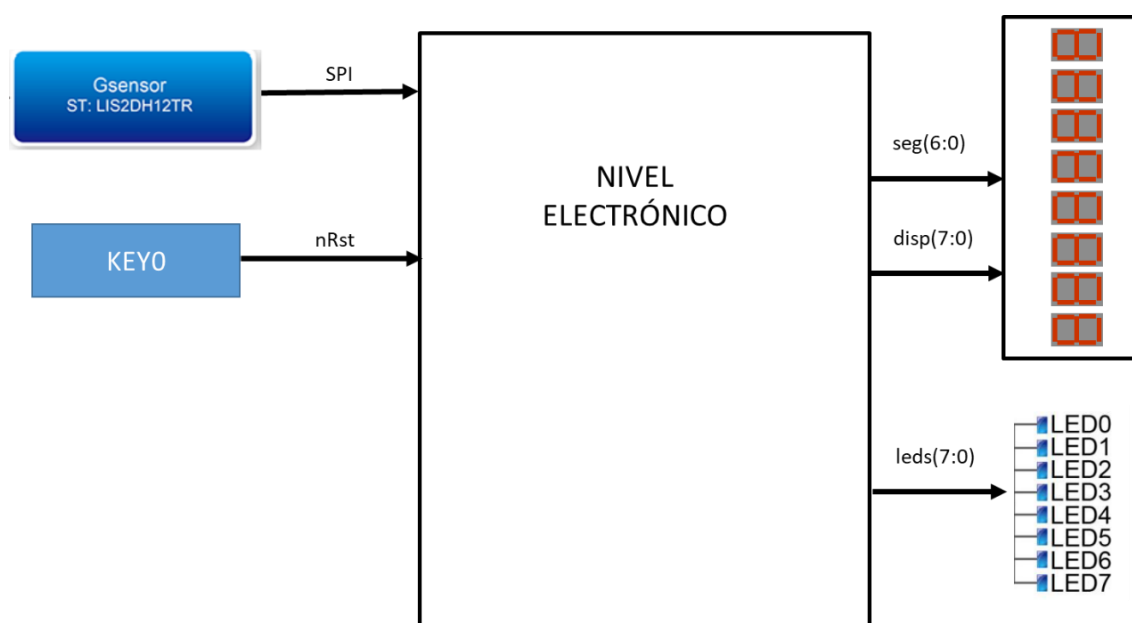


Fig. 1. Interconexión del NIVEL al acelerómetro y a los LEDs y displays de 7 segmentos.

Interfaz con el acelerómetro

El sistema se comunica con el acelerómetro con una interfaz SPI. Las señales de esta interfaz son las siguientes:

Señal	Dirección	Descripción
nCS	salida	Habilita el puerto serie y controlado por SPI
SPC	salida	Señal de reloj de la interfaz SPI
SDI	entrada	Entrada de datos
SDO	salida	Salida de datos

Interfaz con la barra de displays de 7 segmentos

El sistema realiza la visualización de la inclinación en uno de los ejes utilizando una barra de displays de 7 segmentos del tipo cátodo común. La interfaz es la siguiente:

Señal	Dirección	Descripción
seg[7..0]	salida	seg[0] : segmento g seg [1] : segmento f seg [2] : segmento e seg [3] : segmento d seg [4] : segmento c seg [5] : segmento b seg [6] : segmento a seg [7] : segmento punto
mux_disp[4..0]	salida	mux_disp [0] : cátodo del display 0 (LSD) mux_disp [1] : cátodo del display 1 mux_disp [2] : cátodo del display 2 mux_disp [3] : cátodo del display 3 mux_disp [4] : cátodo del display 4

La interfaz permite iluminar solo un display a la vez. El display se selecciona activando (a nivel bajo) el cátodo correspondiente. El display activo se ilumina de acuerdo con el código de 7 segmentos y punto decimal introducido (nivel alto).

Interfaz con la barra de LEDs

El sistema realiza la visualización de la inclinación en el otro eje utilizando una barra de displays de 7 segmentos :

Señal	Dirección	Descripción
Leds_X [7..0]	salida	Leds_X [0]: led 1 Leds_X [1]: led 2 Leds_X [2]: led 3 Leds_X [3]: led 4 Leds_X [4]: led 5 Leds_X [5]: led 6 Leds_X [6]: led 7 Leds_X [7]: led 8

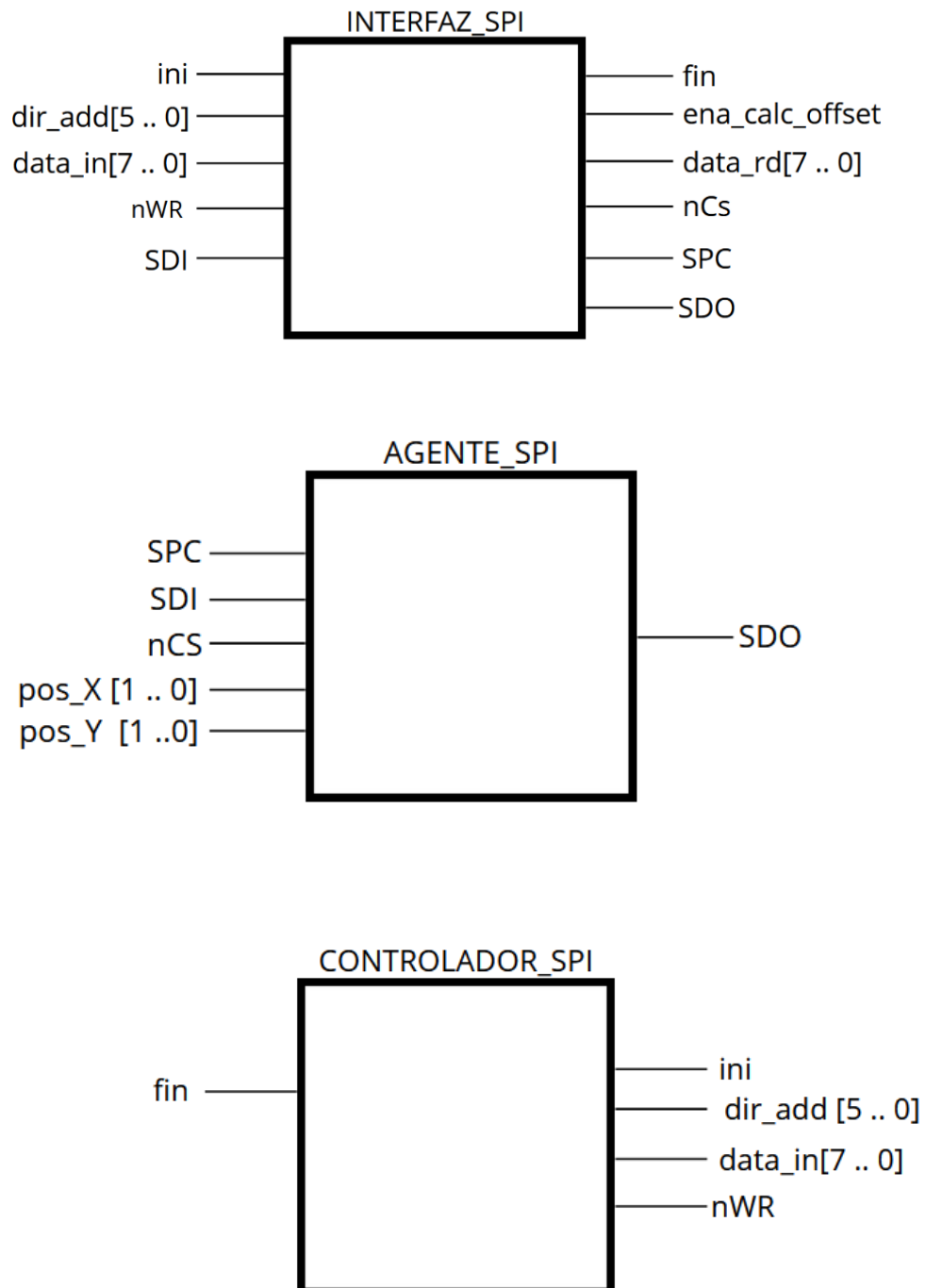
Especificaciones

Ref	Especificación
ESP00	El sistema utilizará una comunicación por SPI a cuatro hilos.
ESP01	El sistema realiza medidas cada 5ms detectando el movimiento de los ejes X e Y.
ESP02	El movimiento se evaluará tomando la posición inicial como referencia. La posición se calculará como la diferencia entre el valor que recibimos del sensor y el valor de referencia.
ESP03	<p>En los LEDs de la tarjeta DECA se representarán los movimientos del eje X :</p> <ul style="list-style-type: none"> • 8 leds encendidos: posición horizontal inicial • Grado de inclinación: se calcula de forma proporcional a 'g' utilizando la siguiente formula: medida_ejeX < (2*N-15)/15 g • Valores de N en función de los leds: <ul style="list-style-type: none"> ○ N = 1 à ON,OFF,OFF,OFF,OFF,OFF,OFF,OFF ○ N = 2 à ON,ON,OFF,OFF,OFF,OFF,OFF,OFF ○ N = 3 à ON,ON,ON, OFF,OFF,OFF,OFF,OFF ○ N = 4 à ON,ON,ON, ON ,OFF,OFF,OFF,OFF ○ N = 5 à ON,ON,ON, ON, ON ,OFF,OFF,OFF ○ N = 6 à ON,ON,ON, ON, ON, ON ,OFF,OFF ○ N = 7 à ON,ON,ON, ON, ON, ON ,ON,OFF ○ N = 8 à ON,ON,ON, ON, ON, ON ,ON,ON ○ N = 9 à OFF,ON,ON,ON,ON,ON,ON,ON ○ N = 10 à OFF,OFF,ON,ON,ON,ON,ON,ON ○ N = 11 à OFF,OFF,OFF,ON,ON,ON,ON,ON ○ N = 12 à OFF,OFF,OFF,OFF,ON,ON,ON,ON ○ N = 13 à OFF,OFF,OFF,OFF,OFF,ON,ON,ON ○ N = 14 à OFF,OFF,OFF,OFF,OFF,OFF,ON,ON ○ N = 15 à OFF,OFF,OFF,OFF,OFF,OFF,OFF,ON
ESP04	En los displays de la tarjeta DECA se representarán los movimientos del eje Y. Considerando que el ledN se corresponde con el displayN y que el equivalente a un LED encendido es un display representando un '8'.
ESP05	La interfaz debe permitir que un controlador ordene transferencias de escrituras de 1 byte o lecturas de 4 bytes cuando el máster esté preparado.
ESP06	El controlador puede indicar la dirección del registro sobre el que desea operar y la indicación de tipo de operación.
ESP07	El dato para enviar será introducido por el controlador.
ESP08	El controlador debe ordenar al máster la escritura en los registros de configuración y la lectura de los datos de medida en X e Y.
ESP09	<p>La configuración del circuito debe ser a través de los registros CTRL_REG1 A CTRL_REG6. Se realiza un avez tras el reset asincrono. Debe ser :</p> <ul style="list-style-type: none"> • Normal mode (10 bits de resolución) • Output Data Rate (ODR) de 200Hz • Rango de medida entre +2g y -2g

	<ul style="list-style-type: none">• Habilitados los ejes X e Y• FIFO deshabilitada.• High-pass filter bypassed.• Todas las interrupciones inhabilitadas.• Block Data Update (BLU) habilitada.• Big/Little Endian en Little Endian.• Self-test inhabilitado.• SPI configurado a cuatro hilos.• “Reboot memory content” en modo normal.
ESP10	El último registro en configurarse debe ser el REG1. Realiza el cambio de modo.
ESP11	Se ignora la condición de overrun.
ESP12	Se realiza el promediado del número de lecturas recibidas durante 320ms, tomándolos como referencia.
ESP13	Una vez realizado el promediado, para cada eje se realiza medidas cada 5ms y calcula el promediado de las lecturas realizadas en los últimos 160ms para determinar la posición actual de la tarjeta.

2 Diseño jerárquico

El diagrama de la Fig. 2 representa el primer nivel de la jerarquía del diseño¹:



¹ Todos los bloques tienen un reloj común, `clk`, y una entrada de reset asíncrono, `rst_n`, activa a nivel bajo. Estas señales no se incluyen en las interfaces por simplicidad.

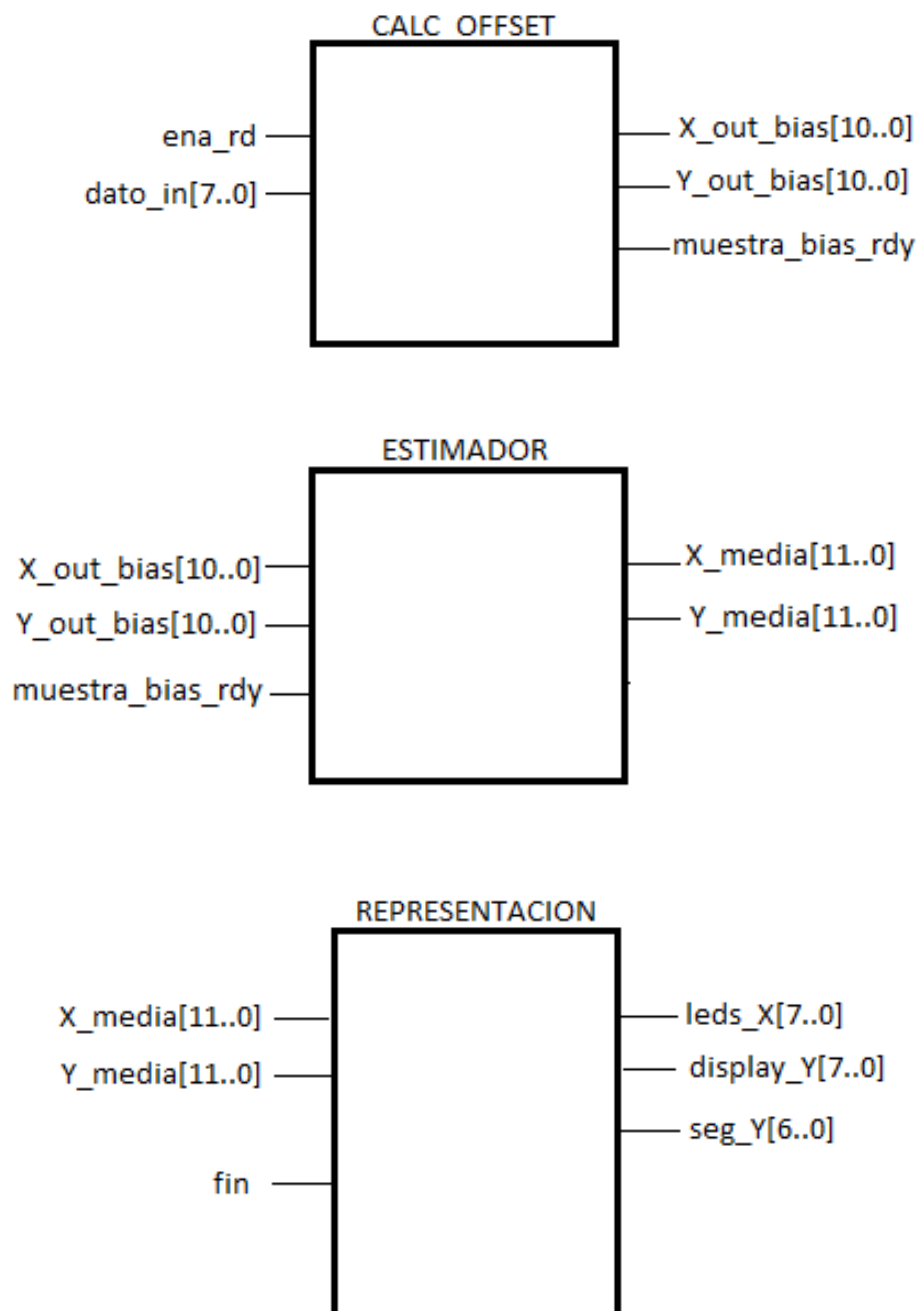


Fig. 2. Diagrama de bloques del primer nivel de la jerarquía de NIVEL.

En los siguientes subapartados se describe la interfaz y la función de cada uno de estos bloques.

Bloque INTERFAZ_SPI

La interfaz SPI es la encargada de generar el Master SPI para la lectura y escritura de datos.

Señal	Dirección	Descripción
ini	in	Se activa cuando se quiere realizar una transferencia.
dir_add	in	Dirección del registro.
data_in	in	Datos que enviamos en la operación de escritura.
nWR	in	Indica la operación que se va a realizar : 0 lectura, 1 escritura.
SDI	inout	Entrada de datos.
fin	buffer	Indica que se ha finalizado la transferencia.
ena_calc_offset	buffer	Habilitación para calcular el offset.
data_rd	buffer	Datos de lectura.
nCS	buffer	Habilita el puerto serie y controlado por SPI.
SPC	buffer	Reloj SPI.
SDO	inout	Salida de datos hacia el sensor.

2.1.1 Bloque AGENTE_SPI

Agente Spi realiza el papel de esclavo para comprobar el funcionamiento del masterSPI, responder a las transferencias efectuadas.

Señal	Dirección	Descripción
pos_X	in	Posición X.
pos_Y	in	Posición Y.
nCS	in	Chip select.
SPC	in	Reloj SPI (5 Mhz).
SDI	in	Slave Datos de entrada conectado a SDO Master.
SDO	buffer	Slave Datos de salida conectado a SDI Master.

Bloque CONTROLADOR_SPI

El Controlador SPI ordena las transferencias, estas son de dos tipos: Escrituras de un byte o Lecturas de cuatro bytes. Solo se puede realizar cualquier transferencia siempre y cuando se indique que el master SPI (Interfaz SPI) está preparado para realizarlo.

Señal	Dirección	Descripción
CUENTA_5ms	natural	Valor genérico de modulo que se necesita para generar 5ms.
fin	in	Indica que el máster está listo para generar transferencia.
ini	buffer	Se activa cada que se quiere realizar una transferencia.
dir_add	buffer	Dirección del registro.
data_in	buffer	Datos para enviar a una escritura.
nWR	buffer	Indica la operación que vamos a realizar.

Bloque CALC_OFFSET

Señal	Dirección	Descripción
N	in positive	Número de registros del banco.
ena_rd	in	Habilitación para calcular el offset.
dato_rd	in	Datos de lectura del sensor.
X_out_bias	buffer	Posición de la tarjeta en el eje X.
Y_out_bias	buffer	Posición de la tarjeta en el eje Y.
muestra_bias_rdy	buffer	Indica que la muestra está disponible.

Bloque ESTIMADOR

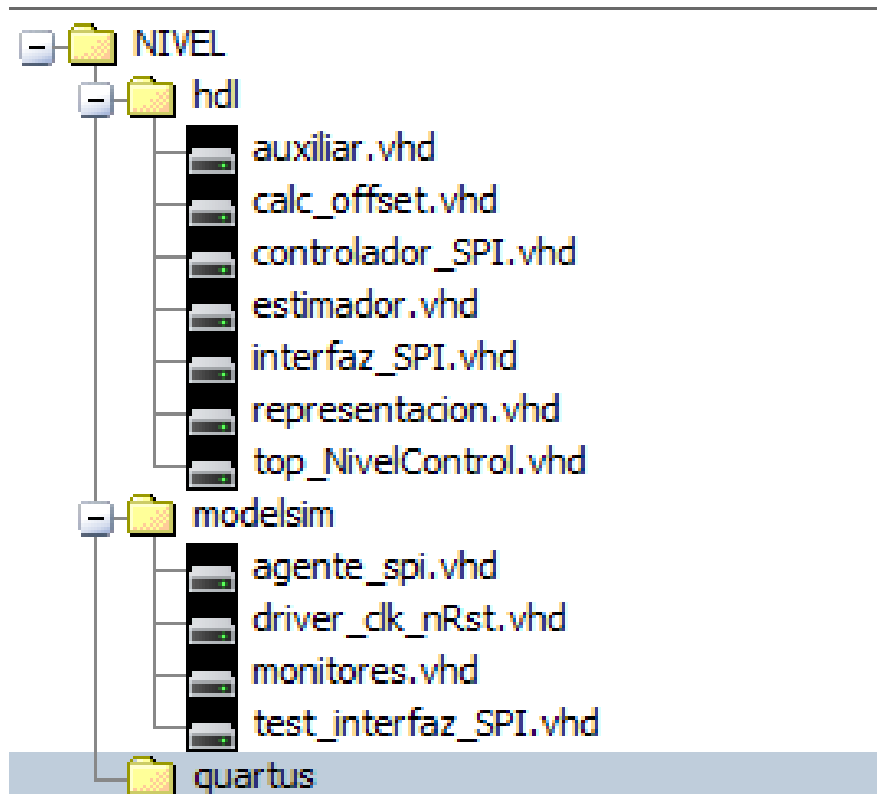
Señal	Dirección	Descripción
N	in positive	Número de registros del banco.
X_out_bias	in	Posición de la tarjeta en el eje X..
Y_out_bias	in	Posición de la tarjeta en el eje Y.
muestra_bias_rdy	in	Indica que la muestra está disponible.
X_media	buffer	Valor medio de las últimas N muestras en X.
Y_media	buffer	Valor medio de las últimas N muestras en Y.

Bloque AUXILIAR

Función	Descripción
ceil_log (x: in natural)	Calcula el menos número natural, N, para el que 2^N es mayor igual que x.

3 Diseño detallado

El proyecto está almacenado en la carpeta NIVEL, la cual a su vez contiene las carpetas hdl, modelsim y quartus. La carpeta *hdl* contiene los ficheros RTL y estructural del diseño. La carpeta *modelsim* contiene el proyecto de simulación (NIVEL.mpf), y los ficheros donde se definen los diferentes testbenches. Finalmente, *quartus* contiene el proyecto para el diseño físico y los ficheros relacionados con éste.



4 Pruebas de verificación funcional de NIVEL

El plan de pruebas de NIVEL consiste en realizar diversas pruebas para verificar que el diseño presentado cumple con las necesidades establecidas.

Test INTERFAZ_SPI

Es un test para comprobar el correcto funcionamiento del módulo INTERFAZ_SPI.

Ubicación de los ficheros del test	En /NIVEL/modelsim/test y /NIVEL/hdl	
Simulación escalada	No	
Ficheros	test_interfaz_SPI.vhd	testbench
	monitores.vhd	Código de verificación automática de los tiempos.
	Agente_SPI.vhd	Código que simula el funcionamiento del sensor.
Descripción del test	<p>Inicialización del reset e inicialización de las entradas. Y asignamos valores a pos_X y pos_Y.</p> <p>En primer lugar, ordenaremos una lectura. Para ello primero indicaremos la dirección del registro, el tipo de transferencia y simularemos un pulso de un período a nivel alto en la señal ini. Se ha de comprobar que se respetan todos los tiempos establecidos por el protocolo SPI, que las escrituras y las lecturas de las líneas SDO y SDI se hacen en los flancos correspondientes, que la interfaz recibe 4 bytes consecutivos del agente y, además, que se reciben los datos esperados.</p> <p>Finalmente ordenaremos una escritura. Para ello, al igual que antes, indicaremos la dirección del registro sobre el que queremos escribir y el dato. Después simularemos un pulso de un período a nivel alto en la señal ini para ordenar el comienzo de la transacción. Se ha de comprobar que se respetan los tiempos anteriormente mencionados y que se hace solo una escritura.</p> <p>En ambos casos hemos de asegurarnos que tanto en la línea SDI como en la SDO, los datos transmitidos son los mismos que los valores dados en el test.</p> <p>Respecto a los tiempos, no debería haber problemas gracias a la frecuencia escogida para el SPC pero, se han establecido una serie de monitores para comprobar de forma automática que se respeta la frecuencia elegida para el SPC, que se cumplen el tiempo a nivel alto y a nivel bajo mínimos, que el $t_{su_cs} > 5\text{ ns}$ y que $t_{hod_cs} > 20\text{ ns}$.</p>	

Test CONTROLADOR_SPI

Ubicación de los ficheros del test	En /NIVEL/modelsim/test y /NIVEL/hdl	
Simulación escalada	Sí	
Ficheros	test_interfaz_SPI.vhd	Se han ido añadiendo los diferentes módulos al mismo fichero de test.
	Monitores.vhd	Código de verificación automática de los tiempos
	Agente_SPI.vhd	Código que simula el funcionamiento del sensor
	Driver_clk_nRst.vhd	Código para la generación automática del clk y el nRst
	Controlador_SPI.vhd	Código del módulo del controlador del master SPI
Descripción del test	<p>Inicialización del reset e inicialización de las entradas. Asignamos valores a pos_X y pos_Y.</p> <p>En este test tenemos que comprobar que el controlador hace de forma automática todas las asignaciones que se han hecho de forma manual en el anterior bloque.</p> <p>En cuanto a tiempos tenemos que comprobar que desde la puesta del nRst a nivel alto, se esperan 5 ms antes de ordenar la primera transacción y, que después de eso, el resto de las transacciones se hacen cada 5 ms.</p> <p>Se comprueba que se realiza la configuración en los registros, modificando los registros 4 y 1; El registro 1 debe ser el último en escribirse.</p> <p>Seguidamente se realizan lecturas indefinidamente.</p>	

Test REPRESENTACION

Ubicación de los ficheros del test	En /NIVEL/modelsim/test y /NIVEL/hdl	
Simulación escalada	Sí	
Ficheros	test_interfaz_SPI.vhd	Se han ido añadiendo los diferentes módulos al mismo fichero de test.
	Monitores.vhd	Código de verificación automática de los tiempos
	Agente_SPI.vhd	Código que simula el funcionamiento del sensor
	Driver_clk_nRst.vhd	Código para la generación automática del clk y el nRst
	Controlador_SPI.vhd	Código del módulo del controlador del master SPI
	Calc_offset.vhd	Incluye la aritmética para el cálculo del offset y de las siguientes medidas referenciadas al mismo.
	Estimador.vhd	Su función es proporcionar una media de las últimas 16 medidas (configurable).
	Auxiliar.vhd	Incluye la función (ceil_log) utilizada tanto en calc_offset como en estimador.
	Top_SPI.vhd	Top del estructural, incluirá todos los módulos del hdl.
Descripción del test	<p>En este test comprobaremos si el cálculo del offset y de las medidas se hace de manera correcta y en los tiempos que han sido especificados.</p> <p>Para facilitar la lectura en el visor de las señales (wave), escalaremos los tiempos.</p> <p>Se tiene que comprobar que se obtiene el offset tras 320 ms y que después se obtiene la media de las últimas medidas cada 160 ms.</p> <p>Daremos diferentes valores a pos_X y pos_Y para que el agente simule movimiento y por tanto no obtengamos siempre el mismo valor de medida.</p> <p>Las medidas que nos devuelve el agente deben ir al calc_offset y de ahí al estimador. Serán las salidas de este último las que usemos para la representación en los leds y los displays. Así pues, ese cambio en las medidas deberá verse reflejado en las señales leds_X y display_Y del bloque de representación.</p>	

5 Diseño físico

En este apartado se documentan los detalles básicos relacionados con el diseño físico del circuito: la asignación de pines, las restricciones de la síntesis y los informes que proporciona Quartus Prime sobre los recursos de la FPGA utilizados y la frecuencia máxima de reloj obtenida.

Asignación de pines

En la siguiente tabla se detalla la asignación de los pines de la interfaz de NIVEL a los pines de la FPGA especificando para cada caso el tipo de pin, el número de pin de la FPGA que se corresponde con el pin del diseño, el banco al que corresponde y el estándar de entrada/salida que utiliza.

Señal	direccion	Pin FPGA	I/O bank	I/O standard
clk	Input	M8	2	2.5-V
rst_n	Input	H21	6	1.5-V Schmitt Trigger
SPC	output	B5	8	1.2-V
SDO	Inout	C6	8	1.2-V
SDI	Inout	D5	8	1.2-V
nCS	Output	E9	8	1.2-V
Display Y[7]	Output	V10	3	3.3-V LVTTL
Display Y[6]	Output	AA6	3	3.3-V LVTTL
Display Y[5]	Output	AB6	3	3.3-V LVTTL
Display Y[4]	Output	R11	3	3.3-V LVTTL
Display Y[3]	Output	AB8	3	3.3-V LVTTL
Display Y[2]	Output	W8	3	3.3-V LVTTL
Display Y[1]	Output	W6	3	3.3-V LVTTL
Display Y[0]	Output	Y5	3	3.3-V LVTTL
Leds_X[7]	Output	C5	8	1.2-V
Leds_X[6]	Output	B4	8	1.2-V
Leds_X[5]	Output	A5	8	1.2-V
Leds_X[4]	Output	C4	8	1.2-V
Leds_X[3]	Output	B7	8	1.2-V
Leds_X[2]	Output	A6	8	1.2-V
Leds_X[1]	Output	C8	8	1.2-V
Leds_X[0]	Output	C7	8	1.2-V
G GPIO0_D17)	Output	W16	4	3.3-V LVTTL
F (GPIO0_D28)	Output	AB11	4	3.3-V LVTT
E (GPIO0_D19)	Output	W15	4	3.3-V LVTTL
D GPIO0_D38)	Output	AB10	4	3.3-V LVTTL
C (GPIO0_D21)	Output	AA15	4	3.3-V LVTTL
B (GPIO0_D34)	Output	W12	4	3.3-V LVTTL
A GPIO0_D25)	Output	AA13	4	3.3-V LVTTL

Restricciones de la síntesis

Se ha utilizado el siguiente fichero sdc:

```
## DATE      "Wed May 18 16:27:13 2022"

##
## DEVICE    "10M50DAF484C6GES"
##

#####
# Time Information
#####

set_time_format -unit ns -decimal_places 3

#####
# Create Clock
#####

create_clock -name {clk} -period 20.000 -waveform { 0.000 10.000 } [get_ports {clk}]
create_clock -name {nRst} -period 1.000 -waveform { 0.000 0.500 } [get_ports {nRst}]

#####
# Create Generated Clock
#####

#####
# Set Clock Latency
#####

#####
# Set Clock Uncertainty
#####

set_clock_uncertainty -rise_from [get_clocks {clk}] -rise_to [get_clocks {clk}] 0.020
set_clock_uncertainty -rise_from [get_clocks {clk}] -fall_to [get_clocks {clk}] 0.020
set_clock_uncertainty -rise_from [get_clocks {clk}] -rise_to [get_clocks {nRst}] 0.030
set_clock_uncertainty -rise_from [get_clocks {clk}] -fall_to [get_clocks {nRst}] 0.030
set_clock_uncertainty -fall_from [get_clocks {clk}] -rise_to [get_clocks {clk}] 0.020
set_clock_uncertainty -fall_from [get_clocks {clk}] -fall_to [get_clocks {clk}] 0.020
set_clock_uncertainty -fall_from [get_clocks {clk}] -rise_to [get_clocks {nRst}] 0.030
set_clock_uncertainty -fall_from [get_clocks {clk}] -fall_to [get_clocks {nRst}] 0.030
set_clock_uncertainty -rise_from [get_clocks {nRst}] -rise_to [get_clocks {clk}] 0.030
set_clock_uncertainty -rise_from [get_clocks {nRst}] -fall_to [get_clocks {clk}] 0.030
set_clock_uncertainty -fall_from [get_clocks {nRst}] -rise_to [get_clocks {clk}] 0.030
set_clock_uncertainty -fall_from [get_clocks {nRst}] -fall_to [get_clocks {clk}] 0.030

#####
# Set Input Delay
#####

#####
# Set Output Delay
#####

#####
# Set Clock Groups
#####

#####
# Set False Path
#####
set_false_path -from [get_ports {SDI nRst}] -to [get_ports {seg_Y[6] SDO SPC display_Y[0] display_Y[1] display_Y[2] display_Y[3] display_Y[4] display_Y[5] display_Y[6] display_Y[7] leds_X[0] leds_X[1] leds_X[2] leds_X[3] leds_X[4]}]

#####
# Set Multipole Path
#####

#####
# Set Maximum Delay
#####

#####
# Set Minimum Delay
#####

#####
# Set Input Transition
#####
```

Compilation Report - top_NivelControl

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Se
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
- Fitter
- Assembler
- TimeQuest Timing Analyzer
 - Summary
 - Parallel Compilation
 - SDC File List
 - Clocks
 - Slow 1200mV 85C Mode
 - Slow 1200mV 0C Model
 - Fast 1200mV 0C Model
 - Multicorner Timing Analy
 - Advanced I/O Timing
 - Clock Transfers
 - Report TCCS

Slow 1200mV 85C Model Fmax Summary

<<Filter>>

	Fmax	Restricted Fmax	Clock Name	Note
1	132.54 MHz	132.54 MHz	clk	

This panel reports FMAX for every clock in the design, regardless of the user-specified clock periods. FMAX is only computed for paths where the source and destination registers are driven by the same clock. Paths of different clocks, including generated clocks, are ignored. For

El fichero es top_NivelControl.out.sdc

Recursos utilizados

A continuación, se muestra la pantalla que genera el compilador (opción *Flow Summary* del *Compilation Report*):

Compilation Report - top_NivelControl

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Se
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
- Fitter
- Flow Messages
- Flow Suppressed Messages
- TimeQuest Timing Analyzer
 - Summary
 - Parallel Compilation
 - Clocks
 - Slow 1200mV 85C Mode
 - Fmax Summary
 - Setup Summary
 - Hold Summary
 - Recovery Summary

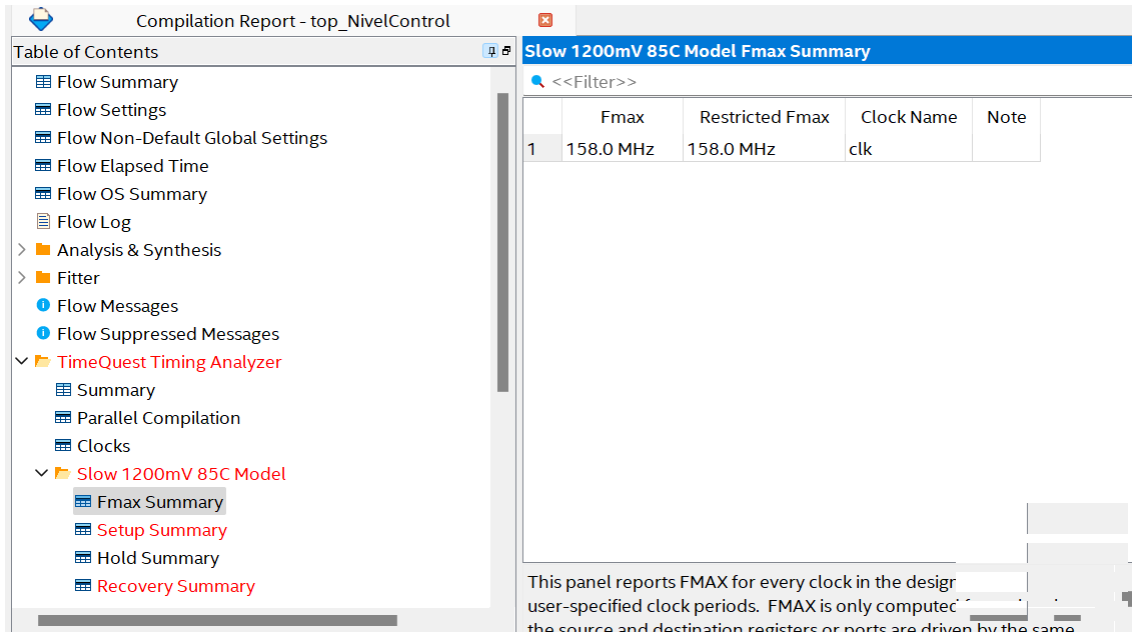
Flow Summary

<<Filter>>

Flow Status	Successful - Wed May 18 16:12:53 2022
Quartus Prime Version	17.1.0 Build 590 10/25/2017 SJ Lite Edition
Revision Name	top_NivelControl
Top-level Entity Name	top_NivelControl
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	456 / 49,760 (< 1 %)
Total registers	184
Total pins	29 / 360 (8 %)
Total virtual pins	0
Total memory bits	660 / 1,677,312 (< 1 %)
Embedded Multiplier 9-bit elements	0 / 288 (0 %)
Total PLLs	0 / 4 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 2 (0 %)

Frecuencia máxima de reloj

A continuación, se muestra la pantalla que genera el compilador (*Fmax Summary*):



The screenshot displays the 'Slow 1200mV 85C Model Fmax Summary' window. On the left is a 'Table of Contents' pane with a tree view showing the hierarchy of the compilation report. The main area on the right contains a table with the following data:

	Fmax	Restricted Fmax	Clock Name	Note
1	158.0 MHz	158.0 MHz	clk	

Below the table, a descriptive text states: 'This panel reports FMAX for every clock in the design user-specified clock periods. FMAX is only computed the source and destination registers or ports are driven by the same'.

6 Bibliografía

- [1] Especificación del diseño: nivel electrónico (NIVEL) [moodle DD2]
- [2] Tarjeta DECA-MAX10 (página web del fabricante). [online]
<https://www.arrow.com/en/products/deca/arrow-development-tools>
- [3] Tarjeta XDECA. Manual de usuario. [moodle DD2-documentacion técnica]
- [4] Datasheet Acelerómetro: LIS2DH12TR.pdf [online]